

**T.C.**  
**SAKARYA ÜNİVERSİTESİ**  
**EĞİTİM BİLİMLERİ ENSTİTÜSÜ**  
**BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ ANABİLİM DALI**  
**BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ BİLİM DALI**

**ÇEVİRİMİÇİ VE YÜZ YÜZE KODLAMA EĞİTİMİNDE OYUNLAŞTIRMA**  
**ÖĞELERİ KULLANIMININ AKADEMİK BAŞARI, MOTİVASYON VE TUTUMA**  
**ETKİSİ**

**DOKTORA TEZİ**

**HÜSEYİN YAŞAR**

**DANIŞMAN**

**PROF. DR. MÜBİN KIYICI**

**MAYIS 2021**



**T.C.**  
**SAKARYA ÜNİVERSİTESİ**  
**EĞİTİM BİLİMLERİ ENSTİTÜSÜ**  
**BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ ANABİLİM DALI**  
**BİLGİSAYAR VE ÖĞRETİM TEKNOLOJİLERİ EĞİTİMİ BİLİM DALI**

**ÇEVİRİMİÇİ VE YÜZ YÜZE KODLAMA EĞİTİMİNDE OYUNLAŞTIRMA**  
**ÖĞELERİ KULLANIMININ AKADEMİK BAŞARI, MOTİVASYON VE TUTUMA**  
**ETKİSİ**

**DOKTORA TEZİ**

**HÜSEYİN YAŞAR**

**DANIŞMAN**

**PROF. DR. MÜBİN KIYICI**

**MAYIS 2021**

## BİLDİRİM

Sakarya Üniversitesi Eğitim Bilimleri Enstitüsü, Tez-Proje Yazım Kılavuzu'na uygun olarak hazırladığım bu çalışmada:

- Tezde yer verilen tüm bilgi ve belgeleri akademik ve etik kurallar çerçevesinde elde ettiğimi ve sunduğumu,
- Yararlandığım eserlere atıfta bulunduğumu ve kaynak olarak gösterdiğimi,
- Kullanılan verilerde herhangi bir deęiřtirmede bulunmadığımı,
- Bu tezin tamamını ya da herhangi bir bölümünü başka bir tez çalışması olarak sunmadığımı

beyan ederim.

Hüseyin YAŞAR

## ÖN SÖZ

Bu çalışmadaki temel amacım, kodlama eğitiminde karşılaşılan sorunların çözümünde, çevrimiçi ve yüz yüze öğrenme ortamlarına oyunlaştırma öğeleri dâhil edilerek, etkinliğinin ortaya konmasıdır. Oyunlaştırmanın bireyler üzerindeki olumlu etkileri ve oyunlaştırma öğeleri kullanımına ilişkin ayrıntılı çalışmaların sınırlılığı, bu çalışmayı yapmamda en büyük etken olmuştur. Bu çalışmamın alana, alanda yapılacak çalışmalara ve derslerinde oyunlaştırma yaklaşımı kullanmak isteyen öğretmenlere katkı sağlamasını umarım.

Bu çalışmanın ortaya çıkmasında desteklerini benden esirgemeyen, fikir aşamasından tamamlanma aşamasına kadar yanımda olan değerli danışmanın Prof. Dr. Mübin KIYICI'ya sonsuz teşekkürlerimi sunarım. Tez izleme komitemde yer alan, fikirleriyle tezin her aşamasında bana yol gösteren Doç. Dr. Fatma SAPMAZ'a ve Dr. Öğr. Üyesi Onur İŞBULAN'a, katkılarından dolayı teşekkür ederim. Tez savunma sınavıma katılım göstererek değerli fikirlerini paylaşan, Prof. Dr. Işıl KABAKÇI YURDAKUL ve Dr. Öğr. Üyesi Zeliha DEMİR KAYMAK'a şükranlarımı sunarım. Ayrıca lisansüstü eğitimin süresince emeklerini unutmayacağım Doç. Dr. Özcan Erkan AKGÜN ve Prof. Dr. Mehmet Barış HORZUM hocalarıma, karşılaştığım sorunlarda yardımlarını benden esirgemeyen Arş. Gör. Murat TOPAL'a, doktora sürecinde beni sabırla destekleyen ve karşılaştığım zorluklarda her zaman yanımda olan eşim, kızım ve oğluma sonsuz teşekkürlerimi sunarım.

## ÖZET

# ÇEVİRİMİÇİ VE YÜZ YÜZE KODLAMA EĞİTİMİNDE OYUNLAŞTIRMA ÖĞELERİ KULLANIMININ AKADEMİK BAŞARI, MOTİVASYON VE TUTUMA ETKİSİ

Hüseyin YAŞAR, Doktora Tezi

Danışman: Prof. Dr. Mübin KIYICI

Sakarya Üniversitesi, 2021

Bu araştırma, çevrimiçi ve yüz yüze kodlama eğitiminde oyunlaştırma öğeleri kullanımının akademik başarı, motivasyon ve tutum üzerindeki etkilerini ortaya koymayı amaçlamaktadır. Bu amaç doğrultusunda nicel araştırma desenlerinden 4x3 karışık (split plot) desen kullanılmıştır. İki faktörden oluşan desenin birinci faktörünü deney ve kontrol olmak üzere 4 grup oluşturmaktadır. Deney gruplarında kodlama eğitimi; oyunlaştırılmış çevrimiçi öğrenme ortamı (OÇÖÖ), oyunlaştırılmış yüz yüze öğrenme ortamı (OYYÖÖ) ve oyunlaştırma öğelerinin kullanılmadığı çevrimiçi öğrenme ortamında (ÇÖÖ) verilmişken, kontrol grubunda ise yüz yüze öğrenme ortamında (YYÖÖ) kodlama eğitimi verilmiştir. Araştırmanın ikinci faktörünü ise tekrarlı ölçümler (ön test, son test, izleme testi) oluşturmaktadır.

Araştırmanın çalışma grubunu, 2017-2018 eğitim öğretim yılında Adıyaman ili Gölbaşı ilçesinde bir Anadolu Lisesinde okuyan ve Seçmeli Bilgisayar Bilimi dersini alan 9. sınıf öğrencileri oluşturmaktadır. Deney ve kontrol grupları 9. sınıf seviyesinde bulunan 5 şube arasından kura ile seçilmiştir. Araştırma, OÇÖÖ grubunda 33, OYYÖÖ grubunda 33, ÇÖÖ grubunda 33 ve YYÖÖ grubunda 33 olmak üzere 132 öğrenci ile yürütülmüştür. Veri toplama aracı olarak; Akademik Başarı Testi, Motivasyon Ölçeği ve Programlama Dillerine Yönelik Tutum Ölçeği kullanılmış, elde edilen veriler karışık ölçümler için iki faktörlü ANOVA kullanılarak analiz edilmiştir.

Elde edilen bulgulara göre tüm grupların akademik başarılarının deneysel işlem süreci sonunda yükseldiği ve izleme ölçümlerinde devam ettiği görülmüştür. Deney gruplarında kullanılan OÇÖÖ, OYYÖÖ ve ÇÖÖ'nun akademik başarıyı arttırmada, kontrol grubunda kullanılan YYÖÖ'dan daha etkili olduğu bulgusuna ulaşılmıştır. Akademik başarının en fazla yükseldiği grup ise OYYÖÖ kullanıldığı grup olmuştur. Ayrıca OÇÖÖ ve ÇÖÖ kullanıldığı gruplarda akademik başarı puanları arasında ise anlamlı bir farklılık

görülmemiştir. İzleme testinden elde edilen bulgulara göre, kalıcılığı sağlamada öğrenme ortamı olarak OÇÖO ve OYYÖO'nun daha etkili olduğu tespit edilmiştir. Motivasyon ile ilgili elde edilen bulgulara göre; OÇÖO ve OYYÖO'nun kullanıldığı gruplarının motivasyon puanlarının yükseldiği görülmüştür. ÇÖO ve YYÖO'nun kullanıldığı gruplarda ise motivasyon puanlarında anlamlı bir değişim gözlemlenmemiştir. OÇÖO ve OYYÖO'nun motivasyonu arttırmada benzer etkiye sahip olduğu bulgusuna ulaşılmıştır. İzleme testi sonuçları incelendiğinde, kalıcılığı sağlama OÇÖO ve OYYÖO'nun benzer etkiye sahip olduğu ve ÇÖO ve YYÖO'dan daha etkili oldukları gözlemlenmiştir. Kodlamaya karşı tutum puanlarına ilişkin bulgular incelendiğinde; tüm gruplarda tutum puanlarının yükseldiği ve izleme ölçümlerinde de devam ettiği görülmüştür. Ayrıca deney gruplarının kodlamaya karşı tutum puanlarının, kontrol grubu tutum puanından yüksek olduğu gözlemlenmiştir. Kodlamaya karşı tutumu arttırmada en etkili öğrenme ortamları benzer etkiye sahip olan OÇÖO ve OYYÖO olmuştur. Ayrıca izleme testinden elde edilen bulgulara göre, deney gruplarında uygulanan deneysel işlemlerin kalıcılığı sağlamada daha etkili olduğu belirlenmiştir. Kalıcılığı sağlamada en etkili öğrenme ortamların ise benzer etkiye sahip olan OÇÖO ve OYYÖO olduğu tespit edilmiştir.

**Anahtar Kelimeler:** Kodlama Eğitimi, Oyunlaştırma, Çevrimiçi Öğrenme, Akademik Başarı, Motivasyon, Kodlamaya Yönelik Tutum

## **ABSTRACT**

### **THE IMPACT OF THE EMPLOYMENT OF GAMIFICATION ELEMENTS IN ONLINE AND FACE-TO-FACE CODING EDUCATION ON ACADEMIC ACHIEVEMENT, MOTIVATION, AND ATTITUDE**

Hüseyin YAŞAR, Doctoral Dissertation

Supervisor: Prof. Dr. Mübin KIYICI

Sakarya University, 2021

The present study aimed to determine the effects of the employment of gamification elements in online and face-to-face coding education on academic achievement, motivation and attitude. In the study, 4x3 split plot design, a quantitative research model, was adopted. The first factor in the design included 4 experiment and control groups. The coding education was conducted in gamified online learning environment (GOLE), gamified face-to-face learning environment (GFFLE), and online learning environment without gamification elements (OLE) in experimental groups, while coding was instructed in face-to-face learning environment (FFLE) in the control group. The second study factor included the repeated measurements (pretest, posttest, follow-up test).

The study group included 9th grade students attending the elective Computer Science Course in an Anatolian High School in Adıyaman province Gölbaşı district during the 2017-2018 academic year. Experimental and control groups were selected randomly from 5 9th grade classes. The study was conducted with 132 students; 33 were in the GOLE group, 33 were in the GFFLE group, 33 were in the OLE group, and 33 were in the FFLE group. Academic Achievement Test, Motivation Scale and Attitude Towards Programming Languages Scale were employed to collect the study data, and the data were analyzed with two-factor ANOVA for mixed measurements.

The study findings demonstrated that the academic achievement increased in all groups at the end of the experimental process and remained high in the follow-up measurements. It was determined that GOLE, GFFLE and OLE employed in the experimental groups were more effective in increasing academic achievement when compared to the FFLE implemented in the control group. The highest increase in academic achievement was observed in the GFFLE group. Furthermore, there was no significant difference between the academic achievement scores of the GOLE and OLE groups. The follow-up test scores



revealed that GOLE and GFFLE were more effective learning environments in ensuring permanence. The findings on student motivation demonstrated that the motivation scores increased in the GOLE and GFFLE groups. No significant change was observed in the motivation scores of the OLE and FFLE groups. It was determined that GOLE and GFFLE had a similar effect on the increase in motivation. The follow-up test results revealed that GOLE and GFFLE had similar effects and these applications were more effective when compared to OLE and FFLE. The findings on the student attitude scores towards coding demonstrated that the attitude scores increased in all groups and remained at the same level in the follow-up measurements. Furthermore, it was observed that the attitude scores of the experimental groups towards coding were higher when compared to those of the control group. The most effective learning environments in increasing the attitude towards coding were GOLE and GFFLE, which exhibited similar effects. Also, the follow-up test findings revealed that the experimental procedures implemented in the experimental groups were more effective in achieving permanence. It was determined that the most effective learning environments to ensure permanence were GOLE and OFFLE, which had similar effects.

**Keywords:** Coding Education, Gamification, Online Learning, Academic Achievement, Motivation, Attitude Towards Coding

## İÇİNDEKİLER

BİLDİRİM .....	i
ÖN SÖZ.....	ii
ÖZET .....	iii
ABSTRACT .....	v
TABLolar LİSTESİ .....	x
ŞEKİLLER LİSTESİ.....	xi
RESİMLER LİSTESİ.....	xii
SİMGELER VE KISALTMALAR .....	xiii
BÖLÜM I .....	1
GİRİŞ.....	1
1.1. Problem cümlesi .....	5
1.2. Alt problemler .....	5
1.3. Önem .....	6
1.4. Sınırlılıklar.....	7
1.5. Tanımlar .....	7
BÖLÜM II.....	8
ARAŞTIRMANIN KURAMSAL ÇERÇEVESİ VE İLGİLİ ARAŞTIRMALAR.....	8
2.1. Araştırmanın kuramsal çerçevesi .....	8
2.1.1. Kodlama (Programlama).....	8
2.1.1.1. Kodlama Eğitimi.....	11
2.1.1.2. Metin tabanlı kodlama.....	13
2.1.1.3. Bilgisayar bilimi dersi .....	15
2.1.2. Oyunlaştırma.....	16
2.1.2.1. Oyunlaştırmanın kuramsal dayanakları .....	19
2.1.2.2. Oyunlaştırma tasarım modelleri ve bileşenleri.....	22

2.1.2.3. Oyuncu tipleri .....	28
2.1.2.4. Eğlence türleri.....	30
2.1.3. Çevrimiçi öğrenme ortamları .....	31
2.2. İlgili arařtırmalar .....	32
2.2.1. Oyunlařtırma ile ilgili arařtırmalar .....	32
2.2.2. Kodlama eğitiminde oyunlařtırma kullanımını ile ilgili arařtırmalar .....	39
2.2.3. Alanyazın Taramasının Sonucu .....	41
<b>BÖLÜM III.....</b>	<b>43</b>
<b>YÖNTEM .....</b>	<b>43</b>
3.1. Arařtırma yöntemi .....	43
3.2. Arařtırmanın çalıřma grubu .....	44
3.3. Veri toplama araçları .....	45
3.3.1. Akademik başarı testi .....	45
3.3.2. Motivasyon Ölçeęi .....	47
3.3.3. Tutum ölçeęi .....	48
3.4. Verilerin toplanması .....	48
3.5. Uygulama süreci .....	48
3.5.1. Ön hazırlık eğitimi ve pilot uygulama .....	49
3.5.2. Oyunlařtırılmıř çevrimiçi öğrenme ortamı .....	50
3.5.2.1. Öğretim materyalleri.....	51
3.5.2.2. Oyunlařtırma tasarım süreci .....	55
3.5.2.3. Oyunlařtırma bileřenleri.....	58
3.5.3. Oyunlařtırılmıř yüz yüze öğrenme ortamı .....	64
3.5.3.1. Öğretim materyalleri.....	64
3.5.3.2. Oyunlařtırma tasarım süreci .....	66
3.5.3.3. Oyunlařtırma bileřenleri.....	68

3.5.4. Çevrimiçi öğrenme ortamı .....	72
3.5.5. Yüz yüze öğrenme ortamı .....	72
3.6. Verilerin analizi .....	73
BÖLÜM IV .....	80
BULGULAR .....	80
4.1. Akademik başarı testi puanlarına ilişkin bulgular .....	80
4.2. Motivasyon puanlarına ilişkin bulgular .....	85
4.3. Tutum puanlarına ilişkin bulgular .....	91
BÖLÜM V .....	98
SONUÇ, TARTIŞMA VE ÖNERİLER .....	98
5.1. Sonuç ve Tartışma .....	98
5.1.1. Akademik başarı testinden elde edilen bulgulara yönelik sonuç ve tartışma .....	98
5.1.2. Motivasyon ölçeğinden elde edilen bulgulara yönelik sonuç ve tartışma .....	102
5.1.3. Tutum ölçeğinden elde edilen bulgulara yönelik sonuç ve tartışma .....	106
5.2. Öneriler .....	108
5.2.1. Araştırma sonuçlarına dayalı öneriler .....	108
5.2.2. İleride yapılabilecek araştırmalara yönelik öneriler .....	109
KAYNAKLAR .....	110
EKLER .....	123

## TABLÖLAR LİSTESİ

Tablo 1. Oyun ve Oyunlaştırma Arasındaki Farklar .....	18
Tablo 2. Araştırmanın Simgesel Gösterimi .....	43
Tablo 3. Deney ve Kontrol Gruplarındaki Öğrencilere Ait Bilgiler.....	44
Tablo 4. Ders Videolarında Kullanılan Şifreler.....	53
Tablo 5. OÇÖÖ'da Kullanılan Eğlenelim Etkinlikleri .....	54
Tablo 6. OYYÖÖ'da Kullanılan Eğlenelim Etkinlikleri .....	65
Tablo 7. ABT, MÖ ve PDYTÖ Ön Test, Son Test ve İzleme Testi Varyans Homojenliği (Levene) Testi Sonuçları .....	74
Tablo 8. ABT Ön Test Puanlarına İlişkin Tek Faktörlü ANOVA Sonuçları .....	75
Tablo 9. MÖ Ön Test Puanlarına İlişkin Tek Faktörlü ANOVA Sonuçları.....	75
Tablo 10. PDYTÖ Ön Test Puanlarına İlişkin Tek Faktörlü ANOVA Sonuçları.....	76
Tablo 11. ABT, MÖ ve PDYTÖ Ön Test, Son Test ve İzleme Testi Ölçümlerinden Elde Edilen Mauchly Küresellik Testi Sonuçları.....	78
Tablo 12. ABT Ön Test, Son Test ve İzleme Testi Puanlarına İlişkin Betimsel İstatistikler .....	80
Tablo 13. ABT Ön Test, Son Test ve İzleme Testi Puanlarına İlişkin Karışık Ölçümler İçin İki Faktörlü ANOVA Sonuçları .....	81
Tablo 14. MÖ Ön Test, Son Test ve İzleme Testi Puanlarına İlişkin Betimsel İstatistikler	86
Tablo 15. MÖ Ön Test, Son Test ve İzleme Testi Puanlarına İlişkin Tekrarlı Ölçümler ANOVA Sonuçları .....	87
Tablo 16. PDYTÖ Ön Test Son Test ve İzleme Testi Puanlarına İlişkin Betimsel İstatistikler .....	92
Tablo 17. PDYTÖ Ön Test, Son Test ve İzleme Testi Puanlarına İlişkin Tekrarlı Ölçümler ANOVA Sonuçları .....	93

## ŞEKİLLER LİSTESİ

Şekil 1. Kodlama süreci (Gundurao et al., 2010) .....	9
Şekil 2. Programlama dili seçimine yönelik yaklaşımlar .....	14
Şekil 3. Bilgisayar bilimi (kur 1, kur 2) öğretim programında yer alan üniteler.....	16
Şekil 4. Akış deneyimi (Csikszentmihalyi ve Rathunde, 2014:28).....	21
Şekil 5. Werbach ve Hunter (2012:78) oyunlaştırma tasarım modeli .....	22
Şekil 6. D6 oyunlaştırma tasarımı aşamaları (Werbach ve Hunter, 2012).....	25
Şekil 7. Uygulama süreci.....	49
Şekil 8. OÇÖÖ öğretim materyalleri.....	52
Şekil 9. OÇÖÖ oyunlaştırma tasarım modeli.....	55
Şekil 10. OÇÖÖ tasarlanırken kullanılan oyuncu (öğrenci) tipleri.....	57
Şekil 11. OÇÖÖ'da kullanılan dinamikler, mekanikler ve bileşenler .....	58
Şekil 12. OÇÖÖ'da kullanılan rozetler .....	60
Şekil 13. OYYÖÖ öğretim materyalleri.....	64
Şekil 14. OYYÖÖ oyunlaştırma tasarım modeli.....	66
Şekil 15. OYYÖÖ tasarlanırken kullanılan oyuncu (Öğrenci) tipleri.....	67
Şekil 16. OYYÖÖ'da kullanılan dinamikler, mekanikler ve bileşenler .....	68
Şekil 17. ABT ön test, son test ve izleme testi puan ortalamaları grafiği .....	85
Şekil 18. MÖ ön test, son test ve izleme testi puan ortalamaları grafiği.....	91
Şekil 19. PDYTÖ ön test, son test ve izleme testi puan ortalamaları grafiği .....	96

## RESİMLER LİSTESİ

Resim 1. Chou (2015) octalysis tasarım modeli.....	27
Resim 2. Bartle (1996) oyuncu tipleri .....	29
Resim 3. Marczewski (2015) oyuncu tipleri .....	30
Resim 4. OÇÖO ekran görüntüsü.....	51
Resim 5. OÇÖO'da kullanılan lider tahtası .....	61
Resim 6. OÇÖO'da kullanılan seviyeler ve rütbeler .....	62
Resim 7. OÇÖO'da kullanılan ilerleme çubuğu .....	63
Resim 8. OYYÖO'da kullanılan lider tahtası .....	70
Resim 9. OYYÖO'da kullanılan seviyeler ve rütbeler .....	71
Resim 10. ÇÖO ekran görüntüsü.....	72

## SİMGELER VE KISALTMALAR

OÇÖO: Oyunlaştırılmış çevrimiçi öğrenme ortamı

OYYÖO: Oyunlaştırılmış yüz yüze öğrenme ortamı

ÇÖO: Çevrimiçi öğrenme ortamı

YYÖO: Yüz yüze öğrenme ortamı

MÖ: Motivasyon ölçeği

PDYTÖ: Programlama dillerine yönelik tutum ölçeği

ABT: Akademik başarı testi

n: Frekans

$\bar{X}$ : Aritmetik ortalama

Ss: Standart sapma

Sd: Serbestlik derecesi

F: F değeri

n: Anlamlılık düzeyi

$\eta^2$ : Eta kare değeri (Etki büyüklüğü)

ANOVA: Varyans analizi



# BÖLÜM I

## GİRİŞ

Geçmiş yıllarda bir öğretmen olarak öğrencilerimize, "Bilgi ve iletişim teknolojileri hangi alanlarda kullanılmaktadır?" diye bir soru yönelttiğinizi varsayalım. Bu soruya öğrencilerin verecekleri cevapları düşündüğümüzde, sınırlı cevap seçeneklerinin olduğu görülecektir. Sorduğumuz soruyu bugünün şartlarında revize ederek: "Bilgi ve iletişim teknolojilerinin kullanılmadığı alanları, kullanılmama nedenleriyle birlikte açıklayınız?" şeklindeki yönelttiğimizde öğrencilerin verecekleri cevaplar benzer şekilde sınırlı olacaktır. Çünkü bilgi toplumuyla birlikte bilgi ve iletişim teknolojileri hayatın her alanına girmiş durumda. Eğitim, sağlık, ulaşım, haberleşme vb. birçok alanda karşılaşılan problemlere bilgi ve iletişim teknolojileri merkezli çözümler üretilmiş ve üretilmeye devam etmektedir. Neredeyse değişim ve yenilik kavramlarını gördüğümüz her alanda bilgi ve iletişim teknolojilerinin etkisini hissetmekteyiz.

Bilgi ve iletişim teknolojilerinin hayatımızda meydana getirdiği yenilik ve değişim hızının, üretim ve kullanım konusunda da aynı oranda olması gerekmektedir (Gülbahar ve Kalelioğlu, 2018). Günümüz dünyasında çocukların ve gençlerin pasif tüketenler olmamaları ve bu dijital dönüşümün bir parçası olmaları kaçınılmazdır (Kandemir, 2017:268). Bunun sağlanması ise bireylerin yaşadığımız yüzyılın önemli becerilerini kazanmaları ve yaşamlarına adapte etmeleriyle olacaktır. 21. yüzyıl becerileri olarak da ifade edilen bu becerilerden biri de kodlamadır. Kodlama, yaşadığımız yüzyılın yeni dilidir ve bu dili bilmeyenlerin bugün okuma-yazma veya matematik bilmeyenlerin karşılaştıkları zorluklarla karşılaşmaları büyük olasılıktır (Kandemir, 2017:267). Bu nedenle problem analizinden, programın kullanıma hazır hale getirilmesi aşamasına kadar ilerleyen kodlama süreci, tüm bireylerin kazanması gereken beceriler arasında görülmektedir. Nitekim kodlama becerisi bireyde; mantık kullanımı, karar verme, eleştirel düşünme ve sorgulama gibi becerilerin de geliştirilmesini sağlamaktadır (Gülbahar ve Kalelioğlu, 2018). Bununla birlikte kodlama becerisi, problem çözme ve sistemli düşünme gibi üst düzey düşünme becerilerini geliştirmekle birlikte problemleri farklı yönleriyle ele alabilme ve pratik çözümler üretebilme yetisi kazandırmaktadır (Yükseltürk ve Altıok, 2015). Bireysel gelişime katkılarının yanı sıra, geleceğin mesleklerinde de kodlama becerisini kullanan

bireylere ihtiyaç duyulacağı belirtilmektedir (Bidwell, 2013). Bilgi toplumuyla birlikte toplumsal yapıdaki değişim, her alanda bireyden istenen bilgi ve becerilerinde değişmesine neden olmuştur. Bilgi ve iletişim teknolojilerini etkin bir şekilde kullanan, problemlere bu teknolojileri kullanarak çözüm üretebilen ve ürettiği çözümü kodlayarak işlevsel programlar yaratan bireylere ihtiyaç duyulmaktadır. Bu ihtiyacın giderilmesinde ise kodlama becerisine sahip bireylerin yetiştirilmesi önemli olarak görülmektedir.

Kodlama becerisinin sahip olduğu güç göz önüne alındığında, kodlama eğitiminin önemi daha iyi anlaşılmaktadır. 21. yüzyıl becerileriyle donatılmış bireylerin yetiştirilmesi ve bilgi toplumuyla birlikte meydana gelen dönüşümün bir parçası olan insan kaynakları ihtiyacının giderilmesi noktasında kodlama eğitiminin önemi gün geçtikçe artmaktadır. Kodlama eğitimi, bulunduğumuz yüzyılda yazılım çalışmalarının sürdürülebilirliğini (Kert ve Uğraş, 2009) ve problem çözmek için gerekli yazılım üretimini sağlayan önemli bir bileşendir. Kodlama eğitiminin bireysel gelişime katkıları göz önüne alındığında, küçük yaşlardan itibaren verilmesi önem arz etmektedir (Kert ve Uğraş, 2009). Dolayısıyla kodlama mantığının öğrencilere erken yaşlarda verilmesi, bireyin diğer alanlardaki başarılarına da katkı sağlayacağı (Karabak ve Güneş, 2013) gibi bireyin geleceğe hazırlanmasında önemli bir fırsat oluşturacaktır.

Günümüz şartlarında kodlama becerisi, çocukluktan itibaren kazandırılması gereken bir beceri olarak karşımıza çıkmaktadır. Fakat yapısı gereği matematiksel, mantıksal vb. birçok beceriyi içinde barındırmasından dolayı, bu becerilere sahip olmayan öğrenciler tarafından öğrenilmesi zor bir alan olarak algılanmaktadır (Gomes ve Mendes, 2007). Bunlar dışında kodlama eğitiminde öğrenciden ve kodlamanın yapısından kaynaklı birçok zorlukla karşılaşıldığı görülmektedir. Bunlar; algoritma oluşturma, programlama dilinin yapısı, öğrenme ve öğretme yöntemleri, hatayı bulma ve ayıklama, motivasyon, sayısal bilgi-beceri eksikliği, kod yazma- koda dökme, kodlamaya karşı tutum ve program geliştirme şeklindedir (Saygıner ve Tüzün, 2018). Problemin analizinden başlayıp, algoritmaların oluşturulması, algoritmaların görselleştirilmesi, kodların yazılması ve hata ayıklama gibi karmaşık birçok süreci içinde barındıran kodlama eğitiminde; karşılaşılan problemlerin ortadan kaldırılması veya en aza indirgenmesi için geleneksel kodlama eğitiminde de değişimlerin olması kaçınılmazdır. Farklı araçların, yöntemlerin, tekniklerin ve öğrenme ortamlarının sürece dâhil edilmesi göz önüne alınabilir.

Bu yöntemlerden biri de oyunlaştırma (gamification) yaklaşımıdır. Nick Pelling tarafından 2003 yılında kullanılmasına rağmen şu anki anlamıyla 2010 yılında benimsenen (Werbach

ve Hunter, 2012) oyunlaştırma kavramı, oyun tasarım bileşenlerinin oyun dışı ortamlarda kullanılması olarak ifade edilmiştir (Deterding, Dixon, Khaled ve Nacke, 2011). Zichermann ve Cunningham (2011:16) ise oyunlaştırmayı, oyun düşüncesinin ve oyun mekaniklerinin kullanıcının ilgisinin çekilmesi ve problemlerinin çözülmesi amacıyla kullanılması süreci olarak tanımlanmıştır. Kapp (2012), oyunlaştırmanın; özellikle bireyin içerikte ilerlemesinin teşvik edilmesinde, motivasyonun sağlanmasında, davranış değişikliğinin oluşturulmasında ve yeniliğe karşı harekete geçirilmesi amacıyla etkili bir şekilde kullanılabileceğini belirtmiştir.

Öğrencilerin motivasyonlarının sağlanmasında oyunlaştırma önemli bir yöntem olarak karşımıza çıkmaktadır. Birçok eğitim alanında olduğu gibi kodlama eğitiminde de karşılaşılan en önemli problemlerden biri, öğrencinin yeterli motivasyonu sağlanmadan öğrenme ortamına dâhil edilmesidir. Nitekim öğrencinin motive edilerek ortama aidiyetinin artırılması ile öğrenme sürecinin kolaylaşacağı ve derinlemesine öğrenmenin gerçekleşeceği ifade edilmektedir (Kumar ve Khurana, 2012). Öğrencilerin, yapısı gereği karmaşık olarak gördükleri ve öğrenmekte zorlandıkları kodlama becerisinin kazandırılmasında, öğrencinin motivasyonunun sağlanması ve öğrenme ortamına etkin olarak dâhil edilmesi gerekmektedir. Bununla birlikte kodlama eğitiminde oyunlaştırmanın etkili ve ciddi bir şekilde kullanılması, öğrencilerin gerçek yaşam problemlerini çözmelerini sağlayacağı belirtilmiştir (Kumar ve Khurana, 2012). Fidan (2016) ise kodlama eğitiminde oyunlaştırma kullanımının eğlenceli ve zevkli bir öğrenme ortamı oluşturduğunu dile getirmiş, Lee ve Hamer (2011) ise oyunlaştırmanın, öğrenci katılımını sağlama, ortama bağlılığını artırma ve motive etme açısından önemli bir strateji olduğunu ifade etmiştir. Alanyazında yapılan araştırmalar; kodlama eğitiminde oyunlaştırma kullanımının motivasyon (Fidan, 2016; Yıldız, 2018; Atabay ve Albayrak, 2020), tutum (Yıldız, 2018; Sağay, 2019) ve öğrenme başarısı (Fidan, 2018; Türk ve Gören, 2017; Atabay ve Albayrak, 2020) üzerine olumlu etkilerinin olduğunu ortaya koymaktadır.

Gelişen teknolojiler ve yaşam şartlarında meydana gelen değişimler, öğrenme ortamlarının da değişimini beraberinde getirmiştir. Özellikle internetin gelişmesiyle, bilgi toplumunun sorumluluklarına sahip olma ve artan bilgi ihtiyacını karşılamak için farklı öğrenme süreçlerine ve ortamlarına olan geçiş hızlanmıştır (Zhang, 2002). Bu değişimin bir parçası olan çevrimiçi öğrenme ortamları; öğrenmenin internet, intranet ve bilgisayar ağları gibi teknolojiler kullanılarak, öğretmen ve öğrencinin zaman ve mekândan bağımsız iletişime geçebildiği ve bireysel öğrenmenin ön planda olduğu öğrenme ortamları (Bilgiç, Duman ve

Seferođlu, 2011) olarak tanımlanmaktadır. Çevrimiçi öğrenme ortamlarının bilgiye istenildiğinde ulaşma, ortam ve zaman kısıtlamasının olmaması, istenildiğinde derse ulaşma, öğrenmede eşit fırsata sahip olma ve maliyet (Balcı, 2011; Özgür H., 2011) gibi birçok avantajı bulunmaktadır.

Çevrimiçi öğrenme ortamları birçok alanda yaygın olarak kullanılmakta ve gün geçtikçe kullanım oranı artmaktadır. Bireysel ve kurumsal olarak birçok eğitim çevrimiçi öğrenme ortamları aracılığıyla yapılmakta ve bu ortamların verimliliğinin artırılmasına yönelik çeşitli yaklaşımların kullanıldığı görülmektedir. Bu yaklaşımlardan biri olan oyunlaştırma; çevrimiçi öğrenme ortamında bağlılığın sağlanması, öğrencilerin motive edilmesi ve derslere devamın artırılması açısından büyük fırsatları içinde barındırmaktadır (Sun ve Rueda, 2012). Öğrencinin motive edilmesi ve karşılaşılan sorunların ortadan kaldırılması için öğrenme ortamı ve oyunlaştırma birbirinin tamamlayıcısı niteliğindedir (Ribeiro, Coelho ve Aguiar, 2011). Nitekim çevrimiçi öğrenme ortamının etkililiğinin artırılması (Tunga ve İnceođlu, 2016), derse katılım problemlerinin ortadan kaldırılması (Yuen, Lee ve Tsang, 2011) ve derse karşı bağlılığın artırılması (Glover, 2013) için oyunlaştırmannın önemli bir çözüm olacağı görülmektedir.

Bilgi toplumuyla birlikte teknoloji alanında meydana gelen hızlı gelişmeler tüm alanlarda önemli değişimleri beraberinde getirmiştir. Dünya genelinde ülkelerin gelişmişlik düzeylerinin temel göstergeleri arasında; bilgi ve iletişim teknolojilerinin her alanda etkin bir şekilde kullanımı, üretimi ve pazarlanması gelmektedir. Temel güç kaynağı olarak görülen bu teknolojilerin üretilmesinde ve kullanılmasında gerekli insan kaynakları ihtiyacının karşılanması önemli hususlardan biridir. Yetişmiş insan gücünün sağlanmasında ise kodlama becerisinin bireylere kazandırılması kaçınılmazdır. Kodlama becerisi dünya genelinde üzerinde düşünölen, öğrencilere kazandırılması amacıyla önemli çalışmaların yürütöldüğü bir beceri olarak karşımıza çıkmaktadır. Fakat bu becerinin kazandırılmasına yönelik yapılan kodlama eğitimlerinde birçok sorunla karşılaşılmaktadır. Kodlama becerisinin önemi de düşünöldüğünde, karşılaşılan sorunların çözüme ulaşması önem arz etmektedir. Bu nedenle araştırmadaki temel amaç; 21. yüzyıl becerileri arasından yer alan kodlama becerisinin öğrencilere kazandırılmasında karşılaşılan sorunların giderilmesi ya da en aza indirgenmesi noktasında, çevrimiçi ve yüz yüze kodlama eğitimlerinde oyunlaştırma öğeleri kullanımının oluşturduğu etkinin belirlenmesidir.

## 1.1. Problem cümlesi

Çevrimiçi ve yüz yüze kodlama eğitiminde oyunlaştırma öğeleri kullanımının 9. sınıf öğrencilerinin akademik başarıları, motivasyonları ve kodlamaya yönelik tutumları üzerine etkisi var mıdır?

## 1.2. Alt problemler

1. Deney ve kontrol gruplarının akademik başarı testi (ABT) puanları arasında anlamlı bir farklılık var mıdır?

1.1. OÇÖO, OYYÖO, ÇÖO ve YYÖO'nun kullanıldığı deney ve kontrol grupları ayrı ayrı incelendiğinde, ABT ön test, son test ve izleme testi puanları arasında anlamlı farklılık var mıdır?

1.2. OÇÖO, OYYÖO, ÇÖO ve YYÖO'nun kullanıldığı deney ve kontrol grupları karşılaştırıldığında, ABT ön test ve son test puanları arasında anlamlı bir farklılık var mıdır?

1.3. OÇÖO, OYYÖO, ÇÖO ve YYÖO'nun kullanıldığı deney ve kontrol grupları karşılaştırıldığında, son test ve izleme testi puanları arasında anlamlı bir farklılık var mıdır?

2. Deney ve kontrol gruplarının motivasyon ölçeği (MÖ) puanları arasında anlamlı bir farklılık var mıdır?

2.1. OÇÖO, OYYÖO, ÇÖO ve YYÖO'nun kullanıldığı deney ve kontrol grupları ayrı ayrı incelendiğinde, MÖ ön test, son test ve izleme testi puanları arasında anlamlı farklılık var mıdır?

2.2. OÇÖO, OYYÖO, ÇÖO ve YYÖO'nun kullanıldığı deney ve kontrol grupları karşılaştırıldığında, MÖ ön test ve son test puanları arasında anlamlı bir farklılık var mıdır?

2.3. OÇÖO, OYYÖO, ÇÖO ve YYÖO'nun kullanıldığı deney ve kontrol grupları karşılaştırıldığında, MÖ son test ve izleme testi puanları arasında anlamlı bir farklılık var mıdır?

3. Deney ve kontrol gruplarının programlama dillerine yönelik tutum ölçeği (PDYTÖ) puanları arasında anlamlı bir farklılık var mıdır?

- 3.1.** OÇÖO, OYYÖO, ÇÖO ve YYÖO'nun kullanıldığı deney ve kontrol grupları ayrı ayrı incelendiğinde, PDYTÖ ön test, son test ve izleme testi puanları arasında anlamlı farklılık var mıdır?
- 3.2.** OÇÖO, OYYÖO, ÇÖO ve YYÖO'nun kullanıldığı deney ve kontrol grupları karşılaştırıldığında, PDYTÖ ön test ve son test puanları arasında anlamlı bir farklılık var mıdır?
- 3.3.** OÇÖO, OYYÖO, ÇÖO ve YYÖO'nun kullanıldığı deney ve kontrol grupları karşılaştırıldığında, PDYTÖ son test ve izleme testi puanları arasında anlamlı bir farklılık var mıdır?

### **1.3. Önem**

21. yüzyıl becerileri arasında yer alan kodlama, bireylerin küçük yaşlardan itibaren kazanması gereken bir beceri olarak görülmektedir. Bireysel gelişime katkıları ve bilgi toplumunun ihtiyaçları düşünüldüğünde, kodlama becerisinin kazandırılmasına yönelik yapılan çalışmaların titizlikle yürütülmesi ve karşılaşılan sorunlara çözüm yolları aranması gerekmektedir. Alanyazında öğrenme ve öğretme yöntemleri, motivasyon, kodlamaya karşı tutum gibi çeşitli sorunlarla karşılaşıldığı ifade edilmektedir (Saygıner ve Tüzün, 2018). Bu sorunların çözümüne katkı sağlayacak farklı öğrenme ortamlarının, motivasyon sağlamaya ve tutum geliştirmeye yönelik farklı yaklaşımların kullanılması önemli olarak görülmektedir. Bu önem doğrultusunda, araştırmada kodlama eğitiminde farklı öğrenme ortamları kullanılmış ve oyunlaştırma öğeleri ile desteklenmiştir. Oyunlaştırılmış çevrimiçi öğrenme ortamı, oyunlaştırılmış yüz yüze öğrenme ortamı, çevrimiçi öğrenme ortamı ve yüz yüze öğrenme ortamı olmak üzere 4 farklı öğrenme ortamı kullanılarak, kodlama eğitiminde karşılaşılan; motivasyon, kodlamaya karşı tutum ve akademik başarı gibi sorunların çözümüne etkisinin belirlenmesi açısından özgün ve gereklidir. Son yıllarda önemi giderek artan çevrimiçi öğrenme ortamlarının kullanılması ve oyunlaştırma öğeleri ile desteklenmesi açısından günceldir. Ayrıca oyunlaştırma öğelerinin kullanımına yönelik; oyunlaştırma tasarımımda hangi aşamaların takip edileceğine, oyuncu tiplerinin ve eğlence türlerinin nasıl belirleneceğine ve hangi oyunlaştırma öğelerinin kullanılacağına yönelik çerçeve sunması açısından *işlevseldir*.

#### **1.4. Sınırlılıklar**

1. Araştırmada toplanan veriler, 2017-2018 eğitim öğretim yılının ikinci döneminde bilgisayar bilimi dersini alan 132 ortaöğretim 9.sınıf öğrencisi ile,
2. Deneysel işlem süreci, 2 hafta pilot uygulama ve 10 hafta deneysel uygulama olmak üzere 12 hafta ile,
3. Araştırmada kullanılan programlama dili Python ile,
4. Kullanılan çevrimiçi öğrenme ortamı ise Moodle ile sınırlıdır.

#### **1.5. Tanımlar**

*Oyunlaştırma:* Problemlerin çözülmesinde, motivasyon ve bağımlılığın sağlanmasında oyun olmayan içerik ve ortamlarda oyun öğeleri ve tasarım ilkelerinin kullanılmasıdır (Deterding et al., 2011; Werbach ve Hunter, 2012; Zichermann ve Cunningham, 2011)

*Programlama (Kodlama):* Analizi ve tasarımı yapılmış problemlerin çözüm adımlarının, özel kodların kullanıldığı bir programlama diliyle dijital ortama aktarılması programlama olarak ifade edilebilir (Eryılmaz, 2003).

*Motivasyon:* Bireyi davranışa iten, davranışın düzeyini belirleyen, davranışa yön veren ve davranışın devamlılığını sağlayan bir etmendir (Deci, Kostner ve Ryan, 2001).

*Tutum:* Bireyin çevresindeki bir konuya karşı öncesinde bir eğilimin olması olarak ifade edilir (Nuhoğlu, 2008).

## BÖLÜM II

### ARAŞTIRMANIN KURAMSAL ÇERÇEVESİ VE İLGİLİ ARAŞTIRMALAR

Bu bölümde araştırmaya konu olan kavramlar ve kuramsal bilgiler yer alacaktır.

#### 2.1. Araştırmanın kuramsal çerçevesi

Bu bölümde araştırmanın kuramsal çerçevesine yer verilecektir. Kodlama eğitimi ve oyunlaştırma hakkında kuramsal çerçeve ortaya konulacaktır.

##### 2.1.1. Kodlama (Programlama)

Kodlama diğer bir adıyla programlama; bilgi ve iletişim teknolojileri araçlarının, bir işlemin yapılabilmesi için kodlanması olarak tanımlanmaktadır (Sayın ve Seferoğlu, 2016). Bir başka tanıma göre ise bilgisayara işini nasıl yapması gerektiğinin anlatılmasıdır (Van-Roy ve Haridi, 2004:1). Günlük hayat becerisi olan yazı yazmanın öğrenilmesi, nasıl insanlar için bir ihtiyaç ise kodlama da yaşadığımız yüz yılda insanların ihtiyaç duyacağı yazmanın yeni ve gelişmiş bir formudur. Çünkü kodlama düşünmenin ve üretmenin bir yoludur (Sayın ve Seferoğlu, 2016).

Kodlama işlemi, programlama dili olarak ifade edilen belirli komutlar vasıtasıyla yapılır. Programlama dili, programı yazan kişinin oluşturduğu algoritmayı bilgisayara aktarması olarak tanımlanabilir (Karabak ve Güneş, 2013). Arslan ve Tanel (2017) ise programlama dilini, bilişim teknolojileri araçlarının belirlenen amaçların sağlanmasında kullanılan komutlar olarak tanımlamıştır. Kesici ve Kocabaş (2007) programlama dillerini alt seviyeli diller ve üst seviyeli diller olarak iki gruba ayırmıştır. Alt seviyeli diller; kodlamada ikili sayma sisteminin kullanıldığı makine dili ve assembly dilleridir. Üst seviyeli diller ise belirli kavramların kullanılarak kodlamaların yapıldığı; Basic, Fortran, Cobol, Pascal, Delphi, Java, Visual Basic, C++, C#, Python gibi dillerdir. Özellikle üst seviyeli dillerdeki gelişmeler günlük yaşamdaki birçok şeyin, programlar yazılarak bilgisayarlar aracılığıyla yapılması girişimlerini arttırmıştır (Yükseltürk ve Altınok, 2017).



Kodlama süreci kodların yazılmasından daha kapsamlı bir süreç olup, programlama dili kullanılarak yapılan kod yazma eylemi, sadece kodlama sürecinin bir parçasından ibarettir. Kodlama süreci; problemin analizi, algoritmaların kurulması, kodların yazılması, programın çalıştırılması ve hata ayıklama gibi aşamalardan oluşmaktadır (Chiu, 2014). Kodlama süreciyle ilgili bir başka sınıflamada Gundurao, Manjunath ve Nachappa (2010) tarafından yapılmıştır. Gundurao et al. (2010) tarafından ifade edilen üç aşamalı kodlama süreci Şekil 1'de sunulmuştur.



Şekil 1. Kodlama süreci (Gundurao et al., 2010)

Şekil 1'de görüldüğü gibi kodlama sürecinin ilk aşaması, problemin analiz edilerek çözüm yollarının belirlendiği aşamadır. Problemin analizinden sonra ise probleminin çözümü aşamalar halinde algoritma denilen yapılara dönüştürülür ve akış diyagramları halinde resmedilir. Son aşamasında ise program; seçilen programlama dili kullanılarak yazılır ve kontrolü sağlanarak son hali verilir. Kodlama sürecini özetlersek; algoritmaların bilgisayar diline çevrilmesi ve analiz edilen problemi çözüp çözmediğinin test edilerek doğrulanması adımlarından oluşur (Gundurao et al., 2010).

Kodlama, teknoloji ve kullanılan teknolojinin bilimsel temelleri olmak üzere iki önemli bileşenden oluşur. Teknoloji kısmı, programlama yapmamıza olanak sağlayan araçlar, teknikler ve standartları ifade eder. Bilimsel temeller ise programlamayı anlamamızı sağlayan kuramsal temellerden oluşur (Van-Roy ve Haridi, 2004:15). Bayman ve Mayer (1988) kodlama sürecinde üç farklı bilgi türünün kavranması ve kullanılması gerektiğini ifade etmiştir. Bunlardan ilki, programlama dilinin kendine özgü dil özelliklerini ve yazım

kurallarını içeren söz dizimsel (syntactic) bilgidir. İkincisi; döngü, koşul ve değişken gibi kavramları içeren kavramsal (conceptual) bilgi, üçüncüsü ise diğer iki bilgi türünün problemin çözümünde işe koşulmasını sağlayan stratejik (strategic) bilgidir (Yükseltürk ve Altınok, 2017). McGill ve Volet (1997) ise yeni bir sınıflandırmaya giderek, Bayman ve Mayer (1988) tarafından ortaya konulan söz dizimsel, kavramsal ve stratejik bilgileri bildirimsel (declarative) ve işlemsel (procedural) bilgi olarak ifade etmiştir. Bildirimsel bilgi programlama dilinin kurallarını bilme, işlemsel bilgi ise kurallara uygun kod yazabilme olarak ifade edilmiştir (McGill ve Volet, 1997).

Kodlama ile ilgili ele alınabilecek bir diğer bilimsel temel ise kodlama paradigmalarıdır. Vujosevic-Janicic ve Tosic (2008) problem çözme temelli 4 paradigma ifade etmiştir. Bunlar; zorunlu paradigma (imperative paradigm), nesneye yönelik paradigma (object-oriented paradigm), işlevsel paradigma (functional paradigm) ve mantıksal paradigma şeklindedir. Kandemir (2017, 2017:269-270) bu paradigmaları aşağıdaki gibi açıklamıştır:

1. Zorunlu paradigma: Bu paradigmaya göre kodlama; bilgisayara alt program, prosedür veya fonksiyon olarak isimlendirilerek verilen komut dizisidir. Daha sonra komut dizisi verilen isim kullanılarak çağrılır. Fortran (1954), Cobol (1959), Basic (1964), Pascal (1970), C (1970) gibi programlama dilleri bu paradigmayı temel alır.
2. Nesneye yönelik paradigma: Olayların benzetiminden geliştirilmiş bir paradigmadır. Gerçek dünyada yer alan her varlık için bir benzetim vardır. Smalltalk (1969), C++ (1983) ve Java (1995) nesneye yönelik paradigmayı temel alan programlama dilleridir.
3. İşlevsel paradigma: Programcının problemi soyutlamasıyla ilgili daha derin düşünmesine olanak tanıyan paradigmadır. Sorunun doğasının düşünülmesini özendirir. LISP (1958), ML(1973), Scheme (1975), Miranda (1982) ve Haskell (1987) programlama dilleri işlevsel paradigmayı temel alır.
4. Mantıksal paradigma: Matematikteki birinci derece mantık hesaplamaları üzerine kuruludur. Problem algoritmik çözümden ziyade mantıksal olarak ifade edilir. Prolog (1970) ve Curry (1997) programlama dilleri bu paradigmayı temel alır.

Bir programlama dili bir veya birden fazla paradigmayı destekleyebilir. C++, Python, PHP, Ruby ve C# birden çok paradigmayı temel alan programlama dillerine örnek olarak verilebilir (Kandemir, 2017:270).

### 2.1.1.1. Kodlama Eğitimi

Kodlama becerisinin sahip olduğu güç düşünüldüğünde, kodlama eğitiminin önemi daha iyi anlaşılmaktadır. Kodlama eğitimi 21. yüzyılda yazılım çalışmalarının sürdürülebilirliğini sağlayan önemli bir eğitim alanıdır (Kert ve Uğraş, 2009). Bununla birlikte bilgi toplumu olma yolunda yetişmiş bireyler önem taşımakta ve bu da nitelikli eğitim ile gerçekleşecektir (Sayın ve Seferoğlu, 2016). Resnick (2013) bir iş sahibi olmada kodlamanın önemli fırsatlar sunduğunu ve bilgisayar programcısı ihtiyacının karşılanmasında da kodlama eğitimine önem vermek gerektiğini belirtmiştir. Nitekim gelecekte mesleklerin büyük oranda, kodlama becerisine sahip olan ve bu beceriyi kullanan bireylere ihtiyaç duyacağı belirtilmektedir (Bidwell, 2013). Dolayısıyla kodlama mantığının öğrencilere erken yaşlarda verilmesi; bireyin diğer alanlardaki başarılarına katkı sağlayacağı (Karabak ve Güneş, 2013) gibi, bireyin geleceğe hazırlanmasında da önemli bir fırsat oluşturacaktır. Ayrıca kodlama eğitiminin bireysel gelişime katkıları dikkate alındığında, küçük yaşlardan itibaren verilmesi önem arz etmektedir (Kert ve Uğraş, 2009).

Phoenix Araştırma Enstitüsü (2011) yaptığı çalışmada 2020 için gerekli olabilecek becerileri sıralamış ve beceriler arasında en dikkat çekenlerden biri de, bilgi işlemsel düşünme becerisidir. Bilgi işlemsel düşünme; problemin çözümü, sistem tasarlanması ve insan davranışlarının anlaşılması için bilgisayar bilimi kavramlarından yararlanılması olarak tanımlanmıştır (Wing, 2006:33). Bilgi işlemsel düşünme becerisi sadece bilgisayar bilimi alanı için değil, tüm alanlar için sahip olunması gereken bir beceridir (Wing, 2006). Gülbahar ve Kalelioğlu (2018) bilgi işlemsel düşünmenin, bir problem çözme süreci olduğunu ifade etmiş ve içerisinde; problemlerin bilgi ve iletişim teknolojileri araçlarıyla çözümlenmesi, verilerin mantıklı olarak düzenlenmesi ve çözümlenmesi, verilerin model ve benzetimlerle sunulması, algoritmik düşünce ile çözümlerin otomatikleştirilmesi, problemlerin çözüme kavuşturulmasında kaynakların etkin kullanılması, bulunan çözümün transferi ve genelleştirilmesi gibi özellikleri barındırdığını belirtmiştir. Belirtilen bu beceriler düşünüldüğünde bilgi işlemsel düşünmenin önemi daha iyi anlaşılmaktadır. Yapılan araştırmalarda bilgi işlemsel becerilerin kazandırılmasında kodlama eğitiminin iyi bir yöntem olduğunu ifade etmektedir (Gülbahar ve Kalelioğlu, 2018).

Bilgi işlem düşünme becerisinin kazandırılmasında 4 yöntem bulunmaktadır (Weinberg, 2013). Bunlar; bilgisayarsız ortamlar, blok tabanlı ortamlar, oyun-robot programlama ve

disiplinler arası uygulamalar şeklindedir. Bilgisayarsız ortamlar, bilgisayar bilimi temel kavramlarının aktif öğrenme yaklaşımı ile etkinlik temelli verilmesi esasına dayanır (Kalelioğlu ve Keskinliç, 2017:159). Blok tabanlı ortamlar ise programlamaya yeni başlayan bireylere kodlama kavramlarının ve bilgi işlemsel düşünme becerisinin Scratch gibi blok tabanlı araçlarla verildiği ortamlardır. Oyun ve robot programlama ortamları, oyun ve robot tasarım araçları kullanılarak bilgi işlemsel düşünme becerisinin kazandırıldığı ortamlardır (Hardnett, 2008). Disiplinler arası ortamlar, özellikle robotik kitleri öğrencilere diğer disiplinlerle çalışma olanağı vermektedir (Kalelioğlu ve Keskinliç, 2017:176).

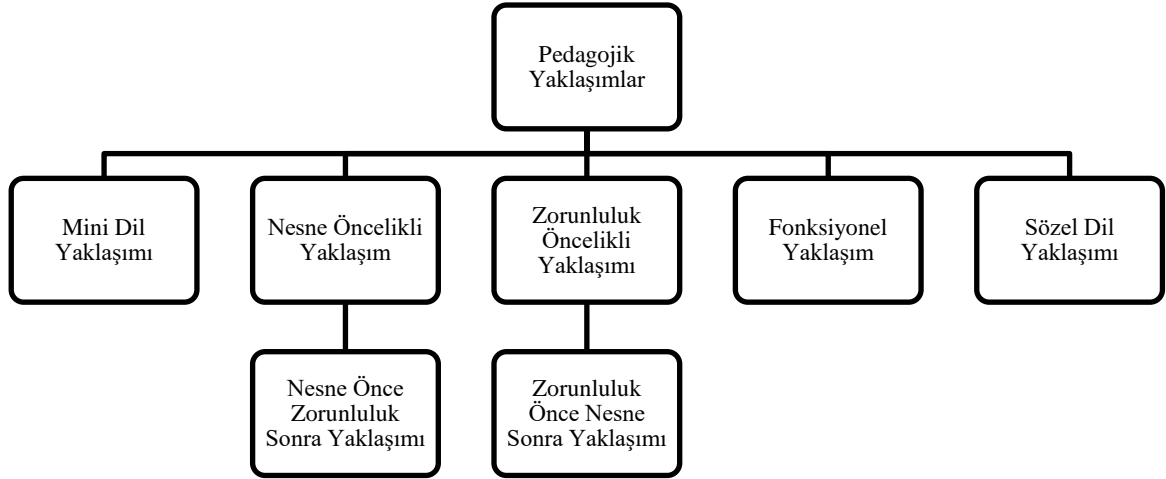
Kodlama eğitiminde çeşitli zorluklar bulunmaktadır. Saygıner ve Tüzün (2018) kodlama eğitimde en temel zorluğun, problemin algoritmasının oluşturulması olduğunu ifade etmiştir. Algoritma oluşturmanın bir problem çözme süreci olması, mantıksal düşünme içermesi, farklı açıdan bakabilme ve sorgulama becerileri gerektirmesi, programın karmaşık yapısı, ezberci yaklaşımlar gibi beceriler gerektirmesi bu zorluğun temel kaynağı olarak ifade edilmiştir. Özellikle kodlama öğrenmeye yeni başlayan bireylerde; problem çözmeye programlama dillerinin özelliklerinden nasıl faydalanılacağı, algoritmaların nasıl tasarlanacağı ve programın kodlanarak nasıl çalıştırılacağı gibi aşamaları içermesinden dolayı öğrenmekte zorluklar yaşanmaktadır (Valente, 1995). Saygıner ve Tüzün (2018) alanyazında, geçmişten günümüze kodlama eğitiminde karşılaşılan zorlukları sıralamışlardır. Bu zorluklar; algoritma oluşturma, programlama dilinin yapısı, öğrenme ve öğretme yöntemleri, hatayı bulma ve ayıklama, motivasyon, sayısal bilgi-beceri eksikliği, kod yazma- koda dökme, kodlamaya karşı tutum ve program geliştirme ortamı ile ilgili zorluklar şeklindedir.

Kodlama eğitiminde karşılaşılan zorlukların aşılmasına yönelik çeşitli çalışmalar yapılmıştır (Yükseltürk ve Altınok, 2017:244). Bu çalışmalarda; kodlama eğitiminde işbirliği unsurlarının kullanımının motivasyon sağladığı, kodlama sürecinde etkili ve sıklıkla geri bildirim verilmesinin başarıyı arttırdığı, etkileşimin artırılmasının başarıyı arttırmada etkili olduğu ve görselleştirme araçlarının kullanılmasının motivasyon ve başarı üzerinde olumlu etki sağladığı ortaya çıkmıştır (Lin ve Zhang, 2009; Nelson ve Rice, 2000; Naps et al, 2002; Malan ve Leitner, 2007). Ayrıca kodlama eğitiminde karşılaşılan zorlukların aşılmasında, metin tabanlı kodlama eğitiminin dışında görsel programlama araçlarının da kullanımı yaygınlaşmıştır. Cooper, Powers, McNally, Goldman, Proulx ve Carlisle (2006) görsel programlama araçlarını; blok tabanlı araçlar, hikâyeleştirilmiş algoritma araçları, akış şemalı algoritma araçları, ürün görselleştirme araçları ve aşamalı kodlama araçları şeklinde

sıralamıştır. Blok tabanlı araçlar, kodlamanın grafiklerin sürükle-bırak yöntemiyle yapılması esasına dayanan araçlar (Alice, Scratch, App inverter vb.), hikâyeleştirilmiş araçlar; karakterler aracılığıyla algoritma canlandırılmasını sağlayan araçlar (Jeroo, Microsoft Kodu, Game Lab vb.), akış şemalı algoritma araçları; kodlamanın işlem sırasına göre bağlantılı olarak yapılmasını sağlayan araçlar (Rapto, Flint vb.), ürün görselleştirme araçları; çıktıların görselleştirilmesine odaklanan araçlar (Lego, Mindstorms, RoboMind vb.), aşamalı kodlama araçları ise; kodlamaya yönelik kavram ve unsurları düzeye göre aşamalı verilmesi esasına dayalı araçlardır (ProfessorJ, RoboLab vb.) (Yükseltürk ve Altınok, 2017:246).

### **2.1.1.2. Metin tabanlı kodlama**

Günümüzde yaygın olarak kullanılan genel amaçlı programlama dillerinin çoğunluğu metin tabanlıdır. Bu nedenle kodlama eğitiminde metin tabanlı programla dillerinin öğretimi hala önemli bir alan olarak karşımıza çıkmaktadır (Kandemir, 2017:268-269). Metin tabanlı programlama dilleri çeşitli paradigmaları desteklemektedir. Bu paradigmlar zorunlu paradigma (Fortran, Cobol, Basic, Pascal, C, Ada), nesneye yönelik paradigma (Smalltalk, C++, Java), işlevsel paradigma (LISP, ML, Scheme, Miranda, Haskell) ve mantıksal paradigma (Prolog, Gödel, Curry) şeklindedir. Ayrıca birden fazla paradigmayı destekleyen, C++, Python, PHP, Ruby ve C# gibi programlama dilleri bulunmaktadır (Kandemir, 2017). Metin tabanlı kodlama öğretiminde, ilk programlama dilinin seçiminde çeşitli pedagojik yaklaşımlar bulunmaktadır. Farooq, Khan, Amed ve Abid (2014) tarafından 5 temel ve 2 alt yaklaşım ifade edilmiş ve Şekil 2'de sunulmuştur.



Şekil 2. Programlama dili seçimine yönelik yaklaşımlar

*Mini dil yaklaşımı:* Acemi programcıların programlamayı zorlanmadan öğrenmeleri için geliştirilmiş bir yaklaşımdır. Bu yaklaşımda, programlama öğretimi görsel ara yüzler ve robotlar kullanılarak yapıldığı için öğrenciler fonksiyonları ve program akışını rahatlıkla öğrenebilirler. Fakat değişken ve parametre gibi kavramlar olmadığı için, gerçek programlama yapısını öğrenmede başarı sağlanamaz. Logo ilk mini programlama diline örnek olarak verilebilir.

*Nesne öncelikli yaklaşım:* Temel odak noktası modülleştirme, kapsülleme, yeniden kullanılabilirlik, öz yineleme, yaratma, nesnelerin ve sınıfların manipülasyonu olan yukarıdan aşağıya bir yaklaşımdır. Nesne yönelimli yaklaşıma yönelik kavramların öğretilmesinde önemli bir yaklaşım olmakla birlikte, programın akışının öğretilmesinde yetersiz kalmaktadır.

- Nesne önce zorunluluk sonra yaklaşımı: Bu yaklaşıma Alice grafik programlama ortamı örnek olarak verilebilir. Bu yaklaşımda kaynak kod yazmadan, üç boyutlu sanal ortamlar oluşturulur. Dolayısıyla söz dizimi hatası düşük olur.

*Zorunluluk öncelikli yaklaşım:* Bu yaklaşımda problemlerin çözümünde fonksiyonların ve prosedür kümelerinin nasıl kullanılacağı öğretilir. Değişken tanımlama, değişkenlere değer atama, döngüler, karar yapıları, fonksiyonlarda parametre ve argüman gibi prosedürel

programlama kavramları üzerinde durulur. Bu yaklaşım geleneksel programlama yaklaşımlarından biridir (Kandemir, 2017:271).

- Zorunluluk önce nesne sonra yaklaşımı: Programlama dili öğretiminde en yaygın kullanılan yaklaşımdır. Bu yaklaşımda öncelikli olarak programlamanın temel kavramları verilir daha sonra nesne yönelimli programlamaya geçilir.

*Fonksiyonel yaklaşım:* Programlama mantığının matematik fonksiyonlarına benzer fonksiyonlarla ifade edildiği bir yaklaşımdır.

*Sözel dil yaklaşımı:* Temel programlama kavramlarını öğretmek için ana programlama dilinin alt kümeleri olarak tanımlanmaktadır. Bu yaklaşımda amaç basit söz dizimi ile program yazarak, öğrencilerin problem çözme becerilerine odaklanmalarını sağlamaktır.

Metin tabanlı programlamaya yeni başlayanlar için ilk dil olarak mini dil ve sözde dil yaklaşımlarını destekleyen programla dilleri kullanılmaktadır. Kandemir (2017), Python programlama dilinin mini dil ve sözde dil olmamasına rağmen programlama öğretimi için oldukça iyi bir dil olduğunu ifade etmiştir. Tamamen özgür ve ücretsiz olması, öğrenmesinin kolay olması, sağlam bir temele dayanması, güçlü kaynaklara sahip olması, internet üzerinde çalışabilir olması, iş bulma kolaylığı sağlaması, çok yönlü olması, güncel ve geniş topluluk desteğine sahip olmasında dolayı araştırmada python programlama dili kullanılmıştır (Aksoy, 2017).

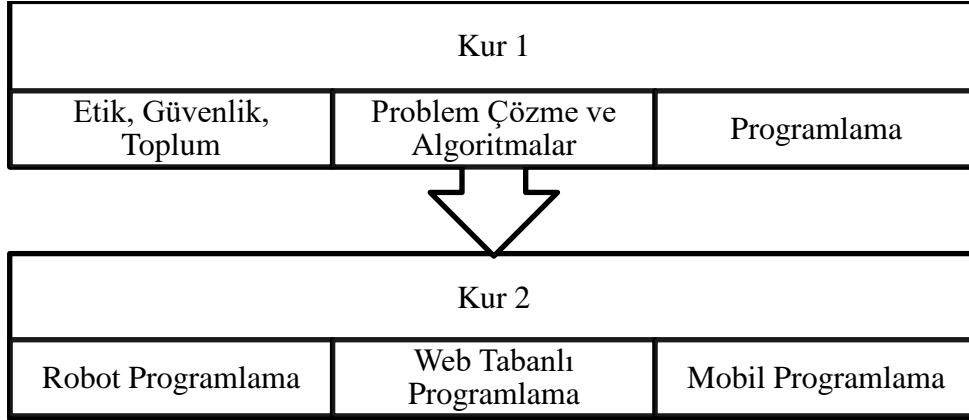
Python programlama dili Guido van Rossum tarafından geliştirilmiştir. İlk sürümü (Python 1.x) 1994 yılında, ikinci sürümü (Python 2.x) 2000 yılında ve halen en son sürüm olarak kullanılan Python 3.x sürümü ise 2008 yılında piyasaya sürülmüştür (Aksoy, 2017). Birçok ortamda kod yazılmasına olanak tanımaktadır. Bunlar; python komut penceresi, IDLE, Jupyter, PyCharm, Vin editörü ve online kod yazma ortamı (www.python.org) şeklindedir.

Python verilen kaynak kodunun tamamını makine diline çevirmek yerine, sadece sırası gelen satırı derleyerek çalıştırdığı için yorumlamalı bir dildir. Bu yönüyle; platformlardan bağımsız olma, dinamik olarak kodlamaya açıklık, küçük boyutlara sahip olma ve uygulama alanlarının dinamikliği gibi avantajları vardır (Aksoy, 2017).

### **2.1.1.3. Bilgisayar bilimi dersi**

Bilgisayar Bilimi Dersi (Kur 1, Kur 2) Talim ve Terbiye Kurulu Başkanlığının (TTKB) 31.08.2016 tarih ve 65 sayılı kararıyla Güzel Sanatlar ve Spor Liselerinde 2016-2017, diğer

ortaöğretim kurumlarında ise 2017-2018 eğitim ve öğretim yılından itibaren uygulanmaya başlamıştır. Bilgisayar Bilimi dersi haftada 2 ders saati olarak "Kur 1" den başlamak üzere kademeli olarak yürütülmektedir (TTKB, 2016). Kur 1 ve kur 2 üniteleri Şekil 3'te sunulmuştur.



Şekil 3. Bilgisayar bilimi (kur 1, kur 2) öğretim programında yer alan üniteler

Şekil 3'te görüldüğü gibi her kur 3 ünitelerden oluşmaktadır. Kur 2'nin uygulanabilmesi için öğrencilerin kur 1'i alması gerekmektedir. Ayrıca kur 2 uygulanırken belirtilen 3 ünitelerden 2 tanesinin veya 1 tanesini seçilebileceği belirtilmiştir.

Bilgisayar bilimi öğretim programı ile öğrencilerde çeşitli becerilerin kazandırılması hedeflenmektedir. Bu beceriler; bilgi işlemsel düşünme, eleştirel düşünme, algoritmik düşünme, matematiksel düşünme, yaratıcı düşünme, problem çözme, algoritma tasarlama, yazılım geliştirme, etkili iletişim kurma, karar verme, çıkarımda bulunma ve iş birliği şeklindedir (TTKB, 2016).

### 2.1.2. Oyunlaştırma

Kullanımı eskilere dayanan oyunlaştırma kavramı; eğitim, sağlık, ticaret vb. birçok alanda hızla yaygınlaşarak kullanılmaktadır. Bu yaygınlaşmanın temelinde teknolojiye gelişimin ve oyunların ortamlardaki başarısının olduğu söylenebilir (Deterding, 2012). Özellikle web ve mobil ortamlarda oyun bileşenlerinin kullanılması ve bağlılığın sağlanması oyunlaştırılmış ortamlara dönüşümü sağlamıştır (Deterding et al., 2011).



Oyunlaştırma kavramı ilk olarak oyun tasarımcısı Nick Pelling tarafından 2003 yılında kullanılmasına rağmen şu anki anlamıyla 2010 yılında benimsenmiştir (Werbach ve Hunter, 2012). Benimsenmesiyle birlikte oyunlaştırmayla ilgili alanyazında birçok tanım yapılmıştır. Werbach ve Hunter (2012:26) oyunlaştırmayı, oyun bileşenlerinin ve oyun tasarım tekniklerinin oyun olmayan içeriklerde kullanılması olarak tanımlamıştır. Zichermann ve Cunningham (2011:16) oyunlaştırmayı, oyun düşüncesinin ve oyun mekaniklerinin kullanıcının ilgisinin çekilmesi ve problemlerinin çözülmesi amacıyla kullanılması süreci olarak tanımlamıştır. Bir başka tanıma göre oyunlaştırma, oyun tasarım bileşenlerinin oyun dışı ortamlarda kullanılması olarak ifade edilmiştir (Deterding et al., 2011). Kapp (2012:10) ise oyunlaştırmayı; oyun tabanlı mekanikler, oyunsal düşünme ve estetikle insanların motive edilmesi, bağlılıklarının sağlanması, öğrenmenin desteklenmesi ve problemlerin çözümü amacıyla kullanılması olarak tanımlamıştır (Kapp, 2012). Alanyazında oyunlaştırmayla ilgili yapılan tanımlar incelendiğinde ortaya çıkan kavramlar; bireyin içsel motivasyonunun sağlanarak ortama bağlılığının sağlanması (Marczewski, 2015: 201), davranış ve eylemlerin ortaya çıkması için motive edilmesi (Kapp, 2012:12), bireylerin problem çözmeye cesaretlendirilmesi (Werbach ve Hunter, 2012:31), oyunlaştırma ile ortamın bireyin olumlu duygusal tepkiler vereceği estetik bir hale dönüştürülmesi (Marczewski, 2015: 40), oyun düşüncesi yani oyun ve oyuna benzer yaklaşımların kullanılması (Marczewski, 2015:14), oyun dışı bağlamların yani gerçek yaşam problemlerinin oyunlaştırılması (Deterding et al., 2011), oyun mekaniklerinin (rozet, seviye, puan vb.) kullanılması ve birey öğrenmesinin desteklenmesi şeklindedir (Kapp, 2012:15).

Kapp (2012) oyunlaştırmının belirli öğrenme hedeflerine ulaşmada etkili olarak kullanılabileceğini fakat her öğrenme durumunda tüm içeriğin ya da öğrenme deneyimlerinin tamamen oyunlaştırılmasının bir anlam ifade etmeyeceğini dile getirmiştir. Özellikle bireyin içerikte ilerlemesinin teşvik edilmesinde, motivasyonun sağlanmasında, davranış değişikliğinin sağlanması ve yeniliğe karşı harekete geçirilmesi amacıyla etkili bir şekilde kullanılabileceğini belirtmiştir.

Oyunlaştırma kavramının daha iyi anlaşılması için oyun ve oyunlaştırma arasındaki farklılığın iyi anlaşılması gerekmektedir. Yılmaz (2015:27) oyun kavramını; *"kendi başına kuralları belirlenmiş bir sistemde insanların eğlenmesi, sosyalleşmesi ve öğrenmesi gibi amaçları, belli hedefleri ve kazanma ile kaybetme gibi sonuçları olan bir uygulamadır."* şeklinde tanımlamıştır. Oyunlaştırmayı ise; *"oyunsu tasarımların ve oyun mekaniklerinin oyun dışı bir sürece eklendiği bir disiplindir."* şeklinde ifade etmiştir. Bu iki tanım

incelendiğinde oyunun bir uygulama, oyunlaştırmanın uygulama dışında bir süreçler topluluğu olduğu görülmektedir. Yılmaz (2015:28) oyun ve oyunlaştırma arasındaki farklılıkları Tablo 1'deki gibi özetlemiştir.

Tablo 1

*Oyun ve Oyunlaştırma Arasındaki Farklar*

Oyun	Oyunlaştırma
Objeler ve onları kullanma kuralları olur.	Kurallar daha çok görevleri tamamlamak içindir.
Kazanma ve kaybetme olur.	Kaybetme genelde yoktur, daha çok aksiyon aldırma için desteklenir.
Bir hikâyesi ve ona göre tasarımı olur.	Asıl uygulandığı süreç neyse onu destekler.
Tümüyle tasarlanmalı ve tek başına çalışmalıdır.	Var olan sürece entegre olur, ayrı çalışır.
Üretimi çok pahalı ve karmaşıktır.	Entegrasyonu basittir.

Alanyazında oyunlaştırmayla ilgili çeşitli sınıflamalar yapılmıştır. Marczewski (2015:15) oyunlaştırmanın içsel ve dışsal olarak iki türünün olduğunu ifade etmiştir. Dışsal oyunlaştırmayı; puan, rozet ve ilerleme çubuğu gibi oyun bileşenlerinin sisteme eklenerek oluşturulduğunu, içsel oyunlaştırmanın ise motivasyonu ve bireyin katılımını sağlamak için davranışların tasarlanmasıyla oluşturulduğunu ifade etmiştir. Benzer bir sınıflandırma ise Kapp (2012) tarafından yapılmıştır. Kapp (2012) oyunlaştırmanın yapısal ve içerik oyunlaştırma olarak iki türünün olduğunu ifade etmiştir. Yapısal oyunlaştırmada içerik üzerinde hiçbir değişiklik yapılmadan bireyin oyun bileşenleri ile motive edilerek içeriğin kullanımı sağlanır. İçerik oyunlaştırmada ise içerik oyuna benzer bir hale dönüştürülür. Kapp (2012) oyunlaştırılmış ortamlarda iki oyunlaştırma türünün ayrı ayrı veya birlikte kullanılabileceğini belirtmiş fakat en etkili yöntemin ise yapısal ve içerik oyunlaştırmanın birlikte kullanılmasıyla olacağını ifade etmiştir. Werbach ve Hunter (2012) göre ise oyunlaştırmanın içsel, dışsal ve davranış değiştirici oyunlaştırma olarak üç türü vardır. İş dünyası ve pazarlama alanlarına yönelik yapılan bu sınıflandırmada; içsel oyunlaştırma kurum içi performansın artırılmasına yönelik bir oyunlaştırma türü iken, dışsal oyunlaştırma

ise kurum ile müşteri arasındaki iletişimi arttırmaya yönelik oyunlaştırma türüdür. Davranış değiştirici oyunlaştırma ise toplumsal refah için yararlı alışkanlıkların kazandırılmasına yönelik kullanılan oyunlaştırma türüdür.

### **2.1.2.1. Oyunlaştırmanın kuramsal dayanakları**

Bu bölümde oyunlaştırmanın kuramsal dayanaklarına yer verilmiştir. Bu kuramsal dayanaklar: İçsel (intrinsic) ve dışsal (extrinsic) isteklendirme, öz belirleme kuramı (self determination theory), akış (Flow) kuramı, fogg davranış modeli (fogg behavior model) ve sosyal öğrenme kuramı (social learning theory) şeklindedir.

*İçsel ve dışsal motivasyon:* Oyunlaştırmanın dayandığı kuramsal dayanaklardan ilki motivasyona dayalı kuramlardır. Motivasyon kavramı Latince "motivus" olarak kullanılan ve eylem sağlayıcı anlamına gelen bir kelimedir (Werbach ve Hunter, 2012).

Alanyazına bakıldığında psikolojik olarak motivasyonun içsel (intrinsic) ve dışsal (extrinsic) motivasyon olarak iki gruba ayrıldığı görülmektedir. Zichermann ve Cunningham (2011:26) içsel motivasyonu, öz benliğimizden türetilen ve dış dünyayla bağlantısı olmayan motivasyon türü olarak ifade etmiştir. Dışsal motivasyonu ise içsel motivasyonun tersine dış dünya tarafından yönlendirilen motivasyon türü olarak belirtmiştir. Werbach ve Hunter (2012, 130-131) içsel motivasyonu dış kaynaklı bir beklenti olmadan kendi iyiliği için eyleme geçme olarak tanımlamışken, dışsal motivasyonu ise kendi dışındaki para, statü, güç, ödül ve patronun isteği gibi sebeplerden dolayı eyleme geçmek olarak tanımlamıştır. İçsel motivasyon içten gelen itici bir güçtür ve dışsal motivasyon gibi dış kaynaklı değildir (Marczewski, 2015). İçsel motivasyon ile dışsal motivasyon arasındaki ilişkinin iyi kavranması gerekmektedir (Kapp, 2012). Burke (2014) oyunlaştırmada dışsal motivasyonu sağlayıcı ödüller yerine içsel motivasyonu sağlayan ödüllerin öncelikli olarak kullanıldığını belirtmiştir. Çünkü oyunlaştırma tasarımında bireyin sadece dışsal motivasyonla eyleme geçirilmesinin tasarımın başarısına engel olacağı dolayısıyla tasarım yapılırken içsel motivasyonunda bulunması gereklidir (Yılmaz, 2015). Oyunlaştırılmış ortamlarda ödül gibi dışsal motivasyon kaynaklarının kullanılması, oyunlaştırma sürecinin amaçladığı davranış değişikliğinin önüne geçecek ve oyunlaştırılmış ortamda istenen davranış değişikliği sağlayamayacaktır (Yılmaz, 2015).

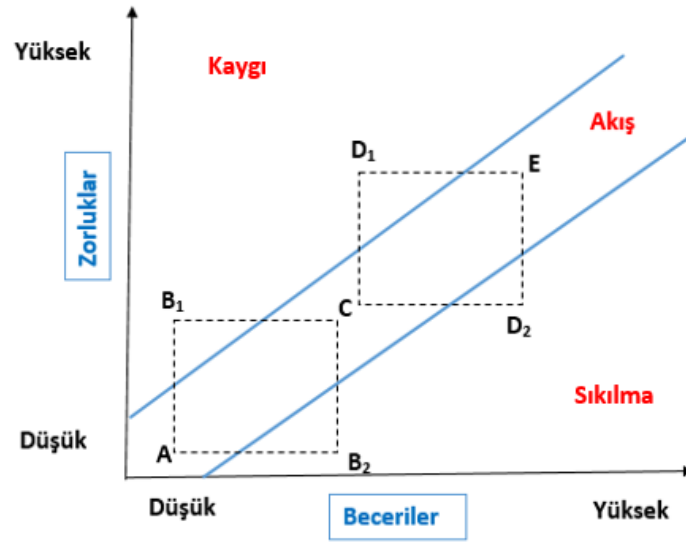
*Öz belirleme kuramı (Self determination theory):* Oyunlaştırmanın kuramsal dayanaklarından bir diğeri de öz belirleme kuramıdır (self determination theory). Deci,

Connell ve Ryan (1989) göre öz belirleme bireyin kendi davranışını seçmesi ve yönetmesi olarak ifade edilmiştir. Öz belirleme kuramı bireyin görevi yerine getirirken motivasyonunu açıklayan makro bir teoridir. Kuramın amacı toplumun bir bütün olarak sağlıklı bir şekilde gelişmesi için gerekli olan faktörlerin belirlenmesidir (Deci ve Ryan, 2000). Bu kurama göre bireyin istek ve motivasyonu, sosyal ve kültürel gibi çevresel faktörlerden etkilenir (Kim, Song, Lockee ve Burton, 2018). Werbach ve Hunter (2012) bireyin kendini geliştirmesinde istek ve içsel tetikleyicilerle birlikte çevresel faktörlerin de destekleyici olması gerektiğini ifade etmiştir. Aksi durumda içsel motivasyonun engelleneceğini belirtmiştir.

Öz belirleme kuramı; yetkinlik ihtiyacı (needs for competence), ilişkili olma (relatedness) ve özerklik (autonomy) olmak üzere 3 faktörden oluşur (Deci ve Ryan, 2000). Yetkinlik ihtiyacı; bireyin karşılaştığı durumlarla başa çıkabilme ve öğreneceğine inanma hissini, ilişkili olma; bireyin başka bireylerle ilişkide olduğunu hissetmesini, özerklik ise bireyin kontrolün kendisinde olması ve davranışlarını yönlendirebilme hissini ifade eder. (Kapp, 2012).

Kapp (2012) öz belirleme kuramını oyunlaştırılmış öğrenme ortamlarında; seçim ve kontrol duygusunun verilmesi, zorluklarla başa çıkma ve hedefi gerçekleştirme noktasında güven sağlama, içerik ve uzmanlaşmak için açık bir yol sunma, aşamalı hedeflerin başarılmasında ödüllendirme, sosyal etkileşim ile bireyler arasından bağlılığın sağlanması gibi amaçlarla kullanılabileceğini belirtmiştir.

*Akış (Flow) Kuramı:* Oyunların başarısının merkezinde akış denilen bir fikir vardır (Zichermann ve Cunningham, 2011:16). Akış kavramı, bireyin dışsal motivasyon sağlayıcı hiçbir unsura gerek kalmadan görevini yüksek içsel motivasyon ile yerine getirmesi durumu olarak tanımlanmıştır (Csikszentmihalyi, 1991). Tanımdan da anlaşılacağı üzere, akışın sağlanmasında içsel motivasyon önemlidir. İçsel motivasyonun yüksek tutulması ve bununla birlikte akışın sağlanmasında, bireyin kapasitesi ile görev arasında dengenin olması gerekmektedir (Csikszentmihalyi, 1991). Verilen görev ile bireyin kapasitesi arasında denge sağlandığı takdirde akış sağlanmış olur (Csikszentmihalyi ve Rathunde, 2014).



Şekil 4. Akış deneyimi (Csikszentmihalyi ve Rathunde, 2014:28)

Şekil 4'te görüldüğü gibi akış, sıkılma ve kaygı arasında bir dengedir. A, C ve E denge hali yani akış durumunu ifade eder. B1 ve D1 kaygı durumunu göstermektedir. Bu durum bireyin zorlandığını dolayısıyla yeni beceriler kazandırılarak akış sağlanması gerekliliğini göstermektedir. B2 ve D2 ise sıkılma durumunu ifade eder. Bu durumda bireyin kapasitesine uygun zorluklar sağlanarak birey akış durumuna getirilmelidir. Bireye zor görevler verildiğinde kaygılanır aksi durumda ise bireyde sıkılma durumu oluşur (Csikszentmihalyi ve Rathunde, 2014). Dolayısıyla oyunlaştırılmış öğrenme ortamları tasarlanırken bireyin kapasitesi dikkate alınarak, verilen görevler bireye uygun olmalıdır. Bireyin kaygılanmasının ve sıkılmasının önüne geçilerek, oyunlaştırılmış ortamda akışının sağlanması gerekmektedir. Bir başka husus ise oyunlaştırılmış öğrenme ortamlarında akış hissinin sağlanmasında içsel motivasyonun gerekliliğidir. Bu sebeple oyunlaştırılmış ortamların tasarlanmasında içsel motivasyon unutulmamalıdır.

*Fogg davranış modeli:* Oyunlaştırmanın kuramsal dayanaklarından biri de fogg davranış modelidir. Bu model davranış değişikliğine, neyin neden olduğunu açıklayan bir modeldir (Yılmaz, 2015:35). Fogg davranış modeline göre davranış değişikliğinin meydana gelmesinde; motivasyon (motivation), yetenek (ability) ve tetikleyiciler (triggers) olarak ifade edilen 3 faktörün bir araya gelmesi gerekmektedir (Fogg, 2009). Yılmaz (2015) oyunlaştırma tasarımının çalışmadığı durumlarda bu faktörlerin kontrol edilerek gerekli düzenlemelerin yapılması gerektiğini belirtmiştir.

*Sosyal öğrenme kuramı:* Sosyal öğrenme kuramı; bireyin başkalarını, başkalarının davranışlarını ve bu davranışların sonuçlarını gözlemleyerek öğrenebileceğini vurgulamaktadır (Kim et al., 2018b). Albert Bandura tarafından geliştirilen kuram; dikkat, hatırd tutma, davranışı meydana getirme ve motivasyon olmak üzere 4 temel süreci kapsamaktadır (Senemoğlu, 2013:229). Dikkat süreci; bireyin model alacağı davranışa dikkat ederek algıladığı aşamayı, hatırd tutma süreci; modellenen davranışın daha sonra hatırlanmak üzere kodlanarak bellekte saklandığı aşamayı, davranış meydana getirme süreci; öğrenilenin performansa dönüştürüldüğü aşamayı, motivasyon süreci ise öğrenilenin performansa dönüştürülmesinin sağlandığı aşamayı ifade eder (Senemoğlu, 2013). Kim et al. (2018b) oyunlarda, oyuncuların diğer oyuncularla ve oyunda yer alan karakterlerle etkileşime girerek ve onları modelleyerek öğrenebileceklerini belirtmiştir. Bu nedenle bireye kazandırılması düşünülen davranışların, avatarlar kullanılarak sanal ortamlarda sunulabilir (Kapp, 2012).

### 2.1.2.2. Oyunlaştırma tasarım modelleri ve bileşenleri

Oyunlaştırılmış ortamların tasarlanması esnasında oyun bileşenlerinin kullanımına yönelik çeşitli modeller bulunmaktadır. Werbach ve Hunter (2012:78) oyunlaştırma tasarlanmasında oyun bileşenlerinin kullanımına yönelik 3 aşamalı piramitsel bir yapı ortaya koymuştur. Ortaya konulan yapı Şekil 5'te sunulmuştur.



Şekil 5. Werbach ve Hunter (2012:78) oyunlaştırma tasarım modeli

Şekil 5 incelendiğinde piramidin birinci seviyesinde dinamikler yer almaktadır. Dinamikler oyunlaştırılmış ortamdaki oyun kurallarından bağımsız tasarım çözümleridir (Yılmaz, 2015:132). Dinamikler oyunlaştırılmış ortamın temelini oluşturur. Oyunlaştırılmış ortamın kurgusunu ve kurallarını oluşturan, kullanıcının etkileşime girmediği soyut unsurlardır (Werbach ve Hunter, 2012:130). Mekanikler ve bileşenler, dinamikler dikkate alınarak oluşturulur. Werbach ve Hunter (2012:78) tarafından ifade edilen dinamikler aşağıda sunulmuştur.

- *Kısıtlamalar*: Oyunlaştırılmış ortamdaki sınırlılıklar ve zorunlu değişimlerdir.
- *Duygular*: Oyunculara uyandırılmak istenen merak, rekabet, hayal kırıklığı ve mutluluk gibi duygulardır.
- *Öyküleştirme*: Oyunlaştırılmış ortamın tutarlı ve devam eden hikâyesidir.
- *İlerleme*: Oyunlaştırılmış ortamda oyuncunun büyümesi ve gelişimidir.
- *İlişkiler*: Oyunlaştırılmış ortamdaki dostluk, statü ve fedakârlık gibi sosyal etkileşimlerdir.

Şekil 5'te sunulan piramidin ikinci seviyesinde yer alan mekanikler oyunlaştırılmış ortama oyuncuların katılımını ve eylemlerin ilerlemesini sağlayan temel süreçlerdir. Her bir mekanik, dinamiklerden birine veya birden fazlasına ulaşmanın yoludur (Werbach ve Hunter, 2012:79-80). Werbach ve Hunter (2012:79) tarafında ifade edilen mekanikler aşağıda sunulmuştur.

- *Zorluklar*: Çaba harcanarak çözülmesi gereken bulmaca veya görevlerdir.
- *Şans*: Rastlantısal olarak ortaya çıkan bileşenlerdir.
- *Rekabet*: Bir oyuncu ve grubun kazanması durumunda diğerinin kaybetmesidir.
- *İşbirliği*: Oyuncuların amaçlarına ulaşmada birlikle çalışmalarınıdır.
- *Geri bildirim*: Oyuncunun nasıl oynadığıyla ilgili bilgilendirilmesidir.
- *Kaynak edinimi*: Yararlı ve biriktirilebilen öğeler edinilmesidir.
- *Ödüller*: Bazı eylem ve davranışlardan dolayı fayda sağlanmasıdır.
- *İşlemler*: Doğrudan veya dolaylı olarak oyuncular arasında yapılan alış verıştır.
- *Sıra*: Farklı oyuncuların oyuna sıralı katılımıdır.

- *Kazanma durumu:* Bir oyuncunun veya grubun kazanması, berabere kalması ve kaybetmesiyle ilgili kavramlardır.

Şekil 5'te sunulan piramidin üçüncü seviyesinde yer alan bileşenler, dinamik ve mekaniklerin özel halleridir. Her bir bileşen birden fazla dinamiğe veya mekaniğe bağlanabilir. Werbach ve Hunter (2012:80) tarafında ifade edilen bileşenler aşağıda sunulmuştur.

- *Başarılar:* Belirlenen hedefler yani başarılması gerekenlerdir.
- *Avatarlar:* Oyuncuların görsel karakterlerle temsilidir.
- *Rozetler:* Başarıların görsel öğelerle ifadesidir.
- *Zorlu Mücadele:* Üst seviyelerde verilen zorlu görevlerdir.
- *Koleksiyonlar:* Rozet vb. öğelerin biriktirilmesidir.
- *Savaş:* Kısa ömürlü mücadelelerdir.
- *İçerik kilidi açma:* Oyuncuların belirli hedefler veya görevleri tamamlamaları durumunda ortaya çıkan görünümdür.
- *Hediyeler:* Diğer oyuncularla kaynakların paylaşım fırsatlarıdır.
- *Lider tahtası:* Oyuncu ilerlemelerinin ve başarılarının görsel olarak sunulmasıdır.
- *Seviyeler:* Oyuncuların ilerleme sürecinde geçmesi gereken aşamalardır.
- *Puan:* Oyuncu ilerlemelerinin sayısal olarak temsil edilmesidir.
- *Görevler:* Amaçlar ve ödüller ile önceden tanımlanmış zorluklardır.
- *Sosyal grafikler:* Oyuncuların oyun içerisinde sosyal ağ olarak gösterilmesidir.
- *Takımlar:* Ortak amaç doğrultusunda birlikte çalışan oyuncular grubudur.
- *Sanal eşyalar:* Gerçek para veya değerinde olan varlıklardır.

Werbach ve Hunter (2012) oyunlaştırma tasarımı yapılırken belirli aşamaların takip edilmesi gerektiğini ifade etmiştir. D6 olarak da ifade edilen oyunlaştırma tasarımının aşamaları Şekil 6'da sunulmuştur.





Şekil 6. D6 oyunlaştırma tasarımı aşamaları (Werbach ve Hunter, 2012)

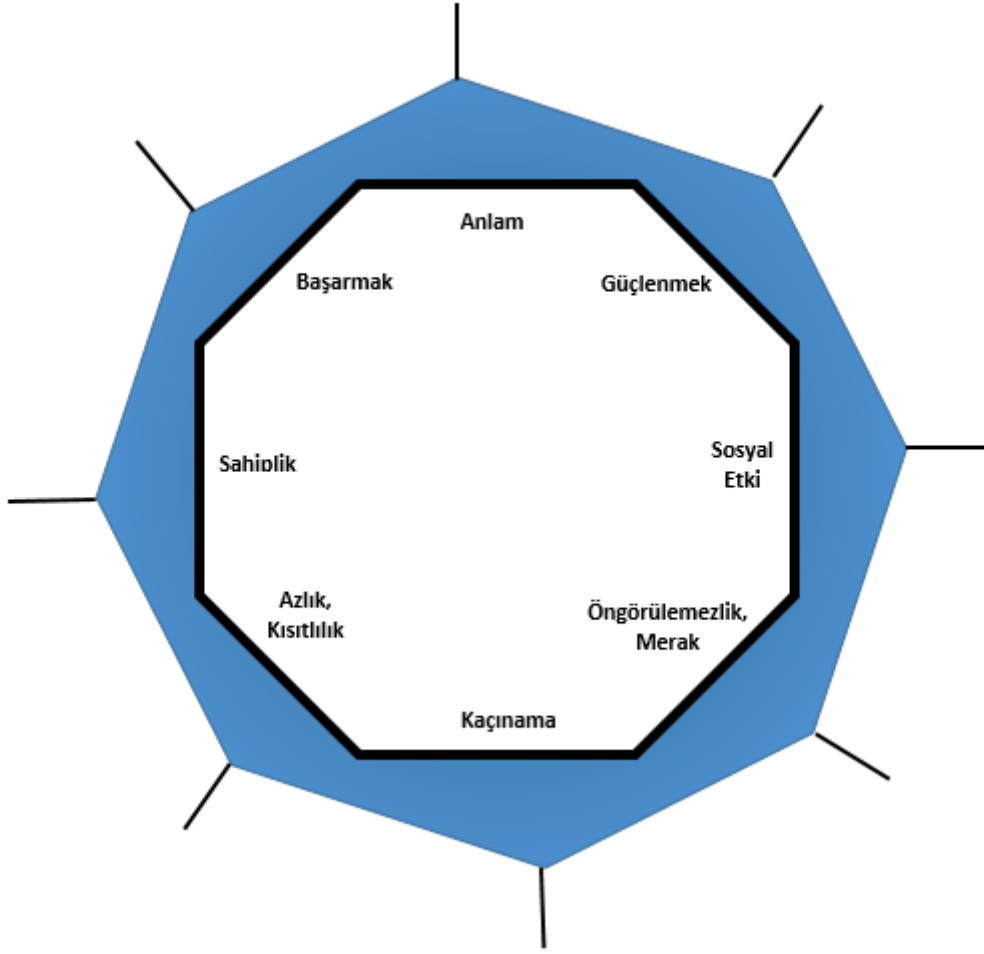
Şekil 6'da sunulan oyunlaştırma tasarım modelinin ilk aşaması oyunlaştırma kullanımındaki genel hedeflerin açıkça ortaya konduğu "Hedefleri belirleme" aşamasıdır. İkinci aşama oyunlaştırma tasarımının genel hedefleri kapsamında ortaya çıkması düşünülen davranışların belirlendiği "Hedef davranışları belirleme" aşamasıdır. Üçüncü aşama oyunlaştırma tasarımında kullanılacak oyuncu tiplerinin ortaya konduğu "Oyuncuları tanımla" aşamasıdır. Dördüncü aşama oyunlaştırma tasarımında kullanılacak oyunlaştırma bileşenlerinin nasıl kullanılacağına belirlendiği "Etkinlik döngülerini planlama" aşamasıdır. Beşinci aşama ise oyunlaştırma tasarımına eğlence unsurlarının dâhil edildiği "Eğlenceyi unutma!" aşamasıdır. Bu aşamada tasarımda hangi eğlence türlerinin kullanılacağına belirlendiği aşmadır. Altıncı ve son aşama ise tasarımda kullanılacak araçların belirlendiği "Uygun araçları kullanma" aşamasıdır.

Oyunlaştırma tasarımıyla ilgili bir diğer model Kapp (2012) tarafından ortaya konmuştur. Kapp (2012) oyunlaştırmanın ince ayrıntılarının bulunduğunu ve dikkatli bir şekilde tasarlanması gerektiğini belirtmiştir. Puan, ödül, rozet vb. bileşenleri kullanarak yapılan uygulamaların oyunlaştırma olmadığını aksine oyunlaştırmanın en az yararlı gözükken bileşenleri olduğunu ifade etmiştir (Kapp, 2012). Kapp (2012) oyunlaştırmayı yapısal ve içerik olarak iki kısma ayırmakla birlikte, oyunlaştırma tasarımı unsurlarını; oyunsal düşünme, oyun mekanikleri ve estetik olarak ifade etmiştir. Oyunlaştırmanın, bu üç unsurun birleşiminden oluştuğunu belirtmiştir. Oyunsal düşünme unsuru günlük yaşam deneyimlerini oyunlaştırılmış bir hale dönüştürürken, mekanik ve estetik unsurları ise

motivasyon sağlama, adanmışlık sağlama, problem çözüme ve öğrenmeyi destekleme gibi işlevleri yerine getirir (Sezgin, Bozkurt, Yılmaz ve Linden, 2018).

Zichermann ve Cunningham (2011:35) tarafından sunulan oyunlaştırma tasarım modelinde tasarım unsurları 3 gruba ayırmıştır. Bunlar; mekanikler, dinamikler ve estetiklerdir. MDE olarak da ifade edilen modelde, mekanikler oyunlaştırma tasarımının işlevsel kısmını oluşturmakla birlikte oyunlaştırma tasarımında tasarımcıya kontrol sağlar. Ayrıca oyuncu davranışlarına rehberlik olanağı sunar. Zichermann ve Cunningham (2011) tarafından önemli olarak değerlendirilen mekanikler; puanlar, seviyeler, rozetler, meydan okuma ve alıştırma şeklindedir. Zichermann ve Cunningham (2011) tarafından sunulan modelin ikinci unsuru olan dinamikler ise oyuncuların mekaniklere verdikleri tepkileri ifade eder. Estetik ise oyuncunun oyunda kendini nasıl hissettiğini ifade etmekle birlikte, oyuncunun mekanik ve dinamiklerle etkileşime girdiklerinde oluşan tümleşik sonuçtur (Zichermann ve Cunningham, 2011).

Chou (2015) oyunlaştırmayı; "oyunlarda bulunan tüm eğlenceli ve ilgi çekici unsurları türetme ve onları gerçek dünyaya ya da üretken faaliyetlere uygulama sanatıdır" şeklinde tanımlamıştır. Tanımdan da anlaşılacağı üzere Chou (2015) oyunlaştırma tasarımında insan odaklı tasarımı ön plana çıkarmaktadır. Chou (2015) tarafından insan odaklı olarak ortaya konulan ve Octalysis olarak isimlendirilen oyunlaştırma tasarım modeli, 8 oyunlaştırma unsurundan oluşmaktadır. Octalysis tasarım modeli Resim 1'de sunulmuştur.



Resim 1. Chou (2015) octalysis tasarım modeli

Resim 1 incelendiğinde oyunlaştırma tasarım modelinin unsurları sekizgenin köşelerine yerleştirilmiş ve aşağıdaki gibi ifade edilmiştir (Chou, 2015).

- *Epik Anlam (Meaning)*: Bireyin kendinden daha büyük bir anlam arayışının motivasyon kaynağı olarak kullanıldığı unsurdur.
- *Gelişim ve başarı (Accomplishment)*: Kendini geliştirmenin, başarmanın ve zorlukları aşmanın motivasyon kaynağı olarak kullanıldığı unsurdur.
- *Yaratıcılığın güçlendirilmesi ve geribildirim (Empowerment)*: Bireyin yaratıcılığının sağlanması, bireye özerklik sağlanması ve geribildirimlerle desteklenmesi motivasyon kaynağıdır.
- *Sahip olma (Ownership)*: Sahip olma ve biriktirme isteğinin motivasyon kaynağı olarak kullanıldığı unsurdur.

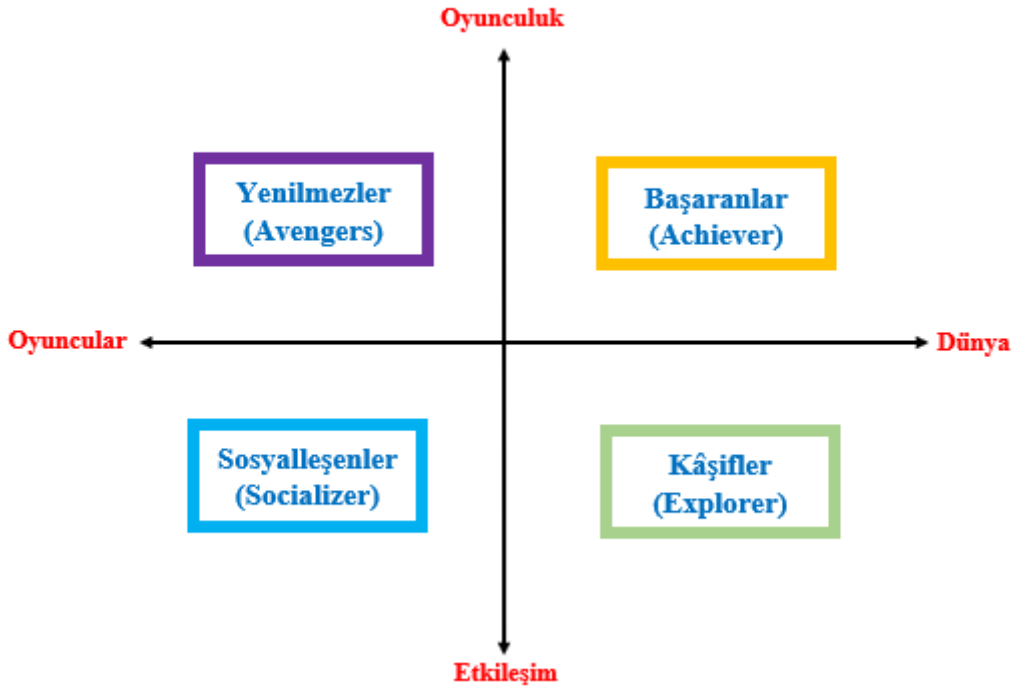
- *Sosyal etki ve ilişki (Social influence)*: Arkadaşlık, rekabet vb. dürtülerin motivasyon kaynağı olarak kullanıldığı unsurdur.
- *Azlık ve sabırsızlık (Scarcity)*: Bir şeye sahip olmayı isteme dürtüsünün motivasyon kaynağı olarak kullanılmasına dayanır.
- *Tahmin edilemezlik ve merak (Unpredictability)*: Daha sonra ne olacağını bulma esasına dayanır.
- *Kayıp ve kaçınma (Avoidance)*: Motivasyonun sağlanmasında olumsuz durumlardan kaçınma dürtüsünün kullanıldığı oyunlaştırma unsurudur.

Chou (2015) tarafından sunulan çerçevenin üst kısmında yer alan oyunlaştırma unsurları pozitif motivasyon sağlayan unsurlar olarak ifade edilirken, alt kısımda yer alan unsurlar ise negatif motivasyon sağlayıcı unsurlar olarak ifade edilmiştir. Ayrıca oyunlaştırma unsurlarında çerçevenin sağ tarafında kalanlar; beynin sağ kısmında yer alan yaratıcılık, kendini geliştirme vb. içsel motivasyon kaynaklı becerileri, sol tarafta kalanlar ise dışsal motivasyon becerileri ifade etmektedir (Economou vd, 2015).

### **2.1.2.3. Oyuncu tipleri**

Oyunlaştırılmış öğrenme ortamlarında ortama dâhil olacak bireylerin, bireysel farklılıklarının dikkate alınarak tasarımın yapılması gerekmektedir. Oyunlaştırılmış öğrenme ortamının azami derecede tüm bireyleri dikkate alarak tasarlanması ortamın başarısını arttıracaktır. Yılmaz (2015:110) oyunlaştırılmış ortamlarda her oyuncunun katılımının, beklentisinin ve motivasyonunun farklı olduğunu bilmesi gerektiğini dolayısıyla oyuncu kitlesinin iyi analiz edilmesi gerektiğini ifade etmiştir.

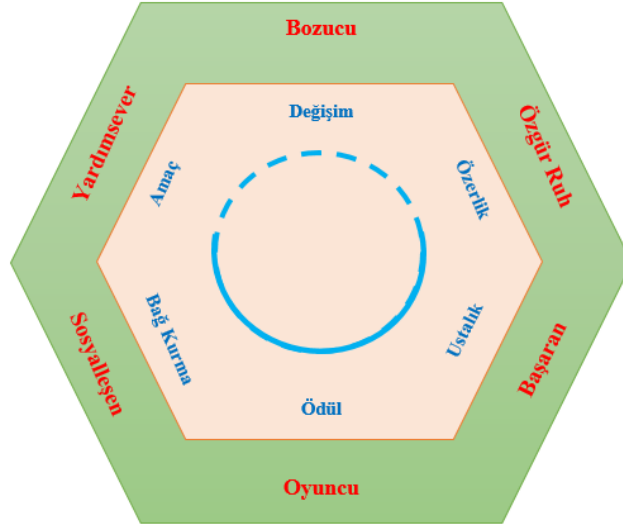
Oyuncu tipleriyle ilgili çeşitli sınıflamalar yapılmıştır. Bartle (1996) oyuncu tiplerini yenilmezler (*avengers*), başarılar (*achiever*), sosyalleşenler (*socializer*) ve kâşifler (*explorer*) olarak dört gruba ayırmış ve Resim 2'de sunulmuştur.



Resim 2. Bartle (1996) oyuncu tipleri

Yenilmezler için önemli olan oyunda karşı tarafı yenmektir. Dolayısıyla oyunlaştırılmış öğrenme ortamı tasarlanırken bu oyuncu tipi için oyun mekaniklerinden rekabet, yarışma ve ödülün mutlaka kullanılması gerekir. Başaranlar, diğer oyuncularından çok görevlerine odaklanırlar. İlerleme durumlarını görmek ve verilen görevleri hızlıca tamamlamak isterler. Sosyalleşenler, diğer oyuncularla işbirliği içerisinde girmek, yardımlaşmak ve ekip olarak ilerlemek isteyen kullanıcı tipidir. Kâşifler ise diğer oyuncularından çok oyunun kendisiyle ilgilenen kullanıcı tipidir. Oyunu keşfederek oyunun tamamına hâkim olmaktan zevk alırlar (Yılmaz, 2015).

Oyunlaştırma ile ilgili bir diğer sınıflandırma Marczewski (2015) tarafından yapılmıştır. Marczewski (2015) yaptığı sınıflandırmada Bartle (1996) tarafından ortaya konulan oyuncu tiplerinin daha gelişmiş bir halini ortaya koyarak 6 tipte oyuncu tanımlamıştır. Bunlar; sosyalleşenler (socialiser) , özgür ruhlar (free spirit), başaranlar (achiever), yardımseverler (philanthropist), oyuncular (player) ve yıkıcılar (disruptor) şeklindedir. Marczewski (2015) tarafından oluşturulan oyuncu tipleri Resim 3'te sunulmuştur.



Resim 3. Marczewski (2015) oyuncu tipleri

Sosyalleşenler, diğer oyuncularla ilişkiye girerek motive olduğu için etkileşime geçmek ve sosyal bağlantı kurmak ister. Özgür ruhlar, özerk olmayı ve kendilerini ifade ederek motive olduğu için yaratmayı ve keşfetmeyi ister. Başaranlar, yeni şeyler öğrenme ve kendilerini geliştirme isteğinden dolayı uzmanlaşarak motive olur ve zorlukların üstesinden gelmek isterler. Yardımcı, başkalarına ve oyunlaştırılmış ortama katkı sağlamayı ister, ortamdaki ödüllerle motive olup ve ödülleri kazanmak isterler. Yıkıcılar, değişimle motive olup olumlu veya olumsuz değişimlerin olmasını ister.

#### 2.1.2.4. Eğlence türleri

Oyunlaştırma tasarımının en önemli unsurlarından biride eğlencedir. Werbach ve Hunter (2012) oyunlaştırma tasarımı yaparken takip edilmesi gereken aşamalardan beşincisi olan "Eğlenceyi unutma!" aşmasında ne tür eğlencenin kullanılacağını belirlemek gerektiğini ifade etmiştir. Yılmaz (2015:94) oyunlaştırılmış ortamın oyuncu açısından cazip olması için motivasyonun sağlanması, oyuncunun kendini iyi hissetmesine olanak tanınması ve oyuncuyu eğlendirmesi gerektiğini ifade etmiştir. Dolayısıyla oyunlaştırma tasarımı yapılırken, eğlence unsurunun oyuncu kitlesini düşünerek dikkatli bir şekilde planlanıp ve hangi tür eğlence tipinin kullanılacağını belirlemek gerekmektedir. Eğlence tipleri 4 gruba ayrılmış ve "The 4 Keys 2 FUN" olarak isimlendirilmiştir (Lazzaro, 2004). Bunlar; zor

eğlence (hard fun), kolay eğlence (easy fun), insanlarla eğlence (people fun) ve ciddi eğlence (serious fun) şeklindedir. Yılmaz (2015:95) eğlence tiplerini aşağıdaki gibi açıklamıştır:

- *Zor eğlence*: Kazanılması zor olan oyunlarda, oyuncunun ilerledikçe eğlenmesine olanak tanıyan eğlence tipidir.
- *Kolay eğlence*: Oyuncuların kafa boşaltarak eğlenmesini amaçlayan ve oyunun kolayca keşfedilmesine olanak tanıyan eğlence türüdür.
- *İnsanlarla eğlence*: Oyuncuların işbirliği içerisinde bir şeyleri başararak eğlenmesini amaçlayan eğlence tipidir.
- *Ciddi eğlence*: Eğlenirken eğlenmeyi ve davranış değişikliği meydana getirmeyi amaçlayan eğlence tipidir.

### **2.1.3. Çevrimiçi öğrenme ortamları**

Bilgi ve iletişim teknolojilerinin etkisiyle öğrenme ve öğrenme ortamlarında da değişimler meydana gelmiştir. Öğrencilere zaman ve mekândan bağımsız imkânlar sunan çevrimiçi öğrenme (Moore ve Kearsley, 1996), bu değişimin bir etkisi olarak karşımıza çıkmaktadır. Çevrimiçi öğrenme, öğrencilerin yüz yüze öğrenme ortamlarından farklı olarak ders kaynaklarına ulaştıkları, etkileşime girdikleri ve öğrenmeyi uzaktan gerçekleştirdikleri öğrenme şeklinde tanımlanmıştır (Çalışkan, 2002). Çevrimiçi öğrenmede, öğrencilerin zaman ve mekân gibi kısıtlamaları olmamakla birlikte, yüz yüze öğrenme ortamının kazandırdığı deneyimleri internet tabanlı öğrenme ortamlarından almaları esastır. Çevrimiçi öğrenmenin eş zamanlı ve eş zamansız olarak iki türü bulunmaktadır. Eşzamanlı çevrimiçi öğrenmede görüntülü ve sesli konferans gibi araçlarla öğretmen ve öğrenci arasında canlı etkileşim sağlanır. Eş zamansız çevrimiçi öğrenmede ise öğretmen ve öğrenci arasındaki etkileşim gecikmeli gerçekleşir (Sözüdoğru, Altınay ve Altınay, 2016). Çevrimiçi öğrenmenin gerçekleştiği ortamlar ise çevrimiçi öğrenme ortamları olarak ifade edilmektedir. Çevrimiçi öğrenme ortamları, öğrenmenin internet, intranet ve bilgisayar ağları gibi teknolojiler kullanılarak, öğretmen ve öğrencinin zaman ve mekândan bağımsız iletişime geçebildiği ve bireysel öğrenmenin ön planda olduğu öğrenme ortamları olarak tanımlanmaktadır (Bilgiç ve diğerleri, 2011). Moodle, Blackboard, Edmode ve Sakai çevrimiçi öğrenme ortamlarına örnek olarak verilebilir.

## 2.2. İlgili arařtırmalar

Bu bölümde ilk olarak oyunlařtırma ile ilgili arařtırmalara yer verilip sonrasında ise kodlama eđitiminde oyunlařtırma kullanımıyla ilgili yapılan arařtırmalar sunulmuřtur.

### 2.2.1. Oyunlařtırma ile ilgili arařtırmalar

Domínguez et al. (2013) çevrimiçi öğrenme ortamına ödül ve rekabet oyunlařtırma öğelerini ekleyerek, oyunlařtırılmıř bir öğrenme ortamı tasarlamıřlardır. Deney ve kontrol gruplarının kullanıldıđı öğrenme ortamı lisans düzeyinde bilgi ve iletiřim teknolojilerinin öğretimi amacıyla test edilmiř ve öğrencilerden nitel ve nicel veriler toplanmıřtır. Deney grubunda yer alan öğrencilerin başarıları kontrol grubuna göre daha yüksek bulunmuř olmasına rađmen motivasyon düzeylerinde düşüřün olduđu gözlemlenmiřtir. Ayrıca deney grubunda yer alan öğrencilerin öğrenme ortamına katılımlarında düşüřün olduđu belirlenmiřtir.

Polat (2014) çalışmasında oyunlařtırma yaklaşımının öğrencilerin genel dil motivasyonlarına ve tutumlarına etkisini incelemiřtir. Deneysel desende yürütölen çalışmada, deney ve kontrol grubu olmak üzere 32 öğrenciyle çalışılmıřtır. Deneysel iřlem süreci altı hafta sürmüř, deney grubunda oyunlařtırılmıř çevrimiçi öğrenme ortamı, kontrol grubunda ise geleneksel çevrimiçi öğrenme ortamı kullanılmıřtır. Veriler açık uçlu ve çoktan seçmeli soruların yer aldıđı bir anket aracılıđıyla toplanmıřtır. Arařtırma sonucunda grupların motivasyon puanlarında anlamlı bir farklılık bulunmazken, deney grubunda yer alan öğrencilerin oyunlařtırma yöntemine karşı olumlu tutum sergiledikleri görölmüřtür.

Hanus ve Fox (2015) iletiřim derslerinde oyunlařtırmanın etkinliđini ortaya koymak için yaptıđı arařtırmada, deney ve kontrol grubundan oluřan 71 öğrenci ile çalışmıřtır. Deney grubunda oyunlařtırılmıř yüz yüze öğrenme ortamı, kontrol grubunda ise geleneksel yüz yüze öğrenme ortamı kullanılmıřtır. Lider tahtası, rozet ve rekabet gibi oyunlařtırma unsurlarının kullanıldıđı arařtırmada elde edilen sonuçlara göre deney grubunda içsel motivasyon, başarı ve memnuniyette düşüřün olduđu görölmüřtür. Arařtırmacılar oyunlařtırma unsurlarının uygulanmasında dikkatli olunması gerektiđini belirtilmiřtir.

Su ve Cheng (2015) yaptıkları arařtırmada oyunlařtırma yaklaşımının fen bilimleri eđitiminde öğrencilerin motivasyonlarına ve başarılarına etkilerini arařtırmıřlardır. Deneysel desende yürütölen arařtırmada 2 deney ve 1 kontrol grubu kullanılmıř, deney gruplarından birinde oyunlařtırılmıř mobil öğrenme ortamı diđerinde ise geleneksel mobil



öğrenme ortamı kullanılmıştır. Kontrol grubunda ise geleneksel sınıf ortamı kullanılmıştır. İlkokul düzeyinde 102 öğrenciyle 6 hafta yürütülen araştırmada oyunlaştırılmış mobil öğrenme ortamının başarı ve motivasyon üzerinde olumlu etkilerinin olduğu sonucuna ulaşılmıştır. Ayrıca motivasyon boyutlarından dikkatin (ARCS-A) diğer boyutlara göre (ARCS-R, ARCS-C, ARCS-S) daha yüksek korelasyona sahip olduğu belirtilmiştir.

Ar (2016) çalışmasında oyunlaştırma ile öğrenmenin meslek lisesi öğrencilerinin başarıları ve öğrenme stratejileri üzerine etkisini ortaya koymayı amaçlamıştır. Tasarım ve geliştirme araştırması olarak, deney ve kontrol grubunda yer alan 65 öğrenci ile yürütülen çalışmada, deney grubunda dersler geleneksel sınıf ortamı ile oluşturulan oyunlaştırılmış web sitesinden yürütülmüştür. Kontrol grubunda ise sadece geleneksel sınıf ortamı kullanılmıştır. Çalışmada nicel verilerin toplanmasında başarı testi ve öğrenme stratejileri ölçeği, nitel verilerin toplanmasında ise görüşme formu kullanılmıştır. Araştırmada elde edilen sonuçlara göre oyunlaştırma ile öğrenme meslek lisesi öğrencilerinin akademik başarılarını arttırmış ve öğrenme stratejileri (bilişsel stratejiler, metabilşsel stratejiler) kullanımını arttırmıştır. Ayrıca oyunlaştırılmış öğrenme öğrenciler tarafından eğlenceli, faydalı ve rekabetçi bulunmuştur.

Buckley ve Doyle (2016) yaptıkları çalışmada oyunlaştırılmış çevrimiçi öğrenme ortamının lisans öğrencilerinin öğrenme durumlarında nasıl bir etki oluşturduğunu belirlemeye çalışmışlardır. Deneysel desende yürütülen araştırmanın örneklemini 100 lisans öğrencisi oluşturmaktadır. PM isimindeki çevrimiçi öğrenme ortamında yürütülen çalışmasının deneysel süreci 4 hafta sürmüş ve çeşitli araçlar kullanılarak öğrencilerin öğrenme düzeyleri belirlenmeye çalışılmıştır. Araştırmada elde edilen sonuçlar; oyunlaştırmanın öğrenme üzerine olumlu etki sağladığını ortaya koymuş ve öğrencilerin katılım durumlarının, oyunlaştırma bileşenlerinin öğrenci motivasyonlarında oluşturduğu etkiye göre değiştiği belirlenmiştir

Fiş Erümit (2016) yaptığı tasarım tabanlı araştırmada bir dersin nasıl oyunlaştırılabileceğini ve oyunlaştırmayla ilgili bir tasarım çerçevesi oluşturmayı amaçlamıştır. Üç aşamada yürütülen çalışmanın birinci aşaması 15 yüksek lisans öğrencisi, ikinci aşaması 7 yüksek lisans öğrencisi ve üçüncü aşaması ise 52 lisans öğrencisiyle tamamlanmıştır. Araştırmadan elde edilen sonuçlara göre uygulama süreci her aşamada öğrenciler tarafından eğlenceli bulunmuştur. Sosyal oyunların kullanıldığı tüm aşamalar öğrenciler tarafından beğenilmiş ve motivasyonlarına olumlu katkı sağlamıştır. Bazı öğrenciler bireysel etkinliklerden bazıları ise grup etkinliklerinden daha fazla motive olmuştur. Kişisel özellikler, yaş, cinsiyet

ve oyun tercihlerinin oyunlaştırma etkinliklerinin oluşturulmasında önemli bir etken olduğu ortaya konmuştur.

Kalkan (2016) çalışmasında oyunlaştırılmış sanal öğrenme ortamının öğrencilerin başarı, akış ve tutuma etkisini ortaya koymaya çalışmıştır. Deneysel desende yürütülen araştırmanın örneklemini 134 öğrenci oluşturmuştur. Second Life 3D sanal ortamının kullanıldığı çalışmada, oyunlaştırılmış ve esnek öğrenme ortamlarında öğrencilere sürat pateni eğitimi verilmiştir. Çalışmada elde edilen sonuçlara göre; oyunlaştırmanın öğrenci başarısında olumlu etki sağladığı, öğrencilerin akış düzeylerini arttırdığı ve oyunlaştırılmış öğrenme ortamının tutum geliştirme potansiyeli oluşturmada önemli rol oynadığı belirlenmiştir.

Meşe (2016) yaptığı çalışmada harmanlanmış öğrenme ortamlarında oyunlaştırma bileşenlerinin etkinliğini ortaya koymaya çalışmıştır. Deneysel desende yapılan çalışma, deney ve kontrol grubu olmak üzere 67 öğrenci ile Eğitimde Bilişim Teknolojileri dersinde yürütülmüştür. Çalışmada deney ve kontrol gruplarında çevrimiçi öğrenme ortamı olarak Moodle kullanılmış ve deney grubunda çevrimiçi öğrenme ortamı oyunlaştırma bileşenleriyle zenginleştirilmiştir. Yüz yüze öğrenme ortamlarında ise deney grubunda, kontrol grubundan farklı olarak oyunlaştırılmış çarkıfelek etkinliği kullanılmıştır. Çalışma sonuçlarına göre; çalışma topluluğu modeli, başarı, motivasyon ve duygu durumu açısından bir farklılık görülmemiş fakat oyunlaştırma bileşenleri ilgi çekici bulunmuştur. Çevrimiçi öğrenme ortamında harcanan zamanın, deney grubu lehine olduğu sonucuna ulaşılmıştır.

Özkan (2016) çalışmasında çevrimiçi İngilizce derslerinin, öğrencilerin motivasyonlarına ve katılımlarına etkisini belirlemeye çalışmakla birlikte hangi oyunlaştırma bileşenlerinin yabancı dil eğitiminde daha etkili olduğunu ortaya koymaya çalışmıştır. Hazırlık sınıfında okuyan 19 öğrenci ile yürütülen çalışmada harmanlanmış öğrenme ortamı kullanılmıştır. Derslerin 2 saati yüz yüze öğrenme ortamında, 2 saati ise oyunlaştırılmış çevrimiçi öğrenme ortamında yürütülmüştür. Çalışma sonuçları oyunlaştırmanın çevrimiçi öğrenme ortamında motivasyona ve derse katılıma olumlu etkilerinin olduğunu ortaya koymaktadır. Ayrıca içerik açma, ilerleme, duygular ve dönüt gibi oyunlaştırma bileşenlerine öğrenciler tarafından daha fazla anlam yüklendiği ortaya konmuştur.

Sarı ve Altun (2016) çalışmalarında, oyunlaştırma unsurlarının öğretim ortamına dahil edilmesinin öğrencilerin ilgi, motivasyon ve derse katılımlarına etkisini öğrenci görüşleriyle tespit etmeye çalışmışlardır. Meslek lisesi 10. sınıfta okuyan ve Temel Elektronik ve Ölçme

dersini alan 27 öğrenci ile yürütülen araştırmada veriler nitel olarak, gözlem ve görüşme yoluyla toplanmıştır. Oyunlaştırılmış öğrenme ortamında oyunlaştırma bileşeni olarak ödül, rozet ve lider tahtası kullanılmıştır. Elde edilen sonuçlara göre öğrencilerin ilgi ve motivasyonlarının yükseldiği görülmüş, ayrıca derse katılma isteklerinin de arttığı belirlenmiştir.

Yıldırım (2016) çalışmasında oyunlaştırma yöntemi ile tasarlanan Öğretim İlke ve Yöntemleri dersinin, öğrencilerin başarılarına, tutumlarına, oyunlaştırmaya yönelik algılarına ve düşüncelerine etkilerini belirlemeye çalışmıştır. Karma desende yürütülen araştırmada deney ve kontrol grubu olmak üzere lisans düzeyinde 96 öğrenciden veriler toplanmıştır. Çalışmada elde edilen sonuçlara göre oyunlaştırma, öğrencilerin başarı ve tutumlarını olumlu etkilemiştir. Dinamikler, mekanikler ve bileşenlerin oyunlaştırılmış ortamın algılanmasında en büyük etkiye sahip olduğu ve oyunlaştırılmış ortamın öğrenciler tarafından olumlu karşılandığı belirlenmiştir.

Tunga (2016) yaptığı araştırmada e-öğrenme ortamlarında oyunlaştırma kullanımının öğrencilerin ders başarılarına ve derse katılım durumlarına etkilerini ortaya koymayı amaçlamıştır. Deneysel desende yürütülen araştırmada, deney grubunda oyunlaştırılmış e-öğrenme ortamı (Moodle), kontrol grubunda ise geleneksel e-öğrenme ortamı kullanılmıştır. Araştırmanın örneklemini lisans düzeyinde 46 öğrenci oluşturmaktadır. Araştırmada elde edilen sonuçlara göre; deney grubunun derse katılım oranlarının daha yüksek olduğu, deney grubu öğrencilerinin daha yüksek görüntüleme ve gönderi sayısına sahip olduğu, oyunlaştırmanın motivasyon üzerine olumlu etki sağladığı, deney ve kontrol gruplarının ön test- son test akademik başarı testi puanlarında anlamlı bir artışın olduğu fakat son test-izleme testi puanlarında kontrol grubunda bir değişim olmazken deney grubunda düşüş olduğu belirlenmiştir.

Çakıroğlu, Başbüyük, Güler, Atabay ve Memiş (2017) yaptıkları araştırmada oyunlaştırılmış Bilgi ve İletişim Teknolojileri dersinin öğrencilerin derse katılımlarına ve akademik performanslarına etkilerini ortaya koymuşlardır. Çalışma eğitim fakültesi sınıf öğretmenliği bölümünde okuyan 37 lisans öğrencisiyle yürütülmüştür. Veriler; katılım ölçeği, etkinlik değerlendirme rubrikleri, oyunlaştırma değerlendirme formu ve klinik görüşmeler aracılığıyla toplanmıştır. Elde edilen sonuçlara göre derse katılımında artışın olduğunu ve bunun da başarının olumlu olarak etkilenmesini sağladığını ortaya koymuştur. Ayrıca oyunlaştırılmış öğrenme ortamının motivasyona oldukça pozitif etki yarattığını göstermiştir.

Ece (2017) yaptığı arařtırmada oyunlařtırılmıř oyun temelli kelime öğretimini, öğrencilerin başarılarına ve motivasyonlarına etkilerini ortaya koymayı amaçlamıřtır. Deneysel desende yürütölen arařtırmanın örneklemini, deney ve kontrol grubu olmak üzere 32 üniversite hazırlık sınıfı öğrencisi oluřturmaktadır. Deney grubunda kelime öğretilimi oyunlařtırılmıř öğrenme ortamında, kontrol grubunda ise geleneksel öğrenme ortamında yapılmıřtır. Arařtırma sonuçları; oyunlařtırılmıř öğrenme ortamının kullanıldıđı deney grubunun motivasyon düzeyinin anlamlı derecede yüksek olduđu göstermektedir. Ayrıca kelime dađarcıđı açısından deney ve kontrol grupları arasında anlamlı bir farklılıđın bulunmadıđı sonucuna ulařılmıřtır.

Sümer (2017) yaptığı arařtırmada; üniversitelerde kullanılan çevrimiçi öğrenme ortamlarında oyunlařtırma unsurları kullanımının, öğrencilerin derse katılım ve ortandan yararlanma durumlarına etkisini belirlemeyi amaçlamıřtır. Karma desende yürütölen arařtırmada 14 haftalık deneysel süreç 295 öğrenci ile yürütölmüřtür. Arařtırmada elde edilen sonuçlar, çevrimiçi öğrenme ortamında oyunlařtırma unsurları kullanımının derse katılımı arttırdıđını ve e-öğrenme davranıřları açısından anlamlı düzeyde artışın olduđunu göstermektedir. Ayrıca oyunlařtırılmıř ortamın ders başarı notları üzerinde etkisinin olmadıđı belirlenmiřtir.

Hüner (2018) çalıřmasında oyunlařtırma yöntemi ve oyun bileřenlerinin öğrencilerin ikinci dil öğrenme başarılarına ve motivasyonlarına etkilerini belirlemeye çalıřmıřtır. Deneysel desende yürütölen arařtırmanın örneklemini deney ve kontrol grubu olmak üzere lise düzeyinde 85 öğrenci oluřturmaktadır. Arařtırma sonuçları, oyunlařtırılmıř öğrenme ortamının öğrencilerin başarılarını arttırdıđını ve derse yönelik motivasyonlarına olumlu katkı sağladıđını göstermektedir.

Kim, Rothrock ve Freivalds (2018a) çalıřmalarında oyunlařtırılmıř web tabanlı öğrenme ortamının, mühendislik öğrencilerinin motivasyonlarına, akademik başarılarına ve derse katılım düzeylerine etkilerini belirlemeyi amaçlamıřtır. Deneysel olarak yürütölen arařtırmada, deney grubuna eğitimler oyunlařtırma bileřenlerinin dâhil edildiđi web ortamında, kontrol grubunda ise oyunlařtırma bileřenini içermeyen web ortamında verilmiřtir. Elde edilen sonuçlara göre, oyunlařtırılmıř öğrenme ortamında eğitim alan öğrencilerin, akademik başarı, ders motivasyonu ve derse katılım düzeylerinin daha yüksek olduđu göstermektedir. Ayrıca oyunlařtırma bileřenlerinden, puan ve lider tahtasının, öğrencileri motive etme düzeylerinin daha yüksek olduđu bulunmuřtur.

Mohammed (2018) yaptığı arařtırmada çevrimiçi öğrenme ortamında (Moodle) oyunlařtırma unsurları kullanımının, öğrenci performanslarına ve algılarına etkilerini ortaya koymayı amaçlamıřtır. Deneysel desende yürütölen arařtırmanın örneklemini deney ve kontrol grubunda yer alan 47 öğrenci oluřturmaktadır. Deney grubunda dersler oyunlařtırılmıř çevrimiçi öğrenme ortamında, kontrol grubunda ise geleneksel çevrimiçi öğrenme ortamında yürütölmüřtür. Arařtırma sonuçları, deney grubunun son test başarı notunun kontrol grubundan anlamlı derecede yüksek olduđu, oyunlařtırılmıř öğrenme ortamının öğrenciler tarafında ilgi çekici ve faydalı olarak göröldüğü ortaya koymaktadır.

Zainuddin (2018) çalışmasında oyunlařtırılmıř ters yüz edilmiř öğrenme ortamının, öğrencilerin performanslarına ve algılanan motivasyonlarına etkilerini ortaya koymayı amaçlamıřtır. Karma desen olarak yürütölen arařtırmanın çalışma grubunu, deney ve kontrol grubu olmak üzere 56 öğrenci oluřturmaktadır. Ayrıca arařtırmanın nitel boyutunda ise 6 öğrenci arařtırmaya dâhil edilmiřtir. Çevrimiçi öğrenme ortamı olarak iSpring'in kullanıldıđı arařtırma sonuçlarına göre; oyunlařtırılmıř sistem öğrenci katılımını arttırmıř, daha iyi motivasyon sađlamıř, rekabet ve kazanma unsurlarıyla öğrencilerin motive olduđu görölmüř ve deney grubunun yetkinlik açısından daha iyi olduđu belirlenmiřtir.

Araya, Arias Ortiz, Botton ve Cristia (2019) çalışmalarında oyunlařtırılmıř çevrimiçi öğrenme ortamının, ilköğretim düzeyinde matematik eğitimi alan öğrencilerin matematik başarılarına etkilerini belirlemeyi amaçlamıřtır. Arařtırmada çevrimiçi öğrenme ortamı olarak ConectaIdeas isimli oyunlařtırma bileřenlerinin kullanıldıđı öğrenme ortamı kullanılmıřtır. Oyunlařtırılmıř çevrimiçi öğrenme ortamında öğrencilere çeřitli problem durumları sunulmuř ve öğrencilerin iki haftalık çalışma süresince çözmeleri istenmiřtir. Bu süreçte öğrenciler bireysel ve grup olarak çalışmıř, ilerleme durumlarını ve arkadaşlarının ilerleme durumları oyunlařtırma bileřenleri vasıtasıyla görebilmiřlerdir. Elde edilen sonuçlara göre, iki haftalık eğitimi alan öğrencilerin başarılarının yükseldiđi belirlenmiřtir.

Bal (2019) eylem arařtırması olarak yürüttüğü çalışmasında, oyunlařtırmanın ortaokul öğrencilerinin yazma becerilerinde nasıl bir etki oluřturduđunu belirlemeyi amaçlamıřtır. Ortaokul 7.sınıf düzeyinde yürütölen arařtırmada, oyunlařtırmanın öğrencilerin Türkçe derslerinde yazma becerilerine olumlu katkı sađladığını ortaya koymuřtur. Ayrıca öğrencilerin motivasyonlarını da olumlu yönde etkilediđi ve öğrencilerin iřbirliđi içerisinde çalışmalarını desteklediđi belirlenmiřtir.

Bayram ve Çalışkan (2019) eylem araştırması olarak yürüttükleri çalışmalarında oyunlaştırma etkinliklerinin, ortaokul 6. sınıf öğrencilerinde yarattığı etkiyi ortaya koymaya çalışmışlardır. Araştırma sonuçlarına göre öğrencilerin katılımları, ilgi ve tutumlarının olumlu yönde olduğu belirlenmiş ayrıca öğrencilerin motivasyonlarının da arttığı ortaya konmuştur.

Huang, Hew ve Lo (2019) yaptıkları çalışmada, oyunlaştırılmış ters yüz öğrenme ortamının öğrencilerin katılımları üzerindeki etkisini incelemiştir. Deney ve kontrol olmak üzere, 96 lisans öğrencisi ile deneysel olarak yürütülen araştırma 10 hafta sürmüştür. Deney grubunda oyunlaştırılmış ters yüz ortamı, kontrol grubunda ise ters yüz ortamı kullanılmıştır. Çevrimiçi öğrenme ortamı olarak Moodle kullanılan çalışmada, oyunlaştırma bileşenleri çevrimiçi öğrenme ortamında sunulmuştur. Elde edilen sonuçlara göre, oyunlaştırılmış ters yüz öğrenme ortamı öğrencilerin derse katılımlarını arttırmıştır.

Kavaklı Duman (2019) yaptığı çalışmada yabancı dil öğretiminde oyunlaştırma yaklaşımının motivasyon ve tutum üzerine etkilerini ortaya koymaya çalışmıştır. ClassDojo kullanılarak yapılan oyunlaştırılmış öğrenme ortamı 9. sınıfta okuyan 104 öğrenci üzerinde uygulanmıştır. Araştırma sonuçlarına göre; katılım ve motivasyonun arttığı belirlenmiş, öğrencilerin dil becerilerinin geliştiği ve öğrencilerin oyunlaştırma yaklaşımına yönelik olumlu tutum sergiledikleri ortaya çıkmıştır.

Ertan (2020) çalışmasında oyunlaştırılmış İngilizce dersinin; öğrencilerin başarı, motivasyon ve tutumları üzerinde etkisini ortaya koymaya çalışmıştır. Araştırma tek gruptan oluşan 36 ön lisans öğrencisi ile İngilizce dersinde yürütülmüştür. Elde edilen sonuçlara göre; oyunlaştırmanın başarıyı arttırmada olumlu etkisinin olduğu, motivasyon ve tutum puanlarının yüksek olduğu belirlenmiştir. Ayrıca öğrencilerin oyunlaştırılmış öğrenme ortamı ve oyun bileşenlerine karşı tutumlarının olumlu olduğu tespit edilmiştir.

Topal (2020) yaptığı çalışmada çevrimiçi öğrenme ortamında oyunlaştırma kullanımının başarı, bağlılık ve öğrenme motivasyonu üzerindeki etkilerini ortaya koymaya çalışmıştır. Faktöriyel desende yürütülen araştırmanın örneklemini, lisans düzeyinde Eğitimde Filmler ortak dersini alan 72 öğrenci oluşturmaktadır. Araştırmada veri toplama aracı olarak; güdülenme ölçeği, çevrimiçi öğrenme ortamlarında öğrenci bağlılık ölçeği, başarı testi, demografik bilgiler formu, görüşme formu ve çevrimiçi öğrenme sistem kayıtları kullanılmıştır. Elde edilen sonuçlara göre; oyunlaştırılmış öğrenme ortamında öğrencilerin sisteme giriş sayılarının anlamlı derecede yüksek olduğu, oyunlaştırmaya yönelik

öğrencilerin tutumlarının olumlu olduğu, tartışma platformunun motivasyonu arttırdığı ve lider tahtasının başarıyı arttırdığı belirlenmiştir.

Taşkın (2020) çalışmasında oyunlaştırılmış ters yüz edilmiş öğrenme ortamının akademik başarı, motivasyon ve katılıma etkisini ortaya koymaya çalışmıştır. Bilgisayar teknolojileri ön lisans bölümlerinde okuyan 54 öğrenci ile yürütülen araştırmada, deney ve kontrol gruplarının katılım ve motivasyon puanları arasında anlamlı bir farklılık görülmemiştir. Deney grubu akademik başarı puanının, kontrol grubundan yüksek olmasına rağmen bu farkın anlamlı olmadığı sonucuna ulaşılmıştır. Ayrıca deney grubunun dersi tamamlama, oturum açma sayısı, derste kalma süresi, dersleri inceleme süresi ve testlere katılım oranlarının kontrol grubundan fazla olduğu belirlenmiştir.

### **2.2.2. Kodlama eğitiminde oyunlaştırma kullanımı ile ilgili araştırmalar**

Kumar ve Khurana (2012) çalışmalarında programlama dilleri öğretiminin eğlenceli olarak sunulması amacıyla oyunlaştırma yöntemini kullanmışlardır. Yüksel lisans öğrenimi gören 207 öğrenciyle yürütülen araştırmada veriler anketler aracılığıyla toplanmıştır. Programlama öğretiminde öğrencilerin motivasyonları ve derse katılımlarının sağlanması amacıyla oyunlaştırmanın nasıl kullanılacağına yönelik yapılan araştırmada, programlama öğretiminde oyunlaştırmanın nasıl kullanılacağına yönelik bir çerçeve oluşturulmuş ve kullanılmıştır. Çalışmada elde edilen sonuçlar, oyunlaştırılmış öğrenme ortamının öğrenciler tarafından memnuniyetle karşılandığını göstermektedir. Ayrıca programlama öğretiminde oyunlaştırmanın etkili ve ciddi bir şekilde uygulanmasının, öğrencilerin gerçek yaşam problemlerini çözmelerinde kolaylık sağlayacağı belirtilmiştir.

Fidan (2016) Scratch ile programlama öğretiminde oyunlaştırma kullanımının öğrencilerin derse katılımı, motivasyonu ve başarılarına etkilerini incelediği araştırmasında; oyunlaştırma unsurları olarak lider tahtası, rozetler, seviyeler ve görevler kullanmıştır. Durum çalışması olarak yürütülen araştırma 37 lisans öğrencisi ile yürütülmüştür. Elde edilen sonuçlar, programlama öğretiminde oyunlaştırma kullanımının öğrenci katılımına olumlu etki sağladığını, öğrenci motivasyonlarını ve ders başarılarını arttırdığını göstermektedir. Ayrıca oyunlaştırmanın, eğitim sürecini daha eğlenceli bir hale dönüştürdüğünü göstermektedir.

Türk ve Gören (2017) yaptıkları çalışmada öğrencilerin, kendi hızlarında programlama öğrenebilmelerine olanak tanıyan oyunlaştırılmış web tabanlı öğrenme ortamı

tasarlamışlardır. Hazırlanan ortam kendi hızında öğrenme, laboratuvar çalışmaları, ödev ve yönetim modüllerinden oluşmaktadır. Hazırlanan oyunlaştırılmış öğrenme ortamı bilgisayar mühendisliği bölümünde okuyan öğrencilerle python programlama dili öğretiminde test edilmiştir. Elde edilen sonuçlara göre öğrenciler rozet kazanmaktan hoşlandıkları gibi diğer arkadaşlarının ilerleme durumlarını da takip etmek istemişlerdir. Ayrıca öğrencilerin ilerlemelerini görmeleri ve çalışmalarlarıyla ilgili anında dönüt almaları motive olmalarını sağlamıştır.

Yıldız (2018) yaptığı çalışmada oyunlaştırılmış blok temelli algoritmik düşünme aktivitelerinin; öğrencilerin tutum, katılım ve becerileri üzerindeki etkilerini incelemiştir. Deneysel desende yürütülen, 2 deney ve 1 kontrol grubunun kullanıldığı araştırma, 88 ortaokul öğrencisiyle yürütülmüştür. Deney 1 grubunda kahoot ve deney 2 grubunda classdojo ortamları oyunlaştırma amacıyla kullanılmıştır. Derslerin yürütülmesinde ise tüm gruplarda code.org kullanılmıştır. Kontrol grubunda oyunlaştırma amacıyla hiçbir ortam kullanılmamıştır. Araştırmada; oyunlaştırmanın derse katılım düzeyini arttırdığı, derse yönelik tutumu olumlu yönde değiştirebildiği ve programlama becerisini arttırdığı sonuçlarına ulaşılmıştır.

Çilengir (2019) blok tabanlı kodlama eğitiminde oyunlaştırma yaklaşımı kullanımının başarı ve motivasyon üzerine etkilerini belirlemeye çalıştığı araştırma, ortaokul 6.sınıf öğrencileriyle yürütülmüştür. Oyunlaştırılmış ortam olarak ClassDojo ve Kahoot kullanılmıştır. Elde edilen sonuçlara göre oyunlaştırma yaklaşımının akademik başarıyı anlamlı derecede yükselttiği fakat motivasyon üzerine anlamlı bir etki yaratmadığı ortaya çıkmıştır.

Ortiz-Rojas, Chiluzia ve Valcke (2019) çalışmasında mühendislik eğitiminde oyunlaştırmanın öğrencilerin öğrenme performanslarında nasıl bir etki oluşturacağını belirlemeyi amaçlamıştır. Deneysel desende yürütülen çalışmada, içsel motivasyon, öz yeterlik, katılım, cinsiyet, oyun deneyimi ve alan gibi değişkenler göz önüne alınmıştır Programlamaya Giriş dersinde oyunlaştırma bileşeni olarak lider tahtası kullanılmıştır. Oyunlaştırılmış öğrenme ortamının geleneksel yüz yüze öğrenme ortamına göre öğrencilerin programlama performanslarını olumlu etkilediği sonucuna ulaşılmıştır.

Sağay (2019) çalışmasında kodlama eğitiminde oyunlaştırma yaklaşımı kullanımının ortaokul öğrencilerinin tutumlarına etkisini belirlemeyi amaçlamıştır. Deneysel desende yürütülen çalışmanın örneklemini 6. sınıfta okuyan 106 ortaokul öğrencisi



oluşturmaktadır. Veri toplama aracı olarak tutum ölçeği kullanılmış ve ölçek çalışma öncesinde ön test, çalışma bitiminde ise son test olarak uygulanmıştır. Elde edilen sonuçlara göre, oyunlaştırma yaklaşımının öğrencilerde olumlu tutum geliştirdiği belirlenmiştir.

Atabay ve Albayrak (2020) çalışmalarında oyunlaştırma yöntemi kullanarak okul öncesi öğrencilere algoritma eğitimi vermişlerdir. Eylem araştırması olarak 12 öğrenci ile yürütülen çalışmada hikâyeler ve oyunlaştırılmış yazılım kullanılmıştır. Oyunlaştırma yönteminin öğrencilerin öğrenmelerine olumlu katkı sağladığı ve problemleri analiz edebildikleri belirlenmiştir. Ayrıca kullanılan oyunlaştırma bileşenlerinden rozet ve ödüllerin motivasyonu arttırmada etkili olduğu sonucuna ulaşılmıştır.

### **2.2.3. Alanyazın Taramasının Sonucu**

Alanyazında yapılan araştırmalar incelendiğinde, oyunlaştırma yaklaşımının yüz yüze ve çevrimiçi olmak üzere çeşitli öğrenme ortamlarında kullanıldığı görülmektedir. Oyunlaştırma bileşenlerinin, araştırmacılar tarafından tasarlandığı araştırmalar olmakla birlikte, Kahoot ve ClassDojo gibi hazır oyunlaştırılmış platformların kullanıldığı çalışmalarda bulunmaktadır. Alanyazında oyunlaştırma öğelerinin kullanımına yönelik bir oyunlaştırma tasarım çerçevesinin belirlenerek, aşamalarının takip edildiği çalışmaların kısıtlı olduğu görülmektedir. Ayrıca oyuncu tipleri ve eğlence türlerinin belirlenerek, oyunlaştırılmış öğrenme ortamının ve bileşenlerinin tasarlandığı çalışmalara ise rastlanmamıştır. Oyunlaştırma yaklaşımının eğitim dışında birçok alanda kullanıldığı alanyazında görülmektedir. Fakat kodlama eğitiminde oyunlaştırma kullanımına yönelik araştırmaların ise kısıtlı olduğu kendini göstermektedir.

Araştırmalarda; ilkokul, ortaokul, lise, lisans ve lisansüstü olmak üzere farklı düzeylerde örneklemeleri kullanıldığı, ağırlıklı olarak; başarı, motivasyon, tutum ve derse katılım gibi değişkenlerin yoğunlukla çalışıldığı görülmektedir. Oyunlaştırma yaklaşımının; başarı (Su ve Cheng,2015; Ar, 2016; Buckley ve Doyle, 2016; Fidan, 2016; Kalkan, 2016; Yıldırım, 2016; Hüner, 2018; Kim et al., 2018a; Mohammed, 2018; Araya et al., 2019; Çilengir, 2019; Ertan, 2020; Topal, 2020) , motivasyon (Su ve Cheng, 2015; Fidan, 2016; Fiş Erümit, 2016; Özkan, 2016; Sarı ve Altun, 2016; Tunga, 2016; Çakıroğlu ve diğerleri, 2017; Ece, 2017; Türk ve Gören, 2017; Hüner, 2018; Kim et al., 2018a; Zainuddin, 2018; Bal, 2019; Bayram ve Çalışkan, 2019; Çilengir, 2019; Kavaklı Duman, 2019; Atabay ve Albayrak, 2020; Ertan, 2020; Topal, 2020), tutum (Polat, 2014; Kalkan, 2016; Yıldırım, 2016; Yıldız, 2018; Bayram

ve Çalışkan, 2019; Kavaklı Duman, 2019; Sağay, 2019; Ertan, 2020; Topal, 2020 ) ve derse katılım (Fidan, 2016; Meşe, 2016; Özkan, 2016; Sarı ve Altun, 2016; Tunga, 2016; Çakıroğlu ve diğerleri, 2017; Sümer, 2017; Kim et al., 2018a; Yıldız, 2018; Zainuddin, 2018; Bayram ve Çalışkan, 2019; Huang et al., 2019; Kavaklı Duman, 2019) üzerine olumlu etkilerinin ifade edildiği araştırmalar yoğunlukta olmakla birlikte, tersi sonuçlara ulaşılan araştırmalarda (Domínguez et al., 2013; Hanus ve Fox, 2015) bulunmaktadır.

## BÖLÜM III

### YÖNTEM

Bu bölümde; araştırmanın modeli, çalışma grubu, veri toplama araçları, verilerin toplanması, uygulama süreci ve verilerin analiziyle ilgili bilgilere yer verilmiştir.

#### 3.1. Araştırma yöntemi

Çevrimiçi ve yüz yüze kodlama eğitiminde oyunlaştırma öğeleri kullanımının akademik başarı, motivasyon ve kodlamaya yönelik tutum üzerindeki etkisini belirlemek amacıyla nicel araştırma desenlerinden karışık desen kullanılmıştır. Split plot faktöriyel desenler olarak da bilinen karışık desenlerde, farklı deneysel işlem koşulları ve tekrarlı ölçümler olmak üzere bağımlı değişken üzerindeki etkisi incelenen iki bağımsız değişken bulunmaktadır (Büyüköztürk, Çakmak, Akgün, Karadeniz, Demirel, 2012:212). 4x3 karışık desenin kullanıldığı araştırmada birinci bağımsız değişkeni, farklı öğrenme ortamlarının kullanıldığı 3 deney ve 1 kontrol grubu oluşmaktadır. Kodlama eğitimi birinci deney grubuna OÇÖO'da, ikinci deney grubuna OYYÖO'da, üçüncü deney grubuna ÇÖO'da ve kontrol grubuna ise hiçbir deneysel işlemin uygulanmadığı YYÖO'da verilmiştir. İkinci bağımsız değişkenimiz ise zamana bağlı tekrarlı ölçümleri (ön test, son test, izleme testi) ifade etmektedir. Araştırmamızın simgesel gösterimi Tablo 2'de sunulmuştur.

Tablo 2

#### *Araştırmanın Simgesel Gösterimi*

Grup	Ön test	Deneysel İşlem	Son Test	İzleme Testi
G1 (Deney 1)	O <sub>1.1</sub>	X <sub>1</sub>	O <sub>1.2</sub>	O <sub>1.3</sub>
G2 (Deney 2)	O <sub>2.1</sub>	X <sub>2</sub>	O <sub>2.2</sub>	O <sub>2.3</sub>
G3 (Deney 3)	O <sub>3.1</sub>	X <sub>3</sub>	O <sub>3.2</sub>	O <sub>3.3</sub>

Tablo 2'de G1, G2, G3 deney grupları, G4 ise kontrol grubudur. O<sub>1.1</sub>, O<sub>2.1</sub>, O<sub>3.1</sub> ve O<sub>4.1</sub> ön test ölçümlerini, O<sub>1.2</sub>, O<sub>2.2</sub>, O<sub>3.2</sub> ve O<sub>4.2</sub> son test ölçümlerini, O<sub>1.3</sub>, O<sub>2.3</sub>, O<sub>3.3</sub> ve O<sub>4.3</sub> ise izleme testi ölçümlerini ifade etmektedir. Deneysel işlem ise X<sub>1</sub>, X<sub>2</sub> ve X<sub>3</sub> ile simgeleştirilmiştir.

### 3.2. Araştırmanın çalışma grubu

Araştırmanın çalışma grubunu 2017-2018 eğitim öğretim yılında Adıyaman ili Gölbaşı ilçesinde bir anadolu lisesinde okuyan ve seçmeli bilgisayar bilimi dersini alan 9. sınıf öğrencileri oluşturmaktadır. Anadolu liselerinde 2017-2018 eğitim öğretim yılı ile birlikte seçmeli bilgisayar bilimi dersi 9.sınıftan itibaren ders çizelgelerine girmiş ve öğrenciler bu derste kodlama eğitimi almaya başlamışlardır. Meslek liseleri dışında diğer ortaöğretim kurumlarında, 9.sınıftan itibaren kademeli olarak kodlama eğitimi verilmeye başlanmıştır.

Okulda 9. sınıf seviyesinde 5 şube ve toplam 170 öğrenci bulunmaktadır. Öğretim yılı başında okul idaresi tarafından öğrenciler yerleştirme puanları dikkate alınarak, şubelere karma olarak dağıtılmıştır. Dolayısıyla tüm sınıflar, yerleştirme puanları açısından birbirine denktir. Çalışma grupları yasal zorunluluklar nedeniyle, tüm sınıflar arasından deney ve kontrol gruplarına kura ile atanmıştır. Bu yönüyle araştırma, hazır gruplardan oluşturulduğu için yarı deneyseldir (Büyüköztürk ve diğerleri, 2012).

Araştırma başlangıcında, çalışma grubu 136 öğrenciden oluşmasına rağmen çeşitli nedenlerle denek kayıpları yaşamıştır. Pilot uygulamadan sonra 3 öğrenci başka okullara nakil olmuş ve deneysel sürecin 5.haftasında ise 1 öğrenci sağlık sebeplerinde dolayı çalışma grubundan çıkarılmıştır. Araştırmanın çalışma grubuna ait bilgileri Tablo 3'te sunulmuştur.

Tablo 3

#### *Deney ve Kontrol Gruplarındaki Öğrencilere Ait Bilgiler*

Grup	Kız	Erkek	Toplam
Deney 1	18	15	33

Deney 2	20	13	33
Deney 3	15	18	33
Kontrol	18	15	33
Toplam	71	61	132

---

Tablo 3 incelendiğinde, arařtırmadaki alıřma grubu 71 kız ve 61 erkek olmak üzere toplam 132 ğrenciden oluřmaktadır.

### **3.3. Veri toplama araları**

Arařtırmada 3 tane veri toplama aracı kullanılmıřtır. Bu aralar; ğrencilerin bařarı dzeylerini belirlemek iin arařtırmacı tarafından geliřtirilen Akademik Bařarı Testi (ABT), derse ynelik motivasyon dzeylerini belirlemek iin Motivasyon leđi (M), kodlamaya karřı tutumlarını lmek iin ise Programlama Dillerine Ynelik Tutum leđi (PDYT) Őeklinindedir.

#### **3.3.1. Akademik bařarı testi**

Arařtırmada kullanılan farklı deneysel kořulların, ğrencilerin bařarılarına etkisini belirlemek amacıyla ABT geliřtirilmiřtir. ABT hazırlanırken, đretim programında yer alan kazanımların ve soruların biliřsel seviyelerini gsteren belirtke tablosu hazırlanmıř ve Ek 1’de sunulmuřtur. Belirtke tablosunda yer alan kazanımlar dikkate alınarak, 36 sorudan oluřan madde havuzu oluřturulmuř ve madde havuzunda yer alan soruların kapsam ve grnř geerliliđi iin 1 lme deđerlendirme uzmanı, 1 BTE uzmanı ve 2 programlama uzmanının grřlerine bařvurulmuřtur. Uzman grřleri dođrultusunda, 2 soru kapsam aısından uygun grlmeyerek testten ıkarılmıř ve bazı soruların madde kkleri ve seeneklerinde de bazı dzenlemeler yapılmıřtır. Yapılan dzenlemeler sonucunda ortaya ıkan bařarı testi taslađı, pilot uygulama ncesinde meslek lisesi bilgisayar alanında eđitim gren 20 đrenciye uygulanarak testin anlaşılabilirliđi ve cevaplanma sresi kontrol edilmiřtir. Testin ortalama 30-40 dakika aralıđında bitirildiđi gzlemlenmiřtir.

Gerekli düzenlemeler yapıldıktan sonra oluşan 34 soruluk başarı testi, güvenilirlik analizi için Hacettepe Üniversitesi Bilgisayar Mühendisliğin Bölümünde eğitim gören ve python eğitimi almış 93 öğrenciye uygulanmıştır. Uygulama sonrasında madde ayırt edicilik (r) ve madde güçlük indeksleri (p) hesaplanmış ve elde edilen değerler Ek 2'de sunulmuştur.

Madde güçlük indeksi, maddeyi doğru olarak yanıtlayanların tüm cevaplayanlara oranı olarak ifade edilir. Bu değer madde; hiç doğru cevaplanmadığı durumda "1" değerini alırken, bütün öğrencilerin doğru cevapladığı durumda ise "0" değerini alır. Madde güçlük indeksinin; 1'e yaklaşması sorunun kolay olduğunu, 0'a yaklaşması sorunun zor olduğunu, 0.5 olması ise sorunun orta güçlükte olduğunu göstermektedir (Atılğan, 2016).

Madde ayırt edicilik indeksi, madde ile ölçülmesi beklenen özelliğin doğru bir şekilde ölçülüp ölçülmediğini yani bilenle bilmeyenin ayırt edilebilirliğinin göstergesi olarak ifade edilir. Bu değer "-1" ile "+1" arasında değer almaktadır. Madde ayırt edicilik indeksinin -1'e yaklaşması yani negatif olması, maddenin bilenle bilmeyeni ters ayırt ettiğini göstermektedir. Bu bilenlerin bu maddeyi yanlış cevapladığının ifadesidir. Madde ayırt edicilik indeksinin +1'e yaklaşması ise maddenin bilenle bilmeyeni, doğru ayırt ettiğinin bir göstergesi olarak ifade edilebilir. Dolayısıyla bir maddenin madde ayırt edicilik indeksinin +1'e yakın olması istenen bir sonuçtur (Atılğan, 2016).

Madde seçimi aşamasında madde güçlüğü ve ayırt ediciliği için uzmanlar tarafından çeşitli değer aralıkları önerilmiştir. Bayrakçeken (2011) madde güçlük indeksinin 0.2-0.8 aralığında, madde ayırt ediciliğinin ise 0.3'ten yüksek olmasının istenen bir durum olduğunu ifade etmiştir. Özgüven (2003) ise madde ayırt edicilik indeksinin 0.2-0.29 aralığında, madde güçlük indeksinin ise 0.15-0.39 ve 0.61-0.85 arasında olan maddelerin teste kullanılabileceğini belirtmiştir. Bu görüşlerde dikkate alınarak araştırmada madde güçlük indeksi 0.85 ve altı olan maddeler, madde ayırt edicilik indeksi ise 0.35 üzerinde olan maddeler teste dâhil edilmiştir. Bu aralıklarda bulunan toplam 28 madde teste dâhil edilmişken, 6 madde (4, 6, 11, 12, 13, 29) testten çıkarılmıştır. Madde analizi sonucunda seçimi yapılan 28 maddeden oluşan testin güvenilirliğini ölçmek için KR-20 iç tutarlılık katsayısı hesaplanmış ve bu değer 0.81 olarak bulunmuştur.

ABT'nin kararlılık katsayısını hesaplamak için test tekrar test yöntemi kullanılmıştır. Test tekrar test yöntemi, başarı testinin farklı zamanlarda aynı gruba iki defa uygulanmasıdır. Bu durumda aynı bireylerden alınan puanlar arasındaki korelasyon katsayısı, testin güvenilirliğini ortaya koymaktadır (Atılğan, 2016). Başarı testi Sakarya Üniversitesi Bilgisayar ve Öğretim

Teknolojileri Bölümünde okuyan ve python eğitimi almış 48 öğrenciye 2 hafta arayla uygulanmış ve uygulamalar arasındaki ilişkiyi ortaya koymak amacıyla Pearson Momentler Çarpım Korelasyon Katsayısı kullanılmıştır. Hesaplanan korelasyon katsayısı 0.81 olarak hesaplanmıştır. Elde edilen korelasyon katsayısının 0.70 ve üzeri olması testin güvenilir bir ölçüm yaptığını ve buda kararlılığının bir göstergesidir (Büyüköztürk ve diğerleri, 2012).

ABT taslak formu üzerinde yapılan düzeltmeler ve analizler sonucunda, test 28 çoktan seçmeli sorudan oluşan son halini almış ve Ek 3'te sunulmuştur. Hazırlanan ABT araştırmada ön test, son test ve izleme testi olarak kullanılmıştır.

### 3.3.2. Motivasyon Ölçeği

Kodlama eğitiminde çevrimiçi öğrenme ortamları ve oyunlaştırma bileşenleri kullanımının öğrencilerin motivasyonlarına etkilerini belirlemek amacıyla; Pintrich, Smith, Garcia ve McKeachie (1993) tarafından geliştirilen Büyüköztürk, Akgün, Özkahveci ve Demirel (2004) tarafından Türkçe'ye uyarlanan ve Büyüköztürk, Akgün, Karadeniz, Kılıç Çakmak ve Demirel (2007) tarafından Türkiye norm çalışması yapılarak 12-18 yaş arasındaki öğrencilere uygun hale getirilen “Güdülenme ve Öğrenme Stratejileri Ölçeği” kullanılmıştır. Ölçeğin kullanımıyla ilgili gerekli izinler alınmış ve Ek 4'te sunulmuştur. Ölçek, motivasyon ve öğrenme stratejileri olmak üzere iki alt ölçekten oluşmaktadır. Araştırmada 7'li likert tipli (1= Benim için kesinlikle yanlış, 7= Benim için kesinlikler doğru) ve 20 maddeden oluşan motivasyon alt ölçeği kullanılmıştır. MÖ değeri, beklenti ve duyuşsal olmak üzere üç alt boyuttan oluşmaktadır. MÖ'nün yapı geçerliliği için yapılan doğrulayıcı faktör analizi sonuçlarına göre ((Ki-Kare ( $\chi^2$ )=10565.09, p=.000, sd=166) ve uyum indeksleri (RMR=0.17, SRMR=0.064, GFI=0.94, AGFI=0.93, RMSEA=0.061, CFI=0.89, NNFI=0.87)) ölçeğinin kabul edilebilir düzeyde geçerliğe sahip olduğu söylenebilir. Ölçeğin güvenilirliğini belirlemek amacıyla cronbach alfa iç tutarlık katsayısı hesaplanmıştır. İç tutarlılık katsayısı değer ve beklenti ana boyutları için .79, duyuşsal ana boyutu için .58 olarak hesaplanmıştır. Bu haliyle ölçeğin güvenilir olduğu ifade edilmiştir. Araştırmada elde edilen verilerden hesaplanan cronbach alfa iç tutarlık katsayısı .82 olarak bulunmuş ve bu değer ölçeğin güvenilirliğini doğrulamaktadır. Araştırma kapsamında kullanılan MÖ örnek maddeleri Ek 5'te sunulmuştur.

### **3.3.3. Tutum ölçeđi**

Kodlama eđitiminde çevrimiçi öğrenme ortamları ve oyunlaştırma öğeleri kullanımının, öğrencilerin kodlamaya yönelik tutumlarına etkilerini belirlemek amacıyla Tapia ve Marsh (2004) tarafından "Matematiđe Yönelik Tutum Ölçeđi" olarak geliştirilen ve Durak (2013) tarafından programlama dillerine uyarlanarak Türkçe'ye çevrilen "Programlama Dillerine Yönelik Tutum Ölçeđi" kullanılmıřtır. Ölçeđin kullanımıyla ilgili gerekli izinler alınmıř ve Ek 6'da sunulmuřtur. Ölçeđin orijinal formunun güvenilirliđini belirlemek amacıyla hesaplanan cronbach alfa iç tutarlık katsayısı .97 olarak belirlenmiřtir. PDYTÖ 5'li likert tipinde (1= Kesinlikle katılmıyorum, 7= Kesinlikle katılıyorum) olup toplam 40 maddeden oluřmaktadır. Uyarlanan ölçeđin pilot çalıřması 165 öğrenciyle yapılmıř ve hesaplanan cronbach alfa iç tutarlık katsayısı .93 olarak bulunmuřtur. Arařtırmada elde edilen verilerden hesaplanan cronbach alfa iç tutarlık katsayısı .81 olarak bulunmuř ve bu deđer ölçeđin güvenilirliđini dođrulamaktadır. Arařtırmada kullandıđımız tutum ölçeđi örnek maddeleri Ek 7'de sunulmuřtur.

### **3.4.Verilerin toplanması**

Arařtırma verileri 2017-2018 eđitim öğretim yılının ikinci döneminde seçmeli bilgisayar bilimi dersini alan 132 öğrenciden oluřan çalıřma grubundan ön test, son test ve izleme testi olarak toplanmıřtır. İlk olarak pilot uygulamadan sonra gerekli düzenlemeler yapılmıř ve deneysel iřlem süreci bařlamadan önce ABT, MÖ, PDYTÖ veri toplama araçları ön test olarak uygulanmıřtır. Deneysel iřlem sürecinin bitiminde aynı veri toplama araçları son test olarak uygulanmıř ve 5 hafta sonra ise aynı veri toplama araçları izleme testi olarak uygulanmıřtır.

### **3.5.Uygulama süreci**

Arařtırmada deneysel iřlemler 2017-2018 eđitim öğretim yılının 2. döneminde 9. sınıflarda okutulan seçmeli bilgisayar bilimi dersinde yürütölmüřtür. Kodlama eđitiminde metin tabanlı programlama dili olan python kullanılmıřtır.

Öğrenme ortamı olarak deney gruplarında OÇÖÖ, OYYÖÖ ve ÇÖÖ, kontrol grubunda ise YYÖÖ kullanılmıřtır. OÇÖÖ ve ÇÖÖ ortamları için Moddle ÖYS kullanılmıř ve sınırsız olarak satın alınan web alanına ([www.kodlamaogreniyorum.net](http://www.kodlamaogreniyorum.net)) yüklenmiřtir.



Araştırma, ön hazırlık eğitimi, pilot uygulama ve deneysel işlem olmak üzere üç aşamada yürütülmüş ve uygulama süreci Şekil 7'de sunulmuştur.



Şekil 7. Uygulama süreci

Şekil 7 incelendiğinde ön hazırlık eğitimi 2017-2018 eğitim öğretim yılının 1. döneminde, pilot uygulama ise 2017-2018 eğitim öğretim yılının 2. döneminde deneysel uygulama ile tümleşik olarak 12 haftalık sürede tamamlanmıştır. Tüm süreç 12 hafta olarak planlanmıştır. Pilot uygulama sonrasında ön testler uygulanmış ve 12 haftalık süreç sonunda ise son testler uygulanmıştır. İzleme testi ise deneysel uygulamanın bitiminden 5 hafta sonra uygulanmıştır.

### 3.5.1. Ön hazırlık eğitimi ve pilot uygulama

Ön hazırlık eğitimi, öğrencilerin çevrimiçi öğrenme ortamı deneyimlerinin artırılmasına yöneliktir. Araştırmada OÇÖO ve ÇÖO gruplarında kullanılan Moodle ÖYS ortamının çalışma grupları belirlenmeden önce, okuldaki tüm 9. sınıf öğrenciler tarafından kullanılabilmesini sağlamak amacıyla 2017-2018 eğitim öğretim yılının 1. döneminde ön hazırlık eğitimi yapılmıştır. Bu süre zarfında öğrencilere ders notları, kodlama etkinlikleri

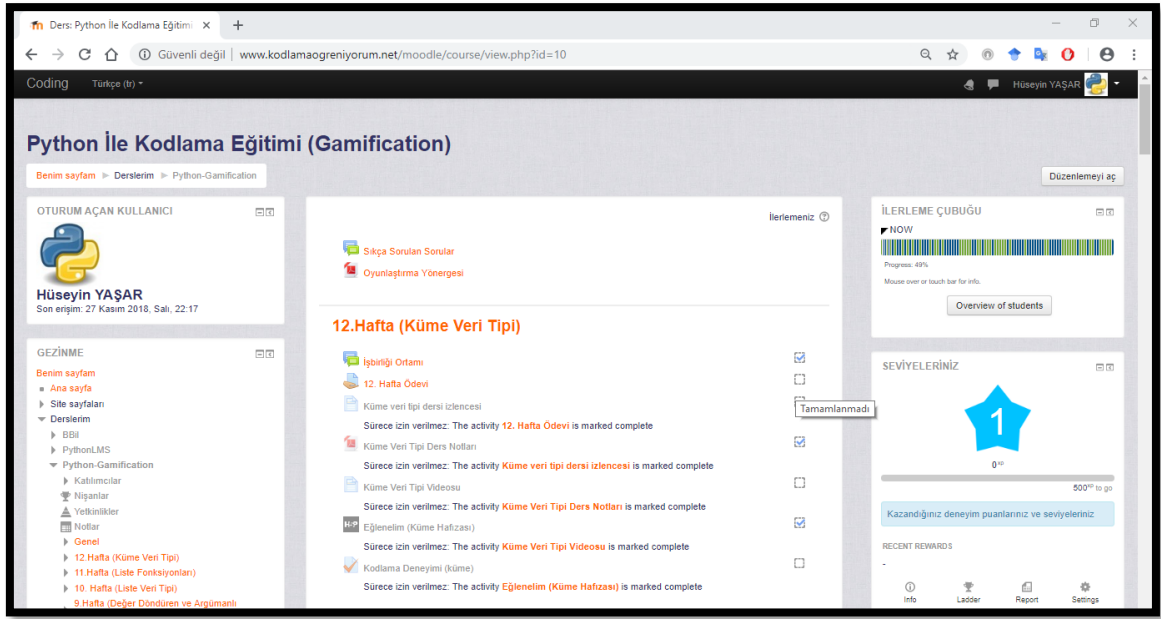
ve sınavlar Moodle ÖYS ortamı kullanılarak sunulmuştur. Yüz yüze öğrenme ortamında yapılan kodlama eğitimi ile birlikte öğrenciler çevrimiçi öğrenme ortamında sunulan kaynakları kullanmış ve verilen görevleri yerine getirmiştir. Ayrıca işbirliği amacıyla oluşturulan forum ortamı tüm öğrencilerin kullanımına sunulmuştur.

Ön hazırlık eğitimin bitiminde, araştırmada kullanılacak öğrenme ortamları kullanılabilir duruma getirilmiştir. Öğrenme ortamlarında kullanılan; öğretim materyalleri, eğlenceli etkinlikleri, kodlama deneyimi etkinlikleri, oyunlaştırma tasarımı süreçleri ve oyunlaştırma bileşenleri uzman görüşü alınarak pilot uygulama için hazır hale getirilmiştir.

Pilot uygulama, 2017-2018 eğitim öğretim yılının ikinci döneminde 2 hafta süreyle uygulanmıştır. Pilot uygulamadaki amaç; deneysel işlem sürecinde deney gruplarında kullanılan, farklı öğrenme ortamlarıyla ilgili olası aksaklıkların belirlenerek, öğrencilerin öğrenme ortamlarına uyumlarının sağlanmasıdır. Pilot uygulama süreci, deneysel işlem süreciyle bir bütün halinde yürütülmüş ve bitiminde öğretim materyalleri ve oyunlaştırma bileşenleriyle ilgili belirlenen aksaklıklar uzman görüşlerine başvurularak giderilmiştir. Pilot uygulama ve deneysel süreçlerin nasıl yürütüldüğüne yönelik OÇÖO ders planları Ek 8'de, OYYÖO ders planları Ek 9'da, ÇÖO ders planları Ek 10'da ve YYÖO ders planları Ek 11'de sunulmuştur.

### **3.5.2. Oyunlaştırılmış çevrimiçi öğrenme ortamı**

OÇÖO'da kodlama eğitimi alacak öğrenciler, birinci deney grubumuzu oluşturmaktadır. Bu grupta kodlama eğitimi oyunlaştırma bileşenlerinin kullanıldığı Moodle ÖYS ortamı kullanılarak yürütülmüş ve görüntüsü Resim 4' te sunulmuştur.

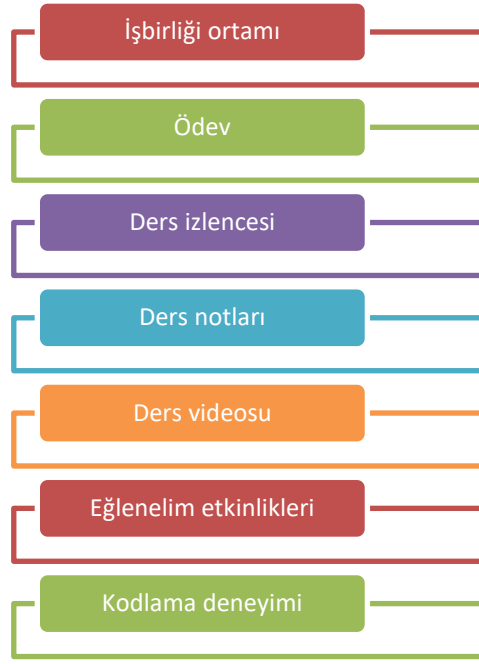


Resim 4. OÇÖO ekran görüntüsü

Resim 4'te görüldüğü gibi OÇÖO'nun orta kısmında öğrencilerin sorularını yöneltebilecekleri *Sıkça Sorulan Sorular* bölümü, öğrencilerin oyunlaştırma süreci hakkında bilgi sahibi olmalarına olanak sağlayan *Oyunlaştırma Yönergesi* ve *Ders materyalleri* öğrencilere sunulmuştur. Ekranın sol kısmında menülere yer verilmiş, sağ kısımda ise oyunlaştırma öğeleri sunulmuştur.

### 3.5.2.1.Öğretim materyalleri

OÇÖO'da öğrencilere haftanın kazanımlarına bağlı olarak hazırlanmış öğretim materyalleri sunulmuş ve uygulama süreci kapsamında haftalık olarak kullanıma açılmıştır. Tüm öğretim materyallerinin öğrenciler tarafından kullanımını sağlamak amacıyla çeşitli kısıtlamalar kullanılmıştır. Bunlar; bir etkinliği tamamlamadan diğer etkinliğe geçememe, haftanın dersi tamamlamadan bir sonraki haftanın dersine geçememe ve kodlama deneyimi etkinliklerine şifre ile erişebilme şeklindedir. OÇÖO'da kullanılan öğretim materyalleri Şekil 8'de sunulmuştur.



Şekil 8. OÖO öğretim materyalleri

*İşbirliği ortamı:* Öğrencilerin işbirliği içerisinde çalışmalarına, kendi aralarında ve ders öğretmeniyle bilgi alış verişinde bulunmalarına olanak sağlayan işbirliği ortamı, haftalık olarak sunulan öğretim materyallerinin üst kısmında yer almaktadır. Moodle ÖYS forum eklentisi kullanılarak hazırlanan ortamda, öğrenciler kendi aralarında ve ders öğretmeniyle iletişim halinde bulunmuştur. Ayrıca öğrenciler kendi aralarında ve ders öğretmeniyle Moodle ÖYS mesajlaşma eklentisi ile özel olarak da iletişim haline geçebilmişlerdir.

*Ödev:* Öğrencilere haftalık olarak bir önceki haftanın konusuyla ilgili ödevler verilmiştir. Verilen ödevlerdeki amaç öğrencilerin öğrendikleriyle ilgili kodlama uygulaması yaparak öğrenilenlerin pekiştirilmesini sağlamaktır. Öğrenciler, verilen ödevleri yaparak bir sonraki ders materyallerine erişebilmişlerdir. Verilen ödevler ders tamamlandıktan sonra ders öğretmene gönderilmiş ve ders öğretmeni tarafından kontrol edilerek geri bildirimler sağlanmıştır.

*Ders izlencesi:* Haftanın dersinde öğrencilere kazandırılması düşünülen kazanımlarla ilgili öğrencilerin bilgi sahibi olmalarını sağlamak amacıyla ders izlencesi bölümü hazırlanmıştır. Öğrenciler bu bölümü okuduktan sonra bir sonraki aşama olan ders notları aşamasına geçebilmişlerdir.

*Ders notları:* Haftanın konusuyla ilgili olarak öğrencilere elektronik kitap halinde ders notları sunulmuştur. Ders notlarında konu anlatımıyla birlikte örnek kodlama uygulamalarına yer verilmiştir. Ders notları hazırlanırken ortaöğretim bilgisayar bilimi kur-1 ders kitabı (Gülbahar, Kalelioğlu ve Karataş, 2017), diğer kaynak kitaplar (Aksoy, 2017) ve internet kaynakları (Özgül, 2017; Coşkun, 2017; İnce, 2017) kullanılmış ve 12 haftalık ders notları Ek 12'de sunulmuştur. Ders notlarını okuyan öğrenciler bir sonraki öğretim materyali olan ders videosuna erişebilmiştir.

*Ders videoları:* Haftanın konusuyla ilgili olarak ders videoları hazırlanmıştır. Ders videoları hazırlanırken, Adobe Captivate 9 programı kullanılmış olup ders planlarında belirtilen kazanımlar dikkate alınmıştır. OYYÖO ve YYÖO için hazırlanan ders sunumları Adobe Captivate 9 programına aktarılarak seslendirilmiş ve örnek kodlama uygulamaları ise programın ekran yakalama özelliği kullanılarak hazırlanmıştır. Hazırlanan 12 haftalık ders videolarının çeşitli bölümlerine şifreler yerleştirilmiş ve öğrencilerin videoyu seyretmeden geçmeleri önlenmiştir. Ders videolarına yerleştirilen şifreler Tablo 4'te sunulmuştur.

Tablo 4

*Ders Videolarında Kullanılan Şifreler*

Haftalar	Şifreler	Haftalar	Şifreler
1.Hafta	jupyter	7.Hafta	defdef
2.Hafta	strings	8.Hafta	parametre
3.Hafta	float	9.Hafta	argüman
4.Hafta	python2018	10.Hafta	listeler
5.Hafta	ifelse	11.Hafta	append
6.Hafta	while	12.Hafta	return

Öğrenciler, Tablo 4'te sunulan ders videosu şifrelerini kullanarak kodlama deneyimi etkinliklerine erişmişlerdir. Dolayısıyla öğrencilerin, ders videolarındaki şifreleri

kullanmadan haftanın son öğretim materyali olan kodlama deneyimi etkinliklerine erişme ve haftanın dersini tamamlama olanakları bulunmamaktadır.

*Eğlenelim etkinliği:* OÇÖO'ya oyunlaştırma kapsamında eğlence unsurlarının dâhil edilmesi amacıyla haftalık olarak eğlenelim etkinlikleri planlanmıştır. Bu etkinliklerdeki amaç anlatılan konuyla ilgili önemli noktaların eğlenerek tekrar edilmesidir. Kullanılan eğlenelim etkinlikleri Tablo 5'te sunulmuştur.

Tablo 5

*OÇÖO'da Kullanılan Eğlenelim Etkinlikleri*

Uygulama haftası	Eğlenelim Etkinliği	Uygulama Haftası	Eğlenelim Etkinliği
1.Hafta	Sürükle Bırak Etkinliği	7.Hafta	Şifreli Bulmaca Etkinliği
2.Hafta	Resmi Yorumla Etkinliği	8.Hafta	Milyoner Etkinliği
3.Hafta	Hafıza Egzersizi Etkinliği	9.Hafta	Sudoku Etkinliği
4.Hafta	Bilgi Yarışması Etkinliği	10.Hafta	Cevabı Sende Etkinliği
5.Hafta	Kod Bulmaca Etkinliği	11.Hafta	Metot Bulmaca Etkinliği
6.Hafta	Kod Asmaca Etkinliği	12.Hafta	Küme Hafızası Etkinliği

Tablo 5'te sunulan eğlenelim etkinlikleriyle ilgili ayrıntılı bilgiler Ek 13'te sunulmuştur.

*Kodlama deneyimi:* OÇÖO'da haftanın konusuyla ilgili kazanım düzeylerini ölçmek amacıyla kodlama deneyimi etkinlikleri kullanılmıştır. Öğrenciler öncelikli olarak kodlama deneyimi etkinliklerine, ders videolarına yerleştirilen şifreleri girerek erişmiş ve soruları cevaplayarak kodlamışlardır. OÇÖO'da öğrencilerin kod yazabilmelerine olanak sağlayan Moodle ÖYS eklentisi olan "Code Runner" kullanılmıştır. Code runner ortamı öğrencilerin python kodlarını yazmalarına ve test etmelerine olanak sağlamaktadır. Öğrenciler soruları kodladıktan sonra ders öğretmenine göndermiş ve ders öğretmeni tarafından kontrol edilmiştir. Kontrol sonrasında ise ders öğretmeni tarafından öğrencilere geri bildirim

sağlanıp kodlama deneyimi etkinliklerini başarılı olarak tamamlayan öğrenciler bir sonraki derse hazır duruma getirilmiştir. Kodlama deneyimi etkinliklerini başarı ile tamamlamayan öğrenciler için ise yeni haftanın dersi açılmamıştır. Kullanılan kodlama deneyimi etkinlikleri Ek 14'te sunulmuştur.

### 3.5.2.2.Oyunlaştırma tasarım süreci

Oyunlaştırmanın yapısal ve içerik olmak üzere iki türünün olduğu Kapp (2013) tarafından ifade edilmiştir. Yapısal oyunlaştırmada öğrencilerin içeriği kullanmaları için puan, ödül, rozet vb. oyunlaştırma bileşenleriyle motive edilerek ödüllendirilir. İçeriğin yapısında hiçbir değişiklik yapılmaz. İçerik oyunlaştırmada ise içeriğin oyuna benzer bir hale getirilmesi vardır. Örneğin içeriğin hikâyeleştirilerek sunulması gibi (Kapp, 2013). Araştırmada içerik üzerinde hiçbir değişikliğe gidilmeyerek yapısal oyunlaştırma kullanılmıştır.

Yapısal oyunlaştırma sürecinin tasarlanmasında Werbach ve Hunter (2012:85) tarafından ortaya konulan ve 6 aşamadan oluşan oyunlaştırma tasarım modeli izlenmiştir. D6 olarak da ifade edilen oyunlaştırma tasarım modeli Şekil 9'da sunulmuştur.



Şekil 9. OÇÖO oyunlaştırma tasarım modeli

*Hedefleri belirleme:* Şekil 9'da sunulan oyunlaştırma tasarım modelinin ilk aşamasında oyunlaştırma kullanımındaki genel hedefler belirlenir. OÇÖO'da oyunlaştırma kullanılmasının genel hedefleri;

- Öğrencilerin akademik başarılarının arttırılması,
- Derse yönelik motivasyonlarının arttırılması,

- Kodlamaya karşı olumlu tutum geliştirilmesi şeklindedir.

*Hedef davranışları belirleme:* Oyunlaştırma tasarım modelinin bu aşamasında genel hedefler kapsamında ortaya çıkması düşünülen davranışlar belirlenir. OÇÖO'da hedef davranışlar;

- ABT son test puanını ön test puanına göre yükseltmek ve izleme testinde de devam ettirmek.
- Kodlama deneyimi etkinliklerinde programları başarılı olarak kodlamak.
- Eğlenelim etkinliklerini başarı ile tamamlamak.
- Ödevleri düzenli olarak yapmak.
- İşbirliği ortamlarında yorum yapmak ve arkadaşlarına yardım etmek.
- MÖ son test puanını ön test puanına göre yükseltmek ve izleme testinde de devam ettirmek.
- PDYTÖ son test puanını ön test puanına göre yükseltmek ve izleme testinde de devam ettirmek.

*Oyuncuları tanımlama:* Yılmaz (2015:110) oyunlaştırılmış ortamlarda her oyuncunun katılımının, beklentisinin ve motivasyonunun farklı olduğunun bilinmesi ve oyuncu kitlesinin iyi analiz edilmesi gerektiğini ifade etmiştir. Bu nedenle oyunlaştırılmış tasarımın hedef kitleye uygun tasarlanması amacıyla oyuncuların özelliklerinin belirlenmesi gerekmektedir. Bartle (1996) oyuncu tiplerini 4 gruba ayırmış ve bunları; yenilmezler, başarılar, sosyalleşenler ve kâşifler şeklinde ifade etmiştir. Marczewski (2015) ise Bartle (1996) tarafından ortaya konulan oyuncu tiplerinin daha gelişmiş bir halini ortaya koyarak 6 tipte oyuncu tanımlamıştır. Bunlar; sosyalleşenler, özgür ruhlar, başarılar, yardım severler, oyuncular ve yıkıcılar şeklindedir. OÇÖO oyunlaştırma tasarımı aşamasında Marczewski (2015) tarafından ortaya konulan oyuncu tiplerinden yararlanılmıştır. Yılmaz (2015:112) bu oyuncu tiplerinin oyunlaştırmaya daha yakın olduğunu belirtmiştir. OÇÖO tasarlanırken öğrenciler için kullanılan oyuncu tipleri ve oyunlaştırma bileşenleri Şekil 10'da sunulmuştur.



Yardım severler	Başaranlar	Oyuncular
<ul style="list-style-type: none"> <li>• İşbirliği ortamı</li> </ul>	<ul style="list-style-type: none"> <li>• Seviyeler</li> <li>• İlerleme durumu</li> </ul>	<ul style="list-style-type: none"> <li>• Puanlar</li> <li>• Hediyeler</li> <li>• Lider tahtası</li> <li>• Rozet</li> </ul>

Şekil 10. OÇÖO tasarlanırken kullanılan oyuncu (öğrenci) tipleri

Şekil 10'da da görüldüğü gibi OÇÖO tasarlanırken Marczewski (2015) tarafında ortaya konulan 6 oyuncu tipinin 3 tanesi kullanılmıştır. Öğrenme ortamı oyunlaştırılırken bu üç oyuncu tipi dikkate alınarak oyunlaştırma bileşenleri kullanılmıştır.

*Etkinlik döngülerini planlama:* Bu aşamada oyunlaştırılmış tasarım sürecinde kullanılan oyunlaştırma bileşenlerinin nasıl kullanılacağı belirlenir. OÇÖO'da kullanılan oyunlaştırma tasarım sürecinde kullanılan oyunlaştırma bileşenlerinin işleyişiyle ilgili oyunlaştırma yönergesi hazırlanmıştır. Oyunlaştırma yönergesi Moodle ÖYS ders ana sayfasına yerleştirilmiş ve deneysel işlem sürecinde öğrencilerin kullanımına sunulmuştur. Öğrenciler, oyunlaştırma bileşenlerinin kullanımıyla ilgili tüm bilgilere oyunlaştırma yönergesinden erişmiştir. OÇÖO'da kullanılan oyunlaştırma yönergesi Ek 15'te sunulmuştur.

*Eğlenceyi Unutma!:* Oyunlaştırma tasarımının en önemli aşamalarında biride eğlence unsurlarının tasarıma dahil edilmesidir. Eğlence unsurları tasarıma dâhil edilirken, hangi tür eğlencenin kullanılacağı belirlenmelidir. Lazzaro (2004) eğlence türleriyle ilgili "The 4 Keys to FUN" olarak ifade ettiği 4 tür ortaya koymuştur. Bunlar; zor eğlence, kolay eğlence, insanlarla eğlence ve ciddi eğlencedir. OÇÖO tasarımında ciddi eğlence kullanılacaktır. Yılmaz (2015:95) ciddi eğlencenin, eğlenmenin yanında davranış değişikliği meydana getiren ve öğrenmeyi hedef alan eğlence türü olduğunu ifade etmiştir.

*Uygun araçları kullanma:* OÇÖO oluştururken çevrimiçi öğrenme ortamı olarak Moodle ÖYS kullanılmıştır. Moodle; 224 ülkede, 104.573 kayıtlı site, 19.282.619 kurs ve 163.341.776 kullanıcı sayısı ile dünya genelinde yaygın olarak kullanılan bir çevrimiçi öğrenme ortamıdır (Moodle, 2019). Moodle çevrimiçi öğrenme ortamının birçok avantajı bulunmaktadır. Bunlar;

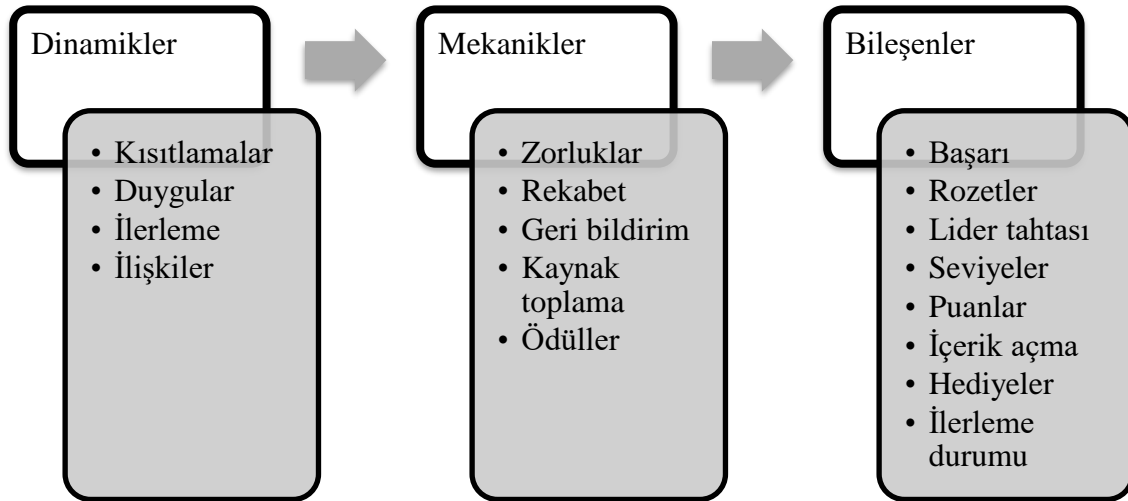
- Açık kaynak kodlu ücretsiz bir ortam olması,
- Öğrenci-öğretmen ve öğrenci-öğrenci etkileşiminde kolaylık sağlaması,

- Eş zamanlı ve eş zamansız mesaj ve sohbet ortamlarına sahip olması,
- Grup iletişimini desteklemesi,
- Tartışma panolarının bulunması,
- Sınav ve not dağıtım sisteminin hızlı olması (Chung ve Ackerman, 2015:217-218) şeklinde sıralanabilir.

Moodle çevrimiçi öğrenme ortamının daha etkin kullanılması amacıyla 1606 eklentiye içinde barındırmaktadır (Moodle, 2019). Araştırmada OÇÖO ve ÇÖO tasarlanırken forum, nişanlar, ranking block, level up, completion progrees, code runner, xp-experience point vb. eklentilerden yararlanılmıştır.

### 3.5.2.3. Oyunlaştırma bileşenleri

OÇÖO'da kullanılan oyunlaştırma bileşenleri belirlenirken Werbach ve Hunter (2012:78-81) tarafından ortaya konulan dinamikler, mekanikler ve bileşenler kullanılmış ve Şekil 11'de sunulmuştur.



Şekil 11. OÇÖO'da kullanılan dinamikler, mekanikler ve bileşenler

Şekil 11'de görüldüğü gibi araştırmada 4 dinamik kullanılmıştır. Yılmaz (2015) dinamiklerin, oyun kurallarından bağımsız tasarım çözümleri olduğunu ve oyunlaştırma ortamının bütününe etkileyen unsurlar olduğunu ifade etmiştir. Werbach ve Hunter (2012)

dinamiklerin oyunlaştırılmış sistemde görülmeyen fakat dikkate alınması ve yönetilmesi gereken önemli unsurlar olduğunu belirtmiştir. Bu nedenle araştırmada kısıtlamalar, duygular, ilerleme ve ilişkiler dinamikleri kullanılmıştır. Araştırmadaki kısıtlamalar; öğretim materyallerinin birini tamamlamadan diğerine erişimin olmaması, kodlama deneyimi etkinliklerine şifre ile erişilebilmesi, kodlama deneyimi etkinliklerinde yeterli başarı puanına erişilmeden bir sonraki haftanın dersinin açılmaması şeklindedir. Araştırmadaki duygular; puan, rozet, hediye kazanımından mutlu olma veya hayal kırıklığına uğrama, lider tahtasındaki sırasından mutlu olma, atlanılan seviye ve kazanılan rütbelere mutlu olma, lider tahtasındaki konumunu ve arkadaşlarının konumunu merak etme, arkadaşlarının rozet durumlarını merak etme şeklinde ifade edilebilir. Araştırmadaki ilerleme; belirli puanlara ulaşıldığında seviye ve rütbe kazanma, ilişkiler ise işbirliği ortamı kullanılarak öğrencilerin sosyal ilişkiler kurması ve yardımlaşması şeklindedir.

Şekil 11'de görüldüğü gibi araştırmada 5 mekanik kullanılmıştır. Yılmaz (2015) mekaniklerin tam ortada yer aldığını ve dinamiklerden aldığı referans ile son kullanıcının gördüğü bileşenleri beslediğini ifade etmiştir. Ayrıca oyuncunun etkileşime geçtiği bileşenleri çalıştıran, motive eden ve ilerletenin mekanikler olduğunu belirtmiştir. Araştırmada zorluklar, rekabet, geri bildirim, kaynak toplama ve ödüller olmak üzere 5 mekanik kullanılmıştır. Zorluklar mekaniği; ödevleri başarı ile yapma, kodlama deneyimi etkinliklerini başarı ile tamamlama durumlarında, rekabet mekaniği; ödül kazanma, rozet kazanma, seviye atlama, rütbe kazanma ve lider tahtasında üst sıralarda yer alma durumlarında, geri bildirim mekaniği; her aşamada yapılanların puanlanması, belirli kriterlerin sağlanmasında rozet verilmesi durumunda, kaynak toplama mekaniği; rozetlerin biriktirilmesi ile hediye kazanma durumunda ve ödüller mekaniği ise lider tahtasındaki konumuna ve rozet sayısına göre hediye kazanma durumunda kullanılmıştır.

Oyunlaştırma bileşenleri ise öğrencilerin direkt etkileşime geçtiği, birçok dinamik ve mekanik ile bağlantılı olan tasarım unsurlarıdır (Yılmaz, 2015). OÇÖO'da 8 tane oyunlaştırma bileşeni kullanılmıştır. Bunlar; başarı, rozetler, lider tahtası, seviyeler, puanlar, içerik açma, hediyeler ve ilerleme durumu şeklindedir.

*Başarı:* OÇÖO'da haftalık olarak dersin hedefleri öğrencilere sunulmuştur. Öğrencilerin her hafta belirlenen hedefleri başarıları istenmiştir. Hedeflenen davranışların ölçülmesinde kodlama deneyimi etkinlikleri kullanılmış ve belirlenen başarı puanını alamayan öğrencilerin dersi tekrar çalışmaları sağlanmıştır.

*Rozetler:* Araştırmada kodcu, işbirlikçi, düzenli, ve etkinlikçi olmak üzere 4 rozet kullanılmıştır. OÇÖO'da rozetler Moodle "Nişanlar" eklentisi kullanılarak entegre edilmiştir. Moodle ortamında her hafta 4 rozet olmak üzere toplam 12 hafta için 48 rozet kullanılmıştır.



*Şekil 12.* OÇÖO'da kullanılan rozetler

Şekil 12'de sunulan rozetlerden kodcu rozeti; kodlama deneyimi etkinliklerini başarı ile tamamlayan öğrencilere, işbirlikçi rozeti; işbirliği ortamına katılım gösteren öğrencilere, düzenli rozeti; haftalık sunulan öğretim materyallerini düzenli olarak tamamlayan ve görevleri yerine getiren öğrencilere, etkinlikçi rozeti ise eğlenelim etkinliklerine katılım göstererek başarı sağlayan öğrencilere verilmiştir. OÇÖO'da öğrencilerin kazandıkları rozet sayıları Ek 16'da sunulmuştur.

*Lider tahtası:* OÇÖO'da öğrenciler, aldıkları puanlara göre haftalık, aylık ve genel olarak üç aşamalı kullanılan lider tahtasında sıralanmıştır. Lider tahtası olarak Moodle eklentilerinde "Rankink block" kullanılmıştır. Ders ana sayfasının sağ paneline yerleştirilen lider tahtasında öğrenciler hem kendi hem de arkadaşlarının sıralamalarını ve puanlarını görmüştür. OÇÖO'da kullanılan lider tahtasının ekran görüntüsü Resim 5'te sunulmuştur.



Sıra	Tam ad	Puanlar
1	İBRAHİM ETHEM	268.3
2	RABİA	265.4
3	ESRA	265.1
4	EYLÜL	264.7
5	ZEHRA	264.3
6	İREM SUDE	263.7
7	ÜNZÜLE	262.6
8	DİLAN	262.5
9	DİLARA	262.4
10	NAZAR	261.7

Resim 5. OÇÖO'da kullanılan lider tahtası

Resim 5'te görüldüğü gibi lider tahtasında en yüksek puanlı 10 öğrencinin isimleri yer almaktadır. Diğer öğrencilerin sıralamalarına ise "Tüm Sıralamaları Gör" düğmesiyle erişilebilmektedir.

*Seviyeler:* OÇÖO'da öğrenciler, kazandıkları deneyim puanlarına göre belirli seviyeler ve rütbeler elde etmiştir. Toplam 10 seviye ve 5 rütbenin kullanıldığı OÇÖO'da öğrenci seviyeleri ve rütbeleri Moodle oyunlaştırma eklentisi olan "Level up" panelinde sunulmuştur. OÇÖO'da kullanılan seviyeler ve rütbeler Resim 6'da sunulmuştur.

1	0 <sup>xP</sup>	
2	500 <sup>xP</sup>	Yeni Kodlayıcı
3	1.000 <sup>xP</sup>	Acemi Kodlayıcı
4	2.000 <sup>xP</sup>	Acemi Kodlayıcı
5	2.500 <sup>xP</sup>	Kodlayıcı
6	3.000 <sup>xP</sup>	Kodlayıcı
7	3.500 <sup>xP</sup>	İyi Kodlayıcı
8	4.000 <sup>xP</sup>	İyi Kodlayıcı
9	6.000 <sup>xP</sup>	Uzman Kodlayıcı
10	6.500 <sup>xP</sup>	Uzman Kodlayıcı

Resim 6. OÇÖO'da kullanılan seviyeler ve rütbeler

Resim 6'da görülen seviyelere göre tüm öğrencilerin 5. seviye ve üstündeki seviyelerde yer aldığı, rütbelere göre ise 10 öğrenci uzman kodlayıcı rütbesine, 14 öğrenci iyi kodlayıcı rütbesine ve 9 öğrenci kodlayıcı rütbesine erişmiştir. Öğrencilerin deneyim puanları, seviyeleri ve rütbeleri Ek 17'de sunulmuştur.

*Puanlar:* OÇÖO'da iki puan türü kullanılmıştır. Birincisi öğrencilerin öğrenme materyallerinden aldığı puanlar diğeri ise OÇÖO deneyimlerinden alınan deneyim puanlarıdır. Öğretim materyallerinden alınan puanlara göre lider tahtası oluşturulmuş ve öğrenciler bu puanlara göre hediyeler kazanmıştır. Deneyim puanlarına göre ise öğrenciler seviyeler atlamış ve rütbelere kazanmıştır. Öğretim materyallerinden kazanılan puanlar; işbirliği ortamına katılım, ödev yapma, ders izlencesini okuma, ders notlarını inceleme, ders videolarını seyretme, eğlenelim etkinliklerine katılım ve kodlama deneyimi etkinliklerine katılım şeklindedir.

*İçerik açma:* OÇÖO'da 3 içerik kısıtlaması kullanılmıştır. İlki, öğretim materyalleri öğrencilere sıralı olarak sunulmuş ve bir materyal tamamlanmadan diğeri materyal öğrencilere açılmamıştır. İkincisi ise kodlama deneyimi etkinliklerine erişim, ders

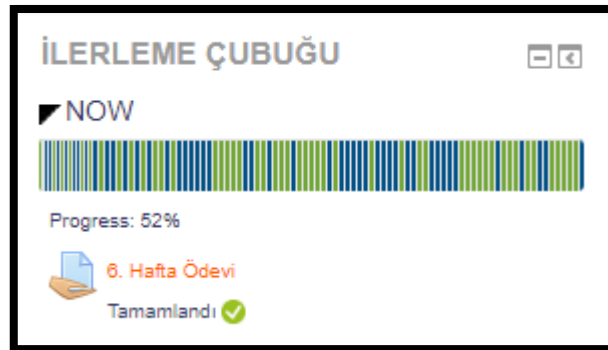
videolarına yerleştirilen şifrelerle yapılmıştır. Öğrenciler ders videosuna yerleştirilen şifreyi girerek kodlama deneyimi etkinliklerine erişmiştir. Üçüncüsü de kodlama deneyimi etkinliklerinde istenen başarı puanını alamayan öğrenciler bir sonraki haftanın dersine erişememiştir. Kodlama deneyimi etkinliklerinde başarı sağlayan öğrencilere ise bir sonraki haftanın içerikleri açılmıştır.

*Hediyeler:* OÇÖO'da öğrenciler, lider tahtasındaki konumlarına göre ve kazandıkları rozet sayılarına göre hediyeler almıştır. Lider tahtasında ödüller verilirken aylık sıralamalar dikkate alınmıştır. Her ay lider tahtasında birinci sırada bulunan öğrencilere hediyeler verilmiştir. Bu hediyeler:

- 1.Ay (1-4 Haftalar): Python ajandası
- 2. Ay (5-8 Haftalar): Flash bellek
- 3. Ay (9-12 Haftalar): Kulaklık şeklindedir.

Kodcu, işbirlikçi, düzenli ve etkinlikçi rozetlerinden 8 ve üzeri kazanan öğrenciler üzerinde rozet resminin bulunduğu kupa bardaklarla ödüllendirilmiştir. 5 öğrenci kodcu rozetinden, 9 öğrenci işbirlikçi rozetinden, 5 öğrenci düzenli rozetinden ve 19 öğrenci etkinlikçi rozetinden hediye kazanmıştır.

*İlerleme durumu:* OÇÖO'da öğrenciler, ilerleme durumlarını ders ana sayfasında sunulan ilerleme çubuğundan takip etmiştir. İlerleme çubuğu olarak Moodle eklentisi olan "Completion Progress" kullanılmıştır. OÇÖO'da kullanılan ilerleme çubuğunun ekran görüntüsü Resim 7'de sunulmuştur.



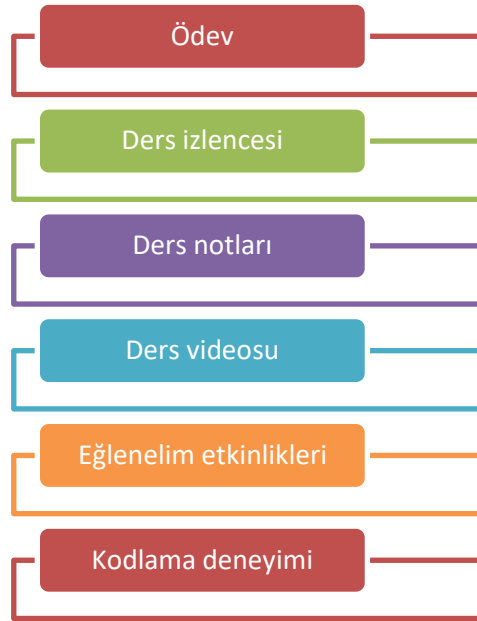
Resim 7. OÇÖO'da kullanılan ilerleme çubuğu

### 3.5.3. Oyunlaştırılmış yüz yüze öğrenme ortamı

OYYÖO'da kodlama eğitimi alacak öğrenciler ikinci deney grubumuzu oluşturmaktadır. Bu grupta kodlama eğitimi oyunlaştırılmış yüz yüze öğrenme ortamında verilmiştir.

#### 3.5.3.1. Öğretim materyalleri

OYYÖO'da haftanın kazanımlarına uygun hazırlanmış öğretim materyalleri kullanılmış ve Şekil 13'te sunulmuştur.



Şekil 13. OYYÖO öğretim materyalleri

*Ödev:* Haftalık olarak öğrencilere bir önceki haftanın konusuyla ilgili ödevler verilmiş ve yeni haftanın konusuna başlanmadan kontrol edilmiştir. Verilen ödevlerdeki amaç öğrencilerin öğrendikleriyle ilgili kodlama uygulaması yaparak öğrenilenlerin pekiştirilmesini sağlamaktır.

*Ders izlencesi:* Haftanın dersinde öğrencilere kazandırılması düşünülen kazanımlar hakkında, öğrencilerin bilgi sahibi olmalarını sağlamak amacıyla hazırlanan ders izlencesi sınıf panosuna asılmış ve ödev kontrolünden sonra öğrencilere okunmuştur.

*Ders notları:* Haftanın konusuyla ilgili olarak öğrencilere elektronik kitap halinde ders notları sunulmuş, ders notlarında konu anlatımıyla birlikte örnek kodlama uygulamalarına



yer verilmiştir. Ders notları etkileşimli tahtaya yüklenmiş ve basılı olarak öğrencilere verilmiştir. Tüm öğrenme ortamlarında aynı ders notları kullanılmıştır (bkz. Ek 12).

*Ders sunumları:* Haftanın konusuyla ilgili olarak ders sunumları hazırlanıp, sunumlar hazırlanırken ders planlarında belirtilen kazanımlar dikkate alınmıştır. Araştırma süresince her hafta 1 tane olmak üzere 12 ders sunumu kullanılmış ve sunumlar MS Power Point programı kullanılarak hazırlanmıştır. OYYÖO ders anlatımı sunumlar kullanılarak yapılmış, kodlama işlemleri ise Python IDLE ortamı kullanılarak yürütülmüştür.

*Eğlenelim etkinliği:* OYYÖO'ya oyunlaştırma kapsamında, eğlence unsurlarının dâhil edilmesi amacıyla haftalık eğlenelim etkinlikleri planlanmıştır. Bu etkinliklerdeki amaç anlatılan konunun önemli noktalarının eğlenerek tekrar edilmesidir. Kullanılan eğlenelim etkinlikleri Tablo 6'da sunulmuştur.

Tablo 6

*OYYÖO'da Kullanılan Eğlenelim Etkinlikleri*

Uygulama haftası	Eğlenelim Etkinliği	Uygulama Haftası	Eğlenelim Etkinliği
1.Hafta	Kes-Yapıştır Etkinliği	7.Hafta	Şifreli Bulmaca Etkinliği
2.Hafta	Çek Kodunu Söyle Çıktım	8.Hafta	Plickers Etkinliği
3.Hafta	Kartları Eşleştirme Etkinliği	9.Hafta	Sudoku Etkinliği
4.Hafta	Bilgi Yarışması Etkinliği	10.Hafta	Flipquiz Etkinliği
5.Hafta	Program Çalıştırma Etkinliği	11.Hafta	Metot Bulmaca Etkinliği
6.Hafta	Kahoot Etkinliği	12.Hafta	Metot Çek-Kodla Etkinliği

Tablo 6'da verilen eğlenelim etkinlikleriyle ilgili ayrıntılı bilgiler Ek 18'de sunulmuştur.

*Kodlama deneyimi:* OYYÖO'da haftanın konusuyla ilgili kazanım düzeylerini ölçmek amacıyla kodlama deneyimi etkinlikleri kullanılmıştır. Öğrencilere, hazırlanan etkinlikler basılı olarak verilmiş ve öğrencilerin IDLE ortamında kodlamaları istenmiştir. Öğrenciler doğru cevapladıkları sorulara göre puan kazanmıştır. Tüm öğrenme ortamlarında aynı kodlama deneyimi etkinlikleri kullanılmıştır (bkz. Ek 14).

### 3.5.3.2. Oyunlaştırma tasarım süreci

OYYÖO'da içerik üzerinde hiçbir değişikliğe gidilmeyerek yapısal oyunlaştırma kullanılmıştır. Yapısal oyunlaştırma sürecinin tasarlanmasında, Werbach ve Hunter (2012:85) tarafından ortaya konulan ve 6 aşamadan oluşan oyunlaştırma tasarım modeli izlenmiştir. D6 olarak da ifade edilen oyunlaştırma tasarım modeli Şekil 14'te sunulmuştur.



Şekil 14. OYYÖO oyunlaştırma tasarım modeli

*Hedefleri belirleme:* Şekil 14'te sunulan oyunlaştırma tasarım modelinin ilk aşamasında oyunlaştırma kullanımındaki genel hedefler belirlenir. OYYÖO'da oyunlaştırma kullanımının genel hedefleri:

- Öğrencilerin akademik başarılarının artırılması,
- Derse yönelik motivasyonlarının artırılması,
- Kodlamaya karşı olumlu tutum geliştirilmesi şeklindedir.

*Hedef davranışları belirleme:* Oyunlaştırma tasarım modelinin bu aşamasında genel hedefler kapsamında ortaya çıkması düşünülen davranışlar belirlenir. OYYÖO'da hedef davranışlar:

- ABT son test puanını ön test puanına göre yükseltmek ve izleme testinde de devam ettirmek.
- Kodlama deneyimi etkinliklerinde programları başarılı olarak kodlamak.
- Eğlenceli etkinliklerini başarı ile tamamlamak.
- Ödevleri düzenli olarak yapmak.
- Derse aktif katılım göstererek soruları cevaplamak.
- Derste arkadaşları ile işbirliği içinde çalışarak yardımcı olmak.
- MÖ son test puanını ön test puanına göre yükseltmek ve izleme testinde de devam ettirmek.
- PDYTÖ son test puanını ön test puanına göre yükseltmek ve izleme testinde de devam ettirmek.

*Oyuncuları tanımlama:* OYYÖO'da Marczewski (2015) tarafından ortaya konulan oyuncu tiplerinden yararlanılmıştır. OYYÖO tasarlanırken öğrenciler için kullanılan oyuncu tipleri ve oyunlaştırma bileşenleri Şekil 15'te sunulmuştur.

Yardım severler	Başaranlar	Oyuncular
<ul style="list-style-type: none"> <li>• İşbirliği ortamı</li> </ul>	<ul style="list-style-type: none"> <li>• Seviyeler</li> </ul>	<ul style="list-style-type: none"> <li>• Puanlar</li> <li>• Hediyeler</li> <li>• Lider tahtası</li> <li>• Rozet</li> </ul>

Şekil 15. OYYÖO tasarlanırken kullanılan oyuncu (Öğrenci) tipleri

Şekil 15'te de görüldüğü gibi OYYÖO tasarlanırken Marczewski (2015) tarafında ortaya konulan 6 oyuncu tipinin 3 tanesi kullanılmıştır. Öğrenme ortamı oyunlaştırılırken bu üç oyuncu tipi, dikkate alınarak oyunlaştırma bileşenleri kullanılmıştır.

*Etkinlik döngülerini planlama:* Bu aşamada oyunlaştırılmış tasarım sürecinde, kullanılan oyunlaştırma bileşenlerinin nasıl kullanılacağı belirlenir. Oyunlaştırma bileşenlerinin işleyişiyle ilgili oyunlaştırma yönergesi hazırlanmış ve sınıf panosuna asılarak araştırma sürecinde öğrencilerin kullanımına sunulmuştur. Öğrenciler oyunlaştırma bileşenlerinin

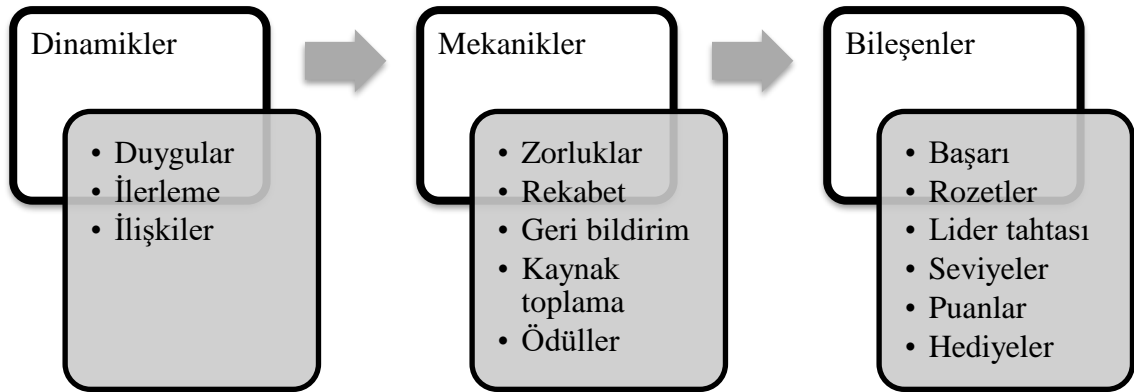
kullanımıyla ilgili tüm bilgilere, sunulan oyunlaştırma yönergesinden erişebilmişlerdir. OYYÖO'da kullanılan oyunlaştırma yönergesi Ek 19'da sunulmuştur.

*Eğlenceyi Unutma!:* OYYÖO tasarımında OÇÖO olduğu gibi ciddi eğlence kullanılmıştır. Yılmaz (2015:95) ciddi eğlencenin eğlenmenin yanında davranış değişiklikleri ve öğrenmeleri hedefleyen eğlence türü olduğunu ifade etmiştir.

*Uygun araçları kullanma:* OYYÖO oluşturulurken öğrenme ortamı olarak yüz yüze öğrenme ortamı kullanılmıştır. Eğlenceli etkinlikleri için çeşitli Web 2.0 araçlarından yararlanılmıştır. Kodlama ortamı olarak Python IDLE ortamı kullanılmıştır. Ayrıca oyunlaştırma bileşenleri, yüz yüze öğrenme ortamına uygun olarak hazırlanıp entegre edilmiş ve OYYÖO oyunlaştırma bileşenleri bölümünde ayrıntılı olarak sunulmuştur.

### 3.5.3.3. Oyunlaştırma bileşenleri

OYYÖO'da kullanılan oyunlaştırma bileşenleri belirlenirken Werbach ve Hunter (2012:78-81) tarafından ortaya konulan dinamikler, mekanikler ve bileşenler kullanılıp Şekil 16'da sunulmuştur.



Şekil 16. OYYÖO'da kullanılan dinamikler, mekanikler ve bileşenler

Şekil 16'da görüldüğü gibi OYYÖO oyunlaştırılırken 3 dinamik kullanılmıştır. Araştırmadaki duygular; puan, rozet, hediye kazanımından mutlu olma veya hayal kırıklığına uğrama, lider tahtasındaki sırasından mutlu olma, atlanılan seviye ve kazanılan rütbelere mutlu olma, lider tahtasındaki konumunu ve arkadaşlarının konumunu merak etme, arkadaşlarının rozet durumlarını merak etme şeklinde ifade edilebilir. Araştırmadaki

ilerleme; belirli puanlara ulaşıldığında seviyeler ve rütbelere kazanma şeklindedir. Araştırmadaki ilişkiler; öğrencilerin işbirliği içerisinde girmesi, sosyal ilişkiler kurması ve yardımlaşması şeklindedir.

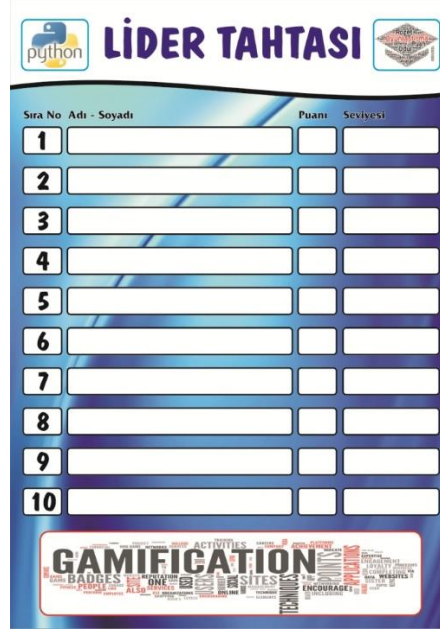
Şekil 16'da görüldüğü gibi araştırmada 5 mekanik kullanılmıştır. Yılmaz (2015) mekaniklerin tam ortada yer aldığını ve dinamiklerden aldığı referans ile son kullanıcının gördüğü bileşenleri beslediğini ifade etmiştir. Ayrıca oyuncunun etkileşime geçtiği bileşenleri çalıştıran, motive eden ve ilerletenin mekanikler olduğunu belirtmiştir. Araştırmada zorluklar, rekabet, geri bildirim, kaynak toplama ve ödüller olmak üzere 5 mekanik kullanılmıştır. Zorluklar mekaniği; ödevleri başarı ile yapma, kodlama deneyimi etkinliklerini başarı ile tamamlama durumlarında, rekabet mekaniği; ödül kazanma, rozet kazanma, seviye atlama, rütbe kazanma ve lider tahtasında üst sıralarda yer alma durumlarında, geri bildirim mekaniği; her aşamada yapılanların puanlanması, belirli kriterlerin sağlanmasında rozetler verilmesi durumlarında, kaynak toplama mekaniği; rozetlerin biriktirilmesi ile hediye kazanma durumunda ve ödüller mekaniği; lider tahtasındaki konumuna ve rozet sayısına göre hediye kazanma durumunda kullanılmıştır.

Oyunlaştırma bileşenleri ise öğrencilerin direkt etkileşime geçtiği, birçok dinamik ve mekanik ile bağlantılı olan tasarım unsurlarıdır (Yılmaz, 2015). OÇÖÖ'da 6 tane oyunlaştırma bileşeni kullanılmıştır. Bunlar; başarı, rozet, lider tahtası, seviye, puan ve hediye şeklindedir.

*Başarı:* OYYÖÖ'da haftalık olarak dersin hedefleri öğrencilere sunulmuştur. Öğrencilerin her hafta belirlenen hedefleri başarıları istenmiştir. Hedeflenen davranışların ölçülmesinde kodlama deneyimi etkinlikleri kullanılmış ve belirlenen başarı puanını alamayan öğrencilere, anlaşılmayan noktalarda destek verilmiştir.

*Rozetler:* OYYÖÖ'da kodcu, işbirlikçi, düzenli ve etkinlikçi olmak üzere 4 rozet kullanılmıştır. Kullanılan rozetler OÇÖÖ kullanılan rozetlerle görsel olarak aynıdır (bkz. Şekil 12). Fakat OYYÖÖ'da rozetler öğrencilere damla rozet olarak verilmiştir. Kodcu rozeti kodlama deneyimi etkinliklerini başarı ile tamamlayan öğrencilere, işbirlikçi rozeti arkadaşlarıyla işbirliği içerisinde giren ve yardımcı olan öğrencilere, düzenli rozeti tüm etkinliklerde başarı gösteren ve etkin katılım sağlayan öğrencilere, etkinlikçi rozeti ise eğlenelelim etkinliklerine katılım göstererek başarı sağlayan öğrencilere verilmiştir. OYYÖÖ'da öğrencilerin kazandıkları rozet sayıları Ek 20'de sunulmuştur.

*Lider tahtası:* OYYÖO'da öğrenciler aldıkları puanlara göre lider tahtasında sıralanmıştır. Lider tahtası olarak 80 cm x 100 cm ebatlarında sert plastikten yapılmış pano kullanılmıştır. Kullanılan lider tahtasında puan ve seviye bilgilerine yer verilmiş ve ekran görüntüsü Resim 8'de sunulmuştur.

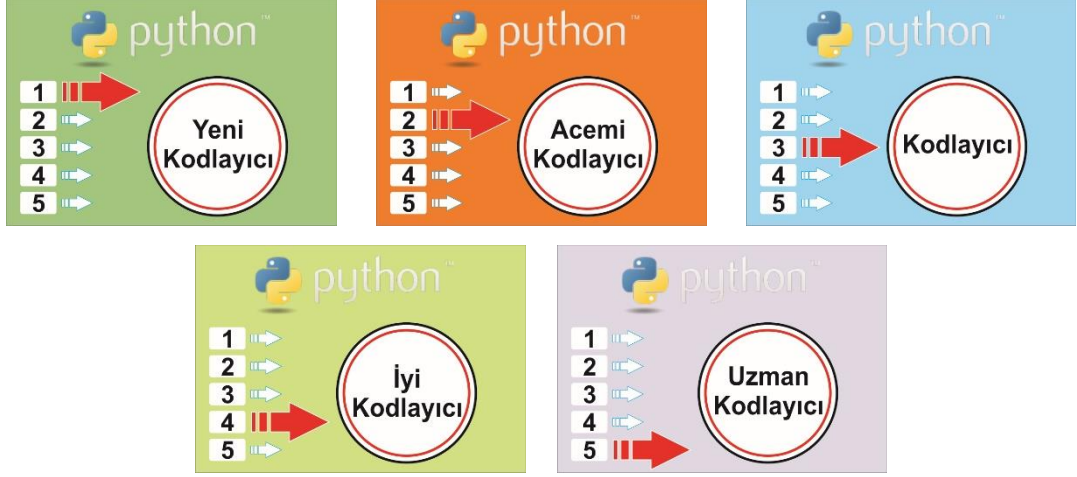


Sıra No	Adı - Soyadı	Puanı	Seviyesi
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			

Resim 8. OYYÖO'da kullanılan lider tahtası

Resim 8'de görülen lider tahtası sınıfta uygun bir yere asılarak öğrencilere sunulmuştur. Haftalık olarak sıralamalar, seviyeler ve puanlar güncellenmiştir.

*Seviyeler:* OYYÖO'da öğrenciler kazandıkları puanlara göre belirli seviyeler ve rütbelere elde etmiştir. Toplam 5 seviye ve 5 rütbenin kullanıldığı OYYÖO'da öğrenciler seviyelerini gösteren boyundan askılı yaka kartlarını ders süresince takmışlardır. OYYÖO'da kullanılan seviyeler ve kartlar Resim 9'da sunulmuştur.



Resim 9. OYYÖO'da kullanılan seviyeler ve rütbeler

Resim 9'daki seviye ve rütbelere göre; araştırma süresince 10 öğrenci uzman kodlayıcı rütbesine, 17 öğrenci iyi kodlayıcı rütbesine ve 6 öğrenci kodlayıcı rütbesine erişmiştir. Puanlar, seviyeler ve rütbeler Ek 21'de sunulmuştur.

*Puanlar:* OYYÖO'da öğrenciler sadece öğretim materyallerinden ve sınıf içerisindeki davranışlarından puan kazanmıştır. Bunlar; ödev yapma, derse katılım, eğlenelim etkinliklerindeki performansları ve kodlama deneyimi etkinliklerindeki başarıları şeklindedir. Öğrenciler, elde edilen puanlara göre seviyeler ve rütbelere kazanmış ve lider tahtasında sunulmuştur.

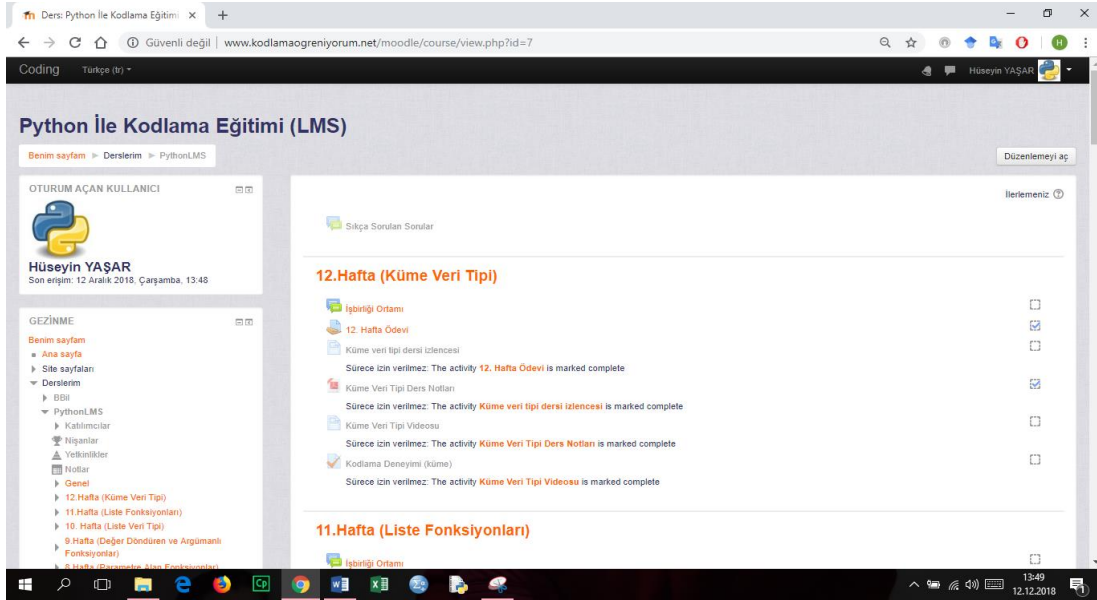
*Hediyeler:* OYYÖO'da öğrenciler, lider tahtasındaki konumlarına göre ve kazandıkları rozet sayılarına göre hediyeler almıştır. Lider tahtasında ödüller verilirken, aylık sıralamalar dikkate alınmıştır. Her ay lider tahtasında birinci sırada bulunan öğrencilere hediyeler verilmiştir. Bu hediyeler;

- 1. Ay (1-4 Haftalar) : Python ajandası
- 2. Ay (5-8 Haftalar): Flash bellek
- 3. Ay (9-12 Haftalar): Kulaklık şeklindedir.

Kodcu, işbirlikçi, düzenli ve etkinlikçi rozetlerinden 8 ve üzeri kazanan öğrenciler, üzerinde rozet resminin bulunduğu kupa bardaklarla ödüllendirilmiştir. 5 öğrenci kodcu rozetinden, 4 öğrenci işbirlikçi rozetinden, 6 öğrenci düzenli rozetinden ve 22 öğrenci etkinlikçi rozetinden gerçek ödül kazanmıştır.

### 3.5.4. Çevrimiçi öğrenme ortamı

ÇÖO'da kodlama eğitimi alacak öğrenciler üçüncü deney grubumuzu oluşturmaktadır. ÇÖO'da kodlama eğitimi Moodle ÖYS ortamında verilmiş ve ortamının görüntüsü Resim 10'da sunulmuştur.



Resim 10. ÇÖO ekran görüntüsü

Resim 10'da görüldüğü gibi ÇÖO'da oyunlaştırma bileşenleri ve eğlenelim etkinlikleri kullanılmamıştır. Öğretim materyalleri OÇÖO'da kullanılanlarla aynıdır.

### 3.5.5. Yüz yüze öğrenme ortamı

Araştırmada YYÖO kontrol grubumuzu oluşturmaktadır. YYÖO'da kodlama eğitimi verilirken hiçbir deneysel işlem uygulanmamıştır. Dersler OYYÖO'da kullanılan öğretim materyalleriyle işlenmiş sadece eğlenelim etkinlikleri ve oyunlaştırma bileşenleri ortamda kullanılmamıştır. Ders kazanımları, ödevler, ders notları, ders sunumları ve kodlama deneyimi etkinlikleri aynı şekilde kullanılmıştır.



### 3.6. Verilerin analizi

Araştırmada elde edilen verilerin analizinde MS Excel 2013 ve SPSS 22 paket programları kullanılmıştır. ABT güvenilirliğinin kontrolünde KR-20 iç tutarlılık katsayısı, kararlılığın kontrolünde ise pearson momentler çarpım korelasyon katsayısı kullanılmıştır. Elde edilen verilerin analizinde kullanılacak yöntemin belirlenmesinde öncelikle parametrik test varsayımları incelenmiştir. Parametrik test varsayımları; verilerin normal dağılım göstermesi, verilerin en az aralık ölçeğinde olması ve grup varyanslarının eşitliği şeklinde ifade edilmiştir (Field, 2013). Bu varsayımlara göre ilk olarak gruplara uygulanan tüm ölçeklerden elde edilen puanların normal dağılım gösterip göstermediği kontrol edilmiştir. Can (2016) normallüğün kontrolü için veri grubunun çarpıklık (skewness) ve basıklık (kurtosis) değerlerine bakılarak, bunların sifıra yakınlığına göre normallik hakkında fikir yürütülebileceğini belirtmiştir. Bu nedenle tüm gruplardan elde edilen çarpıklık (skewness) ve basıklık (kurtosis) değerleri hesaplanmıştır. Elde edilen sonuçlar Ek 22'de sunulmuştur.

Ek 22'de sunulan çarpıklık ve basıklık katsayıları, standart hatalarına bölünerek elde edilmiştir. Çarpıklığı ve basıklığı standart hatasına bölmek, katsayının ortalamasının kaç standart sapma altında ya da üstünde olduğunu verecektir. Hesaplanan çarpıklık ve basıklık katsayısının -1.96 ile +1.96 aralığında olması  $p=0.05$  anlamlılık düzeyine göre dağılımın çarpıklığının ve basıklığının sifirdan anlamlı bir farklılık göstermediğinin kabulüdür (Can, 2016). Ek 22'de sunulan tüm çarpıklık ve basıklık değerlerinin -1.96 ve +1.96 aralığında olduğu görülmektedir. Dolayısıyla ABT, MÖ ve PDYTÖ ön test, son test ve izleme testi ölçümlerinden elde edilen değerlerin normal dağılım gösterdiği çarpıklık ve basıklık değerlerine göre söylenebilir.

Normallik varsayımı dışında parametrik testlerin bir diğer varsayımı verilerin en az aralık ölçeğinde olmasıdır. Veri toplama araçları ABT, MÖ ve PDYTÖ'den elde edilen veriler aralık ölçeğinde olduğundan dolayı bu varsayımda karşılanmıştır.

Parametrik test varsayımlarından bir diğeri ise varyans homojenliğidir. Gruplardan aynı zamanda elde edilen puanların varyanslarının eşitliğine dayanan bu varsayım (Büyüköztürk, 2012) varyans homojenliği testi olan levene testi ile kontrol edilmiş ve sonuçlar 7'de sunulmuştur.

Tablo 7

*ABT, MÖ ve PDYTÖ Ön Test Son Test ve İzleme Testi Varyans Homojenliği (Levene) Testi Sonuçları*

Ölçek	Ölçüm	n	Sd1	Sd2	F	p
ABT	Ön Test	132	3	128	2.061	.109
	Son Test	132	3	128	.722	.540
	İzleme	132	3	128	2.142	.098
MÖ	Ön Test	132	3	128	2.488	.063
	Son Test	132	3	128	1.547	.206
	İzleme	132	3	128	.952	.417
PDYTÖ	Ön Test	132	3	128	.580	.629
	Son Test	132	3	128	2.219	.089
	İzleme	132	3	128	2.418	.069

Tablo 7 incelendiğinde grupların ABT, MÖ ve PDYTÖ ön test puanları açısından varyans homojenliği varsayımını sağladığı görülmektedir (ABT<sub>ön test</sub> için F=2.061, p>.05; MÖ<sub>ön test</sub> için F=2.488, p>.05; PDYTÖ<sub>ön test</sub> için F=.580, p>.05). Son test ölçümlerinde gruplardan elde edilen değerler incelendiğinde ABT, MÖ ve PDYTÖ açısından grupların varyanslarının homojen olduğu görülmüştür (ABT<sub>son test</sub> için F=.722, p>.05; MÖ<sub>son test</sub> için F=1.547, p>.05; PDYTÖ<sub>son test</sub> için F=2.219, p>.05). İzleme testi ölçümlerinin de tüm değişkenler açısından varyans homojenliği varsayımını karşıladığı belirlenmiştir (ABT<sub>izleme testi</sub> için F=2.142, p>.05; MÖ<sub>izleme testi</sub> için F=.952, p>.05; PDYTÖ<sub>izleme testi</sub> için F=2.418, p>.05). Tablo 7'de sunulan Levene testi sonuçlarına göre ABT, MÖ, PDYTÖ değişkenlerinden elde edilen ön test, son test ve izleme testi değerlerinin varyans homojenliği varsayımını karşıladığı görülmüştür.

Parametrik test varsayımlarının kontrolünden sonra grupların ABT, MÖ ve PDYTÖ ön test puanları açısından denklilikleri kontrol edilmiştir. Gruplar arasında ABT ön test puanları açısından anlamlı farklılığın bulunup bulunmadığını belirlemek amacıyla ilişkisiz örneklem için tek faktörlü ANOVA yapılmıştır. Elde edilen sonuçlar Tablo 8'de sunulmuştur.

Tablo 8

*ABT Ön Test Puanlarına İlişkin Tek Faktörlü ANOVA Sonuçları*

Varyansın Kaynağı	Kareler Toplamı	Sd	Kareler Ortalaması	F	<i>p</i>
Gruplar arası	337.933	3	112.835	1.835	.144
Grup içi	7857.810	128	61.389		
Toplam	8195.743	131			

Tablo 8 incelendiğinde deney 1 ( $\bar{X} = 28.78$ ), deney 2 ( $\bar{X} = 32.45$ ), deney 3 ( $\bar{X} = 29.64$ ) ve kontrol ( $\bar{X} = 32.24$ ) grupları ön test puan ortalamaları arasında anlamlı farklılık gözlenmemiştir ( $F_{(3-128)}=1.835$ ,  $p>.05$ ). Bulgular grupların deneysel işlem öncesinde ABT puanları açısından birbirine denk olduğunu göstermektedir.

MÖ ön test puanları açısından gruplar arasında anlamlı farklılığın bulunup bulunmadığını belirlemek amacıyla ilişkisiz örneklem için tek faktörlü ANOVA yapılmıştır. Elde edilen sonuçlar Tablo 9'da sunulmuştur.

Tablo 9

*MÖ Ön Test Puanlarına İlişkin Tek Faktörlü ANOVA Sonuçları*

Varyansın Kaynağı	Kareler Toplamı	Sd	Kareler Ortalaması	F	<i>p</i>
Gruplar arası	2.819	3	.940	1.639	.184

Grup içi	73.370	128	.573
Toplam	76.189	131	

Tablo 9 incelendiğinde deney 1 ( $\bar{X} = 4.10$ ), deney 2 ( $\bar{X} = 4.26$ ), deney 3 ( $\bar{X} = 4.34$ ) ve kontrol ( $\bar{X} = 3.96$ ) grupları ön test puan ortalamaları arasında anlamlı farklılık gözlenmemiştir ( $F_{(3-128)}=1.639$ ,  $p>.05$ ). Bulgular grupların deneysel işlem öncesinde MÖ puanları açısından birbirine denk olduğunu göstermektedir.

PDYTÖ ön test puanları açısından gruplar arasında anlamlı farklılığın bulunup bulunmadığını belirlemek amacıyla ilişkisiz örneklem için tek faktörlü ANOVA yapılmıştır. Elde edilen sonuçlar Tablo 10'da sunulmuştur.

Tablo 10

*PDYTÖ Ön Test Puanlarına İlişkin Tek Faktörlü ANOVA Sonuçları*

Varyansın Kaynağı	Kareler Toplamı	Sd	Kareler Ortalaması	F	<i>p</i>
Gruplar arası	.537	3	.197	1.187	.318
Grup içi	19.296	128	.151		
Toplam	19.833	131			

Tablo 10 incelendiğinde deney 1 ( $\bar{X} = 2.70$ ), deney 2 ( $\bar{X} = 2.83$ ), deney 3 ( $\bar{X} = 2.86$ ) ve kontrol ( $\bar{X} = 2.78$ ) grupları ön test puan ortalamaları arasında anlamlı farklılık gözlenmemiştir ( $F_{(3-128)}=1.187$ ,  $p>.05$ ). Bulgular grupların deneysel işlem öncesinde PDYTÖ puanları açısından birbirine denk olduğunu göstermektedir.

Parametrik test varsayımları ve grupların denkliğine ilişkin bulgular incelendiğinde ABT, MÖ, PDYTÖ ölçme araçlarından elde edilen verilerin analiz edilmesinden sonra karışık ölçümler için iki faktörlü ANOVA (repeated measure ANOVA) kullanılması kararlaştırılmıştır. Tek faktör üzerinde tekrarlı ölçümler için iki faktörlü ANOVA olarak da bilinen istatistiksel modelde gruplara bağlı ilişkisiz ölçümler ve zamana bağlı tekrarlı

ölçümler olarak iki faktörün satır x sütun ortak etkisi ve satır ile sütun temel etkileri test edilir (Büyüköztürk, 2012). Bu modelde grupların ölçme sonuçlarının karşılaştırılması ilişkisiz örneklem için ölçme olurken, tekrarlı ölçmeler ise ilişkili örneklem için ölçme olmaktadır. Bu nedenle iki değişik uygulama barındırdığından karışık ölçümler için iki yönlü varyans analizi olarak da ifade edilir (Can, 2016).

Karışık ölçümler için iki faktörlü ANOVA yapılabilmesi için belirli varsayımların sağlanması gerekmektedir. Bu varsayımlardan ilki verilerin en az aralık ölçeğinde olmasıdır (Büyüköztürk, 2012:79; Can, 2016:248; Taşpınar, 2017:135). ABT, MÖ ve PDYTÖ veri toplama araçları aralık ölçeğinde olduğundan bu varsayım karşılanmıştır.

İkinci varsayım gruplardan bağımlı değişkene ait elde ölçümlerin normal dağılıma sahip olmasıdır (Büyüköztürk, 2012:79; Can, 2016:248; Taşpınar, 2017:135). Parametrik test varsayımları incelenirken kontrol edilen çarpıklık, basıklık istatistikleri verilerin normal dağılıma sahip olduğunu göstermiştir (çarpıklık ve basıklık için bkz. Ek 22). Dolayısıyla bu varsayım da karşılanmıştır.

Üçüncü varsayın gruplardan elde edilen ölçümlerin varyanslarının homojenliği koşuludur (Büyüköztürk, 2012:79; Can, 2016:248; Taşpınar, 2017:135). Bu varsayımda levne varyans homojenliği testi ile incelenmiş ve varyansların homojen olduğuna karar verilmiştir (bkz. Tablo 7).

Dördüncü varsayım grupların ikili ölçüm kombinasyonlarının eşitliğidir (Büyüköztürk, 2012:79; Can, 2016:248; Taşpınar, 2017:135). Bu varsayım kovaryans matrislerinin eşitliği testi (Box's Test Of Equality of Covariance Matrices) ile kontrol edilmiştir. Kovaryans eşitliği çözümlemesi sonucunda ABT açısından grupların ikili kombinasyonlarının birlikte değişimlerinin arasında fark olmadığı belirlenmiş ve kovaryans eşitliği sağlanmıştır (Box  $F_{ABT}=1.364$ ;  $p>.05$ ). MÖ ve PDYTÖ kovaryans eşitliği çözümlemesinde ise kovaryans eşitliği sağlanamamıştır ( $F_{MÖ}=2.403$ ;  $p<.05$ ,  $F_{PDYTÖ}=2.579$ ;  $p<.05$ ). ABT açısından kovaryans eşitliği varsayımı sağlanmışken MÖ ve PDYTÖ açısından kovaryans eşitliği varsayımı sağlanamamıştır. Tabachnick ve Fidell (2013:318) örneklem sayılarının eşit olduğu durumlarda kovaryans matrisleri eşitliği varsayımının gerekli olmadığını belirtmiştir. Can (2018:207) ise kovaryans eşitliği varsayımının sağlanamadığı durumlarda örneklem sayılarına bakılması gerektiğini belirtmiş, örneklem sayıları eşit ise analiz sonuçlarının fazla etkilenmeyeceğini ve analize devam edilebileceğini ifade etmiştir. Örneklem sayıları eşit

olduğundan ABT, MÖ ve PDYTÖ açısından kovaryans eşitliği varsayımının sağlandığı kabul edilerek analize devam edilmiştir.

Beşinci varsayım grup içi faktörün herhangi iki grup için hesaplanan fark puanlarının evrendeki varyanslarının eşitliğidir (Taşpınar, 2017:135). Küresellik (Sphericity) koşulu olarak da ifade edilen varsayımın sağlanıp sağlanmadığı “Mauchly Küresellik Testi” ile test edilir (Tabachnick ve Fidell, 2013; Field, 2013:546). Mauchly Küresellik Testi sonucunun  $p > .05$  çıkması durumunda varsayımın sağlandığı,  $p < .05$  çıkması durumunda ise küresellik varsayımının sağlanmadığı belirtilmektedir (Field, 2013:546). Küresellik varsayımının sağlanıp sağlanmadığına ilişkin yapılan Mauchly Küresellik Testi sonuçları Tablo 11’de sunulmuştur.

Tablo 11

*ABT, MÖ ve PDYTÖ Ön Test Son Test ve İzleme Testi Ölçümlerinden Elde Edilen Mauchly Küresellik Testi Sonuçları*

Ölçek	Gruplar İçi Etki	Mauchly W	$\chi^2$	Sd	p	Epsilon	
						Greenhouse-Geisser Düzeltmesi	Huynh-Feldt Düzeltmesi
ABT	Ölçüm	.661	52.673	2	.000	.747	.771
MÖ	Ölçüm	.332	139.913	2	.000	.600	.616
PDYTÖ	Ölçüm	.756	35.521	2	.000	.804	.832

Tablo 11 incelendiğinde ABT, MÖ ve PDYTÖ'den farklı zamanlarda alınan tekrarlı ölçümler için küresellik varsayımının sağlanamadığı görülmüştür (ABT için  $W_{(2)}=.661$ ,  $p < .05$ ; MÖ için  $W_{(2)}=.332$ ,  $p < .05$ ; PDYTÖ için  $W_{(2)}=.756$ ,  $p < .05$ ). Can (2016:222) küresellik varsayımının sağlanamadığı durumlarda ya epsilon değerine göre serbestlik derecesinde düzeltme yapılmış, ANOVA f değerinin kullanılması gerektiğini ya da çok değişkenli istatistiklerin birisiyle karar verilebileceğini ifade etmiştir. Serbestlik dereceleri üzerinden yapılan düzeltme yöntemleri Greenhouse-Geisser düzeltmesi ve Huynh-Feldt

düzeltilmesi olarak bilinir (Tabachnick ve Fidell, 2013). Bu çalışmada küresellik varsayımının sağlanmadığı ABT, MÖ ve PDYTÖ için Greenhouse-Geisser düzeltmesi kullanılarak varyans analizi sonuçları sunulmuştur.

Elde edilen verilere ilişkin yapılan incelemeler sonucunda ABT, MÖ ve PDYTÖ'nden elde edilen verilerin analizinde karışık ölçümler için iki faktörlü ANOVA kullanabilmenin gerekli koşullarının sağlandığı görülmüştür.

## BÖLÜM IV

### BULGULAR

Bu bölümde çevrimiçi ve yüz yüze kodlama eğitiminde oyunlaştırma öğeleri kullanımının akademik başarı, motivasyon ve tutum üzerine etkilerinin belirlendiği araştırma bulgularına yer verilmiştir.

#### 4.1. Akademik başarı testi puanlarına ilişkin bulgular

Grupların ABT ön test, son test ve izleme testi puanlarının ortalamaları ve standart sapma değerleri hesaplanmış ve elde edilen değerler Tablo 12'de sunulmuştur.

Tablo 12

*ABT Ön Test Son Test ve İzleme Testi Puanlarına İlişkin Betimsel İstatistikler*

Gruplar	Ölçümler	<i>N</i>	$\bar{X}$	<i>Ss</i>	En Düşük Puan	En Yüksek Puan
Deney 1	Ön Test	33	28.78	7.08	17.85	46.41
	Son Test	33	74.54	6.10	64.26	85.68
	İzleme Testi	33	72.70	5.49	60.69	85.68
Deney 2	Ön Test	33	32.45	6.20	21.42	49.98
	Son Test	33	80.27	7.09	64.26	92.82
	İzleme Testi	33	74.97	7.36	64.26	92.82
Deney 3	Ön Test	33	29.64	9.08	10.71	49.98
	Son Test	33	72.16	6.42	57.12	82.11
	İzleme Testi	33	69.99	4.55	57.12	78.54



	Ön Test	33	32.24	8.63	17.85	49.98
Kontrol	Son Test	33	65.34	8.35	46.41	85.68
	İzleme Testi	33	66.96	6.68	57.12	82.11

Tablo 12'deki ABT ön test puan ortalamaları incelendiğinde, deney 1 grubundaki öğrencilerin puan ortalaması ( $\bar{X}=28.78$ ); deney 2 grubundaki öğrencilerin puan ortalaması ( $\bar{X} = 32.45$ ); deney 3 grubundaki öğrencilerin puan ortalaması ( $\bar{X} = 29.64$ ) ve kontrol grubundaki öğrencilerin puan ortalaması ( $\bar{X} = 32.24$ ) olarak bulunmuştur. ABT son test puan ortalamaları incelendiğinde, deney 1 grubundaki öğrencilerin puan ortalaması ( $\bar{X}=74.54$ ); deney 2 grubundaki öğrencilerin puan ortalaması ( $\bar{X} = 80.27$ ); deney 3 grubundaki öğrencilerin puan ortalaması ( $\bar{X} = 72.16$ ) ve kontrol grubundaki öğrencilerin puan ortalaması ( $\bar{X} = 65.34$ ) olarak bulunmuştur. ABT izleme testi puan ortalamaları incelendiğinde, deney 1 grubundaki öğrencilerin puan ortalaması ( $\bar{X}=72.70$ ); deney 2 grubundaki öğrencilerin puan ortalaması ( $\bar{X} = 74.97$ ); deney 3 grubundaki öğrencilerin puan ortalaması ( $\bar{X} = 69.99$ ) ve kontrol grubundaki öğrencilerin puan ortalaması ( $\bar{X} = 66.96$ ) olarak bulunmuştur.

Grupların ABT ön test, son test ve izleme testinden aldıkları puan ortalamaları arasındaki farkın istatistiksel olarak anlamlı olup olmadığı karışık ölçümler için iki faktörlü ANOVA (Split Plot ANOVA) ile incelenmiştir. Elde edilen bulgular Tablo 13'te sunulmuştur ve verilen değerlerde küresellik varsayımının karşılanamaması nedeniyle Greenhouse- Geisser düzeltmesi ile elde edilen bulgulara yer verilmiştir.

Tablo 13

*ABT Ön Test Son Test ve İzleme Testi Puanlarına İlişkin Karışık Ölçümler İçin İki Faktörlü ANOVA Sonuçları*

Varyans Kaynağı	Kareler Toplamı	sd	Kareler Ortalaması	F	p	$\eta^2$
<b>Gruplararası</b>	<b>11669.791</b>	<b>131</b>				

Grup (Deney 1, Deney 2, Deney 3, Kontrol)	3099.811	3	1033.270	15.433	.000	.266
Hata	8569.980	128	66.953			
<b>Grupiçi</b>	<b>157005.757</b>	<b>197.089</b>				
Ölçüm (Ön Test, Son Test, İzleme Testi)	150627.532	1.493	100882.435	1849.510	.000	.935
<b>Grup*Ölçüm</b>	<b>2201.585</b>	<b>4.479</b>	<b>491.502</b>	<b>9.011</b>	<b>.000</b>	<b>.174</b>
Hata	4176.640	191.117	54.545			
Toplam	168675.548	328.089				

Tablo 13 incelendiğinde, grupların ABT ön test, son test ve izleme testi puan ortalamaları üzerinde yapılan ANOVA sonucunda grup etkisinin anlamlı olduğu ve eta kare değerinin büyük etki büyüklüğüne sahip olduğu görülmüştür ( $F_{(3,128)}=15.433$ ;  $p<.05$ ;  $\eta^2=.266$ ). Yani grupların ön test, son test ve izleme ölçümleri arasında ayırım yapmaksızın, ABT'den elde ettikleri puanların arasında anlamlı bir farklılık vardır. Ölçüm etkisi incelendiğinde tüm ölçümler arasındaki farkın anlamlı olduğu ve eta kare değerinin büyük etki büyüklüğüne sahip olduğu görülmüştür ( $F_{(197.089, 191.117)}=1849.510$ ;  $p<.05$ ;  $\eta^2=.935$ ). Buna göre grup ayırımı yapılmaksızın ön test, son test ve izleme testlerinden elde edilen puan ortalamaları arasındaki farkın anlamlı olduğu görülmektedir. Grup ve ölçüm ortak etkisinin de benzer şekilde anlamlı olduğu sonucuna ulaşılmıştır ( $F_{(4.479, 191.117)}=9.011$ ;  $p<.05$ ;  $\eta^2=.935$ ).

Gruplar arasında ve ölçümler arasında oluşan anlamlı farkın kaynağını belirlemek amacıyla Bonferonni çoklu karşılaştırmalar testi kullanılmıştır. Elde edilen bulgular Ek 23'te sunulmuştur.

Ek 23'te verilen Bonferonni ikili karşılaştırma testi sonuçları incelendiğinde deney 1 grubunun ön test puan ortalamaları ( $\bar{X}=28.78$ ) ile son test puan ortalamaları ( $\bar{X}=74.54$ ) arasında anlamlı bir farklılık görülmektedir ( $p<.05$ ). Benzer şekilde deney 1 grubu ön test puan ortalamaları ( $\bar{X}=28.78$ ) ile izleme testi puan ortalamaları ( $\bar{X}=72.70$ ) arasında da anlamlı bir farklılık görülmektedir ( $p<.05$ ). Deney 1 grubu son test puan ortalamaları ( $\bar{X}=74.54$ ) ile izleme testi puan ortalamaları ( $\bar{X}=72.70$ ) karşılaştırıldığında bir azalmanın olduğu fakat bu azalmanın anlamlı olmadığı görülmüştür ( $p>.05$ ). Elde edilen bulgular

dikkate alındığında OÇÖÖ'nun kullanıldığı deney 1 grubunun deneysel işlem sonrasında ABT puanlarının yükseldiği ve izleme ölçümünde ise anlamlı bir değişimin olmadığı gözlemlenmiştir.

Deney 2 grubunun ön test puan ortalamaları ( $\bar{X}=32.45$ ) ile son test puan ortalamaları ( $\bar{X}=80.27$ ) arasında anlamlı bir farklılık görülmektedir ( $p<.05$ ). Benzer şekilde deney 2 grubu ön test puan ortalamaları ( $\bar{X}=32.45$ ) ile izleme testi puan ortalamaları ( $\bar{X}=74.97$ ) arasında anlamlı bir farklılık görülmektedir ( $p<.05$ ). Deney 2 grubu son test puan ortalamaları ( $\bar{X}=80.27$ ) ile izleme testi puan ortalamaları ( $\bar{X}=74.97$ ) karşılaştırıldığında bir azalmanın olduğu ve bu azalmanın anlamlı olduğu görülmüştür ( $p<.05$ ). Elde edilen bulgular dikkate alındığında OYYÖÖ'nun kullanıldığı deney 2 grubunun deneysel işlem sonrasında ABT puanlarının yükseldiği ve izleme ölçümünde ise bir azalmanın olduğu görülmüş ve bu azalmanın anlamlı olduğu tespit edilmiştir ( $p<.05$ ).

Deney 3 grubunun ön test puan ortalamaları ( $\bar{X}=29.64$ ) ile son test puan ortalamaları ( $\bar{X}=72.16$ ) arasında anlamlı bir farklılık görülmektedir ( $p<.05$ ). Benzer şekilde deney 3 grubu ön test puan ortalamaları ( $\bar{X}=29.64$ ) ile izleme testi puan ortalamaları ( $\bar{X}=69.99$ ) arasında anlamlı bir farklılık görülmektedir ( $p<.05$ ). Deney 3 grubu son test puan ortalamaları ( $\bar{X}=72.16$ ) ile izleme testi puan ortalamaları ( $\bar{X}=69.99$ ) karşılaştırıldığında bir azalmanın olduğu fakat bu azalmanın anlamlı olmadığı görülmüştür ( $p>.05$ ). Elde edilen bulgular dikkate alındığında ÇÖÖ'nun kullanıldığı deney 3 grubunun deneysel işlem sonrasında ABT puanlarının yükseldiği ve izleme ölçümünde ise anlamlı bir değişimin olmadığı gözlemlenmiştir.

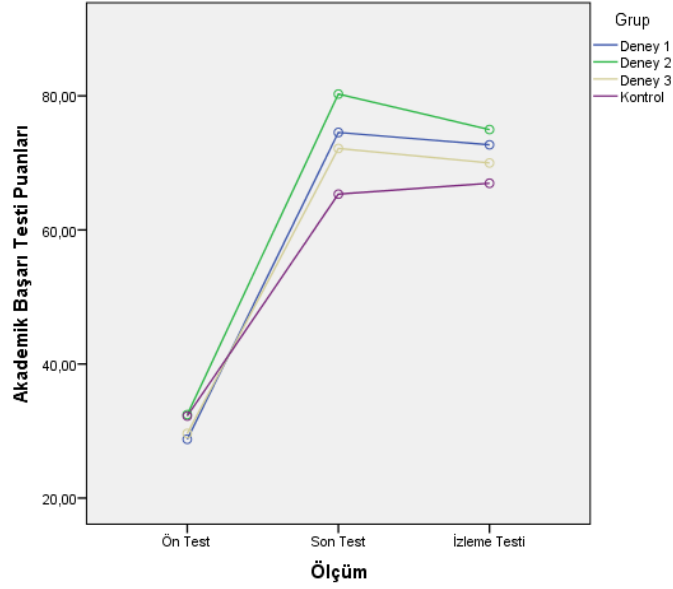
Kontrol grubunun ön test puan ortalamaları ( $\bar{X}=32.24$ ) ile son test puan ortalamaları ( $\bar{X}=65.34$ ) arasında anlamlı bir farklılık görülmektedir ( $p<.05$ ). Benzer şekilde kontrol grubu ön test puan ortalamaları ( $\bar{X}=32.24$ ) ile izleme testi puan ortalamaları ( $\bar{X}=66.96$ ) arasında da anlamlı bir farklılık görülmektedir ( $p<.05$ ). Kontrol grubu son test puan ortalamaları ( $\bar{X}=65.34$ ) ile izleme testi puan ortalamaları ( $\bar{X}=66.96$ ) karşılaştırıldığında bir artışın olduğu fakat bu artışın anlamlı olmadığı görülmüştür ( $p>.05$ ). Elde edilen bulgular dikkate alındığında YYÖÖ'nun kullanıldığı kontrol grubunun deneysel işlem sonrasında ABT puanlarının yükseldiği ve izleme ölçümünde ise anlamlı bir değişimin olmadığı gözlemlenmiştir.

Ek 23 incelendiğinde deney 1, deney 2, deney 3 ve kontrol gruplarının ABT son test ve izleme testi ölçümleri arasındaki farklılaşmayla ilgili bilgi sunmaktadır. Deney 1 ( $\bar{X}=74.54$ ),

deney 2 ( $\bar{X}=80.27$ ), deney 3 ( $\bar{X}=72.16$ ) son test puan ortalamaları ile kontrol grubu son test puan ortalaması ( $\bar{X}=65.34$ ) arasında anlamlı farklılık görülmüştür ( $p<.05$ ). Başka bir ifadeyle kodlama eğitiminde OÇÖÖ, OYYÖÖ ve ÇÖÖ kullanımının akademik başarıyı arttırmada etkili olduğu görülmüştür. Deney 1 son test puan ortalaması ( $\bar{X}=74.54$ ) ile deney 2 grubu son test puan ortalaması ( $\bar{X}=80.27$ ) arasında deney 2 grubu lehine anlamlı farklılık görülmüştür ( $p<.05$ ). Başka bir ifadeyle OYYÖÖ akademik başarıyı arttırmada OÇÖÖ'ya göre daha etkilidir. Deney 2 son test puan ortalaması ( $\bar{X}=80.27$ ) ile deney 3 son test puan ortalaması ( $\bar{X}=72.16$ ) arasında deney 2 grubu lehine anlamlı farklılık bulunmuştur ( $p<.05$ ). Başka bir ifadeyle kodlama eğitiminde OYYÖÖ kullanımının akademik başarıyı arttırmada ÇÖÖ'ya göre daha etkili olduğu görülmüştür. Deney 1 son test puan ortalaması ( $\bar{X}=74.54$ ) ile deney 3 son test puan ortalaması ( $\bar{X}=72.16$ ) arasında anlamlı bir farklılığa rastlanmamıştır ( $p>.05$ ). Başka bir ifadeyle kodlama eğitiminde OÇÖÖ kullanımı ile ÇÖÖ kullanımının akademik başarı üzerinde anlamlı bir etkiye sahip olmadığı bulunmuştur.

Deney 1 grubu izleme testi puan ortalaması ( $\bar{X}=72.70$ ) ile kontrol grubu izleme testi puan ortalaması ( $\bar{X}=66.96$ ) arasında deney 1 grubu lehine anlamlı farklılık bulunmuştur ( $p<.05$ ). Başka OÇÖÖ kullanıldığı deney 1 grubu ile hiçbir deneysel işleme maruz kalmayan kontrol grubu arasında son test puan ortalamalarında görülen anlamlı farklılık izleme testinde de görülmektedir. Deney 2 grubu izleme testi puan ortalaması ( $\bar{X}=74.97$ ) ile deney 3 grubu izleme testi puan ortalaması ( $\bar{X}=69.99$ ) arasında deney 2 grubu lehine anlamlı farklılık görülmüştür ( $p<.05$ ). Başka bir ifadeyle OYYÖÖ kullanıldığı deney 2 grubu ile ÇÖÖ kullanıldığı deney 3 grubu arasında son test puan ortalamalarında görülen anlamlı farklılık izleme testinde de görülmektedir. Deney 2 grubu izleme testi puan ortalaması ( $\bar{X}=74.97$ ) ile kontrol grubu izleme testi puan ortalaması ( $\bar{X}=66.96$ ) arasında deney 2 grubu lehine anlamlı farklılık bulunmuştur ( $p<.05$ ). Başka bir ifadeyle OYYÖÖ kullanıldığı deney 2 grubu ile hiçbir deneysel işleme maruz kalmayan kontrol grubu arasında son test puan ortalamalarında görülen anlamlı farklılık izleme testinde de görülmektedir. Deney 1 ve deney 2 arasında son test ortalamalarında görülen anlamlı farklılık izleme testinde görülmemiştir. Bunun nedeni olarak da OYYÖÖ kullanıldığı deney 2 grubunu son test ve izleme testi puan ortalamaları arasında son test lehine anlamlı farklılık görülmesidir. Deney 1 ve deney 3 grupları izleme testi puan ortalamalarında da son test puan ortalamalarında olduğu gibi anlamlı farklılığa rastlanmamıştır. Deney 3 ve kontrol grupları son test puan ortalamaları arasında deney 3 grubu lehine görülen anlamlı farklılık izleme testi puan ortalamalarında görülmemiştir.

Deney 1, deney 2, deney 3 ve kontrol gruplarının ABT ön test, son test ve izleme testi puan ortalamaları ilişkin grafik Şekil 17'de sunulmuştur.



Şekil 17. ABT ön test, son test ve izleme testi puan ortalamaları grafiği

Şekil 17'de sunulan grafik incelendiğinde tüm grupların son test puan ortalamaları ön test puan ortalamalarına göre yükselmiştir. Ayrıca tüm gruplarda izleme testi puanlarında son test puanlarına göre bir düşüş gözlemlenmiş ve deney 2 grubunda bu düşüşün daha fazla olduğu görülmüştür. Analiz sonucunda elde edilen bulgular grafiği destekler niteliktedir.

#### 4.2. Motivasyon puanlarına ilişkin bulgular

Grupların MÖ ön test, son test ve izleme testi puanlarının ortalamaları ve standart sapma değerleri hesaplanmış ve elde edilen değerler Tablo 14'te sunulmuştur.

Tablo 14

*MÖ Ön Test Son Test ve İzleme Testi Puanlarına İlişkin Betimsel İstatistikler*

Gruplar	Ölçümler	N	$\bar{X}$	Ss	En Düşük Puan	En Yüksek Puan
Deney 1	Ön Test	33	4.10	.60	2.55	5.25
	Son Test	33	4.95	.59	3.60	6.00
	İzleme Testi	33	4.89	.30	3.25	6.10
Deney 2	Ön Test	33	4.26	.94	2.10	5.90
	Son Test	33	5.08	.72	3.40	6.05
	İzleme Testi	33	4.95	.36	4.35	5.55
Deney 3	Ön Test	33	4.34	.81	2.90	5.75
	Son Test	33	4.59	.54	3.60	6.10
	İzleme Testi	33	4.37	.31	3.95	5.10
Kontrol	Ön Test	33	3.96	.63	2.60	5.15
	Son Test	33	4.26	.51	3.25	5.10
	İzleme Testi	33	4.27	.27	3.90	4.75

Tablo 14'teki MÖ ön test puan ortalamaları incelendiğinde, deney 1 grubundaki öğrencilerin puan ortalaması ( $\bar{X}=4.10$ ); deney 2 grubundaki öğrencilerin puan ortalaması ( $\bar{X} = 4.26$ ); deney 3 grubundaki öğrencilerin puan ortalaması ( $\bar{X} = 4.34$ ) ve kontrol grubundaki öğrencilerin puan ortalaması ( $\bar{X} = 3.96$ ) olarak bulunmuştur. MÖ son test puan ortalamaları incelendiğinde, deney 1 grubundaki öğrencilerin puan ortalaması ( $\bar{X}=4.95$ ); deney 2 grubundaki öğrencilerin puan ortalaması ( $\bar{X} = 5.08$ ); deney 3 grubundaki öğrencilerin puan ortalaması ( $\bar{X} = 4.59$ ) ve kontrol grubundaki öğrencilerin puan ortalaması ( $\bar{X} = 4.26$ ) olarak bulunmuştur. MÖ izleme testi puan ortalamaları incelendiğinde, deney 1 grubundaki öğrencilerin puan ortalaması ( $\bar{X}=4.89$ ); deney 2 grubundaki öğrencilerin puan ortalaması

( $\bar{X} = 4.95$ ); deney 3 grubundaki öğrencilerin puan ortalaması ( $\bar{X} = 4.37$ ) ve kontrol grubundaki öğrencilerin puan ortalaması ( $\bar{X} = 4.27$ ) olarak bulunmuştur.

Grupların MÖ ön test, son test ve izleme testinden aldıkları puan ortalamaları arasındaki farkın istatistiksel olarak anlamlı olup olmadığı karışık ölçümler için iki yönlü ANOVA (Split Plot ANOVA) ile incelenmiştir. Elde edilen bulgular Tablo 15'de sunulmuştur. Tablo 15'de verilen değerlerde küresellik varsayımının karşılanamaması nedeniyle Greenhouse-Geisser düzeltmesi ile elde edilen bulgulara yer verilmiştir.

Tablo 15

*MÖ Ön Test Son Test ve İzleme Testi Puanlarına İlişkin Tekrarlı Ölçümler ANOVA Sonuçları*

Varyans Kaynağı	Kareler Toplamı	sd	Kareler Ortalaması	F	p	$\eta^2$
<b>Gruplararası</b>	<b>68.009</b>	<b>131</b>				
Grup (Deney 1, Deney 2, Deney 3, Kontrol)	20.486	3	6.829	18.393	.000	.301
Hata	47.523	128	.371			
<b>Grupiçi</b>	<b>114.49</b>	<b>158.303</b>				
Ölçüm (Ön Test, Son Test, İzleme Testi)	23.121	1.199	19.279	35.359	.000	.216
<b>Grup*Ölçüm</b>	<b>7.670</b>	<b>3.598</b>	<b>2.132</b>	<b>3.910</b>	<b>.006</b>	<b>.084</b>
Hata	83.699	153.506	.545			
<b>Toplam</b>	<b>182.499</b>	<b>289.303</b>				

Tablo 15 incelendiğinde, grupların MÖ ön test, son test ve izleme testi puan ortalamaları üzerinde yapılan ANOVA sonucunda grup etkisinin anlamlı olduğu ve eta kare değerinin büyük etki büyüklüğüne sahip olduğu görülmüştür ( $F_{(3,128)}=18.393$ ;  $p<.05$ ;  $\eta^2=.301$ ). Yani grupların ön test, son test ve izleme ölçümleri arasında ayırım yapmaksızın, MÖ'den elde

ettikleri puanların arasında anlamlı bir farklılık vardır. Ölçüm etkisi incelendiğinde tüm ölçümler arasındaki farkında anlamlı olduğu ve eta kare değerinin büyük etki büyüklüğüne sahip olduğu görülmüştür ( $F_{(1,999, 153,506)}=35.359$ ;  $p<.05$ ;  $\eta^2=.216$ ). Buna göre grup ayrımı yapılmaksızın ön test, son test ve izleme testlerinden elde edilen puan ortalamaları arasındaki farkın anlamlı olduğu görülmektedir. Grup ve ölçüm ortak etkisinin de benzer şekilde anlamlı olduğu sonucuna ulaşılmıştır ( $F_{(3,598, 153,506)}=3.910$ ;  $p<.05$ ;  $\eta^2=.084$ )

Gruplar arasında ve ölçümler arasında oluşan anlamlı farkın kaynağını belirlemek amacıyla Bonferonni çoklu karşılaştırmalar testi kullanılmıştır. Elde edilen bulgular Ek 24'te sunulmuştur.

Ek 24'te verilen Bonferonni ikili karşılaştırma testi sonuçları incelendiğinde deney 1 grubunun ön test puan ortalamaları ( $\bar{X}=4.10$ ) ile son test puan ortalamaları ( $\bar{X}=4.95$ ) arasında anlamlı bir farklılık görülmektedir ( $p<.05$ ). Benzer şekilde deney 1 grubu ön test puan ortalamaları ( $\bar{X}=4.10$ ) ile izleme testi puan ortalamaları ( $\bar{X}=4.89$ ) arasında da anlamlı bir farklılık görülmektedir ( $p<.05$ ). Deney 1 grubu son test puan ortalamaları ( $\bar{X}=4.95$ ) ile izleme testi puan ortalamaları ( $\bar{X}=4.89$ ) karşılaştırıldığında bir azalmanın olduğu fakat bu azalmanın anlamlı olmadığı görülmüştür ( $p>.05$ ). Elde edilen bulgular dikkate alındığında OÇÖO'nun kullanıldığı deney 1 grubunun deneysel işlem sonrasında motivasyon puanlarının yükseldiği ve izleme ölçümünde ise anlamlı bir değişimin olmadığı gözlemlenmiştir.

Deney 2 grubunun ön test puan ortalamaları ( $\bar{X}=4.26$ ) ile son test puan ortalamaları ( $\bar{X}=5.08$ ) arasında anlamlı bir farklılık görülmektedir ( $p<.05$ ). Benzer şekilde deney 2 grubu ön test puan ortalamaları ( $\bar{X}=4.26$ ) ile izleme testi puan ortalamaları ( $\bar{X}=4.95$ ) arasında da anlamlı bir farklılık görülmektedir ( $p<.05$ ). Deney 2 grubu son test puan ortalamaları ( $\bar{X}=5.08$ ) ile izleme testi puan ortalamaları ( $\bar{X}=4.95$ ) karşılaştırıldığında bir azalmanın olduğu ve bu azalmanın anlamlı olmadığı görülmüştür ( $p>.05$ ). Elde edilen bulgular dikkate alındığında OYYÖO'nun kullanıldığı deney 2 grubunun deneysel işlem sonrasında motivasyon puanlarının yükseldiği ve izleme ölçümünde ise motivasyon puanlarında anlamlı bir değişimin olmadığı bulunmuştur ( $p>.05$ ).

Deney 3 grubunun ön test puan ortalamaları ( $\bar{X}=4.34$ ) ile son test puan ortalamaları ( $\bar{X}=4.59$ ) arasında anlamlı bir farklılık görülmemiştir ( $p>.05$ ). Benzer şekilde deney 3 grubu ön test puan ortalamaları ( $\bar{X}=4.34$ ) ile izleme testi puan ortalamaları ( $\bar{X}=4.37$ ) arasında da anlamlı bir farklılık bulunmamıştır ( $p>.05$ ). Deney 3 grubu son test puan ortalamaları ( $\bar{X}=4.59$ ) ile



izleme testi puan ortalamaları ( $\bar{X}=4.37$ ) karşılaştırıldığında bir azalmanın olduğu ve bu azalmanın anlamlı olduğu görülmüştür ( $p<.05$ ). Elde edilen bulgular dikkate alındığında ÇÖÖ'nun kullanıldığı deney 3 grubunun deneysel işlem sonrasında motivasyon puanlarında anlamlı bir değişimin olmadığı ve izleme ölçümünde ise anlamlı bir düşüşün olduğu gözlemlenmiştir.

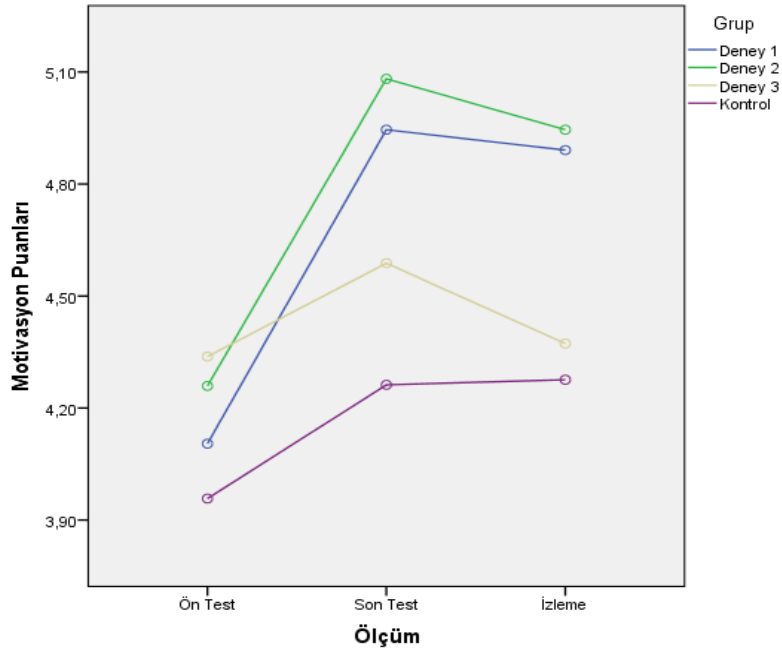
Kontrol grubunun ön test puan ortalamaları ( $\bar{X}=3.96$ ) ile son test puan ortalamaları ( $\bar{X}=4.26$ ) arasında anlamlı bir farklılık görülmemiştir ( $p>.05$ ). Benzer şekilde kontrol grubu ön test puan ortalamaları ( $\bar{X}=3.96$ ) ile izleme testi puan ortalamaları ( $\bar{X}=4.27$ ) arasında da anlamlı bir farklılık bulunmamıştır ( $p>.05$ ). Kontrol grubu son test puan ortalamaları ( $\bar{X}=4.26$ ) ile izleme testi puan ortalamaları ( $\bar{X}=4.27$ ) karşılaştırıldığında bir artışın olduğu fakat bu artışın anlamlı olmadığı görülmüştür ( $p>.05$ ). Elde edilen bulgular dikkate alındığında YYÖÖ'nun kullanıldığı kontrol grubunun deneysel işlem sonrasında motivasyon puanlarında anlamlı bir değişimin olmadığı ve izleme ölçümlerinde de devam ettiği gözlemlenmiştir.

Ek 24 incelendiğinde deney 1, deney 2, deney 3 ve kontrol gruplarının motivasyon son test ve izleme testi ölçümleri arasındaki farklılaşmayla ilgili de bilgi sunmaktadır. Deney 1 ( $\bar{X}=4.95$ ), deney 2 ( $\bar{X}=5.08$ ) son test puan ortalamaları ile kontrol grubu son test puan ortalaması ( $\bar{X}=4.26$ ) arasında anlamlı farklılık görülmüştür ( $p<.05$ ). Başka bir ifadeyle kodlama eğitiminde OÇÖÖ (deney 1) ve OYYÖÖ (deney 2) kullanımının motivasyonu arttırmada etkili olduğu görülmüştür. Deney 2 son test puan ortalaması ( $\bar{X}=5.08$ ) ile deney 3 grubu son test puan ortalaması ( $\bar{X}=4.59$ ) arasında deney 2 grubu lehine anlamlı farklılık görülmüştür ( $p<.05$ ). Başka bir ifadeyle OYYÖÖ motivasyonu arttırmada ÇÖÖ'ya göre daha etkilidir. Deney 1 son test puan ortalaması ( $\bar{X}=4.95$ ) ile deney 2 son test puan ortalaması ( $\bar{X}=5.08$ ) arasında anlamlı bir farklılığa rastlanmamıştır ( $p>.05$ ). Başka bir ifadeyle kodlama eğitiminde OÇÖÖ kullanımı ile OYYÖÖ kullanımının motivasyon üzerinde benzer etkiye sahip olduğu bulunmuş ve aralarında anlamlı farklılığa rastlanmamıştır. Deney 1 son test puan ortalaması ( $\bar{X}=4.95$ ) ile deney 3 son test puan ortalaması ( $\bar{X}=4.59$ ) arasında anlamlı bir farklılığa rastlanmamıştır ( $p>.05$ ). Başka bir ifadeyle kodlama eğitiminde OÇÖÖ kullanımı ile ÇÖÖ kullanımının motivasyon üzerinde benzer etkiye sahip olduğu bulunmuş ve aralarında anlamlı farklılığa rastlanmamıştır. Deney 3 son test puan ortalaması ( $\bar{X}=4.59$ ) ile kontrol grubu son test puan ortalaması ( $\bar{X}=4.27$ ) arasında anlamlı bir farklılığa rastlanmamıştır ( $p>.05$ ). Başka bir ifadeyle kodlama eğitiminde ÇÖÖ kullanımı ile YYÖÖ

kullanımının motivasyon üzerinde benzer etkiye sahip olduğu bulunmuş ve aralarında anlamlı farklılığa rastlanmamıştır.

Deney 1 grubu izleme testi puan ortalaması ( $\bar{X}=4.89$ ) ile deney 3 grubu izleme testi puan ortalaması ( $\bar{X}=4.37$ ) arasında deney 1 grubu lehine anlamlı farklılık bulunmuştur ( $p<.05$ ). Başka bir ifadeyle deneysel işlem sonrasında ölçülen son test puan ortalamaları arasında anlamlı farklılık bulunmayan deney 1 ve deney 3 grupları arasında izleme ölçümlerinde deney 1 grubu lehine anlamlı farklılık oluşmuştur. Deney 1 grubu izleme testi puan ortalaması ( $\bar{X}=4.89$ ) ile kontrol grubu izleme testi puan ortalaması ( $\bar{X}=4.27$ ) arasında deney 1 grubu lehine anlamlı farklılık bulunmuştur ( $p<.05$ ). Başka bir ifadeyle OÇÖO'nun kullanıldığı deney 1 grubu ile YYÖO'nun kullanıldığı kontrol grubu arasında deney 1 grubu lehine olan son test motivasyon puan ortalamaları arasındaki anlamlı farklılık izleme testi puan ortalamalarında da gözlemlenmiştir. Deney 2 grubu izleme testi puan ortalaması ( $\bar{X}=4.95$ ) ile deney 3 grubu izleme testi puan ortalaması ( $\bar{X}=4.37$ ) arasında deney 2 grubu lehine anlamlı farklılık bulunmuştur ( $p<.05$ ). OYYÖO'nun kullanıldığı deney 2 grubu ile ÇÖO'nun kullanıldığı deney 3 grubu arasında deney 2 grubu lehine olan anlamlı motivasyon son test puan ortalamaları izleme testi puan ortalamalarında da devam etmiştir. Deney 2 grubu izleme testi puan ortalaması ( $\bar{X}=4.95$ ) ile kontrol grubu izleme testi puan ortalaması ( $\bar{X}=4.27$ ) arasında deney 2 grubu lehine anlamlı farklılık bulunmuştur ( $p<.05$ ). OYYÖO'nun kullanıldığı deney 2 grubu ile YYÖO'nun kullanıldığı kontrol grubu arasında deney 2 grubu lehine olan anlamlı motivasyon son test puan ortalamaları izleme testi puan ortalamalarında da devam etmiştir. Deney 1 izleme testi puan ortalaması ( $\bar{X}=4.89$ ) ile deney 2 izleme testi puan ortalaması ( $\bar{X}=4.95$ ) arasında anlamlı bir farklılığa rastlanmamıştır ( $p>.05$ ). Başka bir ifadeyle kodlama eğitiminde OÇÖO kullanıldığı deney 1 ile OYYÖO kullanıldığı deney 2 grupları arasında son test motivasyon ölçümlerinde görülmeyen anlamlı farklılığa izleme ölçümlerinde rastlanmamıştır ( $p>.05$ ). Deney 3 izleme testi puan ortalaması ( $\bar{X}=4.37$ ) ile kontrol grubu izleme testi puan ortalaması ( $\bar{X}=4.27$ ) arasında anlamlı bir farklılığa rastlanmamıştır ( $p>.05$ ). Başka bir ifadeyle kodlama eğitiminde ÇÖO kullanıldığı deney 3 ile YYÖO kullanıldığı kontrol grupları arasında son test motivasyon ölçümlerinde görülmeyen anlamlı farklılığa izleme ölçümlerinde rastlanmamıştır ( $p>.05$ ).

Deney 1, deney 2, deney 3 ve kontrol gruplarının motivasyon ön test, son test ve izleme testi puan ortalamaları ilişkin grafik Şekil 18'de sunulmuştur.



Şekil 18. MÖ ön test, son test ve izleme testi puan ortalamaları grafiği

Şekil 18 incelendiğinde tüm grupların son test puan ortalamaları ön test puan ortalamalarına göre yükselmiştir. Ayrıca kontrol grubu dışında diğer gruplarda izleme testi puanlarında son test puanlarına göre bir düşüş gözlemlenmiş ve deney 3 grubunda bu düşüşün daha fazla olduğu görülmüştür. Kontrol grubunda ise izleme testi puan ortalamaları son test puan ortalamalarına göre anlamlı olmamakla birlikte hafif yükselmiştir. Analiz sonucunda elde edilen bulgular grafiği destekler niteliktedir.

### 4.3. Tutum puanlarına ilişkin bulgular

Grupların PDYTÖ ön test, son test ve izleme testi puanlarının ortalamaları ve standart sapma değerleri hesaplanmış ve elde edilen değerler Tablo 16'da sunulmuştur.

Tablo 16

*PDYTÖ Ön Test Son Test ve İzleme Testi Puanlarına İlişkin Betimsel İstatistikler*

Gruplar	Ölçümler	N	$\bar{X}$	Ss	En Düşük Puan	En Yüksek Puan
Deney 1	Ön Test	33	2.70	.34	1.88	3.50
	Son Test	33	4.23	.22	3.65	4.65
	İzleme Testi	33	4.16	.19	3.68	4.53
Deney 2	Ön Test	33	2.83	.35	2.10	3.38
	Son Test	33	4.26	.29	3.78	4.75
	İzleme Testi	33	4.21	.21	3.70	4.53
Deney 3	Ön Test	33	2.86	.48	2.00	4.20
	Son Test	33	3.61	.62	3.23	4.28
	İzleme Testi	33	3.72	.33	3.33	4.15
Kontrol	Ön Test	33	2.78	.37	1.88	4.20
	Son Test	33	3.32	.29	2.58	3.78
	İzleme Testi	33	3.45	.14	3.25	3.85

Tablo 16'daki PDYTÖ ön test puan ortalamaları incelendiğinde, deney 1 grubundaki öğrencilerin puan ortalaması ( $\bar{X}=2.70$ ); deney 2 grubundaki öğrencilerin puan ortalaması ( $\bar{X} = 2.83$ ); deney 3 grubundaki öğrencilerin puan ortalaması ( $\bar{X} = 2.86$ ) ve kontrol grubundaki öğrencilerin puan ortalaması ( $\bar{X} = 2.78$ ) olarak bulunmuştur. PDYTÖ son test puan ortalamaları incelendiğinde, deney 1 grubundaki öğrencilerin puan ortalaması ( $\bar{X}=4.23$ ); deney 2 grubundaki öğrencilerin puan ortalaması ( $\bar{X} = 4.26$ ); deney 3 grubundaki öğrencilerin puan ortalaması ( $\bar{X} = 3.61$ ) ve kontrol grubundaki öğrencilerin puan ortalaması ( $\bar{X} = 3.32$ ) olarak bulunmuştur. PDYTÖ izleme testi puan ortalamaları incelendiğinde, deney 1 grubundaki öğrencilerin puan ortalaması ( $\bar{X}=4.16$ ); deney 2 grubundaki öğrencilerin puan ortalaması ( $\bar{X} = 4.21$ ); deney 3 grubundaki öğrencilerin puan ortalaması

( $\bar{X} = 3.72$ ) ve kontrol grubundaki öğrencilerin puan ortalaması ( $\bar{X} = 3.45$ ) olarak bulunmuştur.

Grupların PDYTÖ ön test, son test ve izleme testinden aldıkları puan ortalamaları arasındaki farkın istatistiksel olarak anlamlı olup olmadığı karışık ölçümler için iki yönlü varyans analizi (Split Plot ANOVA) ile incelenmiştir. Elde edilen bulgular Tablo 17'de sunulmuştur. Tablo 17'de verilen değerlerde küresellik varsayımının karşılanamaması nedeniyle Greenhouse- Geisser düzeltmesi ile elde edilen bulgulara yer verilmiştir.

Tablo 17

*PDYTÖ Ön Test Son Test ve İzleme Testi Puanlarına İlişkin Tekrarlı Ölçümler ANOVA Sonuçları*

Varyans Kaynağı	Kareler Toplamı	sd	Kareler Ortalaması	F	p	$\eta^2$
<b>Gruplararası</b>	<b>32.626</b>	<b>131</b>				
Grup (Deney 1, Deney 2, Deney 3, Kontrol)	22.985	3	7.662	101.723	.000	.705
Hata	9.641	128	.075			
<b>Grupiçi</b>	<b>140.086</b>	<b>212.222</b>				
Ölçüm (Ön Test, Son Test, İzleme Testi)	104.811	1.608	65.192	581.138	.000	.819
<b>Grup*Ölçüm</b>	<b>12.190</b>	<b>4.823</b>	<b>2.527</b>	<b>22.529</b>	<b>.006</b>	<b>.346</b>
Hata	23.085	205.791	.112			
<b>Toplam</b>	<b>172.712</b>	<b>343.222</b>				

Tablo 17 incelendiğinde, grupların (deney 1, deney 2, deney 3, kontrol) PDYTÖ ön test, son test ve izleme testi puan ortalamaları üzerinde yapılan varyans analizi sonucunda grup etkisinin anlamlı olduğu ve eta kare değerinin büyük etki büyüklüğüne sahip olduğu görülmüştür ( $F_{(3,128)}=101.723$ ;  $p<.05$ ;  $\eta^2=.705$ ). Yani grupların ön test, son test ve izleme

ölçümleri arasında ayırım yapmaksızın, PDYTÖ'den elde ettikleri puanların arasında anlamlı bir farklılık vardır. Ölçüm etkisi incelendiğinde tüm ölçümler arasındaki farkında anlamlı olduğu ve eta kare değerinin büyük etki büyüklüğüne sahip olduğu görülmüştür ( $F_{(1.608,205.791)}=581.138$ ;  $p<.05$ ;  $\eta^2=.819$ ). Buna göre grup ayırımı yapılmaksızın ön test, son test ve izleme testlerinden elde edilen puan ortalamaları arasındaki farkın anlamlı olduğu görülmektedir. Grup ve ölçüm ortak etkisinin de benzer şekilde anlamlı olduğu sonucuna ulaşılmıştır ( $F_{(4.823, 205.791)}=22.529$ ;  $p<.05$ ;  $\eta^2=.346$ )

Gruplar arasında ve ölçümler arasında oluşan anlamlı farkın kaynağını belirlemek amacıyla Bonferonni çoklu karşılaştırmalar testi kullanılmıştır. Elde edilen bulgular Ek 25'te sunulmuştur.

EK 25'te verilen Bonferonni ikili karşılaştırma testi sonuçları incelendiğinde deney 1 grubunun ön test puan ortalamaları ( $\bar{X}=2.70$ ) ile son test puan ortalamaları ( $\bar{X}=4.23$ ) arasında anlamlı bir farklılık görülmektedir ( $p<.05$ ). Benzer şekilde deney 1 grubu ön test puan ortalamaları ( $\bar{X}=2.70$ ) ile izleme testi puan ortalamaları ( $\bar{X}=4.16$ ) arasında da anlamlı bir farklılık görülmektedir ( $p<.05$ ). Deney 1 grubu son test puan ortalamaları ( $\bar{X}=4.23$ ) ile izleme testi puan ortalamaları ( $\bar{X}=4.16$ ) karşılaştırıldığında bir azalmanın olduğu fakat bu azalmanın anlamlı olmadığı görülmüştür ( $p>.05$ ). Elde edilen bulgular dikkate alındığında OÇÖO'nun kullanıldığı deney 1 grubunun deneysel işlem sonrasında tutum puanlarının yükseldiği ve izleme ölçümünde ise anlamlı bir değişimin olmadığı gözlemlenmiştir.

Deney 2 grubunun ön test puan ortalamaları ( $\bar{X}=2.83$ ) ile son test puan ortalamaları ( $\bar{X}=4.26$ ) arasında anlamlı bir farklılık görülmektedir ( $p<.05$ ). Benzer şekilde deney 2 grubu ön test puan ortalamaları ( $\bar{X}=2.83$ ) ile izleme testi puan ortalamaları ( $\bar{X}=4.21$ ) arasında da anlamlı bir farklılık görülmektedir ( $p<.05$ ). Deney 2 grubu son test puan ortalamaları ( $\bar{X}=4.26$ ) ile izleme testi puan ortalamaları ( $\bar{X}=4.21$ ) karşılaştırıldığında bir azalmanın olduğu fakat bu azalmanın anlamlı olmadığı görülmüştür ( $p>.05$ ). Elde edilen bulgular dikkate alındığında OYYÖO'nun kullanıldığı deney 2 grubunun deneysel işlem sonrasında tutum puanlarının yükseldiği ve izleme ölçümünde ise anlamlı bir değişimin olmadığı gözlemlenmiştir.

Deney 3 grubunun ön test puan ortalamaları ( $\bar{X}=2.86$ ) ile son test puan ortalamaları ( $\bar{X}=3.61$ ) arasında anlamlı bir farklılık görülmektedir ( $p<.05$ ). Benzer şekilde deney 3 grubu ön test puan ortalamaları ( $\bar{X}=2.86$ ) ile izleme testi puan ortalamaları ( $\bar{X}=3.72$ ) arasında da anlamlı bir farklılık görülmektedir ( $p<.05$ ). Deney 3 grubu son test puan ortalamaları ( $\bar{X}=3.61$ ) ile izleme testi puan ortalamaları ( $\bar{X}=3.72$ ) karşılaştırıldığında izleme testi lehine bir artışın

olduğu fakat bu artışın anlamlı olmadığı görülmüştür ( $p>.05$ ). Elde edilen bulgular dikkate alındığında ÇÖÖ'nun kullanıldığı deney 3 grubunun deneysel işlem sonrasında tutum puanlarının yükseldiği ve izleme ölçümünde ise anlamlı bir değişimin olmadığı gözlemlenmiştir.

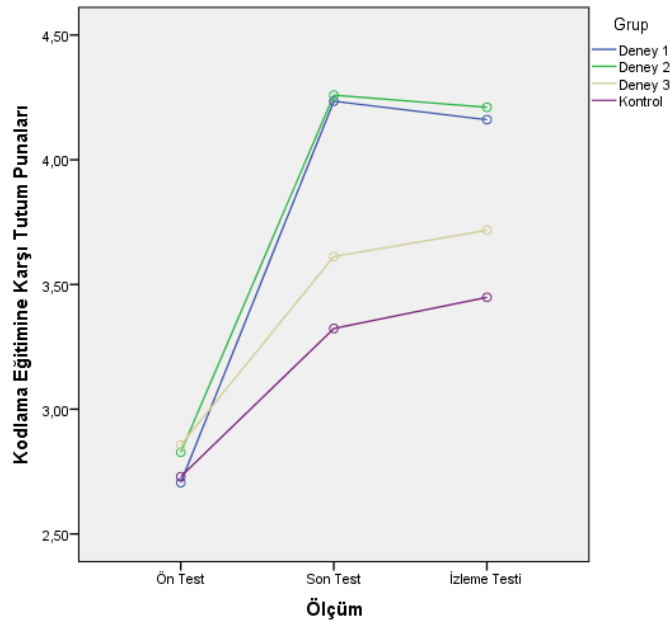
Kontrol grubunun ön test puan ortalamaları ( $\bar{X}=2.78$ ) ile son test puan ortalamaları ( $\bar{X}=3.32$ ) arasında anlamlı bir farklılık görülmektedir ( $p<.05$ ). Benzer şekilde kontrol grubu ön test puan ortalamaları ( $\bar{X}=2.78$ ) ile izleme testi puan ortalamaları ( $\bar{X}=3.45$ ) arasında da anlamlı bir farklılık görülmektedir ( $p<.05$ ). Kontrol grubu son test puan ortalamaları ( $\bar{X}=3.32$ ) ile izleme testi puan ortalamaları ( $\bar{X}=3.45$ ) karşılaştırıldığında izleme testi lehine bir artışın olduğu fakat bu artışın anlamlı olmadığı görülmüştür ( $p>.05$ ). Elde edilen bulgular dikkate alındığında hiçbir deneysel işlemin kullanılmadığı kontrol grubunun tutum puanlarının yükseldiği ve izleme ölçümünde ise anlamlı bir değişimin olmadığı gözlemlenmiştir.

EK 25 incelendiğinde deney 1, deney 2, deney 3 ve kontrol gruplarının tutum son test ve izleme testi ölçümleri arasındaki farklılaşmayla ilgili de bilgi sunmaktadır. Deney 1 ( $\bar{X}=4.23$ ), deney 2 ( $\bar{X}=4.26$ ), deney 3 ( $\bar{X}=3.61$ ) son test tutum puan ortalamaları ile kontrol grubu son test tutum puan ortalaması ( $\bar{X}=3.32$ ) arasında anlamlı farklılık görülmüştür ( $p<.05$ ). Başka bir ifadeyle kodlama eğitiminde OÇÖÖ, OYYÖÖ ve ÇÖÖ kullanımının kodlama eğitime karşı tutumu arttırmada etkili olduğu görülmüştür. Deney 1 son test puan ortalaması ( $\bar{X}=4.23$ ) ile deney 3 grubu son test puan ortalaması ( $\bar{X}=3.61$ ) arasında deney 1 grubu lehine anlamlı farklılık görülmüştür ( $p<.05$ ). Başka bir ifadeyle OÇÖÖ kodlama eğitime karşı tutumu arttırmada ÇÖÖ'ya göre daha etkilidir. Deney 2 son test puan ortalaması ( $\bar{X}=4.26$ ) ile deney 3 grubu son test puan ortalaması ( $\bar{X}=3.61$ ) arasında deney 2 grubu lehine anlamlı farklılık görülmüştür ( $p<.05$ ). Başka bir ifadeyle OYYÖÖ kodlama eğitime karşı tutumu arttırmada ÇÖÖ'ya göre daha etkilidir. Deney 1 son test puan ortalaması ( $\bar{X}=4.23$ ) ile deney 2 son test puan ortalaması ( $\bar{X}=4.26$ ) arasında anlamlı farklılığa rastlanmamıştır ( $p>.05$ ). Başka bir ifadeyle kodlama eğitiminde OÇÖÖ ile OYYÖÖ kullanımının kodlama eğitime karşı tutumu arttırmada bezer etkiye sahip olduğu söylenebilir.

Deney 1 ( $\bar{X}=4.16$ ), deney 2 ( $\bar{X}=4.21$ ), deney 3 ( $\bar{X}=3.72$ ) izleme testi tutum puan ortalamaları ile kontrol grubu son test tutum puan ortalaması ( $\bar{X}=3.45$ ) arasında anlamlı farklılık görülmüştür ( $p<.05$ ). Başka bir ifadeyle kodlama eğitiminde OÇÖÖ, OYYÖÖ ve ÇÖÖ kullanımının kodlama eğitime karşı tutumu arttırmada etkili olduğu son test

ortalamlarında görülmüş ve aynı etkinin izleme testi puan ortalamalarında da devam ettiği bulunmuştur. Deney 1 izleme testi puan ortalaması ( $\bar{X}=4.16$ ) ile deney 3 grubu izleme testi puan ortalaması ( $\bar{X}=3.72$ ) arasında deney 1 grubu lehine anlamlı farklılık görülmüştür ( $p<.05$ ). Başka bir ifadeyle OÇÖO kodlama eğitimine karşı tutumu arttırmada ÇÖO'ya göre daha etkili olduğu son test ölçümlerinde görüldüğü gibi izleme ölçümlerinde de devam etmiştir. Deney 2 izleme testi puan ortalaması ( $\bar{X}=4.21$ ) ile deney 3 grubu izleme testi puan ortalaması ( $\bar{X}=3.72$ ) arasında deney 2 grubu lehine anlamlı farklılık görülmüştür ( $p<.05$ ). Başka bir ifadeyle OYYÖO kodlama eğitimine karşı tutumu arttırmada ÇÖO'ya göre daha etkili olduğu görülmüş ve izleme testi ölçümlerinde de bu etki devam etmiştir. Deney 1 izleme testi puan ortalaması ( $\bar{X}=4.16$ ) ile deney 2 izleme testi puan ortalaması ( $\bar{X}=4.21$ ) arasında anlamlı farklılığa rastlanmamıştır ( $p>.05$ ). Başka bir ifadeyle kodlama eğitiminde OÇÖO (deney 1) ile OYYÖO (deney 2) kullanımının kodlama eğitime karşı tutumu arttırmada bezer etkiye sahip olduğu son test ölçümlerinde görülmüş ve aynı etkinin izleme ölçümlerinde de devam ettiği bulunmuştur.

Deney 1, deney 2, deney 3 ve kontrol gruplarının tutum ön test, son test ve izleme testi puan ortalamaları ilişkin grafik Şekil 19'da sunulmuştur.



Şekil 19. PDYTÖ ön test, son test ve izleme testi puan ortalamaları grafiği



Şekil 19 incelendiğinde tüm grupların son test puan ortalamaları ön test puan ortalamalarına göre yükselmiştir. Ayrıca izleme testi puan ortalamaları incelendiğinde deney 1 ve deney 2 gruplarında anlamlı olmamakla birlikte bir düşüşün olduğu görülmektedir. Deney 3 ve kontrol gruplarında ise izleme testi lehine anlamlı olmamakla birlikte bir yükselişin olduğu görülmektedir. Analiz sonucunda elde edilen bulgular grafiği destekler niteliktedir.

## BÖLÜM V

### SONUÇ, TARTIŞMA VE ÖNERİLER

Bu bölümde araştırma bulgularına yönelik sonuç, tartışma ve önerilere yer verilmiştir.

#### 5.1. Sonuç ve Tartışma

Bu bölümde araştırma bulguları üç başlık altında ele alınmış ve tartışılmıştır. İlk olarak çevrimiçi ve yüz yüze kodlama eğitiminde oyunlaştırma öğeleri kullanımının akademik başarı üzerine etkisi ortaya konmuştur. İkinci olarak çevrimiçi ve yüz yüze kodlama eğitiminde oyunlaştırma öğeleri kullanımının motivasyon üzerinde nasıl bir etki oluşturduğu sunulmuş ve son olarak da kodlamaya karşı tutum üzerinde nasıl bir etki oluşturduğu belirtilmiştir.

##### 5.1.1. Akademik başarı testinden elde edilen bulgulara yönelik sonuç ve tartışma

Çevrimiçi ve yüz yüze kodlama eğitiminde oyunlaştırma öğeleri kullanımının akademik başarı üzerine etkilerini ortaya çıkarmak amacıyla 3 deney ve 1 kontrol grubu kullanılmıştır. Birinci deney grubunda çevrimiçi öğrenme ortamı ve oyunlaştırma bileşenlerinin birlikte kullanıldığı OÇÖÖ, ikinci deney grubunda sadece oyunlaştırma bileşenlerinin kullanıldığı OYYÖÖ, üçüncü deney grubunda sadece çevrimiçi öğrenme ortamının kullanıldığı ÇÖÖ ve hiçbir deneysel işlemin uygulanmadığı kontrol grubunda ise YYÖÖ ile kodlama eğitimi verilmiştir.

Akademik başarı ön test, son test ve izleme testi verileri incelendiğinde; deney 1 grubunun deneysel işlem sonrasında yapılan son test ölçümlerinde akademik başarısının anlamlı derecede yükseldiği ve izleme ölçümlerinde de anlamlı bir değişimin olmadığı görülmüştür. Elde edilen bu sonuca göre, çevrimiçi öğrenme ortamında oyunlaştırma öğeleri kullanımının, kodlama eğitiminde akademik başarıyı yükselttiği ve kalıcılığı sağlamada etkili olduğu görülmektedir. Deney 2 grubunun deneysel işlem sonrasında yapılan son test ölçümlerinde akademik başarısının anlamlı derecede yükseldiği fakat izleme ölçümlerinde ise anlamlı bir düşüşün olduğu görülmüştür. İzleme ölçümleri ile ön test ölçümleri

karşılaştırıldığında izleme ölçümleri lehine anlamlı farklılık devam etmektedir. Bu sonuç, yüz yüze öğrenme ortamında oyunlaştırma öğeleri kullanımının, kodlama eğitiminde akademik başarıyı yükselttiği ve kalıcılığı sağlamada etkili olduğunu ortaya koymaktadır. Deney 3 grubunun deneysel işlem sonrasında yapılan son test ölçümlerinde, akademik başarılarının yükseldiği ve izleme ölçümlerinde de anlamlı bir değişimin olmadığı görülmüştür. Elde edilen bu sonuca göre, kodlama eğitiminde sadece çevrimiçi öğrenme ortamları kullanımının, akademik başarıyı yükselttiği ve kalıcılığı sağlamada etkili olduğu görülmektedir. Hiçbir deneysel işlemin yapılmadığı kontrol grubunda ise son test ölçümlerinde akademik başarının yükseldiği ve izleme ölçümlerinde de devam ettiği görülmüştür.

Gruplar arasındaki ilişkiyi ortaya koymak amacıyla yapılan ikili karşılaştırma testinden elde edilen bulgular incelendiğinde; deney gruplarında kullanılan OÇÖO, OYYÖO ve ÇÖO'nun akademik başarıyı, kontrol grubunda uygulanan YYÖO'dan daha fazla arttırdığı görülmüştür. Bu sonuçlara göre kodlama eğitiminde çevrimiçi öğrenme ortamları ve oyunlaştırma bileşenleri kullanımının akademik başarıyı yükseltmede daha etkili olduğu söylenebilir. Deney grupları karşılaştırıldığında, deney 2 grubunun akademik başarı son test puanlarının deney 1 ve deney 3 gruplarından daha yüksek olduğu görülmüştür. OYYÖO kullanımının akademik başarıyı arttırmada OÇÖO ve ÇÖO'dan daha etkili olduğu belirlenmiştir. Bu sonuçlara göre kodlama eğitiminde geleneksel yüz yüze ortamı ile oyunlaştırma bileşenleri bütünleştirildiğinde akademik başarının daha fazla yükseldiği söylenebilir. Deney 1 grubu ile deney 3 grubu arasında akademik başarı puanları açısından anlamlı bir fark bulunmamıştır. Bu bulguya göre kodlama eğitiminde OÇÖO ve ÇÖO kullanımının akademik başarıyı arttırmada benzer etkiye sahip olduğu belirlenmiştir. Bu sonuca göre çevrimiçi öğrenme ortamına oyunlaştırma bileşenlerinin eklenmesi, akademik başarıyı anlamlı derecede değiştirmeyeceği söylenebilir.

Deneysel işlemde beş hafta sonra elde edilen izleme testi puanları incelendiğinde; deney 1 grubunun izleme testi puan ortalamasının, kontrol grubu izleme testi puan ortalamasından anlamlı derecede yüksek olduğu görülmüştür. İki grup arasında son test ölçümlerinde görülen anlamlı farklılık izleme ölçümlerinde de devam etmiştir. Bu sonuçlara göre kodlama eğitiminde çevrimiçi öğrenme ortamı ile oyunlaştırma bileşenlerinin bütünleştirildiği ortamların kalıcılığı sağlamada geleneksel YYÖO'ya göre daha etkili olduğu söylenebilir. Deney 2 grubu izleme testi puan ortalamalarının kontrol grubu izleme testi puan ortalamalarına göre anlamlı derecede yüksek olduğu görülmüştür. İki grup arasında son test

ölçümlerinde görülen anlamlı farklılık izleme ölçümlerinde de devam etmiştir. Bu sonuca göre; geleneksel yüz yüze öğrenme ortamına oyunlaştırma bileşenlerinin dâhil edilmesiyle kalıcılığın daha iyi sağlanacağı belirlenmiştir. Deney 2 grubunun izleme testi puan ortalamasının deney 3 grubu izleme testi puan ortalamasından anlamlı derecede yüksek olduğu görülmüştür. İki grup arasında son test ölçümlerinde görülen anlamlı farklılık izleme ölçümlerinde de devam etmiştir. Bu sonuca göre; kodlama eğitiminde OYYÖO kullanımının kalıcılığı sağlamada ÇÖO'ya göre daha fazla etkili olduğu belirlenmiştir. Deney 1 grubu ile deney 2 grubu izleme ölçümleri arasında anlamlı farklılığa rastlanmamıştır. İki grup arasında son test ölçümlerinde deney 2 grubu lehine görülen anlamlı farklılığa izleme ölçümlerinde rastlanmamıştır. Bunun nedeni olarak da deney 2 grubunun izleme testi puan ortalamalarının son test puan ortalamalarına göre anlamlı derecede düşmüş olmasıdır. Bu sonuca göre deney 1 ve deney 2 gruplarının kalıcılığı sağlamada benzer etkiye sahip olduğu belirlenmiştir. Deney 1 ve deney 3 grupları arasında izleme testi ölçümlerinde son test ölçümlerinde olduğu gibi anlamlı farklılığa rastlanmamıştır. Dolayısıyla deney 1 ve deney 3 gruplarının da kalıcılığı sağlamada benzer etkiye sahip olduğu ortaya çıkmıştır. Deney 3 ve kontrol grupları arasında izleme ölçümlerinde anlamlı bir farklılığa rastlanmamıştır. Bu bulguya göre deney 3 ve kontrol gruplarının kalıcılığı sağlamada benzer etkiye sahip olduğu belirlenmiştir.

Alanyazın incelendiğinde kodlama eğitiminde OÇÖO, OYYYÖO, ÇÖO ve YYÖO'nun birlikte karşılaştırıldığı araştırmalara rastlanmamaktadır. Bununla birlikte kodlama eğitiminde oyunlaştırma bileşenleri ve çevrimiçi öğrenme kullanımına yönelik araştırmalarda kısıtlıdır. Kumar ve Khurana (2012) kodlama öğretiminin eğlenceli şekilde nasıl yapılacağına dair bir çerçeve oluşturdukları araştırmalarında, oyunlaştırmanın etkili ve ciddi bir şekilde uygulanması durumunda öğrencilerin gerçek yaşam problemlerini çözmelerini sağlayacağı belirtilmiştir. Fidan (2016) Scratch ile programlama öğretiminde oyunlaştırma kullanımının, öğrencilerin akademik başarılarını arttırdığı sonucuna ulaşmıştır. Yıldız (2018) yaptığı araştırmada oyunlaştırılmış blok temelli algoritmik düşünme aktiviteleri kullanımının programlama becerisini arttırdığı sonucuna ulaşmıştır. Araştırmada elde edilen sonuçlar; oyunlaştırma bileşenleri ve çevrimiçi öğrenme ortamlarının kullanımının geleneksel yüz yüze öğrenme ortamına göre akademik başarıyı daha fazla arttırdığını göstermektedir. Dolayısıyla alanyazında yapılan araştırmalarda araştırma sonuçlarımızı destekler niteliktedir.

Kodlama eğitimi dışında oyunlaştırılmış yüz yüze öğrenme ile geleneksel yüz yüze öğrenmenin karşılaştırıldığı araştırmalar incelendiğinde; Ar (2016) yaptığı araştırmada

oyunlaştırılmış öğrenme ortamının akademik başarıyı arttırdığını ortaya koymuştur. Benzer şekilde Hüner (2018) yaptığı araştırmada ikinci dil öğreniminde oyunlaştırılmış yüz yüze öğrenme ortamının akademik başarı üzerinde olumlu etkiye sahip olduğu sonucuna ulaşmıştır. Çakıroğlu ve diğerleri (2017) yaptıkları araştırmada, oyunlaştırılmış yüz yüze öğrenme ortamının akademik başarı üzerinde olumlu etki yarattığını gözlemlemiştir. Alanyazından elde edilen bu sonuçlar, araştırmada elde ettiğimiz sonuçları destekler niteliktedir. Elde edilen bu sonuçların aksine Hanus ve Fox (2015) iletişim dersinde oyunlaştırılmış yüz yüze öğrenme ortamı ile geleneksel yüz yüze öğrenme ortamını karşılaştırdığı araştırmalarında, oyunlaştırılmış yüz yüze öğrenme ortamında akademik başarının düştüğünü gözlemlemiştir. Bu sebeple oyunlaştırma bileşenlerinin kullanımında dikkatli olunması gerektiğini ifade etmişlerdir. Ece (2017) oyunlaştırılmış öğrenme ortamında kelime öğretimi yaptığı araştırmasında, oyunlaştırılmış öğrenme ortamının başarı açısından fark yaratmadığını gözlemlemiştir. Dolayısıyla bu sonuçlarda araştırma sonuçlarımızla tutarlılık göstermemektedir.

Oyunlaştırılmış çevrimiçi öğrenme ile geleneksel çevrimiçi öğrenmenin karşılaştırıldığı araştırmalar incelendiğinde; Domínguez et al. (2013) bilgi ve iletişim teknolojilerinin öğretimi amacıyla kullandığı oyunlaştırılmış çevrimiçi öğrenme ortamının akademik başarı üzerinde olumlu etkisinin olduğunu göstermektedir. Yıldırım (2016) öğretim ilke ve yöntemleri dersinde yaptığı araştırmada oyunlaştırılmış çevrimiçi öğrenme ortamı ile geleneksel çevrimiçi öğrenme ortamını karşılaştırmış ve oyunlaştırılmış öğrenme ortamının, akademik başarıyı arttırdığı sonucuna ulaşmıştır. Tunga (2016) oyunlaştırılmış çevrimiçi öğrenme ortamı ve geleneksel oyunlaştırılmış öğrenme ortamını karşılaştırdığı araştırmasında, oyunlaştırılmış öğrenme ortamının akademik başarıyı arttırdığı fakat kalıcılığı sağlamada etkili olmadığı sonucuna ulaşmıştır. Mohammed (2018) oyunlaştırılmış ve geleneksel çevrimiçi öğrenme ortamlarını karşılaştırdığı araştırmasında, oyunlaştırılmış öğrenme ortamının akademik başarıyı arttırdığını gözlemlemiştir. Alanyazından elde edilen bu sonuçlar araştırmada elde ettiğimiz sonuçları destekler niteliktedir. Elde edilen bu sonuçların aksine Sümer (2017) oyunlaştırılmış ve geleneksel çevrimiçi öğrenme ortamlarını karşılaştırdığı araştırmasında, oyunlaştırılmış öğrenme ortamının akademik başarı üzerinde bir fark yaratmadığını gözlemlemiştir. Dolayısıyla bu sonuçlarda araştırma sonuçlarımızla tutarlılık göstermemektedir.

Alanyazında oyunlaştırma bileşenlerinin kullanıldığı farklı öğrenme ortamları da karşılaştırılmıştır. Su ve Cheng (2015) fen bilimleri dersinde oyunlaştırılmış mobil öğrenme,

geleneksel mobil öğrenme ve geleneksel yüz yüze öğrenme ortamının karşılaştırdığı araştırmalarında, oyunlaştırılmış mobil öğrenmenin akademik başarıyı yükselttiğini gözlemlemişlerdir. Kalkan (2016) oyunlaştırılmış sanal öğrenme ortamının akademik başarıyı arttırdığı sonucuna ulaşmıştır. Zainuddin (2018) yaptığı araştırmada oyunlaştırılmış ters yüz edilmiş öğretimin, öğrencilerin akademik başarıları üzerinde olumlu etkiye sahip olduğu sonucuna ulaşmıştır. Bu araştırmalara göre; oyunlaştırma bileşenlerinin ve çevrimiçi öğrenmenin birlikte kullanıldığı öğrenme ortamlarında akademik başarının yükseldiği görülmektedir. Bunu aksine Meşe (2016) yaptığı araştırmada, harmanlanmış öğrenme ortamlarında oyunlaştırma bileşenlerinin etkinliğini ortaya koymaya çalışmıştır. Oyunlaştırılmış çevrimiçi ve yüz yüze öğrenme ortamının birlikte kullanıldığı deney grubunda akademik başarı açısından bir fark gözlemlenmemiştir.

### **5.1.2. Motivasyon ölçeğinden elde edilen bulgulara yönelik sonuç ve tartışma**

Çevrimiçi ve yüz yüze kodlama eğitiminde oyunlaştırma öğeleri kullanımının motivasyon üzerindeki etkileri incelendiğinde; deney 1 grubunun deneysel işlem sonrasında yapılan son test ölçümlerinde elde edilen motivasyon puanlarının anlamlı derecede yükseldiği ve izleme ölçümlerinde de anlamlı bir değişimin olmadığı görülmüştür. Elde edilen bu bulguya göre; kodlama eğitiminde çevrimiçi öğrenme ortamı ve oyunlaştırma bileşenlerinin birlikte kullanımı derse yönelik motivasyonu yükseltmede ve kalıcılığı sağlamada etkili olduğu görülmektedir. Deney 2 grubunun deneysel işlem sonrasında yapılan son test ölçümlerinde elde edilen motivasyon puanlarının anlamlı derecede yükseldiği ve izleme ölçümlerinde anlamlı bir değişimin olmadığı görülmüştür. Bu bulgu, kodlama eğitiminde yüz yüze öğrenme ortamı ile birlikte oyunlaştırma bileşenleri kullanımının derse yönelik motivasyonu arttırmada ve kalıcılığı sağlamada etkili olduğunu göstermektedir. Deney 3 grubunun deneysel işlem sonrasında yapılan son test ölçümlerinden elde edilen motivasyon puanlarında, anlamlı bir değişimin olmadığı görülmüş ve izleme ölçümlerinde ise anlamlı bir düşüşün olduğu belirlenmiştir. Elde edilen bu bulguya göre; kodlama eğitiminde çevrimiçi öğrenme ortamları kullanımının derse yönelik motivasyonu arttırmada ve kalıcılığı sağlamada etkili olmadığı görülmüştür. Hiçbir deneysel işlemin yapılmadığı kontrol grubunun son test ölçümlerinden elde edilen motivasyon puanlarında, anlamlı bir farklılığa rastlanmamıştır. Bununla birlikte izleme ölçümlerinde anlamlı bir farklılık gözlemlenmemiştir.

Gruplar arasındaki ilişkiyi ortaya koymak amacıyla yapılan ikili karşılaştırma testinden elde edilen bulgular incelendiğinde; OÇÖO ve OYYÖO'nun kullanıldığı deney 1 ve deney 2 grupları ile kontrol grubunun motivasyon puanları arasında deney grupları lehine anlamlı bir farklılık görülmüştür. Elde edilen bu bulgulara göre; kodlama eğitiminde OÇÖO ve OYYÖO kullanımının ders motivasyonunu arttırdığı görülmüştür. Deney 2 grubunun motivasyon son test puan ortalamasının, deney 3 grubu son test puan ortalamasından anlamlı derecede yüksek olduğu gözlemlenmiştir. Başka bir ifadeyle kodlama eğitiminde OYYÖO kullanımının derse yönelik motivasyonu arttırmada ÇÖO'dan daha etkili olduğu belirlenmiştir. Deney 1 grubu ile deney 2 grubu arasında derse yönelik motivasyon puanları açısından anlamlı bir fark bulunmamıştır. Bu bulguya göre kodlama eğitiminde OÇÖO ve OYYÖO kullanımının derse yönelik motivasyonu arttırmada benzer etkiye sahip olduğu belirlenmiştir. Deney 1 grubu ile deney 3 grubu arasında derse yönelik motivasyon puanları açısından anlamlı bir fark bulunmamıştır. Bu bulguya göre kodlama eğitiminde OÇÖO ve ÇÖO kullanımının, derse yönelik motivasyonu arttırmada benzer etkiye sahip olduğu belirlenmiştir. Deney 3 grubu ile kontrol grubu arasında derse yönelik motivasyon puanları açısından anlamlı bir fark bulunmamıştır. Bu bulguya göre kodlama eğitiminde, ÇÖO kullanımının derse yönelik motivasyonu arttırmada etkili olmadığı görülmüştür.

Deneysel işlemten beş hafta sonra elde edilen izleme testi puanları incelendiğinde; deney 1 grubunun izleme testi puan ortalaması ile deney 3 grubu izleme testi puan ortalaması arasında deney 1 grubu lehine anlamlı farklılık görülmüştür. İki grup arasında son test ölçümlerinde, anlamlı farklılığa rastlanmamışken izleme ölçümlerinde deney 1 grubu lehine anlamlı farklılık gözlemlenmiştir. Elde edilen bu bulguya göre kodlama eğitiminde OÇÖO ve ÇÖO kullanımının derse yönelik motivasyonu arttırmada benzer etkiye sahip olmalarına rağmen, kalıcılığı sağlama OÇÖO'nun daha etkili olduğu belirlenmiştir. Deney 1 grubunun izleme testi puan ortalaması ile kontrol grubu izleme testi puan ortalaması arasında deney 1 grubu lehine anlamlı farklılık görülmüştür. İki grup arasında son test ölçümlerinde deney 1 grubu lehine olan anlamlı farklılığa, izleme ölçümlerinde de rastlanmıştır. Elde edilen bu bulguya göre kodlama eğitiminde OÇÖO kullanımının derse yönelik motivasyonu arttırmada ve kalıcılığı sağlamada YYÖO'dan daha etkili olduğu belirlenmiştir. Deney 2 grubunun izleme testi puan ortalaması ile deney 3 grubu izleme testi puan ortalaması arasında deney 2 grubu lehine anlamlı farklılık görülmüştür. İki grup arasında son test ölçümlerinde deney 2 grubu lehine olan anlamlı farklılığa izleme ölçümlerinde de rastlanmıştır. Elde edilen bu bulguya göre kodlama eğitiminde OYYÖO kullanımının derse

yönelik motivasyonu arttırmada ve kalıcılığı sağlamada ÇÖÖ'dan daha etkili olduğu belirlenmiştir. Deney 2 grubunun izleme testi puan ortalaması ile kontrol grubu izleme testi puan ortalaması arasında deney 2 grubu lehine anlamlı farklılık görülmüştür. İki grup arasında son test ölçümlerinde deney 2 grubu lehine olan anlamlı farklılığa, izleme ölçümlerinde de rastlanmıştır. Elde edilen bu bulguya göre kodlama eğitiminde OYYÖÖ kullanımının derse yönelik motivasyonu arttırmada ve kalıcılığı sağlamada YYÖÖ'dan daha etkili olduğu belirlenmiştir. Deney 1 grubunun izleme testi puan ortalaması ile deney 2 grubu izleme testi puan ortalaması arasında anlamlı farklılık görülmemiştir. İki grup arasında son test ölçümlerinde anlamlı farklılığa rastlanmamış ve izleme ölçümlerinde de aynı durum devam etmiştir. Elde edilen bu bulguya göre kodlama eğitiminde OÇÖÖ ve OYYÖÖ kullanımının derse yönelik motivasyonu arttırmada ve kalıcılığı sağlamada benzer etkiye sahip olduğu belirlenmiştir. Deney 3 grubunun izleme testi puan ortalaması ile kontrol grubu izleme testi puan ortalaması arasında anlamlı farklılık görülmemiştir. İki grup arasında son test ölçümlerinde anlamlı farklılığa rastlanmamış ve izleme ölçümlerinde de aynı durum devam etmiştir. Elde edilen bu bulguya göre kodlama eğitiminde ÇÖÖ ve YYÖÖ kullanımının derse yönelik motivasyonu arttırmada ve kalıcılığı sağlamada benzer etkiye sahip olduğu belirlenmiştir.

Alanyazın incelendiğinde kodlama eğitiminde OÇÖÖ, OYYYÖÖ, ÇÖÖ ve YYÖÖ'nun birlikte karşılaştırıldığı araştırmalara rastlanmamaktadır. Bununla birlikte kodlama eğitiminde oyunlaştırma bileşenleri ve çevrimiçi öğrenme kullanımına yönelik araştırmalar da kısıtlıdır. Fidan (2016) Scratch ile programlama öğretiminde oyunlaştırma kullanımının öğrencilerin motivasyonlarını arttırdığı sonucuna ulaşmıştır. Türk ve Gören (2017) yaptıkları çalışmada, öğrencilerin kendi hızlarında programlama öğrenebilmelerine olanak tanıyan oyunlaştırılmış web tabanlı öğrenme ortamı tasarlamışlardır. Python programlama dili öğretiminde, test edilen oyunlaştırılmış öğrenme ortamında öğrencilerin ilerlemelerini görmeleri ve çalışmalarıyla ilgili anında dönüt almaları motive olmalarını sağlamıştır. Dolayısıyla alanyazında yapılan araştırmalarda araştırma sonuçlarımızı destekler niteliktedir.

Kodlama eğitimi dışında oyunlaştırılmış yüz yüze öğrenme ile geleneksel yüz yüze öğrenmenin karşılaştırıldığı araştırmalar incelendiğinde; Sarı ve Altun (2016) oyunlaştırma unsurlarının öğretim ortamına dâhil edilmesinin öğrencilerin motivasyonlarını arttırdığı sonucuna ulaşmışlardır. Ece (2017) oyunlaştırılmış öğrenme ortamında kelime öğretimi yaptığı araştırmasında, oyunlaştırılmış öğrenme ortamının motivasyonu arttırdığını



gözlemlemiştir. Çakıroğlu ve diğerleri (2017) yaptıkları araştırmada; oyunlaştırılmış yüz yüze öğrenme ortamının motivasyon üzerinde olumlu etki yarattığını gözlemlemiştir. Hüner (2018) yaptığı araştırmada, ikinci dil öğreniminde oyunlaştırılmış yüz yüze öğrenme ortamının motivasyonu arttırdığı sonucuna ulaşmıştır. Alanyazından elde edilen bu sonuçlar, araştırmada elde ettiğimiz sonuçları destekler niteliktedir. Elde edilen bu sonuçların aksine Domínguez et al. (2013) bilgi ve iletişim teknolojilerinin öğretimi amacıyla kullandığı oyunlaştırılmış çevrimiçi öğrenme ortamının motivasyonu düşürdüğünü gözlemlemiştir. Benzer şekilde Hanus ve Fox (2015) iletişim dersinde, oyunlaştırılmış yüz yüze öğrenme ortamı ile geleneksel yüz yüze öğrenme ortamını karşılaştırdığı araştırmalarında, oyunlaştırılmış yüz yüze öğrenme ortamında motivasyonun düştüğünü gözlemlemiştir. Dolayısıyla bu sonuçlarda araştırma sonuçlarımızla tutarlılık göstermemektedir.

Oyunlaştırılmış çevrimiçi öğrenme ile geleneksel çevrimiçi öğrenmenin karşılaştırıldığı araştırmalar incelendiğinde; Tunga (2016) oyunlaştırılmış çevrimiçi öğrenme ortamı ve geleneksel oyunlaştırılmış öğrenme ortamını karşılaştırdığı araştırmasında, oyunlaştırılmış öğrenme ortamının motivasyonu arttırdığı sonucuna ulaşmıştır. Alanyazından elde edilen bu sonuçlar araştırmada elde ettiğimiz sonuçları destekler niteliktedir. Elde edilen bu sonuçların aksine Domínguez et al. (2013) araştırmalarında oyunlaştırılmış çevrimiçi öğrenme ortamında motivasyon düşüşünün olduğunu gözlemlemiştir. Polat (2014) yaptığı araştırmada, oyunlaştırma çevrimiçi öğrenme ortamının İngilizce dil eğitimi alan öğrencilerin genel dil motivasyonlarına etkisinin olmadığı sonucuna ulaşmıştır. Dolayısıyla bu sonuçlar araştırma sonuçlarımızla tutarlılık göstermemektedir.

Alanyazında oyunlaştırma bileşenlerinin kullanıldığı farklı öğrenme ortamları da karşılaştırılmıştır. Su ve Cheng (2015) fen bilimleri dersinde; oyunlaştırılmış mobil öğrenme, geleneksel mobil öğrenme ve geleneksel yüz yüze öğrenme ortamının karşılaştırıldığı araştırmaların da oyunlaştırılmış mobil öğrenmenin motivasyonu yükselttiğini gözlemlemiştirlerdir. Özkan (2016) araştırmasında, oyunlaştırılmış harmanlanmış öğrenme ortamında İngilizce derslerinin öğrencilerin motivasyonlarına olumlu etki sağladığını gözlemlemiştir. Fiş Erümit (2016) yaptığı tasarım tabanlı araştırmada, bir dersin nasıl oyunlaştırılabileceğini ve oyunlaştırmayla ilgili bir tasarım çerçevesi oluşturmayı amaçlamıştır. Araştırma sonuçlarına göre oyunlaştırmının motivasyonu olumlu etkilediğini gözlemlemiştir. Zainuddin (2018) yaptığı araştırmada, oyunlaştırılmış ters yüz edilmiş öğretimin öğrencilerin motivasyonları üzerinde olumlu etkiye sahip olduğu sonucuna ulaşmıştır. Bu araştırmalara göre, oyunlaştırma bileşenlerinin ve çevrimiçi öğrenmenin

birlikte kullanıldığı öğrenme ortamlarında motivasyonun yükseldiği görülmektedir. Bunu aksine Meşe (2016) yaptığı araştırmada, harmanlanmış öğrenme ortamlarında oyunlaştırma bileşenlerinin etkinliğini ortaya koymaya çalışmıştır. Oyunlaştırılmış çevrimiçi ve yüz yüze öğrenme ortamının birlikte kullanıldığı deney grubunda motivasyon açısından bir fark gözlemlenmemiştir.

### **5.1.3. Tutum ölçeğinden elde edilen bulgulara yönelik sonuç ve tartışma**

Çevrimiçi ve yüz yüze kodlama eğitiminde oyunlaştırma öğeleri kullanımının kodlamaya karşı tutum üzerindeki etkileri incelendiğinde; deney 1 grubunun deneysel işlem sonrasında yapılan son test ölçümlerinde elde edilen tutum puanlarının anlamlı derecede yükseldiği ve izleme ölçümlerinde anlamlı bir değişimin olmadığı görülmüştür. Elde edilen bu bulguya göre; kodlama eğitiminde çevrimiçi öğrenme ortamı ve oyunlaştırma bileşenlerinin birlikte kullanımı, kodlamaya karşı tutumu yükseltmekte ve kalıcılığı sağlamada etkili olduğu görülmektedir. Deney 2 grubunun deneysel işlem sonrasında yapılan son test ölçümlerinde elde edilen tutum puanlarının anlamlı derecede yükseldiği ve izleme ölçümlerinde anlamlı bir değişimin olmadığı görülmüştür. Bu bulgu kodlama eğitiminde yüz yüze öğrenme ortamı ile birlikte oyunlaştırma bileşenleri kullanımının kodlamaya karşı tutumu arttırmada ve kalıcılığı sağlamada etkili olduğunu göstermektedir. Deney 3 grubunun deneysel işlem sonrasında yapılan son test ölçümlerinden elde edilen kodlamaya karşı tutum puanlarının, yükseldiği ve izleme ölçümlerinde de anlamlı bir değişimin olmadığı belirlenmiştir. Elde edilen bu bulguya göre, kodlama eğitiminde çevrimiçi öğrenme ortamları kullanımının kodlamaya karşı tutumu arttırmada ve kalıcılığı sağlamada etkili olduğu görülmüştür. Hiçbir deneysel işlemin yapılmadığı kontrol grubunun son test ölçümlerinden elde edilen tutum puanlarının yükseldiği ve izleme ölçümlerinde anlamlı bir değişimin olmadığı gözlenmiştir. Elde edilen bu bulguya göre, kodlama eğitiminde yüz yüze öğrenme ortamı kullanımının kodlamaya karşı tutumu arttırmada ve kalıcılığı sağlamada etkili olduğu belirlenmiştir.

Gruplar arasındaki ilişkiyi ortaya koymak amacıyla yapılan ikili karşılaştırma testinden elde edilen bulgular incelendiğinde; OÇÖÖ, OYYÖÖ ve ÇÖÖ kullanıldığı deney 1, deney 2 ve deney 3 grupları ile kontrol grubunun kodlamaya karşı tutum puanları arasında deney grupları lehine anlamlı bir farklılık görülmüştür. Elde edilen bu bulgulara göre; kodlama eğitiminde OÇÖÖ, OYYÖÖ ve ÇÖÖ kullanımının kodlamaya karşı tutumu daha fazla

arttırdığı görülmüştür. Deney 1 grubunun tutum son test puan ortalamasının, deney 3 grubu son test puan ortalamasından anlamlı derecede yüksek olduğu gözlemlenmiştir. Başka bir ifadeyle kodlama eğitiminde OÇÖO kullanımının kodlamaya karşı tutumu arttırmada ÇÖO'dan daha etkili olduğu belirlenmiştir. Deney 2 grubunun tutum son test puan ortalamasının, deney 3 grubu son test puan ortalamasından anlamlı derecede yüksek olduğu gözlemlenmiştir. Başka bir ifadeyle kodlama eğitiminde OYYÖO kullanımının kodlamaya karşı tutumu arttırmada ÇÖO'dan daha etkili olduğu belirlenmiştir. Deney 1 grubu ile deney 2 grubu arasında kodlamaya karşı tutum puanları açısından anlamlı bir fark bulunmamıştır. Bu bulguya göre kodlama eğitiminde OÇÖO ve OYYÖO kullanımının kodlamaya karşı tutumu arttırmada benzer etkiye sahip olduğu belirlenmiştir.

OÇÖO, OYYÖO ve ÇÖO kullanıldığı deney 1, deney 2 ve deney 3 grupları ile kontrol grubunun kodlamaya karşı izleme testi tutum puanları arasında, deney grupları lehine anlamlı bir farklılık görülmüştür. Elde edilen bulguya göre deney grupları ile kontrol grubu son test tutum puanları arasında görülen anlamlı farklılık izleme ölçümlerinde de devam etmiştir. Başka bir ifadeyle kodlama eğitiminde OÇÖO, OYYÖO ve ÇÖO kullanımının kodlamaya karşı tutumu arttırmada ve kalıcılığı sağlamada etkili olduğu belirlenmiştir. Deney 1 grubu izleme testi puan ortalaması ile deney 3 grubu izleme testi tutum puan ortalaması arasında deney 1 grubu lehine anlamlı farklılık görülmüştür. İki grup arasında son test ölçümlerinde görülen anlamlı farklılık izleme ölçümlerinde de devam etmiştir. Elde edilen bu bulguya göre, kodlama eğitiminde OÇÖO kullanımının kodlamaya karşı tutumu arttırmada ve kalıcılığı ÇÖO'dan daha etkili olduğu belirlenmiştir. Deney 2 grubu izleme testi puan ortalaması ile deney 3 grubu izleme testi tutum puan ortalaması arasında deney 2 grubu lehine anlamlı farklılık görülmüştür. İki grup arasında son test ölçümlerinde görülen anlamlı farklılık izleme ölçümlerinde de devam etmiştir. Elde edilen bu bulguya göre kodlama eğitiminde OYYÖO kullanımının kodlamaya karşı tutumu arttırmada ve kalıcılığı ÇÖO'dan daha etkili olduğu belirlenmiştir. Deney 1 grubu izleme testi puan ortalaması ile deney 2 grubu izleme testi tutum puan ortalaması arasında anlamlı farklılığa rastlanmamıştır. Elde edilen bu bulguya göre, kodlama eğitiminde OÇÖO kullanımının kodlamaya karşı tutumu arttırmada ve kalıcılığı sağlamada OYYÖO ile benzer etkiye sahip olduğu belirlenmiştir.

Alanyazın incelendiğinde kodlama eğitiminde; OÇÖO, OYYYÖO, ÇÖO ve YYÖO'nun birlikte karşılaştırıldığı araştırmalara rastlanmamaktadır. Bununla birlikte kodlama eğitiminde oyunlaştırma bileşenleri ve çevrimiçi öğrenme kullanımına yönelik

arařtırmalarda kısıtlıdır. Yıldız (2018) yaptıđı arařtırmada; oyunlařtırma yaklařımının blok temelli algoritmik dūřınme aktivitelele ile birlikte kullanımının, ođrencilerin tutumlarını olumlu yōnde etkilediđini gōzlemlemiřtir. Polat (2014) yaptıđı arařtırmada oyunlařtırma yōntemine karřı ođrencilerin olumlu tutum sergilediđini belirlemiřtir. Yıldırım (2016) yaptıđı arařtırmada oyunlařtırılmıř ođretim ilke ve yōntemleri dersi programını geliřtirip uygulamıřtır. Elde edilen sonuēlara gōre, oyunlařtırılmıř ođrenme ortamı deney grubundaki ođrencilerin tutumlarında olumlu etki yaratmıřtır. Kalkan (2016) oyunlařtırılmıř sanal ođrenme ortamının, tutum geliřtirme potansiyelinde önemli rol oynadıđını gōr÷lmüřtür.

## **5.2. Öneriler**

Bu bölümde arařtırma kapsamında elde edilen sonuēlara yōnelik arařtırmacılara ve uygulayıcılara öneriler sunulmuřtur. Bu öneriler çevrimiēi ve yüz yüze kodlama eđitiminde oyunlařtırma ođeleri kullanımına yōnelik olacaktır. Ayrıca ileride yapılabilecek ēalıřmalara yōnelik arařtırmacılara da belirli öneriler sunulmuřtur.

### **5.2.1. Arařtırma sonuēlarına dayalı öneriler**

- Kodlama eđitiminde oyunlařtırma bileřenleri ve çevrimiēi ođrenmenin kullanıldıđı ođrenme ortamlarının, geleneksel yüz yüze ođrenme ortamına gōre akademik bařarıyı daha fazla yükselttiđi gōr÷lmüřtür. Bu nedenle oyunlařtırma bileřenleri ve çevrimiēi ođrenme ortamları kullanımının, akademik bařarı üzerinde olumlu etkisi dikkate alınarak kodlama eđitimlerinde kullanılması önerilmektedir.
- Oyunlařtırma bileřenlerinin, yüz yüze ve çevrimiēi ođrenme ortamlarında motivasyona olumlu katkı sađladıđı arařtırma sonuēlarında gōzlemlenmiřtir. Bu nedenle kodlama eđitiminde, faklı ođrenme ortamlarıyla birlikte oyunlařtırma bileřenlerinin kullanılması önerilmektedir.
- Oyunlařtırma bileřenleri ve çevrimiēi ođrenmenin, kodlamaya yōnelik olumlu tutum geliřtirme potansiyeline sahip olduđu belirlenmiřtir. Bu nedenle kodlamaya karřı ođrencilerde olumlu tutum geliřtirmek amacıyla, oyunlařtırma bileřenleri ve çevrimiēi ođrenmenin kullanılması önerilmektedir.
- İyi tasarlanmış oyunlařtırılmıř ođrenme ortamları ođrencilerin bařarılarında, motivasyonlarında ve tutumlarında olumlu etki oluřturmaktadır. Bu nedenle

oyunlaştırılmış öğrenme ortamı tasarlanırken, derse uygun bir tasarım modeli belirlenmeli ve aşamaları titizlikle düzenlenmelidir.

### **5.2.2.İleride yapılabilecek arařtırmalara yönelik öneriler**

- Arařtırmada, oyunlařtırma modellerinden Werbach ve Hunter (2012:85) tarafından ortaya konulan ve 6 ařamadan oluřan oyunlařtırma tasarım modeli kullanılmıřtır. İleride yapılacak arařtırmalarda, kodlama eęitiminde oyunlařtırılmıř ders tasarımına yönelik farklı modeller kullanılarak sonular karřılařtırılabilir.
- Yapılan arařtırmada oyunlařtırma bileřeni olarak; bařarı, rozet, lider tahtası, seviye, puan, ierik ama, hediye ve ilerleme durumu kullanılmıřtır. İleride yapılacak arařtırmalarda farklı oyunlařtırma bileřenleri kullanılarak oluřan etkiler karřılařtırılabilir.
- Arařtırmada Kapp (2013) tarafından, oyunlařtırma trlerinden ierik zerinde hibir deęiřiklięe gidilmeyerek kullanılan yapısal oyunlařtırma kullanılmıřtır. İleride yapılacak arařtırmalarda, ierik oyunlařtırma tr kullanılarak kodlama eęitiminde nasıl bir etki oluřturacaęı incelenebilir.
- Arařtırma, kodlama eęitiminde yrtlmřtr. Farklı derslerde oyunlařtırılmıř ders tasarımı kullanılarak arařtırmalar yapılabilir. Dięer derslerde yapılan arařtırmaların sonularıyla, arařtırmada elde edilen sonular karřılařtırılabilir.
- Arařtırmada programlama dili olarak Python, evrimii öğrenme ortamı olarak Moodle kullanılmıřtır. Farklı bir programlama dili ve evrimii öğrenme ortamı kullanılarak ileride arařtırmalar yrtlebilir. Programlama dili ve evrimii öğrenme ortamı deęiřikliklerinde oluřan etkiler, arařtırma sonularıyla karřılařtırılabilir.

## KAYNAKLAR

- Aksoy, A. (2017). *Yeni başlayanlar için python* (2. Baskı). İstanbul: Abaküs Kitap Yayın Dağıtım.
- Ar, N. A. (2016). *Oyunlaştırmayla öğrenmenin meslek lisesi öğrencilerinin akademik başarı ve öğrenme stratejileri kullanımı üzerine etkisi*. (Yüksek Lisans Tezi). YÖK Tez Merkezi veri tabanından erişildi (Tez No: 423538).
- Araya, R., Arias Ortiz, E., Bottan, N. L. ve Cristia, J. P. (2019). *Does gamification in education work?: Experimental evidence from chile*. Inter-American Development Bank. doi: 10.18235/0001777
- Arslan, K. ve Tanel Z. (2017, Mayıs). Arduino ile dinleyen öğrencilerden yapan öğrencilere geçiş. *In 11th International Computer & Instructional Technologies Symposium (ICITS)*, Malatya, Turkey.
- Atabay, E. ve Albayrak, M. (2020). Okul öncesi dönem çocuklarına oyunlaştırma ile algoritma eğitimi verilmesi. *Mühendislik Bilimleri ve Tasarım Dergisi*, 8(3), 856-868. doi: 10.21923/jesd.672232
- Atılğan, H. (2016). Madde ve test istatistikleri. H. Atılğan (Ed.). *Eğitimde ölçme ve değerlendirme* (9. Baskı, ss. 293-314). Ankara: Anı Yayıncılık.
- Bal, M. (2019). Use of digital games in writing education: An action research on gamification. *Contemporary Educational Technology*, 10(3), 246-271. doi: 10.30935/cet.590005
- Balcı, B. (2011). E-öğrenme sistemindeki başarı faktörleri. Yamamoto, G.T., Demiray, U., ve Kesim, M. (Ed.), *Türkiye’de e-öğrenme: Gelişmeler ve uygulamalar içinde Bölüm: 20*, (ss.385-397). Ankara: Efil Yayınevi
- Bartle, R. (1996). Hearts, clubs, diamonds, spades: Players who suit MUDs. *The Journal of Virtual Environments*, 1(1). Erişim adresi: [https://www.researchgate.net/publication/247190693\\_Hearts\\_clubs\\_diamonds\\_spades\\_Players\\_who\\_suit\\_MUDs](https://www.researchgate.net/publication/247190693_Hearts_clubs_diamonds_spades_Players_who_suit_MUDs)
- Bayman, P. ve Mayer, R., (1988), Using conceptual models to teach basic computer programming. *Journal of Educational Psychology*, 80(3), 291-298. doi: 10.1037/0022-0663.80.3.291

- Bayrakçeken, S. (2011). Test geliştirme. E. Karip (Ed.). *Ölçme ve değerlendirme*. (4. Baskı, ss. 294-325). Ankara: PegemA Yayıncılık.
- Bayram, Y. T. ve Çalışkan, H. (2019). Sosyal bilgiler dersinde kullanılan oyunlaştırılmış yaratıcı etkinlikler: Bir eylem araştırması. *Journal of Individual Differences in Education*, 1(1), 30-49. Erişim adresi: <https://dergipark.org.tr/tr/download/article-file/724822>
- Bidwell, A. (2013). Tech companies work to combat computer science education gap. Erişim adresi: <https://www.usnews.com/news/articles/2013/12/27/tech-companies-work-to-combat-computer-science-education-gap>.
- Bilgiç, H. G., Duman, D. ve Seferoğlu, S. S. (2011). Dijital yerlilerin özellikleri ve çevrimiçi ortamların tasarlanmasındaki etkileri. *XIII. Akademik Bilişim Konferansı* (s. 257-263). Malatya: İnönü Üniversitesi.
- Buckley, P. ve Doyle, E. (2016). Gamification and student motivation. *Interactive learning environments*, 24(6), 1162-1175. doi: 10.1080/10494820.2014.964263.
- Burke B., (2014). *Gamify: How gamification motivates people to do extraordinary things*. Brookline, Gartner.
- Büyüköztürk, Ş. (2012). *Sosyal bilimler için veri analizi el kitabı: İstatistik, araştırma deseni spss uygulamaları ve yorum* (16. baskı). Ankara: PegemA Yayıncılık.
- Büyüköztürk, Ş., Akgün, Ö. E., Kahveci, Ö. ve Demirel, F. (2004). Güdülenme ve öğrenme stratejileri ölçeğinin Türkçe formunun geçerlik ve güvenirlik çalışması. *Kuram ve Uygulamada Eğitim Bilimleri*, 4(2), 207-239. Erişim adresi: <https://toad.halileksi.net/sites/default/files/pdf/gudulenme-ve-ogrenme-stratejileri-olcegi-toad.pdf>
- Büyüköztürk, Ş., Akgün, Ö.A., Karadeniz, Ş., Kılıç-Çakmak, E., Demirel, F. (2007). Güdülenme ve öğrenme stratejileri ölçeği'nin 12-18 yaş Türkiye norm çalışması. *Türkiye Bilimsel ve Teknik Araştırma Kurumu (TÜBİTAK)*, Proje No: SOBAG - 104K097.
- Büyüköztürk, Ş., Kılıç Çakmak, E., Akgün, Ö.E., Karadeniz, Ş ve Demirel, F. (2012). *Bilimsel araştırma yöntemleri* (11. Baskı). Ankara:PegemA Yayıncılık.
- Can, A. (2016). *SPSS ile bilimsel araştırma sürecinde nicel veri analizi* (4. Baskı). Ankara: PegemA Yayıncılık.

- Chiu, C.F. (2014). Teaching programming concepts to K-12 teachers with Scratch. *Journalism and Mass Communication*, 4(2), 125-132.
- Chou, Y. (2015). Gamification in education: top 10 gamification case studies that will change our future. Eriřim adresi: <https://yukaichou.com/gamification-examples/octalysis-complete-gamification-framework/>
- Cooper, S., Powers, K., McNally, M., Goldman, K.J. Proulx, V. ve Carlisle M. (2006). Tools for teaching introductory programming: What works? *ACM SIGCSE Bulletin*, 38, 560-561. doi: 10.1145/1121341.1121514
- Cořkun, M. M. (2017). Udey python eęitimleri. Eriřim Adresi: <https://www.udemy.com/user/mustafamuratcokun/>
- Csikszentmihalyi, M. (1991). *Flow: The psychology of optimal experience*. New York: Harper Perennial.
- Csikszentmihalyi, M. ve Rathunde, K. (2014). The development of the person: An experiential perspective on the ontogenesis of psychological complexity. M. Csikszentmihalyi (Ed.). *Applications of flow in human development and education*, (s. 7-79). Dordrecht Heidelberg New York London: Springer.
- Çakıroęlu, Ü., Bařıbüyük, B., Güler, M., Atabay, M., ve Memiř, B. Y. (2017). Gamifying an ICT course: Influences on engagement and academic performance. *Computers in human behavior*, 69, 98-107. doi: 10.1016/j.chb.2016.12.018
- Çalıřkan, H. (2002, Mayıs). Çevrimiçi (Online) eęitimde öęrenci etkileřimi. *Açık ve Uzaktan Eęitim Sempozyumu*, Eskiřehir.
- Çilengir, M. D. (2019). *Oyunlařtırma yaklařımı ile blok tabanlı programlama öęretiminin bařarı ve motivasyona etkisi* (Yüksek Lisans Tezi). YÖK Tez Merkezi veri tabanından eriřildi (Tez No: 594858).
- Deci, E. L. ve Ryan, R. M. (2000). The "what" and "why" of goal pursuits: human needs and the self-determination of behavior. *Psychological Inquiry*, 11 (4), 227-268. doi: 10.1207/S15327965PLI1104\_01
- Deci, E. L., Kostner, R. ve Ryan, R. M. (2001). Extrinsic rewards and intrinsic motivation in education: Reconsidered once again. *Review of Educational Research*, 71, 1–27. doi: 10.3102/00346543071001001



- Deci, E., Connell, J., ve Ryan, R. (1989). Self determination in a work organization. *Journal of Applied Psychology*, 74 (4), 580-590. Eriřim adresi: [http://selfdeterminationtheory.org/SDT/documents/1989\\_DeciConnellRyan.pdf](http://selfdeterminationtheory.org/SDT/documents/1989_DeciConnellRyan.pdf)
- Deterding, S. (2012). Gamification: Designing for motivation. *Interactions*, 19(4), 14-17. Eriřim adresi: [https://www.researchgate.net/profile/Sebastian\\_Deterding/publication/244486331\\_Gamification\\_Designing\\_for\\_motivation/links/0a85e53a049814673c000000.pdf](https://www.researchgate.net/profile/Sebastian_Deterding/publication/244486331_Gamification_Designing_for_motivation/links/0a85e53a049814673c000000.pdf)
- Deterding, S., Dixon, D., Khaled, R. ve Nacke, L. (2011, September). From game design elements to gamefulness: defining gamification. In *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments* (pp. 9-15). ACM.
- Domínguez, A., Saenz-De-Navarrete, J., De-Marcos, L., Fernández-Sanz, L., Pagés, C., ve Martínez-Herráiz, J. J. (2013). Gamifying learning experiences: Practical implications and outcomes. *Computers & Education*, 63, 380-392. doi:10.1016/j.compedu.2012.12.020
- Durak, G. (2013). *Programlama dillerinin çevrimiçi öğretimi: öğrenenlerin tutumlarının, memnuniyetlerinin ve akademik başarılarının incelenmesi* (Doktora Tezi). YÖK Tez Merkezi veri tabanından erişildi (Tez No: 348970).
- Ece, S. (2017). *Oyunlaştırılmış oyun temelli kelime öğretiminin öğrencilerin başarılarına ve motivasyonlarına olan etkisi* (Yüksek Lisans Tezi). YÖK Tez Merkezi veri tabanından erişildi (Tez No: 454559).
- Economou, D., Doumanis, I., Pedersen, F., Kathrani, P., Mentzelopoulos, M. ve Bouki, V. (2015). *Evaluation of a dynamic role-playing platform for simulations based on Octalysis gamification framework*. In *Intelligent Environments (Workshops)*(pp. 388-395). Eriřim adresi: <https://www.semanticscholar.org/paper/Evaluation-of-a-dynamic-role-playing-platform-for-Economou-Doumanis/130259bc92ab1b96c86c377ddf9c685c402d97e0>
- Ertan, K. (2020). *Oyunlaştırılmış ingilizce dersinde başarı, tutum ve motivasyon değişkenlerinin incelenmesi* (Yüksek Lisans Tezi). YÖK Tez Merkezi veri tabanından erişildi (Tez No: 633889).

- Eryılmaz, S. (2003). *Algoritma tasarlama ve programlamaya giriş*. Ankara: Detay Yayıncılık.
- Farooq, M. S., Khan, S. A., Ahmed, F., Islam, S. ve Abid, A. (2014). Choice of Pedagogical Approaches towards First Programming Languages. *Journal of Applied Environmental and Biological Sciences*, 4(7S), 311-317. Erişim adresi: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1091.5021&rep=rep1&type=pdf>
- Fidan, A. (2016). *Scratch ile programlama öğretiminde oyunlaştırmanın öğrenci katılımına etkisi* (Yüksek Lisans Tezi). YÖK Tez Merkezi veri tabanından erişildi (Tez No: 445164).
- Field, A. (2013). *Discovering statistics using SPSS*. Sage publications.
- Fiş Erümit, S. (2016). *Oyunlaştırma yaklaşımlarının eğitimde kullanımı: Tasarım tabanlı bir araştırma* (Doktora Tezi). YÖK Tez Merkezi veri tabanından erişildi (Tez No: 442994).
- Fogg, B. J. (2009). A behavior model for persuasive design. *Proceedings of the 4th international Conference on Persuasive Technology* (pp.40). ACM. doi: 10.1145/1541948.1541999
- Glover, I. (2013). Play as you learn: Gamification as a technique for motivating learners. *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications*. Chesapeake: AACE.
- Gomes, A. ve Mendes, A. J. (2007, October). Learning to program difficulties and solutions. *International conference on Engineering Education*, Portugal, Coimbra.
- Gundurao, H. K., Manjunath, N. S. ve Nachappa, M. N. (2010). *Computer Technology and Computer Programming*. Mumbai, IND: Global Media.
- Gülbahar, Y. ve Kalelioğlu, F. (2018). Bilişim teknolojileri ve bilgisayar bilimi: Öğretim programı güncelleme süreci. *Millî Eğitim Dergisi*, 47(217), 5-23. Erişim adresi: <https://dergipark.org.tr/tr/download/article-file/539818>
- Gülbahar, Y., Kalelioğlu, F. ve Karataş, E. (2017). *Ortaöğretim bilgisayar bilimi ders kitabı kur-1*. Ankara: Milli Eğitim Bakanlığı.

- Hanus M. D. ve Fox, J. (2015). Assessing the effects of gamification in the classroom: A longitudinal study on intrinsic motivation, social comparison, satisfaction, effort, and academic performance. *Computers & Education*,2015,80,152-161. doi: 10.1016/j.compedu.2014.08.019
- Hardnett, C. R. (2008, February). Gaming for middle school students: building virtual worlds. In *Proceedings of the 3rd international conference on Game development in computer science education* (pp. 21-25). ACM.
- Huang, B., Hew, K. F. ve Lo, C. K. (2019). Investigating the effects of gamification-enhanced flipped learning on undergraduate students' behavioral and cognitive engagement. *Interactive Learning Environments*, 27(8), 1106-1126. <https://doi.org/10.1080/10494820.2018.1495653>
- Hüner, O. (2018). *Oyunlaştırmanın ikinci dil eğitiminde akademik başarı ve motivasyon üzerine olan etkileri* (Yüksek Lisans Tezi). YÖK Tez Merkezi veri tabanından erişildi (Tez No: 495526).
- İnce, M. D. (2017, 16 Ağustos). Python dersleri. Erişim Adresi <https://www.pythondersleri.com>.
- Kalelioğlu, F. ve Keskinçılıç, F. (2017). Bilgisayar bilimi eğitimi için öğretim yöntemleri. Y. Gülbahar (Ed.), *Bilgi işlemsel düşünmeden programlamaya* (1. Baskı, pp. 155–170). Ankara: PegemA Yayıncılık
- Kalkan, A. (2016). *3D sanal dünyalarda oyunlaştırmanın ortaokul 5. sınıf öğrencilerinin başarı, akış ve tutumlarına etkisinin araştırılması* (Yüksek Lisans Tezi). YÖK Tez Merkezi veri tabanından erişildi (Tez No: 433813).
- Kandemir, C. M. (2017). Metin Tabanlı Programlama. Y. Gülbahar (Ed.), *Bilgi işlemsel düşünmeden programlamaya* (1. Baskı, pp. 267–292). Ankara: PegemA Yayıncılık
- Kapp, K. M. (2012). *The gamification of learning and instruction: game-based methods and strategies for training and education*. John Wiley & Sons.
- Karabak, D. ve Güneş, A. (2013). Ortaokul birinci sınıf öğrencileri için yazılım geliştirme alanında müfredat önerisi. *Eğitim ve Öğretim Araştırmaları Dergisi*, 21(2-3), 163-169. Erişim adresi: <http://www.jret.org/FileUpload/ks281142/File/21b.karabak.pdf>

- Kavaklı Duman, E. (2019). *Lise öğrencilerinde yabancı dil öğrenimini geliştirmek için oyunlaştırmanın kullanımı* (Yüksek Lisans Tezi). YÖK Tez Merkezi veri tabanından erişildi (Tez No: 584497).
- Kert, S. B. ve Uğraş, T. (2009, Mayıs). Programlama eğitiminde sadelik ve eğlence: Scratch örneği. *In The First International Congress of Educational Research*. Çanakkale, Turkey.
- Kesici T., Kocabaş Z. (2007). *Bilgisayar 2 Ders Kitabı* (2. Baskı). Ankara: Semih Ofset.
- Kim, E., Rothrock, L. ve Freivalds, A. (2018a). An empirical study on the impact of lab gamification on engineering students' satisfaction and learning. *International Journal of Engineering Education*, 34(1), 201-216. doi: 10.1109/FIE.2016.7757442
- Kim, S., Song, K., Lockee, B. ve Burton, J. (2018b). What is gamification in learning and education?. *In Gamification in learning and education* (pp. 25-38). Springer, Cham.
- Kumar, B. and Khurana, P. (2012). Gamification in education - learn computer programming with fun. *International Journal of Computers and Distributed Systems*, 2(1), 46-53. Erişim adresi: [http://www.academia.edu/download/34327392/Gamification\\_in\\_Education.pdf](http://www.academia.edu/download/34327392/Gamification_in_Education.pdf).
- Lazarro, N. (2004). *Why we play games: four keys to more emotion without story*. Player experience research and design for mass market interactive entertainment, XEO Design Inc. Erişim adresi: [http://xeodesign.com/xeodesign\\_whyweplaygames.pdf](http://xeodesign.com/xeodesign_whyweplaygames.pdf)
- Lee, J. J. ve Hammer, J. (2011). Gamification in education: What, how, why bother?. *Academic exchange quarterly*, 15(2), 1-5. Erişim adresi: [https://www.researchgate.net/publication/258697764\\_Gamification\\_in\\_Education\\_What\\_How\\_Why\\_Bother](https://www.researchgate.net/publication/258697764_Gamification_in_Education_What_How_Why_Bother)
- Lin, C. C. ve Zhang, M. (2003). The use of computer animation in teaching discrete structures course. *Midwest Instruction and Computing Symposim (MICS)*.
- Malan, D. J. ve Leitner, H. H. (2007). Scratch for budding computer scientists. *ACM Sigcse Bulletin*, 39(1), 223-227. Erişim adresi: [https://www.researchgate.net/publication/38413358\\_Scratch\\_for\\_Budding\\_Computer\\_Scientists](https://www.researchgate.net/publication/38413358_Scratch_for_Budding_Computer_Scientists)

- Marczewski, A. (2015). *User types. In even ninja monkeys like to play: gamification, game thinking and motivational design* (1st ed., pp. 65-80). CreateSpace Independent Publishing Platform.
- McGill, T. J. ve Volet, S. E. (1997). A conceptual framework for analyzing students' knowledge of programming. *Journal of Research on Computing in Education*, 29(3), 276-197. doi: 10.1080/08886504.1997.10782199
- Meşe, C. (2016). *Harmanlanmış öğrenme ortamlarında oyunlaştırma bileşenlerinin etkinliği* (Doktora Tezi). YÖK Tez Merkezi veri tabanından erişildi (Tez No: 438244).
- Mohammed, D. A. (2018). *E-öğrenimde oyunlaştırma: Bir Irak üniversitesinde öğrenci performansı ve algısı üzerine bir çalışma* (Yüksek Lisans Tezi). YÖK Tez Merkezi veri tabanından erişildi (Tez No: 504525).
- Moodle. (2019). Moodle statistics. <https://moodle.net/stats/> adresinden 02.08.2019 tarihinde erişilmiştir.
- Moore, M. G. ve Kearsley, G. (1996). *Distance education: A systems view*. Boston, MA: Wadsworth Publishing.
- Naps, T. L., Rößling, G., Almstrum, V., Dann, W., Fleischer, R., Hundhausen, C. ve Velázquez-Iturbide, J. Á. (2002, June). Exploring the role of visualization and engagement in computer science education. In *ACM Sigcse Bulletin* (Vol. 35, No. 2, pp. 131-152). ACM.
- Nelson, M. L. ve Rice, D. (2000, October). Introduction to algorithms and problem solving. In *30th Annual Frontiers in Education Conference. Building on A Century of Progress in Engineering Education. Conference Proceedings (IEEE Cat. No. 00CH37135)* (Vol. 2, pp. S2C-5). IEEE.
- Nuhoğlu, H. (2008). The development of an attitude scale for science and technology course. *Elementary Education Online*, 7(3). Erişim adresi: [https://www.researchgate.net/publication/228621279\\_The\\_development\\_of\\_an\\_attitude\\_scale\\_for\\_science\\_and\\_technology\\_course](https://www.researchgate.net/publication/228621279_The_development_of_an_attitude_scale_for_science_and_technology_course)
- OECD (2008). Partnership for 21 Century Skills. Erişim adresi: <https://www.oecd.org/site/educeri21st/40756908.pdf>.

- Ortiz-Rojas, M., Chiluiza, K. ve Valcke, M. (2019). Gamification through leaderboards: An empirical study in engineering education. *Computer Applications in Engineering Education*, 27(4), 777-788. doi: 10.1002/cae.12116
- Özgül, F. (2017, 17 Ağustos). Python Programlama Dili. Erişim adresi <https://python-istihza.yazbel.com/>.
- Özgür, H. (2011). *Syracuse modeli ile e-öğrenme ortamı için tasarlanmış bir dersin öğrencilerin başarısına etkisi: "Trakya Üniversitesi Eğitim Fakültesi Örneği (Doktora Tezi)*. YÖK Tez Merkezi veri tabanından erişildi (Tez No: 300201).
- Özguven, İ.E. (2014). *Psikolojik testler*. Ankara: Nobel Akademik Yayıncılık.
- Özkan, M. (2016). *İngilizce öğrencilerinin e-öğrenmede oyun öğeleriyle motive edilmesi ve derse katılımı (Yüksek Lisans Tezi)*. YÖK Tez Merkezi veri tabanından erişildi (Tez No: 441375).
- Phoenix Araştırma Enstitüsü (2011). Future Work Skills 2020. Erişim adresi: [http://www.iftf.org/uploads/media/SR-1382A\\_UPRI\\_future\\_work\\_skills\\_sm.pdf](http://www.iftf.org/uploads/media/SR-1382A_UPRI_future_work_skills_sm.pdf).
- Pintrich, P.R., Smith, D.A.F., Garcia, T. ve McKeachie, W.J. (1993). Reliability and predictive validity of the motivated strategies for learning questionnaire (MSLQ). *Educational and Psychological Measurement*, 53 (3), 801-814. doi: 10.1177/0013164493053003024
- Polat, Y. (2014). *Bir vaka incelemesi: Oyunlaştırma yöntemi ve ingilizce öğrencilerinin motivasyonu üzerine etkisi (Yüksek Lisans Tezi)*. YÖK Tez Merkezi veri tabanından erişildi (Tez No: 358356).
- Resnick, M. (2013). Learn to code, code to learn. How programming prepares kids for more than math. *EdSurge*, May 2013. Erişim adresi: <https://web.media.mit.edu/~mres/papers/L2CC2L-handout.pdf>.
- Ribeiro, A., Coelho, A. ve Aguiar, A. (2011). Boobo World – A collaborative videogame to introduce programming concepts to children. *4th annual conference in science and art of video games (VideoJogos 2011)*, Porto, Portugal.
- Sağay, M. A. (2019). *İlköğretim öğrencileri için oyunlaştırma yöntemi ile kodlama eğitimi uygulaması (Yüksek Lisans Tezi)*. YÖK Tez Merkezi veri tabanından erişildi (Tez No: 593713).

- Sarı, A. ve Altun, T. (2016). Oyunlaştırma yöntemi ile işlenen bilgisayar derslerinin etkililiğine yönelik öğrenci görüşlerinin incelenmesi. *Türk Bilgisayar ve Matematik Eğitimi Dergisi*, 7(3), 553-577. doi: 10.16949/turkbilmat.277871
- Saygıner, Ş. ve Tüzün, H. (2018). Programlama eğitimi üzerine bir inceleme: yaşanan zorluklar, mevcut uygulamalar ve güncel yaklaşımlar. B. Akkoyunlu, A. İşman ve H. F. Odabaşı (Ed.), *Eğitim Teknolojileri Okumaları 2018* (ss. 693–710). Ankara: PegemA Yayıncılık.
- Sayın, Z. ve Seferoğlu, S. S. (2016). Yeni bir 21. yüzyıl becerisi olarak kodlama eğitimi ve kodlamanın eğitim politikalarına etkisi. *Akademik Bilişim Konferansı*, 3-5.
- Senemoğlu, N. (2013). *Gelişim Öğrenme ve Öğretim, Kuramdan Uygulamaya*. Ankara: Yargı Yayınevi.
- Sezgin, S., Bozkurt, A., Yılmaz, E. A. ve Van der Linden, N. (2018). Oyunlaştırma, eğitim ve kuramsal yaklaşımlar: öğrenme süreçlerinde motivasyon, adanmışlık ve sürdürülebilirlik. *Mehmet Akif Ersoy Üniversitesi Eğitim Fakültesi Dergisi*, (45), 169-189. doi: 10.21764/maeuefd.339909
- Sözüdoğru, O., Altınay, Z. ve Altınay, F. (2016). Eğitimde sosyal ağlar. Y. Akkoyunlu, B., İşman, A. ve Odabaşı, H. F (Ed.), *Eğitim Teknolojileri Okumaları*. Ankara: Salmat Basım Yayıncılık. Erişim adresi: <http://www.tojet.net/>
- Su, C.H. ve Cheng, C.H. (2015). A mobile gamification learning system for improving the learning motivation and achievements. *Journal of Computer Assisted Learning*, 31(3), 553-577. doi: 10.1111/jcal.12088
- Sun, J. C. Y. ve Rueda, R. (2012). Situational interest, computer self-efficacy and self-regulation: Their impact on student engagement in distance education. *British journal of educational technology*, 43(2), 191-204. doi: 10.1111/j.1467-8535.2010.01157.x
- Sümer, M. (2017). *Açık ve uzaktan öğrenme programlarında oyunlaştırma kullanımı* (Doktora Tezi). YÖK Tez Merkezi veri tabanından erişildi (Tez No: 486284).
- Tabachnick, B. G. ve Fidell, L. S. (2013). *Using Multivariate Statistics*. (Sixth edition). Pearson Education.

- Taşkın, N. (2020). *Oyunlaştırmanın ters yüz öğrenme ortamında öğrenim gören öğrencilerin motivasyonuna, katılımına ve akademik başarısına etkisi* (Doktora Tezi). YÖK Tez Merkezi veri tabanından erişildi (Tez No: 628621).
- Taşpınar, M. (2017). *Sosyal bilimlerde SPSS uygulamalı nicel veri analizi* (1. Baskı). Ankara: PegemA Yayıncılık
- Topal, M. (2020). *Oyunlaştırma ile zenginleştirilmiş çevrimiçi öğrenmenin başarı, çevrimiçi bağlılık ve öğrenme motivasyonu üzerindeki etkisi* (Doktora Tezi). YÖK Tez Merkezi veri tabanından erişildi (Tez No: 612138).
- TTKB. (2016). *Ortaöğretim bilgisayar bilimi dersi (kur 1, kur 2) öğretim programı*. Ankara: Talim ve Terbiye Kurulu Başkanlığı.
- Tunga, Y. (2016). *E-öğrenme ortamlarında oyunlaştırma kullanımının öğrencilerin akademik başarısına ve derse katılım durumuna etkisinin incelenmesi* (Yüksek Lisans Tezi). YÖK Tez Merkezi veri tabanından erişildi (Tez No: 437868).
- Türk, Y. ve Gören S. (2017, Eylül). Gamified self-paced e-learning platform for computer science courses. *9th International Conference ICT Innovations (ICT'2017)*, Skopje, Macedonia.
- Valente, J. (1995). Logo as a window into the mind. *Logo Update*, 1(4). Erişim adresi: <http://el.media.mit.edu/logo-foundation/pubs/logoupdate/v4n1.html>.
- Van-Roy, P. ve Haridi, S. (2004). *Concepts, techniques, and models of computer programming*. MIT. Erişim adresi: [https://www.researchgate.net/publication/231874671\\_Concepts\\_Techniques\\_and\\_Models\\_of\\_Computer\\_Programming](https://www.researchgate.net/publication/231874671_Concepts_Techniques_and_Models_of_Computer_Programming)
- Vujošević-Janičić, M. ve Tošić, D. (2008). The role of programming paradigms in the first programming courses. *The Teaching of Mathematics*, (21), 63-83. Erişim adresi: [https://www.researchgate.net/publication/265630430\\_The\\_role\\_of\\_programming\\_paradigms\\_in\\_the\\_first\\_programming\\_courses](https://www.researchgate.net/publication/265630430_The_role_of_programming_paradigms_in_the_first_programming_courses)
- Weinberg, A. E. (2013). *Computational Thinking: An investigation of the existing scholarship and research*. Erişim adresi: <http://people.cs.vt.edu/~kafura/CS6604/Papers/CT-Existing-Scholarship-Research-Dissertation.pdf>.



- Werbach, K. ve Hunter, D. (2012). *For the win: how game thinking can revolutionize your business*. Philadelphia: Wharton Digital Press.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35. doi:10.1145/1118178.1118215
- Yıldırım, İ. (2016). *Oyunlaştırma temelli öğretim ilke ve yöntemleri dersi öğretim programının geliştirilmesi, uygulanması ve değerlendirilmesi* (Doktora Tezi). YÖK Tez Merkezi veri tabanından erişildi (Tez No: 422970).
- Yıldırım, İ. (2017). The effects of gamification-based teaching practices on student achievement and students' attitudes toward lessons. *The Internet and Higher Education*, 33, 86-92. doi: 10.1016/j.iheduc.2017.02.002
- Yıldız, M. (2018). *Oyunlaştırılmış blok tabanlı algoritmik düşünme etkinliklerinin öğrencilerin programlamaya yönelik tutum, katılım ve becerilerine etkisi* (Yüksek Lisans Tezi). YÖK Tez Merkezi veri tabanından erişildi (Tez No: 513250).
- Yılmaz, E. A. (2015). *Oyunlaştırma* (1. Baskı). İstanbul: Abaküs Kitap Yayın Dağıtım
- Yuen, K., Lee, S. ve Tsang, E. Y. (2011). Reasons for dropping out in distance learning. *International Journal of Continuing Education and Lifelong Learning*, 3(2), 25-41. Erişim adresi: <https://search.informit.org/doi/10.3316/informit.367512020520623>
- Yükseltürk, E. ve Altınok, S. (2017). Blok tabanlı programlama. Y. Gülbahar (Ed.), *Bilgi işlemsel düşünmeden programlamaya* (1. Baskı, pp. 241-263). Ankara: PegemA Yayıncılık
- Yükseltürk, E. ve Altınok, S., (2015). Bilişim teknolojileri öğretmen adaylarının bilgisayar programlama öğretimine yönelik görüşleri. *Amasya Üniversitesi Eğitim Fakültesi Dergisi*, 4(1), 50-65. Erişim adresi: <https://dergipark.org.tr/tr/pub/amauefd/issue/1732/21264>
- Zainuddin, Z. (2018). Students' learning performance and perceived motivation in gamified flipped-class instruction. *Computers & Education*, 126, 75-88. doi: 10.1016/j.compedu.
- Zhang, L. F. (2002). Thinking styles: Their relationships with modes of thinking and academic performance. *Educational Psychology*, 22(3), 331-348. doi: 10.1080/01443410220138557

Zichermann, G. ve Cunningham, C. (2011). *Gamification by design: Implementing game mechanics in web and mobile apps*. Sebastopol: O'Reilly Media Inc.

## EKLER

### Ek 1. Belirtke Tablosu

Konu	Kazanımlar	Soru No	Bilgi	Kavrama	Uygulama	Analiz	Sentez
Programlama	1.1 Değişken tanımlama kurallarını bilir. 1.2 Temel veri tiplerini kavrar 1.3 Değişken ve temel veri tiplerini kullanarak programlar geliştirir.	14, 21, 23, 29, 32	1		2	1	1
	1.4 Print() fonksiyonunu kullanarak programlar geliştirir. 1.5 Print() fonksiyonunun parametrelerini bilir. 1.6 Print() fonksiyonunun parametrelerini kullanarak programlar geliştirir.	6, 8, 15, 25, 36		1	1	3	
Program Kontrolü	2.1 İç içe kontrol yapılarını ifade etmeyi kavrar. 2.2 Kontrol yapılarını kullanarak programlar geliştirir.	24, 27, 35				2	1
	2.3 Tekrarlı yapıları kavrar. 2.4 Tekrarlı yapılarda kullanılan fonksiyonları kavrar. 2.5 Tekrarlı yapıları ve tekrarlı yapılarda kullanılan fonksiyonları kullanarak programlar geliştirir.	16, 22, 28, 33		2		2	
	2.6 Fonksiyonun ne olduğunu bilir. 2.7 Fonksiyon türlerini bilir. 2.8 Fonksiyonun nasıl tanımlanacağını kavrar. 2.9 Parametre almayan fonksiyonları kullandığı programlar geliştirir.	5, 12, 18, 26, 30	2			2	1
	2.10 Parametre alan fonksiyonların kullandığı programlar geliştirir. 2.11 İsimli ve sıralı argümanların kullandığı programlar geliştirir. 2.12 Varsayılan argümanların kullandığı programlar geliştirir.	2, 9, 10, 17		1	1	2	
	2.13 Değer döndüren fonksiyonların kullandığı programlar geliştirir. 2.14 Çok sayıda sıralı ve isimli argümanlı fonksiyonların kullandığı programlar geliştirir.	4, 31				2	
	3.1 Liste türünde veri tanımlamayı bilir. 3.2 Liste veri tipini ve metotlarını kullanarak programlar geliştirir.	1, 11, 19	1		2		
	3.3 Liste veri tipini ve metotlarını kullanarak programlar geliştirir. 3.4 Demet veri tipini ve metotlarını kullanarak programlar geliştirir.	7, 34			1	1	
Veri Yapıları	3.5 Küme türünde veri tanımlamayı ve metotlarını bilir. 3.6 Küme veri tipini ve metotlarını kullanarak programlar geliştirir.	3, 13, 20		1	2		

## Ek 2. Madde Analizi Sonuçları

Madde numarası	Madde ayırt edicilik indisi (r)	Madde güçlük indisi (p)	Madde varyansı (p.q)
1	.37	.82	.15
2	.40	.84	.14
3	.37	.71	.21
<b>4</b>	<b>.30</b>	<b>.73</b>	<b>.20</b>
5	.43	.80	.16
<b>6</b>	<b>.10</b>	<b>.26</b>	<b>.19</b>
7	.47	.57	.25
8	.37	.84	.14
9	.37	.47	.25
10	.43	.82	.15
<b>11</b>	<b>.33</b>	<b>.84</b>	<b>.14</b>
<b>12</b>	<b>.33</b>	<b>.80</b>	<b>.16</b>
<b>13</b>	<b>.13</b>	<b>.31</b>	<b>.21</b>
14	.47	.80	.16
15	.37	.84	.14
16	.40	.82	.15
17	.37	.54	.25
18	.40	.81	.16
19	.40	.84	.14
20	.67	.71	.21
21	.47	.81	.16
22	.43	.83	.14
23	.40	.83	.14
24	.43	.76	.18
25	.43	.85	.13
26	.47	.85	.13
27	.47	.55	.25
28	.47	.63	.23
<b>29</b>	<b>.27</b>	<b>.33</b>	<b>.22</b>
30	.50	.84	.14
31	.53	.51	.25
32	.50	.75	.19
33	.53	.81	.16
34	.43	.83	.14

## Ek 3. Akademik Başarı Testi

### AKADEMİK BAŞARI TESTİ

1. Aşağıda tanımlanan veri tiplerinden hangisi liste veri tipidir?
- liste= {"Ali", "Ahmet", 36, 3.14}
  - liste= "Python Programlama Dili"
  - liste= ["Kırmızı", "Sarı", "Turuncu"]
  - liste= set({1,2,3,4})

```
2. def fonk(a,b):  
    sonuc= (a**2) + (b**2)  
    print ("İşlem Sonucu", sonuc)
```

Yukarıda tanımlanan fonksiyon *fonk(5,6)* şeklinde çağrıldığında ekran çıktısı ne olur?

- İşlem Sonucu 11
- İşlem Sonucu 30
- İşlem Sonucu 61
- İşlem Sonucu 45

3. kume=set (["Python", "Java", "Perl"])  
kume.add ("C#")

Yukarıda verilen kod satırında aşağıdaki işlemlerden hangisi yapılmaktadır?

- Listeye eleman ekleme
- Kümeden eleman silme
- Demete eleman ekleme
- Kümeye eleman ekleme

4. Python'da fonksiyon tanımlama ile ilgili aşağıda verilen ifadelerden hangisi **yanlıştır**?

- Yerel ve global olmak üzere iki tip fonksiyon kullanılır.
- Fonksiyon tanımlamada delete ifadesi kullanılır.
- Yerel fonksiyonlar kullanıcı tarafından tanımlanır.
- Fonksiyonların tanımlanması ve çağırılması şeklinde iki aşaması vardır.

5. demet=('Python', 'ile', 'kodlama', 'öğreniyorum')

Yukarıdaki demette "Python" elemanı yerine "Java" elemanını koymamız için yazılması gereken kod satırı aşağıdakilerden hangisidir?

- demet [0]= "Java"
- demet [1]= "Java"
- Demet veri tipinde eleman değişikliği yapılamaz.
- demet [2]= "Java"

6. kullanicadi= "Python"  
print (kullanicadi)

Yukarıdaki kodların ekran çıktısı nedir?

- Print("Python")
- Python
- Kullanicadi
- kullanicadi=Python

7. def topla(sayı, kacar=1):  
 toplam=0  
 i=0  
 while (sayı>i):  
 toplam+=i  
 i+=kacar  
 print ("Toplam:", toplam)

Yukarıda tanımlanan fonksiyon *topla(4)* şeklinde çağrıldığında ekran çıktısı ne olur?

- 6
- 4
- 8
- 10

8. I. pi=3  
II. print("Silindirin hacmi=", hacim)  
III. hacim=pi\*(r\*\*2)\*h  
IV. def hacim (r,h):

Silindirin hacmini hesaplayan fonksiyonun kodları yukarıda verilmiştir. Fonksiyon kodlarının sıralanışı aşağıdaki seçeneklerin hangisinde doğru verilmiştir? (Not: Kod sıralamasında girintileri göz ardı ediniz)

- III-II-IV-I
- IV-III-II-I
- IV-I-II-III
- IV-I-III-II

9. I. akenari=int(input("a kenarını giriniz:"))  
II. print("Prizmanın Hacmi:", hacim)  
III. bkenari=int(input("b kenarını giriniz:"))  
IV. ckenari=int(input("c kenarını giriniz:"))  
V. hacim=a\*b\*c

Dikdörtgenler prizmasının hacmini hesaplayan kodlar yukarıda verilmiştir. Kodların doğru sıralanışı aşağıdaki seçeneklerin hangisinde verilmiştir?

- I, III, IV, II, V
- II, V, III, I, IV
- I, III, IV, V, II
- II, III, I, IV, V

10. `print("Python", "Programlama", "Kodlama", sep="Öğreniyorum")`

Yukarıda yazılan kodun ekran çıktısı nedir?

- A. PythonÖğreniyorumProgramlamaÖğreniyorumKodlama
- B. Python Öğreniyorum
- C. Kodlama Öğreniyorum
- D. 'KodlamaÖğreniyorum'

11. `while True:`

```
parola=input("Lütfen bir parola belirleyiniz:")
if len(parola) < 5:
    print("Parola 5 karakterden az olmamalı!")
else:
    print("Parolanız belirlendi!")
    break
```

Yukarıdaki kod bloğunda `break` komutunun kullanım amacı nedir?

- A. Parola hatalarını düzeltmek
- B. Kullanıcı parola belirlediği zaman programı sonlandırmak
- C. Yanlış parola girildiğinde tekrar parola giriş ekranına yönlendirme yapmak
- D. Programın tekrarlı olarak çalışmasını sağlamak

12. `Fonk1 ("Ahmet", 50, 60)`

`Fonk2 (N1=50, ad="Ahmet", N2=60)`

Yukarıda not ortalaması hesaplayan iki fonksiyon farklı şekillerde çağrılmıştır. Aralarındaki fark nedir?

- A. Aralarında fark yoktur.
- B. `Fonk1` fonksiyonu isimli argümana sahiptir.
- C. `Fonk2` fonksiyonu sıralı argümana sahiptir.
- D. `Fonk2` fonksiyonu isimli argümana sahiptir.

13. `def ortalama():`

```
a=8
b=16
c=12
ort=(a+b+c)/3
print("Sayıların Ortalaması:",ort)
```

Yukarıdaki fonksiyon çağrıldığında ekran çıktısı ne olur?

- A. 12.0
- B. 16.0
- C. 8.0
- D. 36.0

14. `renkler=["Kırmızı", "Mavi", "Siyah"]`

Yukarıda tanımlanan listeye "Yeşil" elemanını eklemek için hangi kod satırı kullanılır?

- A. `renkler.reverse ("Yeşil")`
- B. `renkler.append ('Yeşil')`
- C. `renkler.remove ('Yeşil')`
- D. `renkler.index ('Yeşil')`

15. `a= "değişken"`

```
b= "tanımlıyorum"
c= a+b
c
```

Yukarıdaki komutlar dikkate alındığında `c` değeri nedir?

- A. "değişken tanımlıyorum"
- B. "değişken" + "tanımlıyorum"
- C. 'değişkentanımlıyorum'
- D. 'değişken' + 'tanımlıyorum'

16. `for i in .....(5,10):`  
`print (int)`

Yukarıdaki tekrarlı yapının 5 ile 10 aralığındaki sayıları ekrana yazdırdığı düşünüldüğünde belirtilen boşluğa hangi fonksiyon gelmelidir?

- A. `range`
- B. `break`
- C. `pass`
- D. `continue`

17. I. `ad=input("Adınızı Yazınız:")`

II. `print("Adınız=",ad)`

III. `soyad=input("Soyadınızı Yazınız:")`

IV. `print("Soyadınız=",soyad)`

Kullanıcıdan ad ve soyad bilgisini sırasıyla girmesini isteyen ve ekrana yazdıran python kodların sıralaması aşağıdaki seçeneklerin hangisinde doğru verilmiştir?

- A. I, III, II, IV
- B. III, IV, II, I
- C. II, III, I, IV
- D. I, III, IV, II

18. yazilnotu=int(input("Yazılı notunuzu giriniz:"))  
 sozlunotu=int(input("Sözlü notunuzu giriniz:"))  
 odevnotu=int(input("Ödev notunuzu giriniz:"))  
 ortalama=(yazilnotu+sozlunotu+odevnotu)/3  
 .....ortalama<45:  
 print("Sınıf Tekrar")  
 .....ortalama<55:  
 print("Bütünleme sınavlarına girmelisiniz")  
 .....:  
 print("Başarılı bir şekilde sınıf geçtiniz")

Yukarıdaki programda kullanıcıdan yazılı, sözlü ve ödev notlarını girmesi istenmektedir. Kullanıcının ortalaması hesaplanarak durumu kontrol yapısı içinde değerlendirilmektedir. Kontrol yapısı içinde yer alan boşluklara sırasıyla hangi ifadelerin gelmesi gerekmektedir?

- A. Else- Elif - If
- B. Elif – Else - Else
- C. If – Elif - Elif
- D. If –Elif – Else

19. Print() fonksiyonunun parametreleri ile ilgili verilen ifadelerden hangisi **yanlıştır**?

- A. Sep parametresi karakter dizileri arasında konulacak ayracı belirlemek için kullanılır.
- B. End parametresi karakter dizilerinin sonunda ne yapılacağını belirlemek için kullanılır.
- C. File, karakter dizilerini ekrana yazdırmak için kullanılır
- D. File parametresi verilerin nereye yazılacağını belirlemek için kullanılır.

20. Kullanıcının girdiği sayının küpünü hesaplayan fonksiyon aşağıdaki seçeneklerin hangisinde doğru verilmiştir?

- A. def kup():  
 sayi=int(input("Sayıyı giriniz:"))  
 print("Sayının küpü: ",sayi\*\*3)
- B. def kup():  
 sayi=float(input("Sayıyı giriniz:"))  
 print("Sayının küpü: ",sayi\*3)
- C. def kup():  
 sayi=char(input("Sayıyı giriniz:"))  
 print("Sayının küpü: ",3\*sayi)
- D. def kup():  
 sayi=int(input("Sayıyı giriniz:"))  
 print("Sayının küpü: ",sayi<sup>3</sup>)

21. boy=int(input("Boy uzunluğunuzu giriniz:"))  
 if boy<170:  
 print("Boyunuz kısadır")  
 elif boy<180:  
 print("Boyunuz normal seviyededir")  
 else:  
 print("Boyunuz uzundur")

Yukarıdaki programda 3 kullanıcı sırasıyla 155, 178, 192 değerlerini girmiştir. Programın verdiği çıktılar sırasıyla aşağıdaki seçeneklerin hangisinde doğru olarak verilmiştir?

- A. Uzun-Uzun-Kısa
- B. Kısa-Normal-Uzun
- C. Normal-Uzun-Uzun
- D. Normal-Kısa-Kısa

22. Aşağıdakilerden hangisi döngülerde kullanılacak yapılardan **değildir**?

- A. integer
- B. range
- C. pass
- D. break

23. Dikdörtgenin alanını uzunkenar ve kısakenar değişkenlerini tanımlayarak hesaplayan kod bloğu aşağıdakilerden hangisidir? (Not=uzunkenar=20, kısakenar=5)

- A. uzunkenar="20"  
 kısakenar="5"  
 alan=uzunkenar\*kisakenar
- B. 20  
 5  
 alan=20\*5
- C. uzunkenar==20  
 kısakenar==5  
 alan==uzunkenar\*kisakenar
- D. uzunkenar=20  
 kısakenar=10  
 alan=uzunkenar\*kisakenar

24. Aşağıdaki veri tipleri eşleştirmelerinden hangisi yanlıştır?

- A. Integer-Tamsayı
- B. Float-Ondalık Sayı
- C. Integer=Ondalık Sayı
- D. String-Karakter dizisi

25. yaslar=123456789

```
for yas in yaslar:  
    print(yas)
```

Yukarıdaki kod bloğunun ekran çıktısı ile ilgili aşağıdaki ifadelerden hangisi doğrudur?

- A. 1 den 9'a kadar tüm yaşları ekrana yazar
- B. Yaşların toplamını ekrana yazar
- C. Hata mesajı verir.
- D. Yaş değerlerini dosyaya yazar

26. 

```
def fl(a,b=[]):  
    b.append(a)  
    return b  
print(fl(2,[3,4]))
```

Yukarıda belirtilen fonksiyonun ekran çıktısı nedir?

- A. [3, 4, 2]
- B. [3, 3, 4]
- C. [2, 4, 3]
- D. [2, 3, 4]

27. Bir bilgisayar oyununda öncelikle oyuncudan yaşının girilmesi istenmektedir. Eğer kullanıcının yaşı 12 yaş altında ise sistem kullanıcının oyuna erişimini engellemektedir. Yaşı 12 yaş ve üzeri ise kullanıcının oyuna erişimini sağlamaktadır.

Yukarıdaki işlemi gerçekleştiren kontrol yapısı aşağıdakilerden hangisidir.

- A. 

```
yas=int(input("Yaşınızı Giriniz:"))  
if yas<=12  
    print("Yaşınız oyun için uygun değildir.")  
if:  
    print("Yaşının oyun için uygundur. İyi Eğlenceler")
```
- B. 

```
yas=int(input("Yaşınızı Giriniz:"))  
if yas<12:  
    print("Yaşınız oyun için uygun değildir.")  
else:  
    print("Yaşınız oyun için uygundur. İyi Eğlenceler")
```
- C. 

```
yas=int(input("Yaşınızı Giriniz:"))  
if yas==12:  
    print("Yaşınız oyun için uygun değildir.")  
else:  
    print("Yaşının oyun için uygundur. İyi Eğlenceler")
```
- D. 

```
yas=int(input("Yaşınızı Giriniz:"))  
if yas>12:  
    print("Yaşınız oyun için uygun değildir.")  
else:  
    print("Yaşının oyun için uygundur. İyi Eğlenceler")
```

28. 

```
sayi=.....(input("Sayıyı Giriniz:"))  
.....("Girdiğiniz Sayının Küpü.", sayi**3)
```

Yukarıdaki program kullanıcı tarafından girilen sayının küpünü hesaplayıp ekrana yazdırmaktadır. Yazılan kodlarda boş bırakılan yerlere sırasıyla hangi komutların gelmesi gerekmektedir?

- A. int-input
- B. print-float
- C. print-int
- D. int-print

1	(a)	(b)	(c)	(d)
2	(a)	(b)	(c)	(d)
3	(a)	(b)	(c)	(d)
4	(a)	(b)	(c)	(d)
5	(a)	(b)	(c)	(d)
6	(a)	(b)	(c)	(d)
7	(a)	(b)	(c)	(d)
8	(a)	(b)	(c)	(d)
9	(a)	(b)	(c)	(d)
10	(a)	(b)	(c)	(d)
11	(a)	(b)	(c)	(d)
12	(a)	(b)	(c)	(d)
13	(a)	(b)	(c)	(d)
14	(a)	(b)	(c)	(d)
15	(a)	(b)	(c)	(d)
16	(a)	(b)	(c)	(d)
17	(a)	(b)	(c)	(d)
18	(a)	(b)	(c)	(d)
19	(a)	(b)	(c)	(d)
20	(a)	(b)	(c)	(d)
21	(a)	(b)	(c)	(d)
22	(a)	(b)	(c)	(d)
23	(a)	(b)	(c)	(d)
24	(a)	(b)	(c)	(d)
25	(a)	(b)	(c)	(d)
26	(a)	(b)	(c)	(d)
27	(a)	(b)	(c)	(d)
28	(a)	(b)	(c)	(d)



## Ek 4. Motivasyon Ölçeği Kullanım İzni

Güdülenme ve Öğrenme Stratejileri Ölçeği Kullanım İzni

Inbox x



15 Oct (2 days ago) ☆



Merhabalar Hocam;

Doç. Dr. Mübin KIYICI danışmanlığında yürüttüğüm doktora tezim kapsamında "Güdülenme ve Öğrenme Stratejileri Ölçeğinin Türkçe Formunun Geçerlilik ve Güvenilirlik Çalışması" başlıklı çalışmanızda Türkçe uyarlamasını yapmış olduğunuz "Güdülenme ve Öğrenme Stratejileri Ölçeğini" müsaadenizle kullanmak istiyorum. Saygılarımla.

Hüseyin YAŞAR  
Sakarya Üniversitesi Eğitim Bilimleri Enstitüsü  
Bilgisayar ve Öğretim Teknolojileri A.B.D. Doktora Öğrencisi



16 Oct (1 day ago) ☆



Turkish > English [Translate message](#)

[Turn off for: Turkish](#) x

Sayın Hüseyin Yaşar,

Doktora tez çalışmanız kapsamında geçerlik ve güvenilirlik çalışmasını gerçekleştirdiğimiz MSLQ Türkçe versiyonunu kullanmanızdan mutluluk duyarız. Ölçek ile ilgili gerekli olabilecek bazı bilgiler ekte sunulmuştur. İyi çalışmalar ve başarılar dileriz.

Özcan Erkan AKGÜN



## Ek 5. Motivasyon Ölçeği Örnek Maddeler

### MOTIVASYON ÖLÇEĞİ

Sevgili Öğrenciler;

Bu ölçek, kodlama eğitimine yönelik güdülenmenizi belirlemek amacıyla yapılmaktadır. Ölçekte vermiş olduğunuz cevaplar kesinlikle not vermek ve eleştirmek amacıyla kullanılmayacaktır. Ölçekte yer alan soruları dikkatlice okuyarak sizin için en uygun seçeneği, ifadenin karşısındaki kutucuğa işaretleyiniz. Ölçekten elde edilen sonuçlar bilimsel bir araştırma için kullanılacaktır.

Katkılarınız için teşekkür ederim.

**Hüseyin YAŞAR**

**Öncelikle aşağıdaki soruları cevaplayınız.**

**Cinsiyetiniz:**(  ) Erkek(  ) Kız

**Yaşınız:**(  ) 11 ve altı

(  ) 12

(  ) 13

(  ) 14

(  ) 15

(  ) 16

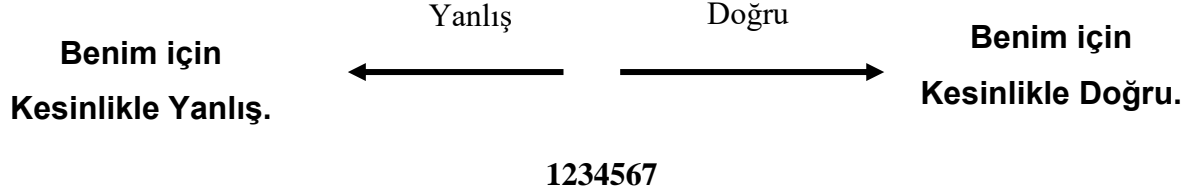
(  ) 17

(  ) 18

(  ) 19 ve üstü

**Ölçek:**

Soruları yanıtlamak için aşağıdaki ölçütleri kullanınız. Soruda geçen ifade sizin için **kesinlikle doğru ise (7)**'yi; sizinle ilgili **kesinlikle yanlışsa (1)**'i işaretleyin. Eğer ifadenin size göre doğruluğu bunlardan farklı ise sizin için en uygun düzeyi gösteren (1)'le (7) arasındaki rakamı işaretleyin.



1.	Bu derste beni zorlayan, aynı zamanda da geliştiren konuları tercih ederim; böylece yeni şeyler öğrenebilirim.	(1) (2) (3) (4) (5) (6) (7)
2.		
3.		
4.	Bu dersten yüksek bir not alacağıma inanıyorum.	(1) (2) (3) (4) (5) (6) (7)
5.		
6.		
7.	Sınav sırasında, başarısız olursam bunun getireceği sonuçları düşünürüm.	(1) (2) (3) (4) (5) (6) (7)
8.		
9.		
10.		
11.	Sınavdayken kendimi huzursuz ve sıkıntılı hissedirim.	(1) (2) (3) (4) (5) (6) (7)
12.		
13.		
14.		
15.		
16.	Bu derste işlenen konular hoşuma gidiyor.	(1) (2) (3) (4) (5) (6) (7)
17.		
18.		
19.		
20.	Dersin zorluğunu, öğretmenini ve becerilerimi dikkate aldığımda, bu derste başarılı olacağımı düşünüyorum.	(1) (2) (3) (4) (5) (6) (7)

## Ek 6. Tutum Ölçeği Kullanım İzni

DoktoraTezMaili x

27 Dec 2017, 19:59 ☆ ↩

Merhabalar Sayın Hocam,

Doç. Dr. Mübin KIYICI danışmanlığında yürüttüğüm doktora tezim kapsamında "Programlama Dillerinin Çevrimiçi Öğretimi: Öğrenenlerin Tutumlarının, Memnuniyetlerinin, ve Akademik Başarılarının İncelenmesi" başlıklı doktora tez çalışmamız kapsamında uyarlayıp Türkiye'ye çevirerek kullandığımız "Programlama Dillerine Yönelik Tutum Ölçeğini" müsaadenizle kullanmak istiyorum.

Kolaylıklar dilerim.

Saygılarımla.

Hüseyin YAŞAR  
Doktora Öğrencisi  
Sakarya Üniversitesi  
Eğitim Bilimleri Enstitüsü  
Bilgisayar ve Öğretim Teknolojileri Eğitimi A.B.D.

27 Dec 2017, 20:15 ☆ ↩

Türkçe → English → [Translate message](#) [Turn off for: Turkish x](#)

Merhaba tabiki kullanabilirsiniz. İyi çalışmalar dilerim.

.....  
Dr Gürhan DURAK  
Balıkesir University Computer Education & Instructional Technologies.....

## Ek 7. Tutum Ölçeği Örnek Maddeler

### PROGRAMLAMA DİLLERİNE YÖNELİK TUTUM ÖLÇEĞİ

Sevgili Öğrenciler;

Aşağıda sizlerin programlama dillerine yönelik tutumlarınızı belirlemek amacıyla bir ölçek sunulmuştur. Ölçekte yer alan maddeleri dikkatlice inceleyerek sizin için en uygun seçeneği işaretleyiniz. Ölçekte vermiş olduğunuz cevaplar kesinlikle not ve eleştirmek amacıyla kullanılmayacak ayrıca elde edilen sonuçlar bilimsel bir araştırma için kullanılacaktır.

Katkılarınız için teşekkür ederim.

**Hüseyin YAŞAR**

**Ölçek:**

Kesinlikle Katılmıyorum	Katılmıyorum	Kararsızım	Katılıyorum	Kesinlikle Katılıyorum
1	2	3	5	5

S.N		Kesinlikle Katılmıyorum	Katılmıyorum	Kararsızım	Katılıyorum	Kesinlikle Katılıyorum
1.	Programlama çok yararlı ve gerekli bir konudur	1	2	3	4	5
2.						
3.						
4.						
5.						
6.						
7.	Okul dışında birçok yolla programlamayı kullanabileceğimi düşünüyorum	1	2	3	4	5
8.	İleri düzey programlama çalışmak yararlıdır	1	2	3	4	5
9.						

10.						
11.	Programlama problemlerini çözerken memnuniyet duyarım	1	2	3	4	5
12.						
13.						
14.						
15.						
16.						
17.						
18.						
19.	Programlama derslerinde sorulara rahatlıkla cevap verebilirim	1	2	3	4	5
20.						
21.						
22.						
23.						
24.						
25.						
26.	Programlama derslerinde kendimi sürekli gerilim altında hissedirim	1	2	3	4	5
27.						
28.						
29.	Program yazarken kendimden emin değilimdir	1	2	3	4	5
30.						
31.						
32.						
33.						
34.						
35.	Program yazmada iyi olduğuma inanıyorum	1	2	3	4	5
36.						
37.						
38.						
39.						
40.	Programlama öğretmekten uzak dururum	1	2	3	4	5

## Ek 8. OÇÖO Ders Planları

Pilot Uygulama 1. Hafta Ders Planı	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	Python Giriş
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Çevrimiçi Öğrenme
<b>Araç gereçler</b>	Oyunlaştırılmış Çevrimiçi Öğrenme Ortamı
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	<ol style="list-style-type: none"><li>1. Python programlama dilini bilir.</li><li>2. Python programlama dilinin avantajlarını bilir.</li><li>3. Python kurulumunu yapabilir.</li><li>4. Python kod yazma ortamlarını bilir.</li></ol>
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"><li>• Öğrencilerin işbirliği ortamında çalışabilmeleri için işbirliği forumu hazırlandı.</li><li>• Oyunlaştırma bileşenleriyle ilgili hazırlıklar yapıldı. Oyunlaştırma yönergesi sisteme yüklendi.</li><li>• Haftanın ödevi hazırlanarak sisteme yüklendi.</li><li>• Ders kazanımları çevrimiçi öğrenme ortamına yüklendi.</li><li>• Ders öncesinde kaynak taraması yapılarak ders notları hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li><li>• Python giriş konusuyla ilgili ders videosu hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li><li>• Öğrencilerin eğlenerek öğrenmelerine olanak tanıyan eğlenelim etkinliği sistemde hazır hale getirildi.</li><li>• Ölçme ve değerlendirme amacıyla kullanılacak 6 sorudan oluşan kodlama deneyimi etkinliği hazırlanarak sisteme yüklendi</li></ul>

## BÖLÜM IV

<b>Derse İşlenişi:</b>	<ol style="list-style-type: none"><li>1. Öğrenciler öncelikli olarak haftanın ödevini yaparak derse başlayacaktır. Haftanın ödevi yapıp gönderildikten sonra ders izlencesi aktif hale gelecektir.</li><li>2. Öğrenciler ders izlencesinde yer alan ders kazanımlarını okuyarak derse başlayacaktır. Kazanımlar okunduktan sonra python giriş ders notları öğrenciler için sistemde aktif hale gelecektir.</li><li>3. Python giriş konusuyla ilgili öğrencilere sunulan ders notları öğrenciler tarafından ayrıntılı olarak incelenecektir. Öğrenciler ders notlarını inceledikten sonra python giriş ders videosu öğrencilerin kullanımına açılacaktır.</li><li>4. Ders anlatım videosunda python programlama dilinin ne olduğu, avantajları, kurulumu, diğer kod yazma ortamları, aritmetiksel operatörler ve kaçış ifadeleri konuları öğrencilere anlatılacaktır. Ayrıca konuyla alakalı örnek kodlamalar yapılacaktır. Ders videosu tamamlandıktan sonra eğlenelim etkinliği öğrencilerin kullanımına açılacaktır.</li><li>5. <b>Sürükle Bırak</b> isimli eğlenelim etkinliği öğrencilerin ders esnasında eğlenerek konuyu pekiştirmelerine olanak sağlayan etkinliklerden oluşmaktadır. Bu etkinliği tamamlayan öğrencilere, kodlama deneyimi etkinliği aktif hale gelecektir.</li><li>6. Öğrenciler, haftanın konusuyla ilgili işbirliği forumunda bilgi alışverişinde bulunabileceklerdir. Ayrıca öğrenme ortamıyla ilgili karşılaştıkları sorunlarla ilgili sıkça sorulan sorular kısmından bilgi alabileceklerdir.</li><li>7. Öğrenciler ayrıca oyunlaştırma süreciyle ilgili sisteme yüklenen oyunlaştırma yönergesinden bilgi alabileceklerdir.</li></ol>
------------------------	--

## BÖLÜM V

<b>Ölçme ve Değerlendirme:</b>	Python giriş konusuyla ilgili 6 sorudan oluşan kodlama deneyimi etkinliği öğrencilerin şifreyi girmesiyle açılacaktır. Haftanın konusuyla alakalı sorular öğrenciler tarafından cevaplanacaktır.
--------------------------------	--



<b>Pilot Uygulama 2. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	Karakter Dizileri, Aritmetiksel Operatörler, Kaçış İfadeleri
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Çevrimiçi Öğrenme
<b>Araç gereçler</b>	Oyunlaştırılmış Çevrimiçi Öğrenme Ortamı
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	<ol style="list-style-type: none"> <li>1. Karakter dizilerinin ne olduğunu bilir.</li> <li>2. Karakter dizileri üzerinde birleştirme ve çoğaltma işlemlerinin nasıl yapılacağını kavrar.</li> <li>3. Type() fonksiyonunu kullanabilir.</li> <li>4. Aritmetik işlem operatörlerini kullanabilir.</li> <li>5. Kaçış ifadelerini kullanabilir.</li> </ol>
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Öğrencilerin işbirliği ortamında çalışabilmeleri için işbirliği forumu hazırlandı.</li> <li>• Oyunlaştırma bileşenleriyle ilgili hazırlıklar yapıldı. Oyunlaştırma yönergesi sisteme yüklendi.</li> <li>• Haftanın ödevi hazırlanarak sisteme yüklendi.</li> <li>• Ders kazanımları çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Ders öncesinde kaynak taraması yapılarak ders notları hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Karakter dizileri konusuyla ilgili ders videosu hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Öğrencilerin eğlenerek öğrenmelerine olanak tanıyan eğlenelim etkinliği sistemde hazır hale getirildi.</li> <li>• Ölçme ve değerlendirme amacıyla kullanılacak 4 sorudan oluşan kodlama deneyimi etkinliği hazırlandı. Hazırlanan etkinlik</li> </ul>

	Moodle eklentilerinden olan Code Runner kullanılarak öğrencilerin kod yazmalarına olanak sağlayacak hale getirildi.
<b>BÖLÜM IV</b>	
<b>Derse İşlenişi:</b>	<p>1. Öğrenciler öncelikli olarak haftanın ödevini yaparak derse başlayacaktır. Haftanın ödevi yapıp gönderildikten sonra ders izlencesi aktif hale gelecektir.</p> <p>2. Öğrenciler ders izlencesinde yer alan ders kazanımlarını okuyarak derse başlayacaktır. Kazanımlar okunduktan sonra karakter dizileri ders notları öğrenciler için sistemde aktif hale gelecektir.</p> <p>3. Karakter dizileriyle ilgili sunulan ders notları öğrenciler tarafından ayrıntılı olarak incelenecektir. Öğrenciler ders notlarını inceledikten sonra ders anlatımı videosu öğrencilerin kullanımına açılacaktır.</p> <p>4. Ders anlatım videosunda karakter dizileri konusu örnek uygulamalar sunularak anlatılmıştır. Ders videosunu tamamlandıktan sonra ders konusuyla ilgili hazırlanan eğlenceli etkinlik öğrencilere açılacaktır.</p> <p>5. <b>Resmi Yorumla</b> isimli eğlenceli etkinlik öğrencilerin ders esnasında eğlenerek konuyu pekiştirmelerine olanak sağlayan etkinliklerden oluşmaktadır. Bu etkinliği tamamlayan öğrencilere kodlama deneyimi etkinliği aktif hale gelecektir.</p> <p>6. Öğrenciler haftanın konusuyla ilgili işbirliği forumunda bilgi alışverişinde bulunabileceklerdir. Ayrıca öğrenme ortamıyla ilgili karşılaştıkları sorunlarla ilgili sıkça sorulan sorular kısmından bilgi alabileceklerdir.</p> <p>7. Öğrenciler ayrıca oyunlaştırma süreciyle ilgili sisteme yüklenen oyunlaştırma yönergesinden bilgi alabileceklerdir.</p>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Karakter dizileri konusuyla ilgili 4 sorudan oluşan kodlama deneyimi etkinliği öğrencilerin şifreyi girmesiyle açılacaktır. Haftanın konusuyla alakalı sorular öğrenciler tarafından

	cevaplanacaktır. Öğrenciler Moodle eklentisi olan Code Runner ortamında yer alan soruları kodlanacaktır.
--	--

<b>Uygulama 1. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	Değişkenler ve Sabitler
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Çevrimiçi Öğrenme
<b>Araç gereçler</b>	Oyunlaştırılmış Çevrimiçi Öğrenme Ortamı
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	<ol style="list-style-type: none"> <li>1. Değişken tanımlama kurallarını bilir.</li> <li>2. Temel veri tiplerini kavrar</li> <li>3. Değişken ve temel veri tiplerini kullanarak programlar geliştirir.</li> </ol>
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Öğrencilerin işbirliği ortamında çalışabilmeleri için işbirliği forumu hazırlandı.</li> <li>• Oyunlaştırma bileşenleriyle ilgili hazırlıklar yapıldı. Oyunlaştırma yönergesi sisteme yüklendi.</li> <li>• Haftanın ödevi hazırlanarak sisteme yüklendi.</li> <li>• Ders kazanımları çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Ders öncesinde kaynak taraması yapılarak ders notları hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Değişkenler konusu ders videosu hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Öğrencilerin eğlenerek öğrenmelerine olanak tanıyan eğlenelim etkinliği sistemde hazır hale getirildi.</li> <li>• Ölçme ve değerlendirme amacıyla kullanılacak 4 sorudan oluşan kodlama deneyimi etkinliği hazırlandı. Hazırlanan etkinlik</li> </ul>

	Moodle eklentilerinden olan Code Runner kullanılarak sınav haline dönüştürüldü.
<b>BÖLÜM IV</b>	
<b>Derse İşlenişi:</b>	<ol style="list-style-type: none"> <li>1. Öğrenciler öncelikli olarak haftanın ödevini yaparak derse başlayacaktır. Haftanın ödevi yapıp gönderildikten sonra ders izlencesi aktif hale gelecektir.</li> <li>2. Öğrenciler ders izlencesinde yer alan ders kazanımlarını okuyarak derse başlayacaktır. Kazanımlar okunduktan sonra ders notları öğrenciler için sistemde aktif hale gelecektir.</li> <li>3. Değişkenlerle ilgili öğrencilere sunulan ders notları öğrenciler tarafından ayrıntılı olarak incelenecektir. Öğrenciler ders notlarını inceledikten sonra ders anlatımı videosu öğrencilerin kullanımına açılacaktır.</li> <li>4. Ders anlatım videosunda değişkenler ve temel veri tipleri konuları öğrencilere anlatılacaktır. Ayrıca değişken tanımlama ve veri tipleriyle ilgili örnek uygulamalar yapılacaktır. Ders videosunu tamamlandıktan sonra ders konusuyla ilgili hazırlanan eğlenelim etkinliği öğrencilere açılacaktır.</li> <li>5. <b>Hafıza Egzersizi</b> isimli eğlenelim etkinliği öğrencilerin ders esnasında eğlenerek konuyu pekiştirmelerine olanak sağlayan etkinliklerden oluşmaktadır. Bu etkinliği tamamlayan öğrencilere kodlama deneyimi etkinliği aktif hale gelecektir.</li> <li>6. Öğrenciler haftanın konusuyla ilgili işbirliği forumunda bilgi alış verişinde bulunabileceklerdir. Ayrıca öğrenme ortamıyla ilgili karşılaştıkları sorunlarla ilgili sıkça sorulan sorular kısmından bilgi alabileceklerdir.</li> <li>7. Öğrenciler ayrıca oyunlaştırma süreciyle ilgili sisteme yüklenen oyunlaştırma yönergesinden bilgi alabileceklerdir.</li> </ol>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Değişkenler konusuyla ilgili 4 sorudan oluşan kodlama deneyimi etkinliği öğrencilerin şifreyi girmesiyle açılacaktır. Haftanın konusuyla alakalı sorular öğrenciler tarafından tamamlanacaktır.

	Öğrenciler Moodle eklentisi olan Code Runner ortamında yer alan soruları kodlanacaktır.
--	---

<b>Uygulama 2. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	1. Print () fonksiyonunun kullanımı 2. Print () fonksiyonunun parametreleri
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Çevrimiçi Öğrenme
<b>Araç gereçler</b>	Oyunlaştırılmış Çevrimiçi Öğrenme Ortamı
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	1. Print() fonksiyonunu kullanarak programlar geliştirir. 2. Print() fonksiyonunun parametrelerini bilir. 3. Print() fonksiyonunun parametrelerini kullanarak programlar geliştirir.
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Öğrencilerin işbirliği ortamında çalışabilmeleri için işbirliği forumu hazırlandı.</li> <li>• Oyunlaştırma bileşenleriyle ilgili hazırlıklar yapıldı. Oyunlaştırma yönergesi sisteme yüklendi.</li> <li>• Haftanın ödevi hazırlanarak sisteme yüklendi.</li> <li>• Ders kazanımları çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Print() fonksiyonu ve parametreleriyle ilgili kaynak taraması yapılarak ders notları hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Print() fonksiyonu ve parametreleriyle ilgili ders videosu hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Öğrencilerin eğlenerek öğrenmelerine olanak tanıyan eğlenceli etkinlikleri sistemde hazır hale getirildi.</li> </ul>

	<ul style="list-style-type: none"> <li>• Ölçme ve değerlendirme amacıyla kullanılacak 4 sorudan oluşan kodlama deneyimi etkinliği hazırlandı. Hazırlanan etkinlik Moodle eklentilerinden olan Code Runner kullanılarak sınav haline dönüştürüldü.</li> </ul>
<b>BÖLÜM IV</b>	
<b>Derse İşlenişi:</b>	<ol style="list-style-type: none"> <li>1.Öğrenciler öncelikli olarak haftanın ödevini yaparak derse başlayacaktır. Haftanın ödevi yapılıp gönderildikten sonra ders izlencesi aktif hale gelecektir.</li> <li>2.Öğrenciler ders izlencesinde yer alan ders kazanımlarını okuyarak derse başlayacaktır. Kazanımlar okunduktan sonra ders notları öğrenciler için sistemde aktif hale gelecektir.</li> <li>3.Print() fonksiyonu ve parametreleriyle ilgili sunulan ders notları öğrenciler tarafından ayrıntılı olarak incelenecektir. Öğrenciler ders notlarını inceledikten sonra ders anlatımı videosu öğrencilerin kullanımına açılacaktır.</li> <li>4.Print() fonksiyonu ve parametreleri ders videosu öğrenciler tarafından seyredilecektir. Örnek uygulamalar öğrenciler tarafından takip edilecektir. . Ders videosunu tamandıktan sonra ders konusuyla ilgili hazırlanan eğlenelim etkinliği öğrencilere açılacaktır.</li> <li>5. <b>Bilgi Yarışması</b> isimli eğlenelim etkinliği öğrencilerin ders esnasında eğlenerek konuyu pekiştirmelerine olanak sağlayan etkinliklerden oluşmaktadır. Bu etkinliği tamamlayan öğrencilere kodlama deneyimi etkinliği aktif hale gelecektir.</li> <li>6.Öğrenciler haftanın konusuyla ilgili işbirliği forumunda bilgi alış verişinde bulunabileceklerdir. Ayrıca öğrenme ortamıyla ilgili karşılaştıkları sorunlarla ilgili sıkça sorulan sorular kısmından bilgi alabileceklerdir.</li> <li>7. Öğrenciler ayrıca oyunlaştırma süreciyle ilgili sisteme yüklenen oyunlaştırma yönergesinden bilgi alabileceklerdir.</li> </ol>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Print() fonksiyonu ve parametreleri konusuyla ilgili 4 sorudan oluşan kodlama deneyimi etkinliği uygulanacaktır. Moodle

	eklentisi olan Code Runner ortamında sunulan sorular öğrenciler tarafından kodlanacaktır.
--	---

<b>Uygulama 3. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	Kontrol Yapıları 1. If kontrol yapısı 2. If-Elif kontrol yapısı 3. If-Else kontrol yapısı 4. If-Elif-Else kontrol yapısı
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Çevrimiçi Öğrenme
<b>Araç gereçler</b>	Oyunlaştırılmış Çevrimiçi Öğrenme Ortamı
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	1. İç içe kontrol yapılarını ifade etmeyi kavrar. 2. Kontrol yapılarını kullanarak programlar geliştirir.
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Öğrencilerin işbirliği ortamında çalışabilmeleri için işbirliği forumu hazırlandı.</li> <li>• Oyunlaştırma bileşenleriyle ilgili hazırlıklar yapıldı.</li> <li>• Haftanın ödevi hazırlanarak sisteme yüklendi.</li> <li>• Kontrol yapılarıyla ilgili ders kazanımları çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Kontrol yapılarıyla ilgili kaynak taraması yapılarak ders notları hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Kontrol yapılarıyla ilgili ders videosu hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Öğrencilerin eğlenerek öğrenmelerine olanak tanıyan eğlenceli etkinlikleri sistemde hazır hale getirildi.</li> </ul>

	<ul style="list-style-type: none"> <li>• Ölçme ve değerlendirme amacıyla kullanılacak 3 sorudan oluşan kodlama deneyimi etkinliği hazırlandı. Hazırlanan etkinlik Moodle eklentilerinden olan Code Runner kullanılarak sınav haline dönüştürüldü.</li> </ul>
<b>BÖLÜM IV</b>	
<b>Derse İşlenişi:</b>	<ol style="list-style-type: none"> <li>1. Öğrenciler kontrol yapıları dersine başlamadan print() fonksiyonu ve parametreleriyle ilgili verilen ödevi tamamlayacaktır. Ödev tamamlandıktan sonra kontrol yapıları kazanımları aktif hale gelecektir.</li> <li>2. Kontrol yapılarla ilgili ders kazanımları okunacaktır. Kazanımlar okunduktan sonra ders notları öğrenciler için sistemde aktif hale gelecektir.</li> <li>3. Kontrol yapılarla ilgili sunulan ders notları öğrenciler tarafından ayrıntılı olarak incelenecektir. Öğrenciler ders notlarını inceledikten sonra ders anlatımı videosu öğrencilerin kullanımına açılacaktır.</li> <li>4. Kontrol yapıları ders videosu öğrenciler tarafından seyredilecektir. Örnek uygulamalar öğrenciler tarafından takip edilecektir. Ders videosunu tamamlandıktan sonra ders konusuyla ilgili hazırlanan eğlenelim etkinliği öğrencilere açılacaktır.</li> <li>5. <b>Kod Bulmaca</b> isimli eğlenelim etkinliği öğrencilerin ders esnasında eğlenerek konuyu pekiştirmelerine olanak sağlayan etkinliklerden oluşmaktadır. Bu etkinliği tamamlayan öğrencilere kodlama deneyimi etkinliği aktif hale gelecektir. Kodlama deneyimi etkinliğine videoya yerleştirilen şifre ile erişilecektir.</li> <li>6. Öğrenciler haftanın konusuyla ilgili işbirliği forumunda bilgi alış verişinde bulunabileceklerdir. Ayrıca öğrenme ortamında karşılaştıkları sorunlarla ilgili sıkça sorulan sorular kısmından bilgi alabileceklerdir.</li> <li>7. Öğrenciler ayrıca oyunlaştırma süreciyle ilgili sisteme yüklenen oyunlaştırma yönergesinden bilgi alabileceklerdir.</li> </ol>



<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Kontrol yapıları konusuyla ilgili 3 sorudan oluşan kodlama deneyimi etkinliği uygulanacaktır. Öğrenciler Moodle eklentisi olan Code Runner ortamında sunulan sorular öğrenciler tarafından kodlanacaktır.

<b>Uygulama 4. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	Tekrarlı Yapılar (Döngüler) 1. Tekrarlı yapıların kullanım amacı 2. For tekrarlı yapısının kullanımı 3. While tekrarlı yapısının kullanımı 4. Tekrarlı yapılarda kullanılan yapılar (range, pass, break, continue)
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Çevrimiçi Öğrenme
<b>Araç gereçler</b>	Oyunlaştırılmış Çevrimiçi Öğrenme Ortamı
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	1. Tekrarlı yapıları kavrar. 2. Tekrarlı yapılarda kullanılan fonksiyonları kavrar. 3. Tekrarlı yapıları ve tekrarlı yapılarda kullanılan fonksiyonları kullanarak programlar geliştirir.
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Öğrencilerin işbirliği ortamında çalışabilmeleri için işbirliği forumu hazırlandı.</li> <li>• Oyunlaştırma bileşenleriyle ilgili hazırlıklar yapıldı.</li> <li>• Haftanın ödevi hazırlanarak sisteme yüklendi.</li> <li>• Tekrarlı yapılarla ilgili ders kazanımları çevrimiçi öğrenme ortamına yüklendi.</li> </ul>

	<ul style="list-style-type: none"> <li>• Tekrarlı yapılarla ilgili kaynak taraması yapılarak ders notları hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Tekrarlı yapılarla ilgili ders videosu hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Öğrencilerin eğlenerek öğrenmelerine olanak tanıyan eğlenelim etkinliği sistemde hazır hale getirildi.</li> <li>• Ölçme ve değerlendirme amacıyla kullanılacak 3 sorudan oluşan kodlama deneyimi etkinliği hazırlandı. Hazırlanan etkinlik Moodle eklentilerinden olan Code Runner kullanılarak sınav haline dönüştürüldü.</li> </ul>
<b>BÖLÜM IV</b>	
<b>Derse İşlenişi:</b>	<ol style="list-style-type: none"> <li>1. Öğrenciler tekrarlı yapılar dersine başlamadan kontrol yapılarıyla ilgili verilen ödevi tamamlayacaktır. Ödev tamamlandıktan sonra tekrarlı yapılar kazanımları aktif hale gelecektir.</li> <li>2. Tekrarlı yapılarla ilgili ders kazanımları okunacaktır. Kazanımlar okunduktan sonra ders notları öğrenciler için sistemde aktif hale gelecektir.</li> <li>3. Tekrarlı yapılarla ilgili sunulan ders notları öğrenciler tarafından ayrıntılı olarak incelenecektir. Öğrenciler ders notlarını inceledikten sonra ders anlatımı videosu öğrencilerin kullanımına açılacaktır.</li> <li>4. Tekrarlı yapılar ders videosu öğrenciler tarafından seyredilecektir. Örnek uygulamalar öğrenciler tarafından takip edilecektir. Ders videosunu tamamlandıktan sonra ders konusuyla ilgili hazırlanan eğlenelim etkinliği öğrencilere açılacaktır.</li> <li>5. <b>Kod Asmaca</b> isimli eğlenelim etkinliği öğrencilerin ders esnasında eğlenerek konuyu pekiştirmelerine olanak sağlayan etkinliklerden oluşmaktadır. Bu etkinliği tamamlayan öğrencilere kodlama deneyimi etkinliği aktif hale gelecektir. Kodlama deneyimi etkinliğine videoya yerleştirilen şifre ile erişilecektir.</li> <li>6. Öğrenciler haftanın konusuyla ilgili işbirliği forumunda bilgi alış verişinde bulunabileceklerdir. Ayrıca öğrenme ortamıyla ilgili</li> </ol>

	karşılaştıkları sorunlarla ilgili sıkça sorulan sorular kısmından bilgi alabileceklerdir. 7. Oyunlaştırma bileşenleriyle ilgili sisteme yüklenen oyunlaştırma yönergesinden bilgi alabileceklerdir.
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Tekrarlı yapılar konusuyla ilgili 3 sorudan oluşan kodlama deneyimi etkinliği uygulanacaktır. Öğrenciler Moodle eklentisi olan Code Runner ortamında sunulan sorular öğrenciler tarafından kodlanacaktır.

<b>Uygulama 5. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	Fonksiyonlar 1. Fonksiyon kavramı ve türleri 2. Fonksiyon tanımlama 3. Parametre almayan fonksiyonların kullanımı
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Çevrimiçi Öğrenme
<b>Araç gereçler</b>	Oyunlaştırılmış Çevrimiçi Öğrenme Ortamı
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	1. Fonksiyonun ne olduğunu bilir. 2. Fonksiyon türlerini bilir. 3. Fonksiyonun nasıl tanımlanacağını kavrar. 4. Parametre almayan fonksiyonları kullanıldığı programlar geliştirir.
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Öğrencilerin işbirliği ortamında çalışabilmeleri için işbirliği forumu hazırlandı.</li> <li>• Oyunlaştırma bileşenleriyle ilgili hazırlıklar yapıldı.</li> </ul>

	<ul style="list-style-type: none"> <li>• Bir önceki haftanın konusuyla ilgili hazırlanan ödev sisteme yüklendi.</li> <li>• Fonksiyonlarla (parametre almayan) ilgili ders kazanımları çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Fonksiyonlarla (parametre almayan) ilgili kaynak taraması yapılarak ders notları hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Fonksiyonlarla (parametre almayan) ilgili ders videosu hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Öğrencilerin eğlenerek öğrenmelerine olanak tanıyan eğlenceli etkinlik sistemi hazır hale getirildi.</li> <li>• Ölçme ve değerlendirme amacıyla kullanılacak 3 sorudan oluşan kodlama deneyimi etkinliği hazırlandı. Hazırlanan etkinlik Moodle eklentilerinden olan Code Runner kullanılarak sınav haline dönüştürüldü.</li> </ul>
<b>BÖLÜM IV</b>	
<b>Derse İşlenişi:</b>	<ol style="list-style-type: none"> <li>1. Öğrenciler fonksiyonlar (parametre almayan) dersine başlamadan tekrarlı yapılarla ilgili verilen ödevi tamamlayacaktır. Ödev tamamlandıktan sonra fonksiyonlar (parametre almayan) kazanımları aktif hale gelecektir.</li> <li>2. Fonksiyonlarla (parametre almayan) ilgili ders kazanımları okunacaktır. Kazanımlar okunduktan sonra ders notları öğrenciler için sistemde aktif hale gelecektir.</li> <li>3. Fonksiyonlarla (parametre almayan) ilgili sunulan ders notları öğrenciler tarafından ayrıntılı olarak incelenecektir. Öğrenciler ders notlarını inceledikten sonra ders anlatımı videosu öğrencilerin kullanımına açılacaktır.</li> <li>4. Fonksiyonlar (parametre almayan) ders videosu öğrenciler tarafından seyredilecektir. Örnek uygulamalar öğrenciler tarafından takip edilecektir. Ders videosu tamamlandıktan sonra ders konusuyla ilgili hazırlanan eğlenceli etkinlik öğrencilere açılacaktır.</li> </ol>

	<p><b>5. Şifreli Bulmaca</b> isimli eğlenelim etkinliği öğrencilerin ders esnasında eğlenerek konuyu pekiştirmelerine olanak sağlayan etkinliklerden oluşmaktadır. Bu etkinliği tamamlayan öğrencilere kodlama deneyimi etkinliği aktif hale gelecektir. Kodlama deneyimi etkinliğine videoya yerleştirilen şifre ile erişilecektir.</p> <p><b>6.</b> Öğrenciler haftanın konusuyla ilgili işbirliği forumunda bilgi alış verişinde bulunabileceklerdir. Ayrıca öğrenme ortamında karşılaştıkları sorunlarla ilgili sıkça sorulan sorular kısmından bilgi alabileceklerdir.</p> <p><b>7.</b> Öğrenciler ayrıca oyunlaştırma süreciyle ilgili sisteme yüklenen oyunlaştırma yönergesinden bilgi alabileceklerdir.</p>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Fonksiyonlar (parametre almayan) konusuyla ilgili 3 sorudan oluşan kodlama deneyimi etkinliği uygulanacaktır. Öğrenciler Moodle eklentisi olan Code Runner ortamında sunulan sorular öğrenciler tarafından kodlanacaktır.

<b>Uygulama 6. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	Fonksiyonlar <b>1.</b> Parametre alan fonksiyonlar <b>2.</b> İsimli ve sıralı argümanlar <b>3.</b> Varsayılan değerli argümanlar
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Çevrimiçi Öğrenme
<b>Araç gereçler</b>	Oyunlaştırılmış Çevrimiçi Öğrenme Ortamı
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	<b>1.</b> Parametre alan fonksiyonların kullanıldığı programlar geliştirir. <b>2.</b> İsimli ve sıralı argümanların kullanıldığı programlar geliştirir.

	3. Varsayılan argümanların kullanıldığı programlar geliştirir.
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Öğrencilerin işbirliği ortamında çalışabilmeleri için işbirliği forumu hazırlandı.</li> <li>• Oyunlaştırma bileşenleriyle ilgili hazırlıklar yapıldı.</li> <li>• Haftanın ödevi hazırlanarak sisteme yüklendi.</li> <li>• Fonksiyonlarla (parametre alan) ilgili ders kazanımları çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Fonksiyonlarla (parametre alan) ilgili kaynak taraması yapılarak ders notları hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Fonksiyonlar(parametre alan) ilgili ders videosu hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Öğrencilerin eğlenerek öğrenmelerine olanak tanıyan eğlenceli etkinlikleri sistemde hazır hale getirildi.</li> <li>• Ölçme ve değerlendirme amacıyla kullanılacak 3 sorudan oluşan kodlama deneyimi etkinliği hazırlandı. Hazırlanan etkinlik Moodle eklentilerinden olan Code Runner kullanılarak sınav haline dönüştürüldü.</li> </ul>
<b>BÖLÜM IV</b>	
<b>Derse İşlenişi:</b>	<ol style="list-style-type: none"> <li>1. Öğrenciler fonksiyonlar (parametre alan) dersine başlamadan fonksiyonlarla (parametre almayan) ilgili verilen ödevi tamamlayacaktır. Ödev tamamlandıktan sonra fonksiyonlar (parametre alan) kazanımları aktif hale gelecektir.</li> <li>2. Fonksiyonlarla (parametre alan) ilgili ders kazanımları okunacaktır. Kazanımlar okunduktan sonra ders notları öğrenciler için sistemde aktif hale gelecektir.</li> <li>3. Fonksiyonlarla (parametre alan) ilgili sunulan ders notları öğrenciler tarafından ayrıntılı olarak incelenecektir. Öğrenciler ders notlarını inceledikten sonra ders anlatımı videosu öğrencilerin kullanımına açılacaktır.</li> <li>4. Fonksiyonlar (parametre alan) ders videosu öğrenciler tarafından seyredilecektir. Örnek uygulamalar öğrenciler tarafından takip edilecektir. Ders videosunu tamamlandıktan</li> </ol>

	<p>sonra ders konusuyla ilgili hazırlanan eğlenceli etkinlik öğrencilere açılacaktır.</p> <p><b>5. Milyoner</b> isimli eğlenceli etkinlik öğrencilerin ders esnasında eğlenmelerine olanak sağlayan etkinliklerden oluşmaktadır. Bu etkinliği tamamlayan öğrencilere kodlama deneyimi etkinliği aktif hale gelecektir. Kodlama deneyimi etkinliğine videoya yerleştirilen şifre ile erişilecektir.</p> <p><b>6.</b> Öğrenciler haftanın konusuyla ilgili işbirliği forumunda bilgi alışverişinde bulunabileceklerdir. Ayrıca öğrenme ortamında karşılaştıkları sorunlarla ilgili sıkça sorulan sorular kısmından bilgi alabileceklerdir.</p> <p><b>7.</b> Öğrenciler ayrıca oyunlaştırma süreciyle ilgili sisteme yüklenen oyunlaştırma yönergesinden bilgi alabileceklerdir.</p>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Fonksiyonlar (parametre alan) konusuyla ilgili 3 sorudan oluşan kodlama deneyimi etkinliği uygulanacaktır. Moodle eklentisi olan Code Runner ortamında sunulan sorular öğrenciler tarafından kodlanacaktır

<b>Uygulama 7. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	<p>Fonksiyonlar</p> <ol style="list-style-type: none"> <li>1. Değer döndüren fonksiyonlar <ol style="list-style-type: none"> <li>a. Global ifadesi</li> <li>b. Return ifadesi</li> <li>c. Pass ifadesi</li> </ol> </li> <li>2. İstenen sayıda sıralı argümanlar</li> <li>3. İstenen sayıda isimli argümanlar</li> </ol>
<b>Süre</b>	2 Ders Saati

<b>Öğretim Yöntem ve Teknikleri</b>	Çevrimiçi Öğrenme
<b>Araç gereçler</b>	Oyunlaştırılmış Çevrimiçi Öğrenme Ortamı
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	<p>1. Değer döndüren fonksiyonların kullanıldığı programlar geliştirir.</p> <p>2. Çok sayıda sıralı ve isimli argümanlı fonksiyonların kullanıldığı programlar geliştirir.</p>
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Öğrencilerin işbirliği ortamında çalışabilmeleri için işbirliği forumu hazırlandı.</li> <li>• Oyunlaştırma bileşenleriyle ilgili hazırlıklar yapıldı.</li> <li>• Haftanın ödevi hazırlanarak sisteme yüklendi.</li> <li>• Fonksiyonlarla (değer döndüren ve argümanlı) ilgili ders kazanımları çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Fonksiyonlarla (değer döndüren ve argümanlı) ilgili kaynak taraması yapılarak ders notları hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Fonksiyonlarla (değer döndüren ve argümanlı) ilgili ders videosu hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Öğrencilerin eğlenerek öğrenmelerine olanak tanıyan eğlencim etkinliği sistemde hazır hale getirildi.</li> <li>• Ölçme ve değerlendirme amacıyla kullanılacak 4 sorudan oluşan kodlama deneyimi etkinliği hazırlandı. Hazırlanan etkinlik Moodle eklentilerinden olan Code Runner kullanılarak sınav haline dönüştürüldü.</li> </ul>
<b>BÖLÜM IV</b>	
<b>Derse İşleniş:</b>	<p>1. Öğrenciler fonksiyonlar (değer döndüren ve argümanlı) dersine başlamadan fonksiyonlarla (parametre alan) ilgili verilen ödevi tamamlayacaktır. Ödev tamamlandıktan sonra fonksiyonlar (değer döndüren ve argümanlı) kazanımları aktif hale gelecektir.</p>



	<p><b>2.</b> Fonksiyonlarla (değer döndüren ve argümanlı) ilgili ders kazanımları okunacaktır. Kazanımlar okunduktan sonra ders notları öğrenciler için sistemde aktif hale gelecektir.</p> <p><b>3.</b> Fonksiyonlarla (değer döndüren ve argümanlı) ilgili sunulan ders notları öğrenciler tarafından ayrıntılı olarak incelenecektir. Öğrenciler ders notlarını inceledikten sonra ders anlatımı videosu öğrencilerin kullanımına açılacaktır.</p> <p><b>4.</b> Fonksiyonlar (değer döndüren ve argümanlı) ders videosu öğrenciler tarafından seyredilecektir. Örnek uygulamalar öğrenciler tarafından takip edilecektir. Ders videosunu tamandıktan sonra ders konusuyla ilgili hazırlanan eğlenelim etkinliği öğrencilere açılacaktır.</p> <p><b>5. Sodoku</b> isimli eğlenelim etkinliği öğrencilerin ders esnasında eğlenmelerine olanak sağlayan etkinliklerden oluşmaktadır. Bu etkinliği tamamlayan öğrencilere kodlama deneyimi etkinliği aktif hale gelecektir. Kodlama deneyimi etkinliğine videoya yerleştirilen şifre ile erişilecektir.</p> <p><b>6.</b> Öğrenciler haftanın konusuyla ilgili işbirliği forumunda bilgi alışverişinde bulunabileceklerdir. Ayrıca öğrenme ortamıyla ilgili karşılaştıkları sorunlarla ilgili sıkça sorulan sorular kısmından bilgi alabileceklerdir.</p>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Fonksiyonlar (değer döndüren ve argümanlı) konusuyla ilgili 4 sorudan oluşan kodlama deneyimi etkinliği uygulanacaktır. Moodle eklentisi olan Code Runner ortamında sunulan sorular öğrenciler tarafından kodlanacaktır

<b>Uygulama 8. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	<b>1.</b> Liste tanımlama

	<p>2. Listelerde kullanılan metotlar</p> <p>3. List () fonksiyonu ile tam sayı liste oluşturma</p> <p>4. * operatörü ile liste oluşturma</p>
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Çevrimiçi Öğrenme
<b>Araç gereçler</b>	Oyunlaştırılmış Çevrimiçi Öğrenme Ortamı
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	<p>1. Liste türünde veri tanımlamayı bilir.</p> <p>2. Liste veri tipini ve metotlarını kullanarak programlar geliştirir.</p>
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Öğrencilerin işbirliği ortamında çalışabilmeleri için işbirliği forumu hazırlandı.</li> <li>• Haftanın ödevi hazırlanarak sisteme yüklendi.</li> <li>• Listeler ilgili ders kazanımları çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Listelerle ilgili kaynak taraması yapılarak ders notları hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Listelerle ilgili ders videosu hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Öğrencilerin eğlenerek öğrenmelerine olanak tanıyan eğlenceli etkinlikleri sistemde hazır hale getirildi.</li> <li>• Ölçme ve değerlendirme amacıyla kullanılacak 3 sorudan oluşan kodlama deneyimi etkinliği hazırlandı. Hazırlanan etkinlik Moodle eklentilerinden olan Code Runner kullanılarak sınav haline dönüştürüldü.</li> </ul>
<b>BÖLÜM IV</b>	
<b>Derse İşleniş:</b>	<p>1. Öğrenciler listeler dersine başlamadan fonksiyonlar (değer döndüren ve argümanlı) ilgili verilen ödevi tamamlayacaktır. Ödev tamamlandıktan sonra listeler kazanımları aktif hale gelecektir.</p>

	<p>2. Listeler ilgili ders kazanımları okunacaktır. Kazanımlar okunduktan sonra ders notları öğrenciler için sistemde aktif hale gelecektir.</p> <p>3. Listelerle ilgili sunulan ders notları öğrenciler tarafından ayrıntılı olarak incelenecektir. Öğrenciler ders notlarını inceledikten sonra ders anlatımı videosu öğrencilerin kullanımına açılacaktır.</p> <p>4. Listeler ders videosu öğrenciler tarafından seyredilecektir. Örnek uygulamalar öğrenciler tarafından takip edilecektir. Ders videosunu tamamlandıktan sonra ders konusuyla ilgili hazırlanan eğlenelim etkinliği öğrencilere açılacaktır.</p> <p>5. <b>Cevabı Sende</b> isimli eğlenelim etkinliği öğrencilerin ders esnasında eğlenmelerine olanak sağlayan etkinliklerden oluşmaktadır. Bu etkinliği tamamlayan öğrencilere kodlama deneyimi etkinliği aktif hale gelecektir. Kodlama deneyimi etkinliğine videoya yerleştirilen şifre ile erişilecektir.</p> <p>6. Öğrenciler haftanın konusuyla ilgili işbirliği forumunda bilgi alış verişinde bulunabileceklerdir. Ayrıca öğrenme ortamında karşılaştıkları sorunlarla ilgili sıkça sorulan sorular kısmından bilgi alabileceklerdir.</p>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Listeler konusuyla ilgili 3 sorudan oluşan kodlama deneyimi uygulanacaktır. Moodle eklentisi olan Code Runner ortamında sunulan sorular öğrenciler tarafından kodlanacaktır

<b>Uygulama 9. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	<p>1. List() fonksiyonu</p> <p>2. Liste öğelerine erişme</p> <p>3. Liste öğelerini değiştirme</p>

	<p>4. Liste öğelerini birleştirme</p> <p>5. Demetler</p>
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Çevrimiçi Öğrenme
<b>Araç gereçler</b>	Oyunlaştırılmış Çevrimiçi Öğrenme Ortamı
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	<p>1. Liste veri tipini ve metotlarını kullanarak programlar geliştirir.</p> <p>2. Demet veri tipini ve metotlarını kullanarak programlar geliştirir.</p>
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Öğrencilerin işbirliği ortamında çalışabilmeleri için işbirliği forumu hazırlandı.</li> <li>• Oyunlaştırma bileşenleriyle ilgili hazırlıklar yapıldı.</li> <li>• Haftanın ödevi hazırlanarak sisteme yüklendi.</li> <li>• Liste metotları ve demetler ders kazanımları çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Liste metotları ve demetlerle ilgili kaynak taraması yapılarak ders notları hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Liste metotları ve demetlerle ilgili ders videosu hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Öğrencilerin eğlenerek öğrenmelerine olanak tanıyan eğlenceli etkinlik sistemi hazır hale getirildi.</li> <li>• Ölçme ve değerlendirme amacıyla kullanılacak 3 sorudan oluşan kodlama deneyimi etkinliği hazırlandı. Hazırlanan etkinlik Moodle eklentilerinden olan Code Runner kullanılarak sınav haline dönüştürüldü.</li> </ul>
<b>BÖLÜM IV</b>	
<b>Derse İşleniş:</b>	<p>1. Öğrenciler Liste metotları ve demetler dersine başlamadan listelerle ilgili verilen ödevi tamamlayacaktır. Ödev tamamlandıktan sonra Liste metotları ve demetler aktif hale gelecektir.</p>

	<p>2. Liste metotları ve demetler ilgili ders kazanımları okunacaktır. Kazanımlar okunduktan sonra ders notları öğrenciler için sistemde aktif hale gelecektir.</p> <p>3. Liste metotları ve demetler ilgili sunulan ders notları öğrenciler tarafından ayrıntılı olarak incelenecektir. Öğrenciler ders notlarını inceledikten sonra ders anlatımı videosu öğrencilerin kullanımına açılacaktır.</p> <p>4. Liste metotları ve demetler ders videosu öğrenciler tarafından seyredilecektir. Örnek uygulamalar öğrenciler tarafından takip edilecektir. Ders videosunu tamamlandıktan sonra ders konusuyla ilgili hazırlanan eğlenceli etkinlik öğrencilere açılacaktır.</p> <p>5. <b>Metot Bulmaca</b> isimli eğlenceli etkinlik öğrencilerin ders esnasında eğlenmelerine olanak sağlayan etkinliklerden oluşmaktadır. Bu etkinliği tamamlayan öğrencilere kodlama deneyimi etkinliği aktif hale gelecektir. Kodlama deneyimi etkinliğine videoya yerleştirilen şifre ile erişilecektir.</p> <p>6. Öğrenciler haftanın konusuyla ilgili işbirliği forumunda bilgi alış verişinde bulunabileceklerdir. Ayrıca öğrenme ortamıyla ilgili karşılaştıkları sorunlarla ilgili sıkça sorulan sorular kısmından bilgi alabileceklerdir.</p> <p>7. Öğrenciler ayrıca oyunlaştırma süreciyle ilgili sisteme yüklenen oyunlaştırma yönergesinden bilgi alabileceklerdir.</p>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Liste metotları ve demetler konusuyla ilgili 3 sorudan oluşan kodlama deneyimi etkinliği uygulanacaktır. Moodle eklentisi olan Code Runner ortamında sunulan sorular öğrenciler tarafından kodlanacaktır

<b>Uygulama 10. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf

<b>Konu</b>	<p>1. Küme veri tipi</p> <p>2. Küme veri tipi metotları</p> <ul style="list-style-type: none"> <li>a. clear()</li> <li>b. add()</li> <li>c. update()</li> <li>d. difference()</li> <li>e. discard()</li> <li>f. intersection()</li> <li>g. isdisjoint()</li> <li>h. issubset()</li> <li>i. issuperset()</li> <li>j. union()</li> <li>k. symmetric_difference()</li> </ul> <p>3. Dondurulmuş kümeler</p>
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Çevrimiçi Öğrenme
<b>Araç gereçler</b>	Oyunlaştırılmış Çevrimiçi Öğrenme Ortamı
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	<p>1. Küme türünde veri tanımlamayı ve metotlarını bilir.</p> <p>2. Küme veri tipini ve metotlarını kullanarak programlar geliştirir.</p>
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Öğrencilerin işbirliği ortamında çalışabilmeleri için işbirliği forumu hazırlandı.</li> <li>• Oyunlaştırma bileşenleriyle ilgili hazırlıklar yapıldı.</li> <li>• Haftanın ödevi hazırlanarak sisteme yüklendi.</li> <li>• Küme veri tipi ders kazanımları çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Küme veri tipiyle ilgili kaynak taraması yapılarak ders notları hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Küme veri tipiyle ilgili ders videosu hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> </ul>

	<ul style="list-style-type: none"> <li>• Öğrencilerin eğlenerek öğrenmelerine olanak tanıyan eğlenelim etkinliği sistemde hazır hale getirildi.</li> <li>• Ölçme ve değerlendirme amacıyla kullanılacak 3 sorudan oluşan kodlama deneyimi etkinliği hazırlandı. Hazırlanan etkinlik Moodle eklentilerinden olan Code Runner kullanılarak sınav haline dönüştürüldü.</li> </ul>
<b>BÖLÜM IV</b>	
<b>Derse İşlenişi:</b>	<ol style="list-style-type: none"> <li>1. Öğrenciler küme veri tipi dersine başlamadan liste metotları ve demetler ilgili verilen ödevi tamamlayacaktır. Ödev tamamlandıktan sonra Liste metotları ve demetler aktif hale gelecektir.</li> <li>2. Küme veri tipiyle ilgili ders kazanımları okunacaktır. Kazanımlar okunduktan sonra ders notları öğrenciler için sistemde aktif hale gelecektir.</li> <li>3. Küme veri tipiyle ilgili sunulan ders notları öğrenciler tarafından ayrıntılı olarak incelenecektir. Öğrenciler ders notlarını inceledikten sonra ders anlatımı videosu öğrencilerin kullanımına açılacaktır.</li> <li>4. Küme veri tipiyle ders videosu öğrenciler tarafından seyredilecektir. Örnek uygulamalar öğrenciler tarafından takip edilecektir. Ders videosunu tamamlandıktan sonra ders konusuyla ilgili hazırlanan eğlenelim etkinliği öğrencilere açılacaktır.</li> <li>5. Küme Hafızası isimli eğlenelim etkinliği öğrencilerin ders esnasında eğlenmelerine olanak sağlayan etkinliklerden oluşmaktadır. Bu etkinliği tamamlayan öğrencilere kodlama deneyimi etkinliği aktif hale gelecektir. Kodlama deneyimi etkinliğine videoya yerleştirilen şifre ile erişilecektir.</li> <li>6. Öğrenciler haftanın konusuyla ilgili işbirliği forumunda bilgi alış verişinde bulunabileceklerdir. Ayrıca öğrenme ortamıyla ilgili karşılaştıkları sorunlarla ilgili sıkça sorulan sorular kısmından bilgi alabileceklerdir.</li> <li>7. Öğrenciler ayrıca oyunlaştırma süreciyle ilgili sisteme yüklenen oyunlaştırma yönergesinden bilgi alabileceklerdir.</li> </ol>

**BÖLÜM V****Ölçme ve  
Değerlendirme:**

Küme veri tipi konusyla ilgili 3 sorudan oluşan kodlama deneyimi etkinliği uygulanacaktır. Moodle eklentisi olan Code Runner ortamında sunulan sorular öğrenciler tarafından kodlanacaktır



## Ek 9. OYYÖO Ders Planları

Pilot Uygulama 1. Hafta Ders Planı	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	Python Giriş
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Anlatım, Soru-Cevap, Gösterip Yaptırma
<b>Araç gereçler</b>	Bilgisayar, Etkileşimli Tahta, Python Kod Editörü, Oyunlaştırma Bileşenleri
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	<ol style="list-style-type: none"><li>1. Python programlama dilini bilir.</li><li>2. Python programlama dilinin avantajlarını bilir.</li><li>3. Python kurulumunu yapabilir.</li><li>4. Python kod yazma ortamlarını bilir.</li></ol>
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"><li>• Oyunlaştırma yönergesi hazırlandı.</li><li>• Kullanılacak oyunlaştırma bileşenleri hazırlandı.</li><li>• Haftanın ödevi hazırlandı.</li><li>• Ders öncesinde kaynak taraması yapılarak öğrenci çalışma yaprakları hazırlandı.</li><li>• Konuyla ilgili ders sunumu hazırlandı.</li><li>• Python kurulum dosyası hazırlanır.</li><li>• Python kodlarının yazılabileceği diğer ortamlar hazırlanır.</li><li>• Eğlence unsurunun öğrenme ortamına aktarılabilmesi amacıyla eğlenelim etkinliği hazırlandı.</li><li>• Ölçme ve değerlendirme amacıyla kullanılacak 6 sorudan oluşan kodlama deneyimi etkinliği hazırlandı.</li></ul>
<b>BÖLÜM IV</b>	
<b>Derse İşlenişi:</b>	<ol style="list-style-type: none"><li>1. Ödevlerin kontrolü yapılacak ve öğrencilere ödevle ilgili sorular yöneltilecek.</li></ol>

	<p>2. Python programla dilini daha önce duyulup duyulmadığı öğrencilere sorulur ve öğrencilerin cevaplama istenir.</p> <p>3. Python tanıtımı yapılarak ne gibi avantajlarının bulunduğu anlatılır. Ayrıca python kullanan şirketlere örnekler verilir.</p> <p>4. Python programlama dili öğrencilerle birlikte bilgisayarlara kurulur.</p> <p>5. Program kurulduktan sonra kod yazmada kullanılacak IDLE ortamı açılır ve öğrencilerinde açması istenir.</p> <p>6. Python kodlarının yazımında kullanılacak diğer ortamlar (<b>Jupyter-iPython, PyCharm</b>) öğrencilere gösterilir.</p> <p>7. Pythonda program dosyasının nasıl açılacağı ve kaydedileceği öğrencilere gösterilir.</p> <p>8. Öğrencilerle birlikte bir program dosyası açılır ve örnek kodlar yazıldıktan sonra kaydedilir. Akabinde yazılan örnek kodlar çalıştırılır. Bu şekilde öğrencilerinde programın nasıl çalıştığını görmesi sağlanır.</p> <p>9. <b>Kes-Yapıştır</b> isimli eğlenceli etkinliği uygulanır.</p>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Konunun öğrenciler tarafında anlaşılıp anlaşılmadığının kontrolü için 6 sorudan oluşan kodlama deneyimi etkinliği uygulanır.

<b>Pilot Uygulama 2. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	Karakter Dizileri, Aritmetiksel Operatörler, Kaçış İfadeleri
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Anlatım, Soru-Cevap, Gösterip Yaptırma
<b>Araç gereçler</b>	Bilgisayar, Etkileşimli Tahta, Python Kod Editörü, Çalışma Yaprakları, Oyunlaştırma Bileşenleri

<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	<ol style="list-style-type: none"> <li>1. Karakter dizilerinin ne olduğunu bilir.</li> <li>2. Karakter dizileri üzerinde birleştirme ve çoğaltma işlemlerinin nasıl yapılacağını kavrar.</li> <li>3. Type() fonksiyonunu kullanabilir.</li> <li>4. Aritmetik işlem operatörlerini kullanabilir.</li> <li>5. Kaçış ifadelerini kullanabilir.</li> </ol>
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Öğrenci puanları hesaplanarak lider tahtası hazırlanır. Ayrıca rozet kazananlara rozetleri verilir.</li> <li>• Haftanın ödevi hazırlanır.</li> <li>• Ders öncesinde kaynak taraması yapılarak öğrenci çalışma yaprakları hazırlandı.</li> <li>• Konuyla ilgili ders sunumu hazırlandı.</li> <li>• Karakter dizileri, aritmetiksel operatörler ve kaçış ifadeleriyle ilgili öğrencilere sunulacak örnekler hazırlandı.</li> <li>• Eğlence unsurunun öğrenme ortamına aktarılabilmesi amacıyla eğlenelim etkinliği hazırlandı.</li> <li>• Ölçme ve değerlendirme amacıyla kullanılacak 4 sorudan oluşan kodlama deneyimi etkinliği hazırlandı.</li> </ul>
<b>BÖLÜM IV</b>	
<b>Derse İşlenişi:</b>	<ol style="list-style-type: none"> <li>1. Lider tahtasında gerekli düzenlemeler yapılır. Rozetler dağıtılır.</li> <li>2. Haftanın ödevleri kontrol edilir.</li> <li>3. Karakter dizileri anlatılır. Tırnak kullanımı, birleştirme, çoğaltma ve type() fonksiyonun kullanımı örneklerle anlatılır.</li> <li>4. Karakter dizileriyle ilgili örnekler gösterilir ve öğrencilerinde yapmaları istenir.</li> <li>5. Aritmetiksel operatörler ve kaçış ifadeleri örneklerle öğrencilere sunulur.</li> <li>6. Aritmetiksel operatörler ve kaçış ifadelerinin kullanımıyla ilgili öğrencilere sorular yöneltilir ve öğrencilerin kodlaması istenir.</li> <li>7. <b>Çek Kodunu Söyle Çıktını</b> isimli eğlenelim etkinliği uygulanır.</li> </ol>

<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Karakter dizileri, aritmetiksel operatörler ve kaçış ifadeleriyle ilgili kodlama deneyimi etkinliği uygulanır.

<b>Uygulama 1. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	Değişkenler ve Sabitler
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Anlatım, Soru-Cevap, Gösterip Yaptırma
<b>Araç gereçler</b>	Bilgisayar, Etkileşimli Tahta, Python Kod Editörü, Çalışma Yaprakları, Oyunlaştırma Bileşenleri
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	<ol style="list-style-type: none"> <li>1.Değişken tanımlama kurallarını bilir.</li> <li>2.Temel veri tiplerini kavrar</li> <li>3.Değişken ve temel veri tiplerini kullanarak programlar geliştirir.</li> </ol>
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Öğrenci puanları hesaplanarak lider tahtası hazırlanır. Ayrıca rozet kazananlara rozetleri verilir.</li> <li>• Haftanın ödevi hazırlanır.</li> <li>• Ders öncesinde kaynak taraması yapılarak öğrenci çalışma yaprakları hazırlandı.</li> <li>• Konuyla ilgili ders sunumu hazırlandı.</li> <li>• Değişken tanımlamayla ilgili öğrencilere sunulacak örnekler hazırlandı.</li> <li>• Eğlence unsurunun öğrenme ortamına aktarılabilmesi amacıyla eğlenelim etkinliği hazırlandı.</li> <li>• Ölçme ve değerlendirme amacıyla kullanılacak 4 sorudan oluşan kodlama deneyimi etkinliği hazırlandı.</li> </ul>

<b>BÖLÜM IV</b>	
<b>Derse İşlenişi:</b>	<ol style="list-style-type: none"> <li>1. Lider tahtasında gerekli düzenlemeler yapılır. Rozetler dağıtılır.</li> <li>2. Haftanın ödevleri kontrol edilir.</li> <li>3. Değişken kavramının ne olduğu hakkında öğrencilerin fikirlerinin alınmasına yönelik sorular sorulur.</li> <li>4. Anlatım yöntemi kullanılarak değişken kavramı anlatılır. Değişken kavramıyla ilgili olarak anlatılacaklar; <ol style="list-style-type: none"> <li>a. Değişken kavramının ne olduğu,</li> <li>b. Değişken tanımlamada hangi operatörün kullanıldığı,</li> <li>c. Tanımlanan değişkenler üzerinde hangi işlemlerin yapılacağı ve nasıl kullanılacağı şeklindedir.</li> </ol> </li> <li>5. Değişken kavramının daha iyi anlaşılabilmesi için çeşitli değişken tanımlama örnekleri gösterilir.</li> <li>6. Değişken uzunluğunu hesaplamak için kullanılan <i>len()</i> fonksiyonunu nasıl kullanıldığı gösterilir ve öğrencilerinde örneklerde tanımladıkları değişkenlerin uzunluklarını hesaplamaları istenir. Ayrıca birden fazla değişkenin uzunluklarının hesaplanması ve aritmetik olarak işleme tabi tutulması gibi işlemler örneklerle anlatılır.</li> <li>7. Değişken tanımlama kuralları anlatılır. Yanlış tanımlanan değişkenler öğrencilere sunulur. Öğrencilerin bu kurallar dikkate alınarak örnek değişken tanımlamaları istenir.</li> <li>8. Gösterip yaptırma yöntemi ile öğrencilere değişken tanımlamayla ilgili örnekler verilerek öğrencilerinde değişken tanımlamalarına olanak tanınır.</li> <li>9. Temel veri tipleri anlatılır ve her veri tipi için örnekler sunulur. (Integer, Float, Boolean, String, Liste)</li> <li>10. Kartları Eşleştirme isimli eğlenceli etkinlik uygulanır.</li> </ol>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Değişkenler ve temel veri tipleri konusuyla ilgili öğrencilerin uygulama yapmalarına olanak tanıyarak konun pekiştirilmesini sağlayacak 4 sorudan oluşan kodlama deneyimi etkinliği

	uygulanır. Kısa sınav konu kazanımlarını ölçen örnek durumlara yönelik kodlama etkinliklerini içermektedir.
<b>Uygulama 2. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	1. Print () fonksiyonunun kullanımı 2. Print () fonksiyonunun parametreleri
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Anlatım, Soru-Cevap, Gösterip Yaptırma
<b>Araç gereçler</b>	Bilgisayar, Etkileşimli Tahta, Python Kod Editörü, Çalışma Yaprakları, Oyunlaştırma Bileşenleri
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	1. Print() fonksiyonunu kullanarak programlar geliştirir. 2. Print() fonksiyonunun parametrelerini bilir. 3. Print() fonksiyonunun parametrelerini kullanarak programlar geliştirir.
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Öğrenci puanları hesaplanarak lider tahtası hazırlanır. Ayrıca rozet kazananlara rozetleri verilir.</li> <li>• Haftanın ödevi hazırlandı.</li> <li>• Print() fonksiyonunun kullanımıyla ilgili kaynaklar tarandı.</li> <li>• Taranan kaynaklar dikkate alınarak öğrencilere verilmek üzere çalışma kâğıtları hazırlandı.</li> <li>• Ders anlatımında kullanılmak üzere sunum hazırlandı.</li> <li>• Print() fonksiyonunun kullanımıyla ilgili öğrencilere sunulmak üzere örnekler hazırlandı.</li> <li>• Eğlence unsurunun öğrenme ortamına aktarılabilmesi amacıyla eğlenelim etkinliği hazırlandı.</li> <li>• Print() fonksiyonuyla ilgili 4 sorudan oluşan kodlama deneyimi etkinliği hazırlandı.</li> </ul>

<b>BÖLÜM IV</b>	
<b>Derse İşlenişi:</b>	<ol style="list-style-type: none"> <li>1. Lider tahtasında gerekli düzenlemeler yapılır. Rozetler dağıtılır.</li> <li>2. Haftanın ödevi kontrol edildi.</li> <li>3. Print() fonksiyonunun kelime anlamının öğrencilerde ne çağrıştırdığı sorulur ve öğrencilerin cevaplamaları istenir.</li> <li>4. Anlatım yöntemi Print() fonksiyonunu konusu anlatılır. <ol style="list-style-type: none"> <li>a. Print() fonksiyonunun ne amaçla ve nasıl kullanıldığı,</li> <li>b. Print() fonksiyonunun değişkenlerle birlikte kullanımı,</li> <li>c. Print() fonksiyonunda tırnak kullanımları,</li> <li>d. Print() fonksiyonunda birden fazla parametre kullanımı,</li> </ol> </li> <li>5. Print() fonksiyonuyla ilgili örnekler sunulur ve öğrencilerinde verilen örnekleri kodlamaları istenir.</li> <li>6. Print() fonksiyonunun parametreleri anlatılarak parametrelerin nasıl kullanılacağı öğrencilere gösterilir. <ol style="list-style-type: none"> <li>a. Sep parametresi,</li> <li>b. End parametresi,</li> <li>c. File parametresi,</li> <li>d. Flush parametresi,</li> </ol> </li> <li>7. Print() fonksiyonunun parametreleriyle ilgili olarak öğrencilerin benzer kodlar yazmaları sağlanır.</li> <li>8. Print fonksiyonu ve parametreleriyle ilgili örnek durumlar verilerek öğrencilerin kodlamaları istenir.</li> <li>9. Bilgi Yarışması isimli eğlenceli etkinlik uygulanır.</li> </ol>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Print() fonksiyonu ve parametreleriyle ilgili 4 sorudan oluşan kodlama deneyimi etkinliği uygulanır. Öğrencilerin sorularda verilen durumları kodlamaları istenir.

<b>Uygulama 3. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	Kontrol Yapıları 1. If kontrol yapısı 2. If-Elif kontrol yapısı 3. If-Else kontrol yapısı 4. If-Elif-Else kontrol yapısı
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Anlatım, Soru-Cevap, Gösterip Yaptırma
<b>Araç gereçler</b>	Bilgisayar, Etkileşimli Tahta, Python Kod Editörü, Çalışma Yaprakları, Oyunlaştırma Bileşenleri
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	1. İç içe kontrol yapılarını ifade etmeyi kavrar. 2. Kontrol yapılarını kullanarak programlar geliştirir.
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Öğrenci puanları hesaplanarak lider tahtası hazırlanır. Ayrıca rozet kazananlara rozetleri verilir.</li> <li>• Haftanın ödevi hazırlandı.</li> <li>• Kontrol yapılarıyla ilgili kaynaklar tarandı.</li> <li>• Kontrol yapılarıyla ilgili çalışma yaprakları hazırlandı.</li> <li>• Kontrol yapıları sunumu hazırlandı.</li> <li>• Kontrol yapılarıyla ilgili örnek problemler hazırlandı.</li> <li>• Eğlence unsurunun öğrenme ortamına aktarılabilmesi amacıyla eğlenelim etkinliği hazırlandı.</li> <li>• Kontrol yapılarıyla ilgili 3 sorudan oluşan kodlama deneyimi etkinliği hazırlandı</li> </ul>
<b>BÖLÜM IV</b>	
<b>Derse İşlenişi:</b>	1. Lider tahtasında gerekli düzenlemeler yapılır. Rozetler dağıtılır. 2. Ödevler kontrol edildi.



	<p><b>3.</b> Günlük hayatta karşılaşılan koşullu durumlara örnekler verildi ve öğrencilere örnek vermeleri istendi.</p> <p><b>4.</b> Anlatım yöntemi kullanılarak <b>If</b> kontrol yapısının kullanımı anlatılır ve bir adet örnek problem durumunun nasıl kodlanacağı öğrenciler gösterilir.</p> <p><b>5. If-Elif</b> kontrol yapısı anlatılır. If ile if- elif kontrol yapıları arasındaki farkı ortaya koymak amacıyla örnek problem durumunun nasıl kodlanacağı gösterilir.</p> <p><b>6. If-Else</b> kontrol yapısı anlatılır. If – else kontrol yapısıyla ilgili örnek problem durumunun nasıl kodlanacağı gösterilir.</p> <p><b>7. If-Elif-Else</b> kontrol yapısı anlatılarak örnek problem durumunun nasıl kodlanacağı gösterilir.</p> <p><b>8.</b> Örnek problem durumları verilerek öğrencilerin hangi kontrol yapılarını tercih edecekleri sorulur. Tercih edilen kontrol yapılarının neden tercih edildiği tartışılır. Tartışma sonunda en uygun kontrol yapısı kullanılarak problem durumu öğrencilerle birlikte kodlanır.</p> <p><b>9. Program Çalıştırma</b> isimli eğlenceli etkinliği uygulanır.</p>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Kontrol yapılarının kullanımıyla ilgili 3 sorudan oluşan kodlama deneyimi etkinliği uygulanır.

<b>Uygulama 4. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	<p>Tekrarlı Yapılar (Döngüler)</p> <p><b>1.</b> Tekrarlı yapıların kullanım amacı</p> <p><b>2.</b> For tekrarlı yapısının kullanımı</p> <p><b>3.</b> While tekrarlı yapısının kullanımı</p> <p><b>4.</b> Tekrarlı yapılarda kullanılan yapılar (range, pass, break, continue)</p>

<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Anlatım, Soru-Cevap, Gösterip Yaptırma
<b>Araç gereçler</b>	Bilgisayar, Etkileşimli Tahta, Python Kod Editörü, Çalışma Yaprakları, Oyunlaştırma Bileşenleri
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	<ol style="list-style-type: none"> <li>1. Tekrarlı yapıları kavrar.</li> <li>2. Tekrarlı yapılarda kullanılan fonksiyonları kavrar.</li> <li>3. Tekrarlı yapıları ve tekrarlı yapılarda kullanılan fonksiyonları kullanarak programlar geliştirir.</li> </ol>
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Öğrenci puanları hesaplanarak lider tahtası hazırlanır. Ayrıca rozet kazananlara rozetleri verilir.</li> <li>• Haftanın ödevi hazırlanır.</li> <li>• Tekrarlı yapılarla ilgili kaynaklar taranır.</li> <li>• Çalışma yaprağı hazırlanır.</li> <li>• Tekrarlı yapıların anlatımı için ders sunumu hazırlanır.</li> <li>• Örnek problem durumları hazırlanır</li> <li>• Eğlence unsurunun öğrenme ortamına aktarılabilmesi amacıyla eğlenelim etkinliği hazırlandı.</li> <li>• Tekrarlı yapılarla ilgili 3 sorudan oluşan kodlama deneyimi etkinliği hazırlanır.</li> </ul>
<b>BÖLÜM IV</b>	
<b>Derse İşlenişi:</b>	<ol style="list-style-type: none"> <li>1. Ödevler kontrol edilir.</li> <li>2. Tekrarlı yapıların ne amaçla kullanıldığı ve ne tür avantajlar sağlayacağı sorusu öğrencilere sorulur.</li> <li>3. Tekrarlı yapıların kullanılması gereken problemi kontrol yapılarıyla kodladığımız zaman ne tür bir hatanın oluşacağı öğrencilere gösterilir.</li> <li>4. <b>For</b> tekrarlı yapısı anlatılır. Nasıl kullanılacağıyla ilgili problem kodlanır ve öğrencilerinde kodlaması istenir.</li> <li>5. <b>While</b> tekrarlı yapısı anlatılır. Nasıl kullanılacağıyla ilgili problem kodlanır ve öğrencilerinde kodlaması istenir.</li> </ol>

	<p><b>6. For-While</b> tekrarlı yapılarının kullanım farkının ortaya konması için örnek problem iki tekrarlı yapı kullanılarak kodlanır.</p> <p><b>7.</b> Tekrarlı yapılarda kullanılan range, pass, break, continue yapılarının kullanımları örneklerle anlatılır.</p> <p><b>8. Kahoot</b> isimli eğlenceli etkinlik uygulanır.</p>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Tekrarlı yapılarla ilgili 3 sorudan oluşan ve öğrenme kazanımlarını kapsayan kodlama deneyimi etkinliği uygulanır. Kodlama deneyimi etkinliği örnek problem durumlarının öğrencilerin kodlamasına yöneliktir.

<b>Uygulama 5. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	<p>Fonksiyonlar</p> <ol style="list-style-type: none"> <li>1. Fonksiyon kavramı ve türleri</li> <li>2. Fonksiyon tanımlama</li> <li>3. Parametre almayan fonksiyonların kullanımı</li> </ol>
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Anlatım, Soru-Cevap, Gösterip Yaptırma
<b>Araç gereçler</b>	Bilgisayar, Etkileşimli Tahta, Python Kod Editörü, Çalışma Yaprakları, Oyunlaştırma Bileşenleri
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	<ol style="list-style-type: none"> <li>1. Fonksiyonun ne olduğunu bilir.</li> <li>2. Fonksiyon türlerini bilir.</li> <li>3. Fonksiyonun nasıl tanımlanacağını kavrar.</li> <li>4. Parametre almayan fonksiyonları kullandığı programlar geliştirir.</li> </ol>

### BÖLÜM III

<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"><li>• Öğrenci puanları hesaplanarak lider tahtası hazırlanır. Ayrıca rozet kazananlara rozetleri verilir.</li><li>• Haftanın ödevi hazırlanır.</li><li>• Fonksiyonlar konusuyla ilgili kaynak taraması yapılır.</li><li>• Çalışma yaprağı hazırlanır.</li><li>• Fonksiyon kavramı, fonksiyon tanımlama ve parametre almayan fonksiyonlarla ilgili ders sunumu hazırlanır</li><li>• Parametre almayan fonksiyon türlerine ilişkin örnek problem durumları hazırlanır.</li><li>• Eğlence unsurunun öğrenme ortamına aktarılabilmesi amacıyla eğlenelim etkinliği hazırlandı.</li><li>• Fonksiyon tanımlamayla ilgili 3 soruluk kodlama deneyimi etkinliği hazırlanır.</li></ul>
------------------------	---

### BÖLÜM IV

<b>Derse İşlenişi:</b>	<ol style="list-style-type: none"><li>1. Lider tahtasında gerekli düzenlemeler yapılır. Rozetler dağıtılır.</li><li>2. Ödev kontrolü yapılır.</li><li>3. Fonksiyon kavramı hakkında öğrencilerin ne düşündükleri sorulur. Öğrencilerin cevaplarına göre fonksiyon kavramı günlük hayattan örneklerle anlatılır.</li><li>4. Fonksiyon türleri anlatılır ve asıl anlatılacak konunun tanımlayacağımız özel fonksiyonlar olduğu ifade edilir.</li><li>5. Fonksiyon tanımlamada def ifadesinin görevi ve fonksiyonun nasıl tanımlanacağı örnek durumlar kodlanarak gösterilir ve öğrencilerinde belirtilen örneği kodlamaları istenir.</li><li>6. Öğrencilerle birlikte üç örnek problem durumuna yönelik kullanılabilecek fonksiyon tanımlamaları yapılır. Burada öğrencilere problem durumlarına yönelik ne tür fonksiyonlara ihtiyaç duyulduğu sorulur. Öğrencilerin görüşlerinin uygulanabilirliği sorgulanır.</li><li>7. <b>Şifreli Bulmaca</b> isimli eğlenelim etkinliği uygulanır.</li></ol>
------------------------	--

<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Parametre almayan fonksiyonlarla ilgili 3 sorudan oluşan kodlama deneyimi etkinliği uygulanır. Etkinlikte öğrencilere verilen problemler için fonksiyon tanımlamaları istenir.

<b>Uygulama 6. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	Fonksiyonlar 1. Parametre alan fonksiyonlar 2. İsimli ve sıralı argümanlar 3. Varsayılan değerli argümanlar
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Anlatım, Soru-Cevap, Gösterip Yaptırma
<b>Araç gereçler</b>	Bilgisayar, Etkileşimli Tahta, Python Kod Editörü, Çalışma Yaprakları, Oyunlaştırma Bileşenleri
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	1. Parametre alan fonksiyonların kullanıldığı programlar geliştirir. 2. İsimli ve sıralı argümanların kullanıldığı programlar geliştirir. 3. Varsayılan argümanların kullanıldığı programlar geliştirir.
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Öğrenci puanları hesaplanarak lider tahtası hazırlanır. Ayrıca rozet kazananlara rozetleri verilir.</li> <li>• Haftanın ödevi hazırlanır.</li> <li>• Parametre alan fonksiyonlarla ilgili kaynak taraması yapılır.</li> <li>• Çalışma kâğıdı hazırlanır.</li> <li>• Parametre alan fonksiyonlar, isimli ve sıralı argümanlar, varsayılan değerli argümanlar konularını kapsayan ders sunumu hazırlanır</li> <li>• Örnek program durumları hazırlanır.</li> </ul>

	<ul style="list-style-type: none"> <li>• Eğlence unsurunun öğrenme ortamına aktarılabilmesi amacıyla eğlenelim etkinliği hazırlandı.</li> <li>• Kazanımlarla ilgili 3 sorudan oluşan kodlama deneyimi etkinliği hazırlanır.</li> </ul>
<b>BÖLÜM IV</b>	
<b>Derse İşlenişi:</b>	<ol style="list-style-type: none"> <li>1. Lider tahtasında gerekli düzenlemeler yapılır. Rozetler dağıtılır.</li> <li>2. Ödev kontrolü yapılır.</li> <li>3. Parametre kavramının ne anlama geldiği sorusu öğrencilere yöneltilir. Ayrıca parametre alan ve almayan fonksiyonlar arasında ne gibi farklılıkların olduğu tartışılır.</li> <li>4. Parametre alan fonksiyonlar anlatılır. Parametresi olarak kodlanmış problem durumu, parametre alan fonksiyonlar kullanılarak tekrar kodlanır. Aralarındaki farkın öğrenciler tarafından görülmesi sağlanır.</li> <li>5. Argüman kavramı anlatılarak, isimli ve sıralı argümanlarla ilgili örnek problem durumları kodlanarak öğrencilere gösterilir. Öğrencilerinde kodlaması sağlanır.</li> <li>6. Varsayılan değerli argümanların kullanıldığı fonksiyonlar anlatılır. Öğrencilerle birlikte örnek problem durumları kodlanır.</li> <li>7. <b>Plickers</b> isimli eğlenelim etkinliği uygulanır.</li> </ol>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Parametre alan fonksiyonların kullanımı ve argüman tipleriyle ilgili 3 sorudan oluşan kodlama deneyimi etkinliği uygulanır. Etkinlik belirtilen kazanımları ölçmeye yönelik kodlama durumlarından oluşmaktadır.

<b>Uygulama 7. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	Fonksiyonlar 1. Değer döndüren fonksiyonlar

	<p>a. Global ifadesi</p> <p>b. Return ifadesi</p> <p>c. Pass ifadesi</p> <p>2. İstenen sayıda sıralı argümanlar</p> <p>3. İstenen sayıda isimli argümanlar</p>
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Anlatım, Soru-Cevap, Gösterip Yaptırma
<b>Araç gereçler</b>	Bilgisayar, Etkileşimli Tahta, Python Kod Editörü, Çalışma Yaprakları, Oyunlaştırma Bileşenleri
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	<p>1. Değer döndüren fonksiyonların kullanıldığı programlar geliştirir.</p> <p>2. Çok sayıda sıralı ve isimli argümanlı fonksiyonların kullanıldığı programlar geliştirir.</p>
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Öğrenci puanları hesaplanarak lider tahtası hazırlanır. Ayrıca rozet kazananlara rozetleri verilir.</li> <li>• Haftanın ödevi hazırlanır.</li> <li>• Değer döndüren fonksiyonlar, istenen sayıda isimli ve sıralı argüman konularıyla ilgili kaynak taraması yapılır.</li> <li>• Çalışma kâğıdı hazırlanır.</li> <li>• Değer döndüren fonksiyonlarla ilgili ders sunumu hazırlanır.</li> <li>• İstenen sayıda isimli ve sıralı argüman kullanımıyla ilgili ders sunumu hazırlanır.</li> <li>• Örnek problem durumları hazırlanır.</li> <li>• Eğlence unsurunun öğrenme ortamına aktarılabilmesi amacıyla eğlenelim etkinliği hazırlandı.</li> <li>• Değer döndüren fonksiyonlar ve istenen sayıda isimli-sıralı argüman kullanımına yönelik 4 sorudan oluşan kodlama deneyimi etkinliği hazırlanır.</li> </ul>

<b>BÖLÜM IV</b>	
<b>Derse İşlenişi:</b>	<ol style="list-style-type: none"> <li>1. Lider tahtasında gerekli düzenlemeler yapılır. Rozetler dağıtılır.</li> <li>2. Ödev kontrolü yapılır.</li> <li>3. Değer döndürme işlemi hakkında öğrencilerin ne düşündüğü sorulur. Örnek verilen problem durumuyla ilgili değer döndürme işleminin nasıl yapılacağı gösterilir.</li> <li>4. Range ifadesi örnek problem durumu kodlanarak gösterilir ve öğrencilerin kodlamaları istenir.</li> <li>5. Global ifadesi örnek problem durumu kodlanarak gösterilir ve öğrencilerin kodlamaları istenir.</li> <li>6. Pass ifadesi örnek problem durumu kodlanarak gösterilir ve öğrencilerin kodlamaları istenir.</li> <li>7. İstenen sayıda sıralı argüman kullanımıyla ilgili örnek problem durumu öğrencilerle birlikte kodlanır.</li> <li>8. İstenen sayıda isimli argüman kullanımıyla ilgili örnek problem durumu öğrencilerle birlikte kodlanır.</li> <li>9. <b>Sudoku</b> isimli eğlenelim etkinliği uygulanır.</li> </ol>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Kazanımları kapsayan 4 sorudan oluşan kodlama deneyimi etkinliği uygulanır. Öğrencilerden değer döndüren fonksiyon kullanımı, istenen sayıda isimli ve sıralı argüman kullanımıyla ilgili problem durumlarını kodlamaları istenir.

<b>Uygulama 8. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	<ol style="list-style-type: none"> <li>1. Liste tanımlama</li> <li>2. Listelerde kullanılan metotlar</li> <li>3. List () fonksiyonu ile tam sayı liste oluşturma</li> <li>4. * operatörü ile liste oluşturma</li> </ol>
<b>Süre</b>	2 Ders Saati



<b>Öğretim Yöntem ve Teknikleri</b>	Anlatım, Soru-Cevap, Gösterip Yaptırma
<b>Araç gereçler</b>	Bilgisayar, Etkileşimli Tahta, Python Kod Editörü, Çalışma Yaprakları, Oyunlaştırma Bileşenleri
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	<ol style="list-style-type: none"> <li>1. Liste türünde veri tanımlamayı bilir.</li> <li>2. Liste veri tipini ve metotlarını kullanarak programlar geliştirir.</li> </ol>
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Öğrenci puanları hesaplanarak lider tahtası hazırlanır. Ayrıca rozet kazananlara rozetleri verilir.</li> <li>• Haftanın ödevi hazırlanır.</li> <li>• Liste tanımlama ve metotlarıyla ilgili kaynak taraması yapılır.</li> <li>• Listelerle ilgili çalışma kâğıdı hazırlanır.</li> <li>• Listeler ve kullanılan metotlarla ilgili ders sunumu hazırlanır.</li> <li>• Örnek problem durumları hazırlanır.</li> <li>• Eğlence unsurunun öğrenme ortamına aktarılabilmesi amacıyla eğlenelim etkinliği hazırlandı.</li> <li>• Listelerle ilgili 3 sorudan oluşan kodlama deneyimi etkinliği hazırlanır.</li> </ul>
<b>BÖLÜM IV</b>	
<b>Derse İşlenişi:</b>	<ol style="list-style-type: none"> <li>1. Lider tahtasında gerekli düzenlemeler yapılır. Rozetler dağıtılır.</li> <li>2. Ödev kontrolü yapılır.</li> <li>3. Liste veri tipinin ne amaçla kullanıldığı ve ne gibi avantajlarının olduğu öğrencilere sorulur. Öğrencilerin cevapları tartışılır.</li> <li>4. Liste tanımlamanın nasıl yapılacağı örneklerle gösterilir. Belirtilen örneklere benzer tanımlamaların öğrenciler tarafından yapılması istenir.</li> <li>5. Liste metotları sırasıyla örneklerle anlatılır. Örneklerin öğrenciler tarafından kodlanması sağlanır. <ol style="list-style-type: none"> <li>a. Append()</li> <li>b. Remove()</li> <li>c. Del()</li> <li>d. Reverse()</li> </ol> </li> </ol>

	<p>e. Sort() f. Index() g. Count()</p> <p>6. List() fonksiyonunu kullanarak liste tanımlamanın nasıl yapılacağı örneklerle gösterilir.</p> <p>7. * operatörü kullanılarak liste tanımlamanın nasıl yapılacağı örneklerle gösterilir.</p> <p>8. <b>Flipquiz</b> isimli eğlenceli etkinlik uygulanır.</p>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Kazanımları kapsayan 3 sorudan oluşan kodlama deneyimi etkinliği uygulanır. Sınavda öğrencilerden liste tanımlama ve liste metotlarının kullanımlarını içeren örnek problem durumlarını kodlamaları istenir.

<b>Uygulama 9. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	<p>1. List() fonksiyonu</p> <p>2. Liste öğelerine erişme</p> <p>3. Liste öğelerini değiştirme</p> <p>4. Liste öğelerini birleştirme</p> <p>5. Demetler</p>
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Anlatım, Soru-Cevap, Gösterip Yaptırma
<b>Araç gereçler</b>	Bilgisayar, Etkileşimli Tahta, Python Kod Editörü, Çalışma Yaprakları, Oyunlaştırma Bileşenleri
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	<p>1. Liste veri tipini ve metotlarını kullanarak programlar geliştirir.</p> <p>2. Demet veri tipini ve metotlarını kullanarak programlar geliştirir.</p>

<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Öğrenci puanları hesaplanarak lider tahtası hazırlanır. Ayrıca rozet kazananlara rozetleri verilir.</li> <li>• Haftanın ödevi hazırlanır.</li> <li>• Liste temel fonksiyonları ve demetlerle ilgili kaynak taraması yapılır</li> <li>• Çalışma kâğıdı hazırlanır.</li> <li>• Liste temel fonksiyonları ve demetlerle ilgili ders sunumu hazırlanır.</li> <li>• Ders anlatımı esnasında kullanılmak üzere örnek problem durumları hazırlanır.</li> <li>• Eğlence unsurunun öğrenme ortamına aktarılabilmesi amacıyla eğlenelim etkinliği hazırlandı.</li> <li>• Liste temel fonksiyonları ve demetlerle ilgili kodlama deneyimi etkinliği hazırlanır.</li> </ul>
<b>BÖLÜM IV</b>	
<b>Derse İşlenişi:</b>	<ol style="list-style-type: none"> <li>1. Lider tahtasında gerekli düzenlemeler yapılır. Rozetler dağıtılır.</li> <li>2. Ödev kontrolü yapılır.</li> <li>3. Listelerde ne gibi işlemlerin yapılabileceği öğrencilere sorulur ve öğrencilerin cevaplama istenir.</li> <li>4. List() fonksiyonunun kullanımı örnek üzerinde anlatılır ve öğrencilerim belirtilen örneği kodlaması istenir.</li> <li>5. Liste öğelerine erişimin nasıl olacağı anlatılır ve iki örnek öğrencilerle birlikte kodlanır.</li> <li>6. Liste öğelerinin nasıl değiştirileceği örneklerle anlatılır.</li> <li>7. Liste öğelerinin nasıl birleştirileceği örneklerle anlatılır.</li> <li>8. Demet veri tipinin ne olduğu anlatılır ve nasıl tanımlanacağı gösterilir. Öğrencilerinde demet veri tipinde örnek tanımlama yapmaları istenir.</li> <li>9. <b>Metot Bulmaca</b> isimli eğlenelim etkinliği uygulanır.</li> </ol>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Liste veri tipi metotları ve demet veri tipiyle ilgili 3 sorudan oluşan kodlama deneyimi etkinliği uygulanır.

<b>Uygulama 10. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	<p>1. Küme veri tipi</p> <p>2. Küme veri tipi metotları</p> <ul style="list-style-type: none"> <li>a. clear()</li> <li>b. add()</li> <li>c. update()</li> <li>d. difference()</li> <li>e. discard()</li> <li>f. intersection()</li> <li>g. isdisjoint()</li> <li>h. issubset()</li> <li>i. issuperset()</li> <li>j. union()</li> <li>k. symmetric_difference()</li> </ul> <p>3. Dondurulmuş kümeler</p>
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Anlatım, Soru-Cevap, Gösterip Yaptırma
<b>Araç gereçler</b>	Bilgisayar, Etkileşimli Tahta, Python Kod Editörü, Çalışma Yaprakları, Oyunlaştırma Bileşenleri
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	<p>1. Küme türünde veri tanımlamayı ve metotlarını bilir.</p> <p>2. Küme veri tipini ve metotlarını kullanarak programlar geliştirir.</p>
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Öğrenci puanları hesaplanarak lider tahtası hazırlanır. Ayrıca rozet kazananlara rozetleri verilir.</li> <li>• Haftanın ödevi hazırlanır.</li> </ul>

	<ul style="list-style-type: none"> <li>• Küme tanımlama ve küme metotlarıyla ilgili kaynak taraması yapılır.</li> <li>• Küme tanımlama ve metotlarının kullanımıyla ilgili çalışma kâğıtları hazırlanır.</li> <li>• Küme tanımlama ve metotlarıyla ilgili ders sunumu hazırlanır.</li> <li>• Kümelerle ilgili örnek problem durumları hazırlanır.</li> <li>• Eğlence unsurunun öğrenme ortamına aktarılabilmesi amacıyla eğlenelim etkinliği hazırlandı.</li> <li>• Küme tanımlama ve metotlarla ilgili kodlama deneyimi etkinliği hazırlanır.</li> </ul>
<b>BÖLÜM IV</b>	
<b>Derse İşlenişi:</b>	<ol style="list-style-type: none"> <li>1. Lider tahtasında gerekli düzenlemeler yapılır. Rozetler dağıtılır.</li> <li>2. Ödev kontrolü yapılır.</li> <li>3. Matematikte kümenin ne olduğu öğrencilere sorulur ve öğrencilerin cevaplaması istenir.</li> <li>4. Kümelerin nasıl tanımlanacağı örnekler üzerinden anlatılır ve öğrencilerinde kodlaması istenir.</li> <li>5. Küme metotları sırasıyla anlatılır. Anlatılan metotlarla ilgili örnek kodlamalar yapılır. Öğrencilerinde örnek kodlamaları yapmaları istenir. <ul style="list-style-type: none"> <li>a. clear()</li> <li>b. add()</li> <li>c. update()</li> <li>d. difference()</li> <li>e. discard()</li> <li>f. intersection()</li> <li>g. isdisjoint()</li> <li>h. issubset()</li> <li>i. issuperset()</li> <li>j. union()</li> <li>k. symmetric_difference()</li> </ul> </li> </ol>

	<p>6. Dondurulmuş kümenin ne olduğu anlatılır ve örnek durum kodlanır.</p> <p>7. <b>Metot Çek Kodla</b> isimli eğlenceli etkinlik uygulanır.</p>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Küme veri tipi ve metotlarıyla ilgili 3 sorudan oluşan kodlama deneyimi etkinliği uygulanır.

## Ek 10. ÇÖÖ Ders Planları

Pilot Uygulama 1. Hafta Ders Planı	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	Python Giriş
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Çevrimiçi Öğrenme
<b>Araç gereçler</b>	Çevrimiçi Öğrenme Ortamı
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	<ol style="list-style-type: none"><li>1. Python programlama dilini bilir.</li><li>2. Python programlama dilinin avantajlarını bilir.</li><li>3. Python kurulumunu yapabilir.</li><li>4. Python kod yazma ortamlarını bilir.</li></ol>
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"><li>• Öğrencilerin işbirliği ortamında çalışabilmeleri için işbirliği forumu hazırlandı.</li><li>• Haftanın ödevi hazırlanarak sisteme yüklendi.</li><li>• Ders kazanımları çevrimiçi öğrenme ortamına yüklendi.</li><li>• Ders öncesinde kaynak taraması yapılarak ders notları hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li><li>• Python giriş konusuyla ilgili ders videosu hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li><li>• Ölçme ve değerlendirme amacıyla kullanılacak 6 sorudan oluşan kodlama deneyimi etkinliği hazırlanarak sisteme yüklendi</li></ul>
<b>BÖLÜM IV</b>	
<b>Derse İşleniş:</b>	<ol style="list-style-type: none"><li>1. Öğrenciler öncelikli olarak haftanın ödevini yaparak derse başlayacaktır. Haftanın ödevi yapıp gönderildikten sonra ders izlencesi aktif hale gelecektir.</li><li>2. Öğrenciler ders izlencesinde yer alan ders kazanımlarını okuyarak derse başlayacaktır. Kazanımlar okunduktan sonra</li></ol>

	<p>python giriş ders notları öğrenciler için sistemde aktif hale gelecektir.</p> <p><b>3.</b> Python giriş konusuyla ilgili öğrencilere sunulan ders notları öğrenciler tarafından ayrıntılı olarak incelenecektir. Öğrenciler ders notlarını inceledikten sonra python giriş ders videosu öğrencilerin kullanımına açılacaktır.</p> <p><b>4.</b> Ders anlatım videosunda python ne olduğu, avantajları, kurulumu, diğer kod yazma ortamları, aritmetiksel operatörler ve kaçış ifadeleri konuları öğrencilere anlatılacaktır. Ayrıca konuyla alakalı örnek kodlamalar yapılacaktır. Ders videosunu tamandıktan sonra ders konusuyla ilgili hazırlanan kodlama deneyimi etkinliği öğrencilere açılacaktır. Kodlama deneyimi etkinliğine giriş yapabilmek için videoya yerleştirilen şifrenin öğrenciler tarafından girilmesi gerekecektir.</p> <p><b>5.</b> Öğrenciler haftanın konusuyla ilgili işbirliği forumunda bilgi alışverişinde bulunabileceklerdir. Ayrıca öğrenme ortamıyla ilgili karşılaştıkları sorunlarla ilgili sıkça sorulan sorular kısmından bilgi alabileceklerdir.</p>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Python giriş konusuyla ilgili 6 sorudan oluşan kodlama deneyimi etkinliği öğrencilerin şifreyi girmesiyle açılacaktır. Haftanın konusuyla alakalı sorular öğrenciler tarafından tamamlanacaktır.

<b>Pilot Uygulama 2. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	Karakter Dizileri, Aritmetiksel Operatörler, Kaçış İfadeleri
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Çevrimiçi Öğrenme
<b>Araç gereçler</b>	Çevrimiçi Öğrenme Ortamı



<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	<ol style="list-style-type: none"> <li>1. Karakter dizilerinin ne olduğunu bilir.</li> <li>2. Karakter dizileri üzerinde birleştirme ve çoğaltma işlemlerinin nasıl yapılacağını kavrar.</li> <li>3. Type() fonksiyonunu kullanabilir.</li> <li>4. Aritmetik işlem operatörlerini kullanabilir.</li> <li>5. Kaçış ifadelerini kullanabilir.</li> </ol>
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Öğrencilerin işbirliği ortamında çalışabilmeleri için işbirliği forumu hazırlandı.</li> <li>• Haftanın ödevi hazırlanarak sisteme yüklendi.</li> <li>• Ders kazanımları çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Ders öncesinde kaynak taraması yapılarak ders notları hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Karakter dizileri konusuyla ilgili ders videosu hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Ölçme ve değerlendirme amacıyla kullanılacak 4 sorudan oluşan kodlama deneyimi etkinliği hazırlandı. Hazırlanan etkinlik Moodle eklentilerinden olan Code Runner kullanılarak öğrencilerin kod yazmalarına olanak sağlayacak hale getirildi.</li> </ul>
<b>BÖLÜM IV</b>	
<b>Derse İşlenişi:</b>	<ol style="list-style-type: none"> <li>1. Öğrenciler öncelikli olarak haftanın ödevini yaparak derse başlayacaktır. Haftanın ödevi yapıp gönderildikten sonra ders izlencesi aktif hale gelecektir.</li> <li>2. Öğrenciler ders izlencesinde yer alan ders kazanımlarını okuyarak derse başlayacaktır. Kazanımlar okunduktan sonra karakter dizileri ders notları öğrenciler için sistemde aktif hale gelecektir.</li> <li>3. Karakter dizileriyle ilgili sunulan ders notları öğrenciler tarafından ayrıntılı olarak incelenecektir. Öğrenciler ders notlarını inceledikten sonra ders anlatımı videosu öğrencilerin kullanımına açılacaktır.</li> </ol>

	<p>4. Ders anlatım videosunda karakter dizileri konusu örnek uygulamalar sunularak anlatılmıştır. Ders videosunu tamamlandıktan sonra ders konusuyla ilgili hazırlanan kodlama deneyimi etkinliği öğrencilere açılacaktır. Kodlama deneyimi etkinliğine giriş yapabilmek için videoya yerleştirilen şifrenin öğrenciler tarafından girilmesi gerekecektir.</p>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	<p>Python giriş konusuyla ilgili 4 sorudan oluşan kodlama deneyimi etkinliği öğrencilerin şifreyi girmesiyle açılacaktır. Haftanın konusuyla alakalı sorular öğrenciler tarafından tamamlanacaktır. Öğrenciler Moodle eklentisi olan Code Runner ortamında yer alan sorular kodlanacaktır.</p>

<b>Uygulama 1. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	Değişkenler ve Sabitler
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Çevrimiçi Öğrenme
<b>Araç gereçler</b>	Çevrimiçi Öğrenme Ortamı
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	<ol style="list-style-type: none"> <li>1. Değişken tanımlama kurallarını bilir.</li> <li>2. Temel veri tiplerini kavrar</li> <li>3. Değişken ve temel veri tiplerini kullanarak programlar geliştirir.</li> </ol>
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Öğrencilerin işbirliği ortamında çalışabilmeleri için işbirliği forumu hazırlandı.</li> </ul>

	<ul style="list-style-type: none"> <li>• Haftanın ödevi hazırlanarak sisteme yüklendi.</li> <li>• Ders kazanımları çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Ders öncesinde kaynak taraması yapılarak ders notları hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Değişkenler konusu ders videosu hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Ölçme ve değerlendirme amacıyla kullanılacak 4 sorudan oluşan kısa sınav hazırlandı. Hazırlanan kısa sınav Moodle eklentilerinden olan Code Runner kullanılarak sınav haline dönüştürüldü.</li> </ul>
<b>BÖLÜM IV</b>	
<b>Derse İşlenişi:</b>	<ol style="list-style-type: none"> <li>1. Öğrenciler öncelikli olarak haftanın ödevini yaparak derse başlayacaktır. Haftanın ödevi yapıp gönderildikten sonra ders izlencesi aktif hale gelecektir.</li> <li>2. Öğrenciler ders izlencesinde yer alan ders kazanımlarını okuyarak derse başlayacaktır. Kazanımlar okunduktan sonra ders notları öğrenciler için sistemde aktif hale gelecektir.</li> <li>3. Değişkenlerle ilgili öğrencilere sunulan ders notları öğrenciler tarafından ayrıntılı olarak incelenecektir. Öğrenciler ders notlarını inceledikten sonra ders anlatımı videosu öğrencilerin kullanımına açılacaktır.</li> <li>4. Ders anlatım videosunda değişkenler temel veri tipleri konuları öğrencilere anlatılacaktır. Ayrıca değişken tanımlama ve veri tipleriyle ilgili örnek uygulamalar yapılacaktır. Ders videosunu tamamlandıktan sonra ders konusuyla ilgili hazırlanan kodlama deneyimi etkinliği öğrencilere açılacaktır. Kodlama deneyimi etkinliğine giriş yapabilmek için videoya yerleştirilen şifrenin öğrenciler tarafından girilmesi gerekecektir.</li> </ol>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Değişkenler ve temel veri tipleri konusuyla ilgili 4 sorudan oluşan kodlama deneyimi etkinliği uygulanacaktır. Etkinlik 4 sorudan oluşmaktadır. Moodle eklentisi olan Code Runner ortamında sorulan sorular öğrenciler tarafından kodlanacaktır.

<b>Uygulama 2. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	1. Print () fonksiyonunun kullanımı 2. Print () fonksiyonunun parametreleri
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Çevrimiçi Öğrenme
<b>Araç gereçler</b>	Çevrimiçi Öğrenme Ortamı
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	1. Print() fonksiyonunu kullanarak programlar geliştirir. 2. Print() fonksiyonunun parametrelerini bilir. 3. Print() fonksiyonunun parametrelerini kullanarak programlar geliştirir.
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Öğrencilerin işbirliği ortamında çalışabilmeleri için işbirliği forumu hazırlandı.</li> <li>• Haftanın ödevi hazırlanarak sisteme yüklendi.</li> <li>• Ders kazanımları çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Print() fonksiyonu ve parametreleriyle ilgili kaynak taraması yapılarak ders notları hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Print() fonksiyonu ve parametreleriyle ilgili ders videosu hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Ölçme ve değerlendirme amacıyla kullanılacak 4 sorudan oluşan kodlama deneyimi etkinliği hazırlandı. Hazırlanan etkinlik Moodle eklentilerinden olan Code Runner kullanılarak sınav haline dönüştürüldü.</li> </ul>

<b>BÖLÜM IV</b>	
<b>Derse İşlenişi:</b>	<p>1. Öğrenciler öncelikli olarak haftanın ödevini yaparak derse başlayacaktır. Haftanın ödevi yapıp gönderildikten sonra ders izlencesi aktif hale gelecektir.</p> <p>2. Öğrenciler ders izlencesinde yer alan ders kazanımlarını okuyarak derse başlayacaktır. Kazanımlar okunduktan sonra ders notları öğrenciler için sistemde aktif hale gelecektir.</p> <p>3. Print() fonksiyonu ve parametreleriyle ilgili sunulan ders notları öğrenciler tarafından ayrıntılı olarak incelenecektir. Öğrenciler ders notlarını inceledikten sonra ders anlatımı videosu öğrencilerin kullanımına açılacaktır.</p> <p>4. Print() fonksiyonu ve parametreleri ders videosu öğrenciler tarafından seyredilecektir. Örnek uygulamalar öğrenciler tarafından takip edilecektir. Ders videosunu tamamlandıktan sonra ders konusuyla ilgili hazırlanan kodlama deneyimi etkinliği öğrencilere açılacaktır. Kodlama deneyimi etkinliğine giriş yapabilmek için videoya yerleştirilen şifrenin öğrenciler tarafından girilmesi gerekecektir.</p>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Print() fonksiyonu ve parametreleri konusuyla ilgili 4 sorudan oluşan kodlama deneyimi etkinliği uygulanacaktır. Moodle eklentisi olan Code Runner ortamında sunulan sorular öğrenciler tarafından kodlanacaktır.

<b>Uygulama 3. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	<p>Kontrol Yapıları</p> <ol style="list-style-type: none"> <li>1. If kontrol yapısı</li> <li>2. If-Elif kontrol yapısı</li> <li>3. If-Else kontrol yapısı</li> </ol>

	<b>4. If-Elif-Else kontrol yapısı</b>
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Çevrimiçi Öğrenme
<b>Araç gereçler</b>	Çevrimiçi Öğrenme Ortamı
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	<ol style="list-style-type: none"> <li>1. İç içe kontrol yapılarını ifade etmeyi kavrar.</li> <li>2. Kontrol yapılarını kullanarak programlar geliştirir.</li> </ol>
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Öğrencilerin işbirliği ortamında çalışabilmeleri için işbirliği forumu hazırlandı.</li> <li>• Haftanın ödevi hazırlanarak sisteme yüklendi.</li> <li>• Kontrol yapılarıyla ilgili ders kazanımları çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Kontrol yapılarıyla ilgili kaynak taraması yapılarak ders notları hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Kontrol yapılarıyla ilgili ders videosu hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Ölçme ve değerlendirme amacıyla kullanılacak 3 sorudan oluşan kodlama deneyimi etkinliği hazırlandı. Hazırlanan etkinlik Moodle eklentilerinden olan Code Runner kullanılarak sınav haline dönüştürüldü.</li> </ul>
<b>BÖLÜM IV</b>	
<b>Derse İşlenişi:</b>	<ol style="list-style-type: none"> <li>1. Öğrenciler kontrol yapıları dersine başlamadan print() fonksiyonu ve parametreleriyle ilgili verilen ödevi tamamlayacaktır. Ödev tamamlandıktan sonra kontrol yapıları kazanımları aktif hale gelecektir.</li> <li>2. Kontrol yapılarla ilgili ders kazanımları okunacaktır. Kazanımlar okunduktan sonra ders notları öğrenciler için sistemde aktif hale gelecektir.</li> <li>3. Kontrol yapılarla ilgili sunulan ders notları öğrenciler tarafından ayrıntılı olarak incelenecektir. Öğrenciler ders notlarını</li> </ol>

	<p>inceledikten sonra ders anlatımı videosu öğrencilerin kullanımına açılacaktır.</p> <p>4. Kontrol yapıları ders videosu öğrenciler tarafından seyredilecektir. Örnek uygulamalar öğrenciler tarafından takip edilecektir. Ders videosu tamamlandıktan sonra kontrol yapılarıyla ilgili hazırlanan kodlama deneyimi etkinliği öğrencilere açılacaktır. Kodlama deneyimi etkinliğine giriş yapabilmek için videoya yerleştirilen şifrenin öğrenciler tarafından girilmesi gerekecektir.</p>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Kontrol yapıları konusuyla ilgili 3 sorudan oluşan kodlama deneyimi etkinliği uygulanacaktır. Öğrenciler Moodle eklentisi olan Code Runner ortamında sunulan sorular öğrenciler tarafından kodlanacaktır.

<b>Uygulama 4. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	<p>Tekrarlı Yapılar (Döngüler)</p> <ol style="list-style-type: none"> <li>1. Tekrarlı yapıların kullanım amacı</li> <li>2. For tekrarlı yapısının kullanımı</li> <li>3. While tekrarlı yapısının kullanımı</li> <li>4. Tekrarlı yapılarda kullanılan yapılar (range, pass, break, continue)</li> </ol>
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Çevrimiçi Öğrenme
<b>Araç gereçler</b>	Çevrimiçi Öğrenme Ortamı
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	<ol style="list-style-type: none"> <li>1. Tekrarlı yapıları kavrar.</li> <li>2. Tekrarlı yapılarda kullanılan fonksiyonları kavrar.</li> </ol>

	3. Tekrarlı yapıları ve tekrarlı yapılarda kullanılan fonksiyonları kullanarak programlar geliştirir.
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Öğrencilerin işbirliği ortamında çalışabilmeleri için işbirliği forumu hazırlandı.</li> <li>• Haftanın ödevi hazırlanarak sisteme yüklendi.</li> <li>• Tekrarlı yapılarla ilgili ders kazanımları çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Tekrarlı yapılarla ilgili kaynak taraması yapılarak ders notları hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Tekrarlı yapılarla ilgili ders videosu hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Ölçme ve değerlendirme amacıyla kullanılacak 3 sorudan oluşan kodlama deneyimi etkinliği hazırlandı. Hazırlanan etkinlik Moodle eklentilerinden olan Code Runner kullanılarak sınav haline dönüştürüldü.</li> </ul>
<b>BÖLÜM IV</b>	
<b>Derse İşlenişi:</b>	<ol style="list-style-type: none"> <li>1. Öğrenciler tekrarlı yapılar dersine başlamadan kontrol yapılarıyla ilgili verilen ödevi tamamlayacaktır. Ödev tamamlandıktan sonra tekrarlı yapılar kazanımları aktif hale gelecektir.</li> <li>2. Tekrarlı yapılarla ilgili ders kazanımları okunacaktır. Kazanımlar okunduktan sonra ders notları öğrenciler için sistemde aktif hale gelecektir.</li> <li>3. Tekrarlı yapılarla ilgili sunulan ders notları öğrenciler tarafından ayrıntılı olarak incelenecektir. Öğrenciler ders notlarını inceledikten sonra ders anlatımı videosu öğrencilerin kullanımına açılacaktır.</li> <li>4. Tekrarlı yapılar ders videosu öğrenciler tarafından seyredilecektir. Örnek uygulamalar öğrenciler tarafından takip edilecektir. Ders videosu tamamlandıktan sonra kontrol yapılarıyla ilgili hazırlanan kodlama deneyimi etkinliği öğrencilere açılacaktır.</li> </ol>



	5. Kodlama deneyimi etkinliğine öğrenciler videoya yerleştirilen şifreyi girerek erişebileceklerdir.
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Tekrarlı yapılar konusuyla ilgili 3 sorudan oluşan kodlama deneyimi etkinliği uygulanacaktır. Öğrenciler Moodle eklentisi olan Code Runner ortamında sunulan sorular öğrenciler tarafından kodlanacaktır.

<b>Uygulama 5. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	Fonksiyonlar 1. Fonksiyon kavramı ve türleri 2. Fonksiyon tanımlama 3. Parametre almayan fonksiyonların kullanımı
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Çevrimiçi Öğrenme
<b>Araç gereçler</b>	Çevrimiçi öğrenme ortamı
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	1. Fonksiyonun ne olduğunu bilir. 2. Fonksiyon türlerini bilir. 3. Fonksiyonun nasıl tanımlanacağını kavrar. 4. Parametre almayan fonksiyonları kullanıldığı programlar geliştirir.
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Öğrencilerin işbirliği ortamında çalışabilmeleri için işbirliği forumu hazırlandı.</li> <li>• Bir önceki haftanın konusuyla ilgili hazırlanan ödev sisteme yüklendi.</li> </ul>

	<ul style="list-style-type: none"> <li>• Fonksiyonlarla (parametre almayan) ilgili ders kazanımları çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Fonksiyonlarla (parametre almayan) ilgili kaynak taraması yapılarak ders notları hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Fonksiyonlarla (parametre almayan) ilgili ders videosu hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Ölçme ve değerlendirme amacıyla kullanılacak 3 sorudan oluşan kodlama deneyimi etkinliği hazırlandı. Hazırlanan etkinlik Moodle eklentilerinden olan Code Runner kullanılarak sınav haline dönüştürüldü.</li> </ul>
<b>BÖLÜM IV</b>	
<b>Derse İşlenişi:</b>	<ol style="list-style-type: none"> <li>1. Öğrenciler fonksiyonlar (parametre almayan) dersine başlamadan tekrarlı yapılarla ilgili verilen ödevi tamamlayacaktır. Ödev tamamlandıktan sonra fonksiyonlar (parametre almayan) kazanımları aktif hale gelecektir.</li> <li>2. Fonksiyonlarla (parametre almayan) ilgili ders kazanımları okunacaktır. Kazanımlar okunduktan sonra ders notları öğrenciler için sistemde aktif hale gelecektir.</li> <li>3. Fonksiyonlarla (parametre almayan) ilgili sunulan ders notları öğrenciler tarafından ayrıntılı olarak incelenecektir. Öğrenciler ders notlarını inceledikten sonra ders anlatımı videosu öğrencilerin kullanımına açılacaktır.</li> <li>4. Fonksiyonlar (parametre almayan) ders videosu öğrenciler tarafından seyredilecektir. Örnek uygulamalar öğrenciler tarafından takip edilecektir. Ders videosu tamamlandıktan sonra fonksiyonlarla (parametre almayan) ilgili hazırlanan kodlama deneyimi etkinliği öğrencilere açılacaktır.</li> <li>5. Kodlama deneyimi etkinliğine öğrenciler videoya yerleştirilen şifreyi girerek erişebileceklerdir.</li> </ol>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Fonksiyonlar (parametre almayan) konusuyla ilgili 3 sorudan oluşan kodlama deneyimi etkinliği uygulanacaktır. Öğrenciler

	Moodle eklentisi olan Code Runner ortamında sunulan sorular öğrenciler tarafından kodlanacaktır.
--	--

<b>Uygulama 6. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	Fonksiyonlar 1. Parametre alan fonksiyonlar 2. İsimli ve sıralı argümanlar 3. Varsayılan değerli argümanlar
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Çevrimiçi Öğrenme
<b>Araç gereçler</b>	Çevrimiçi Öğrenme Ortamı
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	1. Parametre alan fonksiyonların kullanıldığı programlar geliştirir. 2. İsimli ve sıralı argümanların kullanıldığı programlar geliştirir. 3. Varsayılan argümanların kullanıldığı programlar geliştirir.
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Öğrencilerin işbirliği ortamında çalışabilmeleri için işbirliği forumu hazırlandı.</li> <li>• Haftanın ödevi hazırlanarak sisteme yüklendi.</li> <li>• Fonksiyonlarla (parametre alan) ilgili ders kazanımları çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Fonksiyonlarla (parametre alan) ilgili kaynak taraması yapılarak ders notları hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Fonksiyonlarla (parametre alan) ilgili ders videosu hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Ölçme ve değerlendirme amacıyla kullanılacak 3 sorudan oluşan kodlama deneyimi etkinliği hazırlandı. Hazırlanan etkinlik</li> </ul>

	Moodle eklentilerinden olan Code Runner kullanılarak sınav haline dönüştürüldü.
<b>BÖLÜM IV</b>	
<b>Derse İşlenişi:</b>	<p>1. Öğrenciler fonksiyonlar (parametre alan) dersine başlamadan fonksiyonlarla (parametre almayan) ilgili verilen ödevi tamamlayacaktır. Ödev tamamlandıktan sonra fonksiyonlar (parametre alan) kazanımları aktif hale gelecektir.</p> <p>2. Fonksiyonlarla (parametre alan) ilgili ders kazanımları okunacaktır. Kazanımlar okunduktan sonra ders notları öğrenciler için sistemde aktif hale gelecektir.</p> <p>3. Fonksiyonlarla (parametre alan) ilgili sunulan ders notları öğrenciler tarafından ayrıntılı olarak incelenecektir. Öğrenciler ders notlarını inceledikten sonra ders anlatımı videosu öğrencilerin kullanımına açılacaktır.</p> <p>4. Fonksiyonlar (parametre alan) ders videosu öğrenciler tarafından seyredilecektir. Örnek uygulamalar öğrenciler tarafından takip edilecektir. Ders videosu tamamlandıktan sonra fonksiyonlar (parametre alan) ilgili hazırlanan kodlama deneyimi etkinliği öğrencilere açılacaktır.</p> <p>5. Öğrenciler kodlama deneyimi etkinliğine videoya yerleştirilen şifreyi girerek erişebileceklerdir.</p>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Fonksiyonlar (parametre alan) konusuyla ilgili 3 sorudan oluşan kodlama deneyimi etkinliği uygulanacaktır. Moodle eklentisi olan Code Runner ortamında sunulan sorular öğrenciler tarafından kodlanacaktır

<b>Uygulama 7. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	Fonksiyonlar

	<ol style="list-style-type: none"> <li>1. Değer döndüren fonksiyonlar <ol style="list-style-type: none"> <li>a. Global ifadesi</li> <li>b. Return ifadesi</li> <li>c. Pass ifadesi</li> </ol> </li> <li>2. İstenen sayıda sıralı argümanlar</li> <li>3. İstenen sayıda isimli argümanlar</li> </ol>
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Çevrimiçi Öğrenme
<b>Araç gereçler</b>	Çevrimiçi Öğrenme Ortamı
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	<ol style="list-style-type: none"> <li>1. Değer döndüren fonksiyonların kullanıldığı programlar geliştirir.</li> <li>2. Çok sayıda sıralı ve isimli argümanlı fonksiyonların kullanıldığı programlar geliştirir.</li> </ol>
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Öğrencilerin işbirliği ortamında çalışabilmeleri için işbirliği forumu hazırlandı.</li> <li>• Haftanın ödevi hazırlanarak sisteme yüklendi.</li> <li>• Fonksiyonlarla (değer döndüren ve argümanlı) ilgili ders kazanımları çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Fonksiyonlarla (değer döndüren ve argümanlı) ilgili kaynak taraması yapılarak ders notları hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Fonksiyonlarla (değer döndüren ve argümanlı) ilgili ders videosu hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Ölçme ve değerlendirme amacıyla kullanılacak 4 sorudan oluşan kodlama deneyimi etkinliği hazırlandı. Hazırlanan etkinlik Moodle eklentilerinden olan Code Runner kullanılarak sınav haline dönüştürüldü.</li> </ul>
<b>BÖLÜM IV</b>	
<b>Derse İşleniş:</b>	<ol style="list-style-type: none"> <li>1. Öğrenciler fonksiyonlar (değer döndüren ve argümanlı) dersine başlamadan fonksiyonlarla (parametre alan) ilgili verilen ödevi</li> </ol>

	<p>tamamlayacaktır. Ödev tamamlandıktan sonra fonksiyonlar (değer döndüren ve argümanlı) kazanımları aktif hale gelecektir.</p> <p><b>2.</b> Fonksiyonlarla (değer döndüren ve argümanlı) ilgili ders kazanımları okunacaktır. Kazanımlar okunduktan sonra ders notları öğrenciler için sistemde aktif hale gelecektir.</p> <p><b>3.</b> Fonksiyonlarla (değer döndüren ve argümanlı) ilgili sunulan ders notları öğrenciler tarafından ayrıntılı olarak incelenecektir. Öğrenciler ders notlarını inceledikten sonra ders anlatımı videosu öğrencilerin kullanımına açılacaktır.</p> <p><b>4.</b> Fonksiyonlar(değer döndüren ve argümanlı) ders videosu öğrenciler tarafından seyredilecektir. Örnek uygulamalar öğrenciler tarafından takip edilecektir. Ders videosu tamamlandıktan sonra kontrol yapılarıyla ilgili hazırlanan kodlama deneyimi etkinliği öğrencilere açılacaktır.</p> <p><b>5.</b> Kodlama deneyimi etkinliğine öğrenciler videoya yerleştirilen şifreyi girerek erişebileceklerdir.</p>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	<p>Fonksiyonlar (değer döndüren ve argümanlı) konusuyla ilgili 4 sorudan oluşan kodlama deneyimi etkinliği uygulanacaktır. Moodle eklentisi olan Code Runner ortamında sunulan sorular öğrenciler tarafından kodlanacaktır</p>

<b>Uygulama 8. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	<ol style="list-style-type: none"> <li>1. Liste tanımlama</li> <li>2. Listelerde kullanılan metotlar</li> <li>3. List () fonksiyonu ile tam sayı liste oluşturma</li> <li>4. * operatörü ile liste oluşturma</li> </ol>
<b>Süre</b>	2 Ders Saati

<b>Öğretim Yöntem ve Teknikleri</b>	Çevrimiçi Öğrenme
<b>Araç gereçler</b>	Çevrimiçi Öğrenme Ortamı
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	<ol style="list-style-type: none"> <li>1. Liste türünde veri tanımlamayı bilir.</li> <li>2. Liste veri tipini ve metotlarını kullanarak programlar geliştirir.</li> </ol>
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Öğrencilerin işbirliği ortamında çalışabilmeleri için işbirliği forumu hazırlandı.</li> <li>• Haftanın ödevi hazırlanarak sisteme yüklendi.</li> <li>• Listeler ilgili ders kazanımları çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Listelerle ilgili kaynak taraması yapılarak ders notları hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Listelerle ilgili ders videosu hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Ölçme ve değerlendirme amacıyla kullanılacak 3 sorudan oluşan kodlama deneyimi etkinliği hazırlandı. Hazırlanan etkinlik Moodle eklentilerinden olan Code Runner kullanılarak sınav haline dönüştürüldü.</li> </ul>
<b>BÖLÜM IV</b>	
<b>Derse İşlenişi:</b>	<ol style="list-style-type: none"> <li>1. Öğrenciler listeler dersine başlamadan fonksiyonlar (değer döndüren ve argümanlı) ilgili verilen ödevi tamamlayacaktır. Ödev tamamlandıktan sonra listeler kazanımları aktif hale gelecektir.</li> <li>2. Listeler ilgili ders kazanımları okunacaktır. Kazanımlar okunduktan sonra ders notları öğrenciler için sistemde aktif hale gelecektir.</li> <li>3. Listelerle ilgili sunulan ders notları öğrenciler tarafından ayrıntılı olarak incelenecektir. Öğrenciler ders notlarını inceledikten sonra ders anlatımı videosu öğrencilerin kullanımına açılacaktır.</li> </ol>

	<p>4. Listeler ders videosu öğrenciler tarafından seyredilecektir. Örnek uygulamalar öğrenciler tarafından takip edilecektir. Ders videosu tamamlandıktan sonra kontrol yapılarıyla ilgili hazırlanan kodlama deneyimi etkinliği öğrencilere açılacaktır.</p> <p>5. Kodlama deneyimi etkinliğine öğrenciler videoya yerleştirilen şifreyi girerek erişebileceklerdir.</p>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Listeler konusuyla ilgili 3 sorudan oluşan kodlama deneyimi uygulanacaktır. Moodle eklentisi olan Code Runner ortamında sunulan sorular öğrenciler tarafından kodlanacaktır

<b>Uygulama 9. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	<ol style="list-style-type: none"> <li>1. List() fonksiyonu</li> <li>2. Liste öğelerine erişme</li> <li>3. Liste öğelerini değiştirme</li> <li>4. Liste öğelerini birleştirme</li> <li>5. Demetler</li> </ol>
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Çevrimiçi Öğrenme
<b>Araç gereçler</b>	Çevrimiçi Öğrenme Ortamı
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	<ol style="list-style-type: none"> <li>1. Liste veri tipini ve metotlarını kullanarak programlar geliştirir.</li> <li>2. Demet veri tipini ve metotlarını kullanarak programlar geliştirir.</li> </ol>
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Öğrencilerin işbirliği ortamında çalışabilmeleri için işbirliği forumu hazırlandı.</li> <li>• Haftanın ödevi hazırlanarak sisteme yüklendi.</li> </ul>



	<ul style="list-style-type: none"> <li>• Liste metotları ve demetler ders kazanımları çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Liste metotları ve demetlerle ilgili kaynak taraması yapılarak ders notları hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Liste metotları ve demetlerle ilgili ders videosu hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Ölçme ve değerlendirme amacıyla kullanılacak 3 sorudan oluşan kodlama deneyimi etkinliği hazırlandı. Hazırlanan etkinlik Moodle eklentilerinden olan Code Runner kullanılarak sınav haline dönüştürüldü.</li> </ul>
<b>BÖLÜM IV</b>	
<b>Derse İşlenişi:</b>	<ol style="list-style-type: none"> <li>1. Öğrenciler Liste metotları ve demetler dersine başlamadan listelerle ilgili verilen ödevi tamamlayacaktır. Ödev tamamlandıktan sonra Liste metotları ve demetler aktif hale gelecektir.</li> <li>2. Liste metotları ve demetler ilgili ders kazanımları okunacaktır. Kazanımlar okunduktan sonra ders notları öğrenciler için sistemde aktif hale gelecektir.</li> <li>3. Liste metotları ve demetler ilgili sunulan ders notları öğrenciler tarafından ayrıntılı olarak incelenecektir. Öğrenciler ders notlarını inceledikten sonra ders anlatımı videosu öğrencilerin kullanımına açılacaktır.</li> <li>4. Liste metotları ve demetler ders videosu öğrenciler tarafından seyredilecektir. Örnek uygulamalar öğrenciler tarafından takip edilecektir. Ders videosu tamamlandıktan sonra kontrol yapılarıyla ilgili hazırlanan kodlama deneyimi etkinliği öğrencilere açılacaktır.</li> <li>5. Kodlama deneyimi etkinliğine öğrenciler videoya yerleştirilen şifreyi girerek erişebileceklerdir.</li> </ol>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Liste metotları ve demetler konusuyla ilgili 3 sorudan oluşan kodlama deneyimi etkinliği uygulanacaktır. Moodle eklentisi olan

	Code Runner ortamında sunulan sorular öğrenciler tarafından kodlanacaktır
--	---

<b>Uygulama 10. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	<p>1. Küme veri tipi</p> <p>2. Küme veri tipi metotları</p> <ul style="list-style-type: none"> <li>a. clear()</li> <li>b. add()</li> <li>c. update()</li> <li>d. difference()</li> <li>e. discard()</li> <li>f. intersection()</li> <li>g. isdisjoint()</li> <li>h. issubset()</li> <li>i. issuperset()</li> <li>j. union()</li> <li>k. symmetric_difference()</li> </ul> <p>3. Dondurulmuş kümeler</p>
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Çevrimiçi Öğrenme
<b>Araç gereçler</b>	Çevrimiçi Öğrenme Ortamı
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	<p>1. Küme türünde veri tanımlamayı ve metotlarını bilir.</p> <p>2. Küme veri tipini ve metotlarını kullanarak programlar geliştirir.</p>
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Öğrencilerin işbirliği ortamında çalışabilmeleri için işbirliği forumu hazırlandı.</li> <li>• Haftanın ödevi hazırlanarak sisteme yüklendi.</li> </ul>

	<ul style="list-style-type: none"> <li>• Küme veri tipi ders kazanımları çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Küme veri tipiyle ilgili kaynak taraması yapılarak ders notları hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Küme veri tipiyle ilgili ders videosu hazırlandı ve çevrimiçi öğrenme ortamına yüklendi.</li> <li>• Ölçme ve değerlendirme amacıyla kullanılacak 3 sorudan oluşan kodlama deneyimi etkinliği hazırlandı. Hazırlanan etkinlik Moodle eklentilerinden olan Code Runner kullanılarak sınav haline dönüştürüldü.</li> </ul>
<b>BÖLÜM IV</b>	
<b>Derse İşlenişi:</b>	<ol style="list-style-type: none"> <li>1. Öğrenciler küme veri tipi dersine başlamadan liste metotları ve demetler ilgili verilen ödevi tamamlayacaktır. Ödev tamamlandıktan sonra Liste metotları ve demetler aktif hale gelecektir.</li> <li>2. Küme veri tipiyle ilgili ders kazanımları okunacaktır. Kazanımlar okunduktan sonra ders notları öğrenciler için sistemde aktif hale gelecektir.</li> <li>3. Küme veri tipiyle ilgili sunulan ders notları öğrenciler tarafından ayrıntılı olarak incelenecektir. Öğrenciler ders notlarını inceledikten sonra ders anlatımı videosu öğrencilerin kullanımına açılacaktır.</li> <li>4. Küme veri tipiyle ders videosu öğrenciler tarafından seyredilecektir. Örnek uygulamalar öğrenciler tarafından takip edilecektir. Ders videosu tamamlandıktan sonra kontrol yapılarıyla ilgili hazırlanan kodlama deneyimi etkinliği öğrencilere açılacaktır.</li> <li>5. Kodlama deneyimi etkinliğine öğrenciler videoya yerleştirilen şifreyi girerek erişebileceklerdir.</li> </ol>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Küme veri tipi konusuyla ilgili 3 sorudan oluşan kodlama deneyimi etkinliği uygulanacaktır. Moodle eklentisi olan Code

	Runner ortamında sunulan sorular öğrenciler tarafından kodlanacaktır
--	--

## Ek 11. YYÖO Ders Planları

Pilot Uygulama 1. Hafta Ders Planı	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	Python Giriş
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Anlatım, Soru-Cevap, Gösterip Yaptırma
<b>Araç gereçler</b>	Bilgisayar, Etkileşimli Tahta, Python Kod Editörü
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	<ol style="list-style-type: none"><li>1. Python programlama dilini bilir.</li><li>2. Python programlama dilinin avantajlarını bilir.</li><li>3. Python kurulumunu yapabilir.</li><li>4. Python kod yazma ortamlarını bilir.</li></ol>
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"><li>• Haftanın ödevi hazırlandı.</li><li>• Ders öncesinde kaynak taraması yapılarak öğrenci çalışma yaprakları hazırlandı.</li><li>• Konuyla ilgili ders sunumu hazırlandı.</li><li>• Python kurulum dosyası hazırlanır.</li><li>• Python kodlarının yazılabileceği diğer ortamlar hazırlanır.</li><li>• Ölçme ve değerlendirme amacıyla kullanılacak 6 sorudan oluşan kodlama deneyimi etkinliği hazırlandı.</li></ul>
<b>BÖLÜM IV</b>	
<b>Derse İşleniş:</b>	<ol style="list-style-type: none"><li>1. Python programla dilini daha önce duyulup duyulmadığı öğrencilere sorulur ve öğrencilerin cevaplaması istenir.</li><li>2. Python tanıtımı yapılarak ne gibi avantajlarının bulunduğu anlatılır. Ayrıca python kullanan şirketlere örnekler verilir.</li><li>3. Python programlama dili öğrencilerle birlikte bilgisayarlara kurulur.</li></ol>

	<p>4. Program kurulduktan sonra kod yazmada kullanılacak IDLE ortamı açılır ve öğrencilerinde açması istenir.</p> <p>5. Python kodlarının yazımında kullanılacak diğer ortamlar (<b>Jupyter-iPython, PyCharm</b>) öğrencilere gösterilir.</p> <p>6. Python'da program dosyasının nasıl açılacağı ve kaydedileceği öğrencilere gösterilir.</p> <p>7. Öğrencilerle birlikte bir program dosyası açılır ve örnek kodlar yazıldıktan sonra kaydedilir. Akabinde yazılan örnek kodlar çalıştırılır. Bu şekilde öğrencilerinde programın nasıl çalıştığını görmesi sağlanır.</p>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Konunun öğrenciler tarafında anlaşılıp anlaşılmadığının kontrolü için 6 sorudan oluşan kısa sınav uygulanır.

<b>Pilot Uygulama 2. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	Karakter Dizileri, Aritmetiksel Operatörler, Kaçış İfadeleri
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Anlatım, Soru-Cevap, Gösterip Yaptırma
<b>Araç gereçler</b>	Bilgisayar, Etkileşimli Tahta, Python Kod Editörü, Çalışma Yaprakları
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	<ol style="list-style-type: none"> <li>1. Karakter dizilerinin ne olduğunu bilir.</li> <li>2. Karakter dizileri üzerinde birleştirme ve çoğaltma işlemlerinin nasıl yapılacağını kavrar.</li> <li>3. Type() fonksiyonunu kullanabilir.</li> <li>4. Aritmetik işlem operatörlerini kullanabilir.</li> <li>5. Kaçış ifadelerini kullanabilir.</li> </ol>

<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Haftanın ödevi hazırlandı.</li> <li>• Ders öncesinde kaynak taraması yapılarak öğrenci çalışma yaprakları hazırlandı.</li> <li>• Konuyla ilgili ders sunumu hazırlandı.</li> <li>• Karakter dizileri, aritmetiksel operatörler ve kaçış ifadeleriyle ilgili öğrencilere sunulacak örnekler hazırlandı.</li> <li>• Ölçme ve değerlendirme amacıyla kullanılacak 4 sorudan oluşan kodlama deneyimi etkinliği hazırlandı.</li> </ul>
<b>BÖLÜM IV</b>	
<b>Derse İşlenişi:</b>	<ol style="list-style-type: none"> <li>1. Haftanın ödevi kontrol edilir.</li> <li>2. Karakter dizisinin ne olduğu öğrencilere sorulur.</li> <li>3. Karakter dizileri anlatılır. Tırnak kullanımı, birleştirme, çoğaltma ve type() fonksiyonunun kullanımı örneklerle anlatılır.</li> <li>4. Karakter dizileriyle ilgili örnekler gösterilir ve öğrencilerinde yapmaları istenir.</li> <li>5. Aritmetiksel operatörler ve kaçış ifadeleri örneklerle öğrencilere sunulur.</li> <li>6. Aritmetiksel operatörler ve kaçış ifadelerinin kullanımıyla ilgili öğrencilere sorular yöneltilir ve öğrencilerin kodlaması istenir.</li> </ol>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Karakter dizileri, aritmetiksel operatörler ve kaçış ifadeleriyle ilgili kodlama deneyimi etkinliği uygulanır.

<b>Uygulama 1. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	Değişkenler ve Sabitler
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Anlatım, Soru-Cevap, Gösterip Yaptırma

<b>Araç gereçler</b>	Bilgisayar, Etkileşimli Tahta, Python Kod Editörü, Çalışma Yaprakları
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	<ol style="list-style-type: none"> <li>1. Değişken tanımlama kurallarını bilir.</li> <li>2. Temel veri tiplerini kavrar</li> <li>3. Değişken ve temel veri tiplerini kullanarak programlar geliştirir.</li> </ol>
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Haftanın ödevi hazırlandı.</li> <li>• Ders öncesinde kaynak taraması yapılarak öğrenci çalışma yaprakları hazırlandı.</li> <li>• Konuyla ilgili ders sunumu hazırlandı.</li> <li>• Değişken tanımlamayla ilgili öğrencilere sunulacak örnekler hazırlandı.</li> <li>• Ölçme ve değerlendirme amacıyla kullanılacak 4 sorudan oluşan kodlama deneyimi etkinliği hazırlandı.</li> </ul>
<b>BÖLÜM IV</b>	
<b>Derse İşlenişi:</b>	<ol style="list-style-type: none"> <li>1. Haftanın ödevi kontrol edilir.</li> <li>2. Değişken kavramının ne olduğu hakkında öğrencilerin fikirlerinin alınmasına yönelik sorular sorulur.</li> <li>3. Anlatım yöntemi kullanılarak değişken kavramı anlatılır. Değişken kavramıyla ilgili olarak anlatılacaklar; <ol style="list-style-type: none"> <li>a. Değişken kavramının ne olduğu,</li> <li>b. Değişken tanımlamada hangi operatörün kullanıldığı,</li> <li>c. Tanımlanan değişkenler üzerinde hangi işlemlerin yapılacağı ve nasıl kullanılacağı şeklindedir.</li> </ol> </li> <li>4. Değişken kavramının daha iyi anlaşılabilmesi için çeşitli değişken tanımlama örnekleri gösterilir.</li> <li>5. Değişken uzunluğunu hesaplamak için kullanılan <i>len()</i> fonksiyonunu nasıl kullanıldığı gösterilir ve öğrencilerinde örneklerde tanımladıkları değişkenlerin uzunluklarını hesaplamaları istenir. Ayrıca birden fazla değişkenin</li> </ol>



	<p>uzunluklarının hesaplanması ve aritmetik olarak işleme tabi tutulması gibi işlemler örneklerle anlatılır.</p> <p><b>6.</b> Değişken tanımlama kuralları anlatılır. Yanlış tanımlanan değişkenler öğrencilere sunulur. Öğrencilerin bu kurallar dikkate alınarak örnek değişken tanımlamaları istenir.</p> <p><b>7.</b> Gösterip yaptırma yöntemi ile öğrencilere değişken tanımlamayla ilgili örnekler verilerek öğrencilerinde değişken tanımlamalarına olanak tanınır.</p> <p><b>8.</b> Temel veri tipleri anlatılır ve her veri tipi için örnekler sunulur. (Integer, Float, Boolean, String, Liste)</p>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Değişkenler ve temel veri tipleri konusuyla ilgili öğrencilerin uygulama yapmalarına olanak tanıyarak konunun pekiştirilmesini sağlayacak 4 sorudan oluşan kodlama deneyimi etkinliği uygulanır. Kısa sınav konu kazanımlarını ölçen örnek durumlara yönelik kodlama etkinliklerini içermektedir.

<b>Uygulama 2. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	<ol style="list-style-type: none"> <li>Print () fonksiyonunun kullanımı</li> <li>Print () fonksiyonunun parametreleri</li> </ol>
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Anlatım, Soru-Cevap, Gösterip Yaptırma
<b>Araç gereçler</b>	Bilgisayar, Etkileşimli Tahta, Python Kod Editörü, Çalışma Yaprakları
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	<ol style="list-style-type: none"> <li>Print() fonksiyonunu kullanarak programlar geliştirir.</li> <li>Print() fonksiyonunun parametrelerini bilir.</li> </ol>

	3. Print() fonksiyonunun parametrelerini kullanarak programlar geliştirir.
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Haftanın ödevi hazırlandı.</li> <li>• Print() fonksiyonunun kullanımıyla ilgili kaynaklar tarandı.</li> <li>• Taranan kaynaklar dikkate alınarak öğrencilere verilmek üzere çalışma kâğıtları hazırlandı.</li> <li>• Ders anlatımında kullanılmak üzere sunum hazırlandı.</li> <li>• Print() fonksiyonunun kullanımıyla ilgili öğrencilere sunulmak üzere örnekler hazırlandı.</li> <li>• Print() fonksiyonuyla ilgili 4 sorudan oluşan kodlama deneyimi etkinliği hazırlandı.</li> </ul>
<b>BÖLÜM IV</b>	
<b>Derse İşlenişi:</b>	<ol style="list-style-type: none"> <li>1. Haftanın ödevi kontrol edildi.</li> <li>2. Print() fonksiyonunun kelime anlamının öğrencilerde ne çağrıştırdığı sorulur ve öğrencilerin cevaplamaları istenir.</li> <li>3. Anlatım yöntemi Print() fonksiyonunu konusu anlatılır. <ol style="list-style-type: none"> <li>a. Print() fonksiyonunun ne amaçla ve nasıl kullanıldığı,</li> <li>b. Print() fonksiyonunun değişkenlerle birlikte kullanımı,</li> <li>c. Print() fonksiyonunda tırnak kullanımları,</li> <li>d. Print() fonksiyonunda birden fazla parametre kullanımı,</li> </ol> </li> <li>4. Print() fonksiyonuyla ilgili örnekler sunulur ve öğrencilerinde verilen örnekleri kodlamaları istenir.</li> <li>5. Print() fonksiyonunun parametreleri anlatılarak parametrelerin nasıl kullanılacağı öğrencilere gösterilir. <ol style="list-style-type: none"> <li>a. Sep parametresi,</li> <li>b. End parametresi,</li> <li>c. File parametresi,</li> <li>d. Flush parametresi,</li> </ol> </li> </ol>

	<p>6. Print() fonksiyonunun parametreleriyle ilgili olarak öğrencilerin benzer kodlar yazmaları sağlanır.</p> <p>7. Print fonksiyonu ve parametreleriyle ilgili örnek durumlar verilerek öğrencilerin kodlamaları istenir.</p>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Print() fonksiyonu ve parametreleriyle ilgili 4 sorudan oluşan kodlama deneyimi etkinliği uygulanır. Öğrencilerin sorularda verilen durumları kodlamaları istenir.

<b>Uygulama 3. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	<p>Kontrol Yapıları</p> <ol style="list-style-type: none"> <li>1. If kontrol yapısı</li> <li>2. If-Elif kontrol yapısı</li> <li>3. If-Else kontrol yapısı</li> <li>4. If-Elif-Else kontrol yapısı</li> </ol>
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Anlatım, Soru-Cevap, Gösterip Yaptırma
<b>Araç gereçler</b>	Bilgisayar, Etkileşimli Tahta, Python Kod Editörü, Çalışma Yaprakları
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	<ol style="list-style-type: none"> <li>1. İç içe kontrol yapılarını ifade etmeyi kavrar.</li> <li>2. Kontrol yapılarını kullanarak programlar geliştirir.</li> </ol>
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Haftanın ödevi hazırlandı.</li> <li>• Kontrol yapılarıyla ilgili kaynaklar tarandı.</li> <li>• Kontrol yapılarıyla ilgili çalışma yaprakları hazırlandı.</li> <li>• Kontrol yapıları sunumu hazırlandı.</li> <li>• Kontrol yapılarıyla ilgili örnek problemler hazırlandı</li> </ul>

	<ul style="list-style-type: none"> <li>• Kontrol yapılarıyla ilgili 3 sorudan oluşan kodlama deneyimi etkinliği hazırlandı</li> </ul>
<b>BÖLÜM IV</b>	
<b>Derse İşlenişi:</b>	<ol style="list-style-type: none"> <li>1. Ödevler kontrol edildi.</li> <li>2. Günlük hayatta karşılaşılan koşullu durumlara örnekler verildi ve öğrencilere örnek vermeleri istendi.</li> <li>3. Anlatım yöntemi kullanılarak <b>If</b> kontrol yapısının kullanımı anlatılır ve bir adet örnek problem durumunun nasıl kodlanacağı öğrenciler gösterilir.</li> <li>4. <b>If-Elif</b> kontrol yapısı anlatılır. If ile if- elif kontrol yapıları arasındaki farkı ortaya koymak amacıyla örnek problem durumunun nasıl kodlanacağı gösterilir.</li> <li>5. <b>If-Else</b> kontrol yapısı anlatılır. If – else kontrol yapısıyla ilgili örnek problem durumunun nasıl kodlanacağı gösterilir.</li> <li>6. <b>If-Elif-Else</b> kontrol yapısı anlatılarak örnek problem durumunun nasıl kodlanacağı gösterilir.</li> <li>7. Örnek problem durumları verilerek öğrencilerin hangi kontrol yapılarını tercih edecekleri sorulur. Tercih edilen kontrol yapılarının neden tercih edildiği tartışılır. Tartışma sonunda en uygun kontrol yapısı kullanılarak problem durumu öğrencilerle birlikte kodlanır.</li> </ol>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Kontrol yapılarının kullanımıyla ilgili 3 sorudan oluşan kodlama deneyimi etkinliği uygulanır.

<b>Uygulama 4. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	Tekrarlı Yapılar (Döngüler) <ol style="list-style-type: none"> <li>1. Tekrarlı yapıların kullanım amacı</li> <li>2. For tekrarlı yapısının kullanımı</li> </ol>

	<p>3. While tekrarlı yapısının kullanımı</p> <p>4. Tekrarlı yapılarda kullanılan yapılar (range, pass, break, continue)</p>
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Anlatım, Soru-Cevap, Gösterip Yaptırma
<b>Araç gereçler</b>	Bilgisayar, Etkileşimli Tahta, Python Kod Editörü, Çalışma Yaprakları
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	<p>1. Tekrarlı yapıları kavrar.</p> <p>2. Tekrarlı yapılarda kullanılan fonksiyonları kavrar.</p> <p>3. Tekrarlı yapıları ve tekrarlı yapılarda kullanılan fonksiyonları kullanarak programlar geliştirir.</p>
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Haftanın ödevi hazırlanır.</li> <li>• Tekrarlı yapılarla ilgili kaynaklar taranır.</li> <li>• Çalışma yaprağı hazırlanır.</li> <li>• Tekrarlı yapıların anlatımı için ders sunumu hazırlanır.</li> <li>• Örnek problem durumları hazırlanır</li> <li>• Tekrarlı yapılarla ilgili 3 sorudan oluşan kodlama deneyimi etkinliği hazırlanır.</li> </ul>
<b>BÖLÜM IV</b>	
<b>Derse İşlenişi:</b>	<p>1. Ödevler kontrol edilir.</p> <p>2. Tekrarlı yapıların ne amaçla kullanıldığı ve ne tür avantajlar sağlayacağı sorusu öğrencilere sorulur.</p> <p>3. Tekrarlı yapıların kullanılması gereken problemi kontrol yapılarıyla kodladığımız zaman ne tür bir hatanın oluşacağı öğrencilere gösterilir.</p> <p>4. <b>For</b> tekrarlı yapısı anlatılır. Nasıl kullanılacağıyla ilgili problem kodlanır ve öğrencilerinde kodlaması istenir.</p> <p>5. <b>While</b> tekrarlı yapısı anlatılır. Nasıl kullanılacağıyla ilgili problem kodlanır ve öğrencilerinde kodlaması istenir.</p>

	<p><b>6. For-While</b> tekrarlı yapılarının kullanım farkının ortaya konması için örnek problem iki tekrarlı yapı kullanılarak kodlanır.</p> <p><b>7.</b> Tekrarlı yapılarda kullanılan range, pass, break, continue yapılarının kullanımları örneklerle anlatılır.</p>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	<p>Tekrarlı yapılarla ilgili 3 sorudan oluşan ve öğrenme kazanımlarını kapsayan kodlama deneyimi etkinliği uygulanır. Kodlama deneyimi etkinliği örnek problem durumlarının öğrencilerin kodlamasına yöneliktir.</p>

<b>Uygulama 5. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	<p>Fonksiyonlar</p> <ol style="list-style-type: none"> <li>1. Fonksiyon kavramı ve türleri</li> <li>2. Fonksiyon tanımlama</li> <li>3. Parametre almayan fonksiyonların kullanımı</li> </ol>
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Anlatım, Soru-Cevap, Gösterip Yaptırma
<b>Araç gereçler</b>	Bilgisayar, Etkileşimli Tahta, Python Kod Editörü, Çalışma Yaprakları
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	<ol style="list-style-type: none"> <li>1. Fonksiyonun ne olduğunu bilir.</li> <li>2. Fonksiyon türlerini bilir.</li> <li>3. Fonksiyonun nasıl tanımlanacağını kavrar.</li> <li>4. Parametre almayan fonksiyonların kullanıldığı programlar geliştirir.</li> </ol>
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Haftanın ödevi hazırlanır.</li> <li>• Fonksiyonlar konusuyla ilgili kaynak taraması yapılır.</li> </ul>

	<ul style="list-style-type: none"> <li>• Çalışma yaprağı hazırlanır.</li> <li>• Fonksiyon kavramı, fonksiyon tanımlama ve parametre almayan fonksiyonlarla ilgili ders sunumu hazırlanır</li> <li>• Parametre almayan fonksiyon türlerine ilişkin örnek problem durumları hazırlanır</li> <li>• Fonksiyon tanımlamayla ilgili 3 soruluk kodlama deneyimi etkinliği hazırlanır.</li> </ul>
<b>BÖLÜM IV</b>	
<b>Derse İşlenişi:</b>	<ol style="list-style-type: none"> <li>1. Ödev kontrolü yapılır.</li> <li>2. Fonksiyon kavramı hakkında öğrencilerin ne düşündükleri sorulur. Öğrencilerin cevaplarına göre fonksiyon kavramı günlük hayattan örneklerle anlatılır.</li> <li>3. Fonksiyon türleri anlatılır ve asıl anlatılacak konunun tanımlayacağımız özel fonksiyonlar olduğu ifade edilir.</li> <li>4. Fonksiyon tanımlamada def ifadesinin görevi ve fonksiyonun nasıl tanımlanacağı örnek durumlar kodlanarak gösterilir ve öğrencilerinde belirtilen örneği kodlamaları istenir.</li> <li>5. Öğrencilerle birlikte üç örnek problem durumuna yönelik kullanılabilecek fonksiyon tanımlamaları yapılır. Burada öğrencilere problem durumlarına yönelik ne tür fonksiyonlara ihtiyaç duyulduğu sorulur. Öğrencilerin görüşlerinin uygulanabilirliği sorgulanır.</li> </ol>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Parametre almayan fonksiyonlarla ilgili 3 sorudan oluşan kodlama deneyimi etkinliği uygulanır. Etkinlikte öğrencilere verilen problemler için fonksiyon tanımlamaları istenir.

<b>Uygulama 6. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	Fonksiyonlar

	<ol style="list-style-type: none"> <li>1. Parametre alan fonksiyonlar</li> <li>2. İsimli ve sıralı argümanlar</li> <li>3. Varsayılan değerli argümanlar</li> </ol>
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Anlatım, Soru-Cevap, Gösterip Yaptırma
<b>Araç gereçler</b>	Bilgisayar, Etkileşimli Tahta, Python Kod Editörü, Çalışma Yaprakları
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	<ol style="list-style-type: none"> <li>1. Parametre alan fonksiyonların kullanıldığı programlar geliştirir.</li> <li>2. İsimli ve sıralı argümanların kullanıldığı programlar geliştirir.</li> <li>3. Varsayılan argümanların kullanıldığı programlar geliştirir.</li> </ol>
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Haftanın ödevi hazırlanır.</li> <li>• Parametre alan fonksiyonlarla ilgili kaynak taraması yapılır.</li> <li>• Çalışma kâğıdı hazırlanır.</li> <li>• Parametre alan fonksiyonlar, isimli ve sıralı argümanlar, varsayılan değerli argümanlar konularını kapsayan ders sunumu hazırlanır</li> <li>• Örnek program durumları hazırlanır.</li> <li>• Kazanımlarla ilgili 3 sorudan oluşan kodlama deneyimi etkinliği hazırlanır.</li> </ul>
<b>BÖLÜM IV</b>	
<b>Derse İşlenişi:</b>	<ol style="list-style-type: none"> <li>1. Ödev kontrolü yapılır.</li> <li>2. Parametre kavramının ne anlama geldiği sorusu öğrencilere yöneltilir. Ayrıca parametre alan ve almayan fonksiyonlar arasında ne gibi farklılıkların olduğu tartışılır.</li> <li>3. Parametre alan fonksiyonlar anlatılır. Parametresi olarak kodlanmış problem durumu, parametre alan fonksiyonlar kullanılarak tekrar kodlanır. Aralarındaki farkın öğrenciler tarafından görülmesi sağlanır.</li> </ol>



	<p>4. Argüman kavramı anlatılarak, isimli ve sıralı argümanlarla ilgili örnek problem durumları kodlanarak öğrencilere gösterilir. Öğrencilerinde kodlaması sağlanır.</p> <p>5. Varsayılan değerli argümanların kullanıldığı fonksiyonlar anlatılır. Öğrencilerle birlikte örnek problem durumları kodlanır.</p>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Parametre alan fonksiyonların kullanımı ve argüman tipleriyle ilgili 3 sorudan oluşan kodlama deneyimi etkinliği uygulanır. Etkinlik belirtilen kazanımları ölçmeye yönelik kodlama durumlarından oluşmaktadır.

<b>Uygulama 7. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	<p>Fonksiyonlar</p> <p>1. Değer döndüren fonksiyonlar</p> <p style="padding-left: 40px;">a. Global ifadesi</p> <p style="padding-left: 40px;">b. Return ifadesi</p> <p style="padding-left: 40px;">c. Pass ifadesi</p> <p>2. İstenen sayıda sıralı argümanlar</p> <p>3. İstenen sayıda isimli argümanlar</p>
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Anlatım, Soru-Cevap, Gösterip Yaptırma
<b>Araç gereçler</b>	Bilgisayar, Etkileşimli Tahta, Python Kod Editörü, Çalışma Yaprakları
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	<p>1. Değer döndüren fonksiyonların kullanıldığı programlar geliştirir.</p> <p>2. Çok sayıda sıralı ve isimli argümanlı fonksiyonların kullanıldığı programlar geliştirir.</p>

<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Haftanın ödevi hazırlanır.</li> <li>• Değer döndüren fonksiyonlar, istenen sayıda isimli ve sıralı argüman konularıyla ilgili kaynak taraması yapılır.</li> <li>• Çalışma kâğıdı hazırlanır.</li> <li>• Değer döndüren fonksiyonlarla ilgili ders sunumu hazırlanır.</li> <li>• İstenen sayıda isimli ve sıralı argüman kullanımıyla ilgili ders sunumu hazırlanır.</li> <li>• Örnek problem durumları hazırlanır.</li> <li>• Değer döndüren fonksiyonlar ve istenen sayıda isimli-sıralı argüman kullanımına yönelik 4 sorudan oluşan kodlama deneyimi etkinliği hazırlanır.</li> </ul>
<b>BÖLÜM IV</b>	
<b>Derse İşlenişi:</b>	<ol style="list-style-type: none"> <li>1. Ödev kontrolü yapılır.</li> <li>2. Değer döndürme işlemi hakkında öğrencilerin ne düşündüğü sorulur. Örnek verilen problem durumuyla ilgili değer döndürme işleminin nasıl yapılacağı gösterilir.</li> <li>3. Range ifadesi örnek problem durumu kodlanarak gösterilir ve öğrencilerin kodlamalar istenir.</li> <li>4. Global ifadesi örnek problem durumu kodlanarak gösterilir ve öğrencilerin kodlamalar istenir.</li> <li>5. Pass ifadesi örnek problem durumu kodlanarak gösterilir ve öğrencilerin kodlamalar istenir.</li> <li>6. İstenen sayıda sıralı argüman kullanımıyla ilgili örnek problem durumu öğrencilerle birlikte kodlanır.</li> <li>7. İstenen sayıda isimli argüman kullanımıyla ilgili örnek problem durumu öğrencilerle birlikte kodlanır.</li> </ol>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Kazanımları kapsayan 4 sorudan oluşan kodlama deneyimi etkinliği uygulanır. Öğrencilerden değer döndüren fonksiyon kullanımı, istenen sayıda isimli ve sıralı argüman kullanımıyla ilgili problem durumlarını kodlamaları istenir.

<b>Uygulama 8. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	<ol style="list-style-type: none"> <li>1. Liste tanımlama</li> <li>2. Listelerde kullanılan metotlar</li> <li>3. List () fonksiyonu ile tam sayı liste oluşturma</li> <li>4. * operatörü ile liste oluşturma</li> </ol>
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Anlatım, Soru-Cevap, Gösterip Yaptırma
<b>Araç gereçler</b>	Bilgisayar, Etkileşimli Tahta, Python Kod Editörü, Çalışma Yaprakları
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	<ol style="list-style-type: none"> <li>1. Liste türünde veri tanımlamayı bilir.</li> <li>2. Liste veri tipini ve metotlarını kullanarak programlar geliştirir.</li> </ol>
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Haftanın ödevi hazırlanır.</li> <li>• Liste tanımlama ve metotlarıyla ilgili kaynak taraması yapılır.</li> <li>• Listelerle ilgili çalışma kâğıdı hazırlanır.</li> <li>• Listeler ve kullanılan metotlarla ilgili ders sunumu hazırlanır.</li> <li>• Örnek problem durumları hazırlanır.</li> <li>• Listelerle ilgili 3 sorundan oluşan kodlama deneyimi etkinliği hazırlanır.</li> </ul>
<b>BÖLÜM IV</b>	
<b>Derse İşlenişi:</b>	<ol style="list-style-type: none"> <li>1. Ödev kontrolü yapılır.</li> <li>2. Liste veri tipinin ne amaçla kullanıldığı ve ne gibi avantajlarının olduğu öğrencilere sorulur. Öğrencilerin cevapları tartışılır.</li> <li>3. Liste tanımlamanın nasıl yapılacağı örneklerle gösterilir. Belirtilen örneklere benzer tanımlamaların öğrenciler tarafından da yapılması istenir.</li> <li>4. Liste metotları sırasıyla örneklerle anlatılır. Örneklerin öğrenciler tarafından kodlanması sağlanır.</li> </ol>

	<ul style="list-style-type: none"> <li>a. Append()</li> <li>b. Remove()</li> <li>c. Del()</li> <li>d. Reverse()</li> <li>e. Sort()</li> <li>f. Index()</li> <li>g. Count()</li> </ul> <p>5. List() fonksiyonunu kullanarak liste tanımlamanın nasıl yapılacağı örneklerle gösterilir.</p> <p>6. * operatörü kullanılarak liste tanımlamanın nasıl yapılacağı örneklerle gösterilir.</p>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Kazanımları kapsayan 3 sorudan oluşan kodlama deneyimi etkinliği uygulanır. Sınavda öğrencilerden liste tanımlama ve liste metotlarının kullanımlarını içeren örnek problem durumlarını kodlamaları istenir.

<b>Uygulama 9. Hafta Ders Planı</b>	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	<ol style="list-style-type: none"> <li>1. List() fonksiyonu</li> <li>2. Liste öğelerine erişme</li> <li>3. Liste öğelerini değiştirme</li> <li>4. Liste öğelerini birleştirme</li> <li>5. Demetler</li> </ol>
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Anlatım, Soru-Cevap, Gösterip Yaptırma
<b>Araç gereçler</b>	Bilgisayar, Etkileşimli Tahta, Python Kod Editörü, Çalışma Yaprakları

<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	<ol style="list-style-type: none"> <li>1. Liste veri tipini ve metotlarını kullanarak programlar geliştirir.</li> <li>2. Demet veri tipini ve metotlarını kullanarak programlar geliştirir.</li> </ol>
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Haftanın ödevi hazırlanır.</li> <li>• Liste temel fonksiyonları ve demetlerle ilgili kaynak taraması yapılır</li> <li>• Çalışma kâğıdı hazırlanır.</li> <li>• Liste temel fonksiyonları ve demetlerle ilgili ders sunumu hazırlanır.</li> <li>• Ders anlatımı esnasında kullanılmak üzere örnek problem durumları hazırlanır.</li> <li>• Liste temel fonksiyonları ve demetlerle ilgili kodlama deneyimi etkinliği hazırlanır.</li> </ul>
<b>BÖLÜM IV</b>	
<b>Derse İşlenişi:</b>	<ol style="list-style-type: none"> <li>1. Ödev kontrolü yapılır.</li> <li>2. Listelerde ne gibi işlemlerin yapılabileceği öğrencilere sorulur ve öğrencilerin cevaplaması istenir.</li> <li>3. List() fonksiyonunun kullanımı örnek üzerinde anlatılır ve öğrencilerin belirtilen örneği kodlaması istenir.</li> <li>4. Liste öğelerine erişimin nasıl olacağı anlatılır ve iki örnek öğrencilerle birlikte kodlanır.</li> <li>5. Liste öğelerinin nasıl değiştirileceği örneklerle anlatılır.</li> <li>6. Liste öğelerinin nasıl birleştirileceği örneklerle anlatılır.</li> <li>7. Demet veri tipinin ne olduğu anlatılır ve nasıl tanımlanacağı gösterilir. Öğrencilerinde demet veri tipinde örnek tanımlama yapmaları istenir.</li> </ol>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Liste veri tipi metotları ve demet veri tipiyle ilgili 3 sorudan oluşan kodlama deneyimi etkinliği uygulanır.

Uygulama 10. Hafta Ders Planı	
<b>BÖLÜM I</b>	
<b>Dersin Adı</b>	Bilgisayar Bilimi
<b>Sınıf</b>	Anadolu Lisesi / 9.Sınıf
<b>Konu</b>	<p>1. Küme veri tipi</p> <p>2. Küme veri tipi metotları</p> <ul style="list-style-type: none"> <li>a. clear()</li> <li>b. add()</li> <li>c. update()</li> <li>d. difference()</li> <li>e. discard()</li> <li>f. intersection()</li> <li>g. isdisjoint()</li> <li>h. issubset()</li> <li>i. issuperset()</li> <li>j. union()</li> <li>k. symmetric_difference()</li> </ul> <p>3. Dondurulmuş kümeler</p>
<b>Süre</b>	2 Ders Saati
<b>Öğretim Yöntem ve Teknikleri</b>	Anlatım, Soru-Cevap, Gösterip Yaptırma
<b>Araç gereçler</b>	Bilgisayar, Etkileşimli Tahta, Python Kod Editörü, Çalışma Yaprakları
<b>BÖLÜM II</b>	
<b>Öğrenme Kazanımları:</b>	<p>1. Küme türünde veri tanımlamayı ve metotlarını bilir.</p> <p>2. Küme veri tipini ve metotlarını kullanarak programlar geliştirir.</p>
<b>BÖLÜM III</b>	
<b>Derse Hazırlık:</b>	<ul style="list-style-type: none"> <li>• Haftanın ödevi hazırlanır.</li> <li>• Küme tanımlama ve küme metotlarıyla ilgili kaynak taraması yapılır.</li> </ul>

	<ul style="list-style-type: none"> <li>• Küme tanımlama ve metotlarının kullanımıyla ilgili çalışma kâğıtları hazırlanır.</li> <li>• Küme tanımlama ve metotlarıyla ilgili ders sunumu hazırlanır.</li> <li>• Kümelerle ilgili örnek problem durumları hazırlanır.</li> <li>• Küme tanımlama ve metotlarla ilgili kodlama deneyimi etkinliği hazırlanır.</li> </ul>
<b>BÖLÜM IV</b>	
<b>Derse İşlenişi:</b>	<ol style="list-style-type: none"> <li>1. Ödev kontrolü yapılır.</li> <li>2. Matematikte kümenin ne olduğu öğrencilere sorulur ve öğrencilerin cevaplaması istenir.</li> <li>3. Kümelerin nasıl tanımlanacağı örnekler üzerinden anlatılır ve öğrencilerinde kodlaması istenir.</li> <li>4. Küme metotları sırasıyla anlatılır. Anlatılan metotlarla ilgili örnek kodlamalar yapılır. Öğrencilerinde örnek kodlamaları yapmaları istenir. <ul style="list-style-type: none"> <li>a. clear()</li> <li>b. add()</li> <li>c. update()</li> <li>d. difference()</li> <li>e. discard()</li> <li>f. intersection()</li> <li>g. isdisjoint()</li> <li>h. issubset()</li> <li>i. issuperset()</li> <li>j. union()</li> <li>k. symmetric_difference()</li> </ul> </li> <li>5. Dondurulmuş kümenin ne olduğu anlatılır ve örnek durum kodlanır.</li> </ol>
<b>BÖLÜM V</b>	
<b>Ölçme ve Değerlendirme:</b>	Küme veri tipi ve metotlarıyla ilgili 3 sorudan oluşan kodlama deneyimi etkinliği uygulanır.

## Ek 12. Ders Notları

### Pilot Uygulama 1. Hafta Ders Notları

#### Python'un Tarihçesi

Python programlama dilinin ilk sürümü 1989 yılında Guido van Rossum tarafından geliştirilen nesne yönelimli, yorumlanabilen, modüler ve etkileşimli bir programlama dilidir. İlk sürümü python 1.0 1994 yılında yayınlandı. İlk başlarda pek tanınmayan python kısa sürede ilgi gördü.

Python 2.0 sürümü 2000 yılında yayınlandı ve 2001 yılında kar amacı gütmeyen Python Yazılım Vakfı kurularak Python 2.1 sürümü piyasaya sürüldü.

Python 3.0 sürümü ise 3 Aralık 2008 tarihinde piyasaya sürüldü ve daha önceki sürümlerinden bağımsız bir ürün olarak geliştirildi. Bizde dersimizi Python 3.x sürümüyle yürüteceğiz.

#### Python kullanmanın avantajları

Python kullanmanın birçok avantajı vardır. Bunlar;

- Tamamen ücretsiz olarak indirilip ücretsiz bir şekilde kullanılabilen bir dildir.
- Nesnelere dayalı bir dil olmasından dolayı öğrenmesi kolaydır. Diğer programlama dillerine nazaran kısa kodlar kullanılır. Ayrıca kod yazımında girintilime yapıldığından dolayı kodların okunabilirliği yüksektir.
- Neredeyse tüm işletim sistemlerinde çalıştığı için sağlam bir temele sahiptir.
- Çok güçlü kütüphanelere ve desteğe sahiptir. Bundan dolayı kısa sürede çok az kod yazarak çok güzel programlar yazılabilir.
- Birçok alanda kullanıldığından öğrenenlerin iş bulma olanağı yüksektir.
- İnternet üzerinden de çalışma olanağına sahiptir. Sadece bilgisayar ortamında çalışan programlar değil ayrıca internet üzerinden çalışan programlarda yazılabilmektedir.
- Her türlü ölçekte programlar yazılabilir. Her kesime hitap eden çok yönlü bir programlama dilidir.
- Sürekli güncellenen ve geniş topluluk desteğine sahip olan bir programlama dilidir.



## Python 3.6.1 sürümünün kurulumu

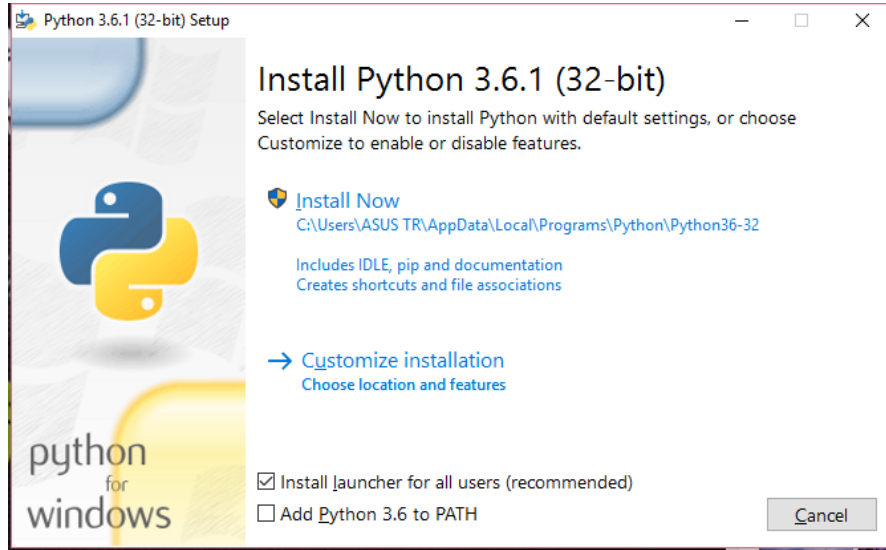
Python Windows işletim sistemine kurulumu için ilk olarak

<https://www.python.org/downloads/> adresinden programı indirmemiz gereklidir.



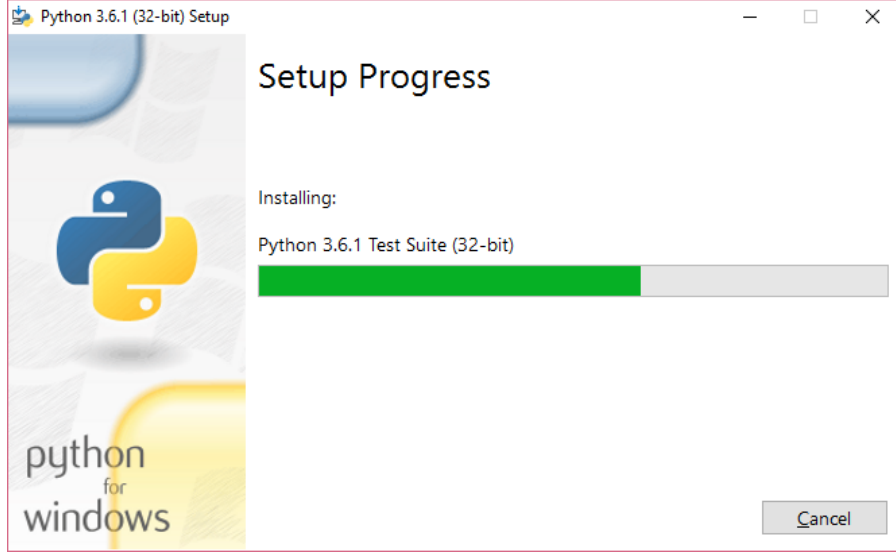
Yukarıda da gördüğünüz gibi Python'un iki sürümü bulunmaktadır. Biz dersimizde Python 3.x kullanacağımızdan dolayı Python 3.6.1 sürümünü indireceğiz.

İndirdiğimiz python-3.6.1.exe dosyasını çalıştırdığımız zaman karşımıza aşağıdaki gibi bir ekran gelecektir.

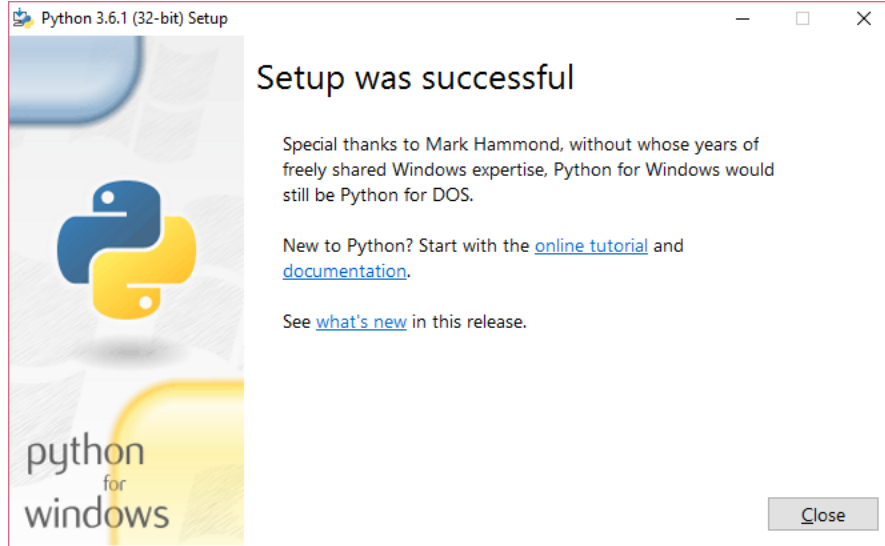


Yukarıdaki ekranda **Add Python 3.6 to PATH** seçeneği aktif olarak gelmemektedir. Kurulum esnasında bu seçeneğin seçilerek kurulum yapılması, programa bilgisayarın her yerinden erişilmesini sağlayacaktır. Bizde kurulum esnasında bu seçeneği seçerek ve **Install Now** linkine tıklayarak kurulumla devam edeceğiz.

Aşağıdaki ekranda görüldüğü gibi program bilgisayarınıza kurulmaya başlayacaktır.



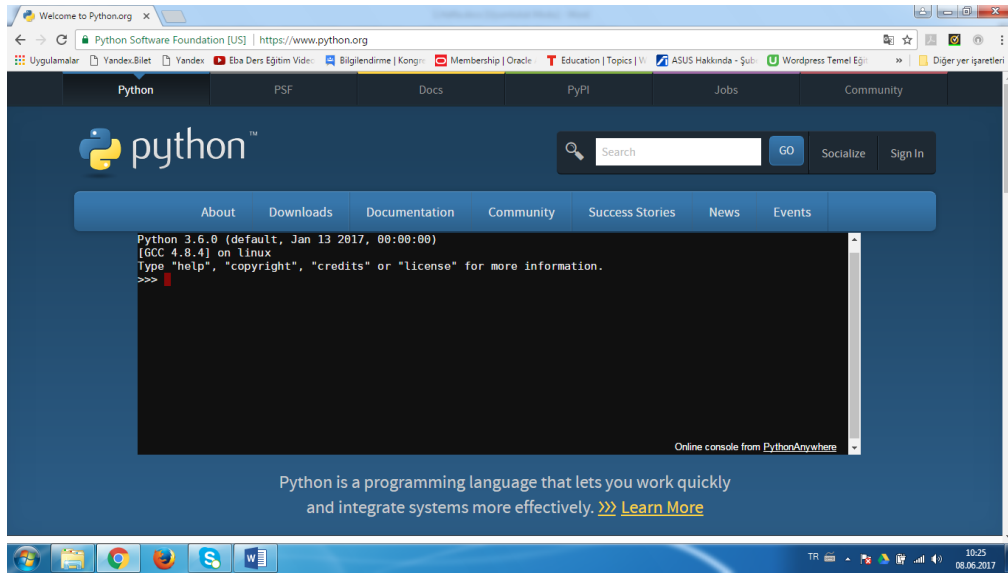
Kurulum tamamlandığında ise aşağıdaki gibi bir ekran karşımıza çıkacaktır.



### Python'da kod yazmak için kullanılan araçlar

- Windows ortamında başlangıç düğmesine tıklayıp çalıştır kısmına **cmd** yazıp komut penceresine ulaştıktan sonra komut ekranında **python** yazılarak kodlama işlemlerine başlanabilir.
- Python bilgisayara kurulduktan sonra **IDLE** programı da bilgisayarınıza yüklenmiş olacaktır. IDLE programını kullanarak python ile kod yazabilirsiniz.
- Jupyter-iPython** Programını kullanarak da python kodlarını yazabilirsiniz.

- d. [www.python.org](http://www.python.org) adresinde yer alan çevrimiçi kod editörünü kullanarak da kodlama işlemlerine başlayabilirsiniz.



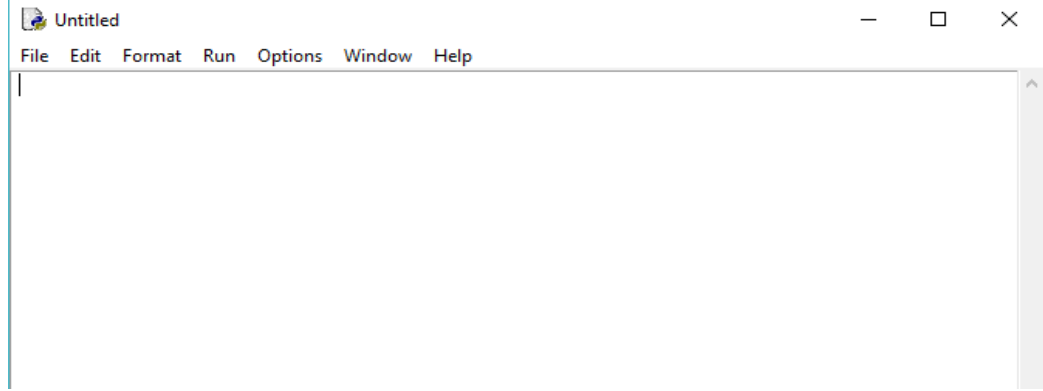
Python ile kod yazabilmemize olanak tanıyan bazı seçenekleri yukarıda sunduk. Dersimizde özellikle IDLE programını ve çevrimiçi kod editörünü yoğun bir şekilde kullanacağız.

### Program Dosyası Oluşturma, Kaydetme ve Çalıştırma

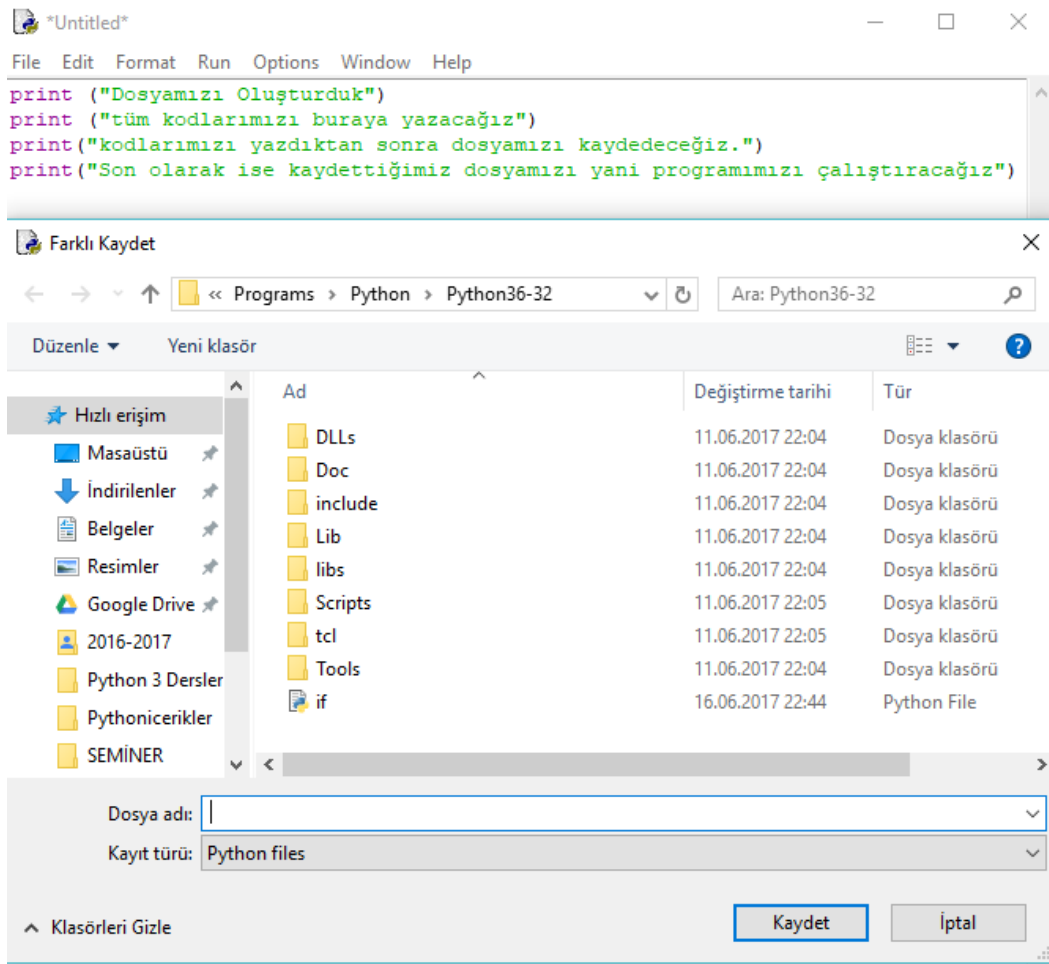
Şimdiye kadar kodlarımızı yazmak için etkileşimli kabuğu kullandık. Etkileşimli kabuk python kodlarımızı çalıştırmak ve test etmek amaçlı kullanılan bir yapıdır.

```
Python 3.6.1 Shell
File Edit Shell Debug Options Window Help
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 17:54:52) [MSC v.1900 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>> print ("Etkileşimli Kabukta")
Etkileşimli Kabukta
>>> print("kodlarımızı test ediyoruz")
kodlarımızı test ediyoruz
>>> print("ve ufak tefek python kodlarımızı çalıştırıyoruz")
ve ufak tefek python kodlarımızı çalıştırıyoruz
>>> |
```

Yukarıda etkileşimli kabuğu görmekteyiz. Biz şimdiye kadar IDLE ortamında kodlarımızı çalıştırdık. Fakat asıl programlarımızı dosya oluşturarak yazacağız. Dosya oluşturmak için **File-New File** seçeneği seçilerek aşağıdaki kodları yazacağımız ekrana ulaşılır.



Bu ekranda programımız için kullanacağımız kodlar yazılır. Kodlar yazıldıktan sonra dosyamızı kaydedeceğiz. Kaydetmek için **File-Save** seçeneğini kullanacağız.



**File-Save** seçeneğinden sonra yukarıdaki gibi bir ekran karşımıza çıkacak. Burada dosyamıza bir isim vereceğiz.

```
Python 3.6.1 Shell
File Edit Shell Debug Options Window Help
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 17:54:52) [MSC v.1900 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>> print ("Etkileşimli Kabukta")
Etkileşimli Kabukta
>>> print("kodlarımızı test ediyoruz")
kodlarımızı test ediyoruz
>>> print("ve ufak tefek python kodlarımızı çalıştırıyoruz")
ve ufak tefek python kodlarımızı çalıştırıyoruz
>>>
RESTART: C:/Users/apghpc/AppData/Local/Programs/Python/Python36-32/dosyaacma.py

Dosyamızı Oluşturduk
tüm kodlarımızı buraya yazacağız
kodlarımızı yazdıktan sonra dosyamızı kaydedeceğiz.
Son olarak ise kaydettiğimiz dosyamızı yani programımızı çalıştıracamız
>>> |
```

Yukarıda programımızın çalışmış hali gözükmektedir. Programımızı çalıştırmak için ise Run-Run Modüle menüsü kullanılır ya da F5 tuşu tercih edilir.

### **Kullanıcıdan Veri Almak ( *Input () Foksiyonu* )**

Pythonda kullanıcıdan veri almak için input () foksiyonu kullanılır. Örnek olarak bir dosya oluşturup aşağıdaki kodu yazalım.

```
input.py - C:/Users/apghpc/AppData/Local/Programs/Python/Python36-32/input.py (3.6.1)
File Edit Format Run Options Window Help
ad=input ("Adınızı Yazınız:")
soyad=input ("Soyadınızı Yazınız:")
print ("Adınız=", ad)
print ("Soyadınız=", soyad)
|
```

Yukarıda görüldüğü gibi kullanıcıdan input fonksiyonu kullanılarak ad soyad bilgilerini girmesi istenmiştir. Daha sonra kullanıcının girdiği bildiler print foksiyonu kullanılarak ekrana yazdırılmıştır.

```
Python 3.6.1 Shell
File Edit Shell Debug Options Window Help
Python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 17:54:52) [MSC v.1900 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>> ad=input("Adınızı Giriniz")
Adınızı Giriniz
>>>
RESTART: C:/Users/apghpc/AppData/Local/Programs/Python/Python36-32/input.py
Adınızı Yazınız:Hüseyin
Soyadınızı Yazınız:Yaşar
Adınız= Hüseyin
Soyadınız= Yaşar
>>> |
```

Dosyamızı kaydedip programı çalıştırdıktan sonra ise yukarıdaki gibi görülmektedir. Bilgiler kullanıcı tarafından girilmiş ve girilen değerler ekrana yazdırılmıştır.

**Örnek:** Kullanıcıdan dikdörtgenin kısa ve uzun kenarını girmesi istendikten sonra program dikdörtgenin alanını hesaplayacaktır.

```
input.py - C:/Users/apghpc/AppData/Local/Programs/Python/Pyth
File Edit Format Run Options Window Help
kısakenar=int(input("Kısa Kenarı Giriniz:"))
uzunkenar=int(input("Uzun Kenarı Giriniz:"))
alan=kısakenar*uzunkenar
print ("Dikdörtgenin Alanı:", alan)
```

Yukarıdaki kodu yazarak kullanıcıdan kısa ve uzun kenar değerleri alınır. Dikkat ederseniz input fonksiyonu int veri tipi olarak tanımlanmıştır. Yani tamsayı değer olarak ifade edilmiştir. Değerler alındıktan sonra dikdörtgenin alanı hesaplanmıştır. Programın çıktısı ise aşağıdaki gibidir.

```
>>>
RESTART: C:/Users/apghpc/AppData/Local/Programs/Python/Python36-32/input.py
Kısa Kenarı Giriniz:5
Uzun Kenarı Giriniz:10
Dikdörtgenin Alanı: 50
```

### Veri Tipi Dönüşümleri

Input fonksiyonu ile alınan veriler karakter dizisi (string) olarak alınmaktadır. Örneğin girilen sayının karesini bulan bir program yazalım.

```
*input.py - C:/Users/apghpc/AppData/Local/Programs/Python/Python36
File Edit Format Run Options Window Help
sayi=input("Lütfen bir sayı giriniz....:")
print("Girdiğiniz Sayının Karesi:",sayi**2)
```

Yukarıdaki gibi programı kaydedip çalıştırdığımız zaman aşağıdaki gibi hata mesajı verecektir.

```
Lütfen bir sayı giriniz....:5
Traceback (most recent call last):
  File "C:/Users/apghpc/AppData/Local/Programs/Python/Python36-32/input.py", line 2, in <module>
    print("Girdiğiniz Sayının Karesi:",sayi**2)
TypeError: unsupported operand type(s) for **: 'str' and 'int'
>>> |
```

Yukarıdaki hata mesajının sebebi input fonksiyonu ile alınan değerin string olması ve bunun üzerinde matematiksel işlem yapılamamasıdır. Bu sorunu çözmek için string veri tipini int(integer) veri tipine dönüştürmemiz gerekmektedir. Aşağıdaki gibi kodlarımızı düzenleyelim.

```
input.py - C:/Users/apghpc/AppData/Local/Programs/Python/Python36-32/input.py (3.6.1)
File Edit Format Run Options Window Help
sayi=int(input("Lütfen bir sayı giriniz....:"))
print("Girdiğiniz Sayının Karesi:",sayi**2)
```

Yukarıda görüldüğü gibi veri tipini integer olarak dönüştürdüğümüz zaman programımız sorunsuz olarak aşağıdaki gibi çalışacaktır.

```
>>>
RESTART: C:/Users/apghpc/AppData/Local/Programs/Python/Python36-32/input.py
Lütfen bir sayı giriniz....:5
Girdiğiniz Sayının Karesi: 25
>>> |
```

## Pilot Uygulama 2. Hafta Ders Notları

### Karakter Dizileri (Strings)

Bir veya birden fazla karakterden oluşan yapılara karakter dizisi (string) denir. Karakter dizilerinin temel özellikleri;

- Çift (") veya tek (') tırnaklar arasında yazılırlar.
- Karakter dizilimleridir.
- Dizi (array) şeklinde davranırlar.
- Elemanları değiştirilemez (immutable).
- Dilimlenebilirler (sliceable)

Programlama dillerinde verinin tipini bilmek son derece önemlidir. Python programlama dili içinde bu geçerlidir. Verinin tipini bilmemiz veriye yapabileceğimiz işlemleri belirlemektedir.

Örneğin python kod ortamında aşağıdaki kodu yazalım.

```
>>> Python Öğreniyorum.
```

*File "<stdin>", line 1*

## *Python Öğreniyorum.*

^

### *SyntaxError: invalid syntax*

Görüldüğü gibi karakter dizisini tırnak içinde yazmadığımızdan dolayı hata mesajı verilecektir.

Tırnak içinde yazılan her şey karakter dizisidir. Aşağıdaki tabloda yer alan karakter dizisi örneklerini bakalım.

<b>Kod</b>	<b>Kod Çıktısı</b>
>>> "h"	<i>'h'</i> Tek karakterlik karakter dizisi
>>> ""	<i>' '</i> İçi boş bir karakter dizisi
>>> " "	<i>' '</i> İçerisinde bir adet boşluk karakteri barındıran karakter dizisidir. Boşluk karakteri barındırdığı için dolu bir karakter dizisidir. Yukarıdakinden farklıdır.
>>> "Merhaba"	<i>'Merhaba'</i>
>>> "Bilgisayar Bilimi"	<i>'Bilgisayar Bilimi'</i>
>>> "Python ile kodlama öğreniyorum"	<i>'Python ile kodlama öğreniyorum'</i>
>>> "&&154%%"	<i>'&amp;&amp;154%%'</i> İçerisinde özel karakterler ve sayılar barındırmasına rağmen bir karakter dizisidir.



## Type fonksiyonuyla verinin tipini belirle

Herhangi bir verinin türünden kesin olarak emin olmak istediğimizde type fonksiyonu kullanılır. Type fonksiyonunun kullanımı aşağıdaki gibidir.

```
>>> type("Python")  
  
<class 'str'>
```

Görüldüğü gibi "Python" öğemizin türünü 'str' olarak gösterdi. Fonksiyonlarda değerler parantez içinde yazılır. Fonksiyonun parantez içinde yer alan değerlerine *parametre* denir.

## Karakter dizileri üzerinde işlemler

Karakter dizilerini "+" yani toplama işaretini kullanarak birleştirebiliriz. Aşağıdaki tabloda yer alan örnekleri inceleyelim.

Birleştirme örnekleri	
<pre>&gt;&gt;&gt; "Anadolu" + "Lisesi"  'AnadoluLisesi'</pre>	İki kelimeyi bitişik olarak birleştirmiştir.
<pre>&gt;&gt;&gt; "Anadolu" + " " + "Lisesi"  'Anadolu Lisesi'</pre>	Kelimeler arasına boşluk bırakmak için çift tırnak arasına bir boşluk karakteri kullanılmıştır.
<pre>&gt;&gt;&gt; "Anadolu" + " Lisesi"  'Anadolu Lisesi'</pre>	Lisesi karakter dizisinin başına bir adet boşluk bırakılmıştır.
<pre>&gt;&gt;&gt; "www" "." "meb" "." "gov" "." "tr"  'www.meb.gov.tr'</pre>	Karakter dizileri arasına + koymadan da birleştirme işlemleri yapabilirsiniz. Python sizin karakter dizilerinizi birleştirmek istediğinizi anlayacaktır.

Karakter dizilerini "\*" yani çarpma işaretini kullanarak karakter dizilerini tekrarlayabiliriz. Aşağıdaki tabloda yer alan örnekleri inceleyelim.

Çarpma örnekleri	
<pre>&gt;&gt;&gt; "Python" * 3 'PythonPythonPython'</pre>	Karakter dizisi çarpılan sayısal değer kadar tekrarlamıştır.
<pre>&gt;&gt;&gt; " Python" *2 'Python Python'</pre>	Kelimededen önce 1 adet boşluk bırakıldığı için
<pre>&gt;&gt;&gt; "#"*5 '#####'</pre>	Belirtilen işaret sayı adedinde tekrarlanmıştır.

### Aritmetik İşlem Operatörleri

Python programlama dilinde toplama, çıkarma, çarpma, bölme vb. işlemleri yapabilmek için çeşitli operatörler kullanılmaktadır. Bu operatörler ve kullanım örnekleri aşağıdaki tabloda verilmiştir.

Operatör	Kullanım Amacı	Örnek
+	<b>Toplama</b>	<pre>&gt;&gt;&gt; 12+8 20 &gt;&gt;&gt; 25+34 59 &gt;&gt;&gt; "25"+"34" '2534'</pre> <p>Yukarıdaki örnekte 25 ve 34 tırnak içinde verildiğinde sayı olarak değil karakter dizi olarak algılanır. Dolayısıyla sayılar birleştirilir.</p> <pre>&gt;&gt;&gt; 25+"25" HATA.....</pre>

		Yukarıdaki örnekte 25 ve "25" birbirinde farklıdır. Biri sayı diğeri ise karakter dizisidir. Dolayısıyla sistem hata mesajı verir.
-	<b>Çıkarma</b>	>>> 12-8 4
*	<b>Çarpma</b>	>>> 12*8 96 >>> "12"*8 '1212121212121212' Yukarıdaki örnekte 12 sayısı tırnak içinde verildiğinden çarpma işlemi yerine tekrarlama işlemi gerçekleştirilir.
/	<b>Bölme</b>	>>> 22/7 3,142857142857143 >>> 12*4 3
//	<b>Tam sayı bölme</b>	>>> 22//7 3 Bölme işleminin tam kısmını almak için // işareti kullanılır
**	<b>Üs alma</b>	>>> 5**3 125 >>> 2**7 128 Çift çarpı işareti ile ilk sayı taban, ikinci sayı ise üs olarak kullanılır (5 <sup>3</sup> , 2 <sup>7</sup> ).
%	<b>Kalanı Hesaplama</b>	>>> 45%7 3 45 sayısının 7 sayısına bölümünden kalanı bulabilmek için % işareti kullanılır.

## Kaçış Dizileri

### Ters Taksim (\)

Aşağıdaki örneği inceleyelim. Örnekte görüldüğü gibi print fonksiyonu kullanılarak ekrana karakter dizisi yazdırılmıştır. Python'da karakter dizileri yazdırılırken tek, çift ve üç tırnak arasına yazılacağını bir önceki derste belirtmiştik. Tek tırnak içinde karakter dizisini yazdığımız ve ayrıca ayraç işaretini kullandığımız zaman ekrana istediğimiz sonucu yazdıramamaktayız. Bu sebeple taksim işaretini kullanmamız gerekecektir.

```
>>> print('Yarın Gölbaşı\'na gidiyorum.')
Yarın Gölbaşı'na gidiyorum.
```

Taksim işaretini kullanmadığımız durumda ise program aşağıdaki gibi hata verecektir.

```
>>> print('Yarın Gölbaşı'na gidiyorum.')
SyntaxError: invalid syntax
```

### Satır başı (\n)

Satır başı yapmak için "\n" ifadesi kullanılmaktadır.

```
>>> print("kodlama\n öğrenmek\n istiyorum")
kodlama
 öğrenmek
 istiyorum
```

Yukarıdaki örnekte de görüldüğü gibi \n ifadesini kullandığımız yerlerde satır başı işlemi yapılmıştır.

```
>>> ders="Bilgisayar Bilimi"
>>> print(ders, "\n", "-"*len(ders), sep="")
Bilgisayar Bilimi
-----
```

Yukarıdaki örnekte ise ders adında bir değişken tanımlanmış ve ardından print fonksiyonu kullanılarak bu değişken ekrana yazdırılmıştır. \n ifadesi ile bir alt satıra inilerek, len fonksiyonuyla başlık uzunluğu kadar "-" karakteri yazdırılmıştır.

### Sekme (\t)

Sekme koymak amacıyla \t ifadesi kullanılır. Şimdi aşağıdaki örnekleri inceleyelim.

```
>>> print("Hüseyin\tYAŞAR")
Hüseyin YAŞAR
```

```
>>> print("bir", "iki", "üç", sep="\t")
bir      iki      üç
```

### Zil Sesi (\a)

Bip benzeri bir ses verir. Aşağıdaki örnekte 10 defa bip sesi duyulur fakat genelde Windows ortamında çalışmaktadır.

```
>>> print("\a"*10)
```

### Aynı Satır Başı (\r)

Aynı satırın başına dönülmesini sağlar.

```
>>> print("Merhaba \r Arkadaşlar!")

Arkadaşlar!
```

## Uygulama 1. Hafta Ders Notları

### Değişkenler

Python programlama dilinde değerlerin bir isme atanması sonucu oluşan yapılara değişken denir. Değişkenler saklama kapları gibi çalışır. Onların içine bir şeyler koyar; sonra oraya koyduğumuzu alır ve üzerinde işlemlerimizi yaparız. Daha sonra istersek tekrar eski yerine koyabiliriz. Değişkenleri tanımlarken = operatörü kullanılır. Aşağıdaki örnekleri birlikte inceleyelim.

```
>>> x="Kodlama"
>>> y="Öğreniyorum"
>>> x+y

'KodlamaÖğreniyorum'
```

Yukarıdaki örnekte x ve y değişkenleri tanımlanmış ve kod satırına x+y yazdığımız zaman tanımlanan değişkenlerin birleşimini vermektedir.

```
>>> a=10
>>> b=7
>>> c=5
>>> d=a*b

70

>>> d/c

14
```

Yukarıdaki örnekte a,b,c değişkenleri tanımlanmıştır. d değişkeni ise a ve b değişkenlerinin çarpımı olarak

	<i>tanımlanmıştır. Görüldüğü gibi d değişkenini tanımlarken ve d sayısını c sayısına bölerken sayıları kullanmadan direkt değişken isimleri kullanılmıştır.</i>	
--	---	--

### **Len() fksiyonu**

Her programlama dilinde kullanıcıdan belirli tiplerde veriler alınır. Kullanıcı adı, şifre bunlara örnek olarak verilebilir. Kullanıcıdan alınan veriler karakter dizisi olarak algılanacak ve tırnak içine alınacaktır.

```
>>> len("pythonkullanici")
15
>>> len("@a1b2c3d4e5xyz")
14
```

Yukarıda bir kullanıcıdan alınan kullanıcı adı ve şifre bilgisi bulunmaktadır. Eğer kullanıcı adı ve şifre belli bir karakter uzunluğunda olacaksa bu uzunlukları toplayarak toplam karakter uzunluğunu bulabiliriz.

```
>>> len("pythonkullanici") + len("@a1b2c3d4e5xyz")
29
```

Yukarıdaki örneklerde kullanıcı adı ve parola değerleri her defasında yeniden yazılmıştır. Burada kullanıcı adı ve parola değişkenleri tanımlanarak bu sıkıntıdan kurtulabiliriz. Programın her yerinde bu değişkenleri kullanabiliriz.

```
>>> kullanıcı_adi = "pythonkullanici"
>>> parola = "@a1b2c3d4e5xyz"
>>> len (kullanıcı_adi)
15
>>> len (parola)
15
>>> len (kullanıcı_adi) + len (parola)
29
```

Yukarıdaki örnekte görüldüğü gibi değişkenler tanımlandıktan sonra len () fonksiyonu değişken adlarını kullanarak karakter uzunluklarını hesaplamıştır.

### Değişken adı belirleme kuralları

- Değişken adları sayı ile başlayamaz. Aşağıdaki örnekte yer alan kullanım yanlıştır.

```
>>> 3_numaralı_kullanıcı = "Ahmet"
```

- Değişken aritmetik operatörlerle başlayamaz. Aşağıdaki örnekte yer alan kullanım yanlıştır.

```
>>> +numara = "145256"
```

- Değişken adları bir harf veya \_ işareti ile başlamalıdır.

```
>>> numara = "145256"
```

```
>>> _numara = "145256"
```

- Değişken adlarında Türkçe karakter kullanılabilir fakat bir sorunla karşılaşmamak için kullanmaktan kaçınılmalıdır.

- Aşağıdaki kelimeler değişken adı olarak kullanılamaz. Bu kelimeler python'da özel anlam ifade etmektedir. Kullanmaya çalıştığımız zaman program hata verecektir.

```
['False', 'None', 'True', 'and', 'as', 'assert', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']
```

- Değişken adlarında kelimeler arasında boşluk bırakılmaz. Aşağıdaki kullanım yanlıştır.

```
>>> kullanıcı_adı = "145256"
```

- Değişken adının değişken değerini betimlemesi önemlidir. Örneğin bir okuldaki öğrenci sayısı için **ogrenci\_sayısı** gibi bir değişken tanımlamak uygun olacaktır.

- Değişken adları çok uzun veya çok kısa olmamalıdır. Makul bir değerde olmalıdır. Örneğin;

```
>>> a = "145256" (KISA)
```

```
>>> golbasi_anadolu_lisesi_9sinif_ogrenci_sayisi = "145256" (UZUN)
```

```
>>> ogrenci_sayisi = "145256" (UYGUN)
```

## Değişken tanımlama örnekleri

1	Dikdörtgenin alanını değişken tanımlayarak hesaplayalım.	<pre>&gt;&gt;&gt; uzun_kenar = "10" &gt;&gt;&gt; kısa_kenar = "5" &gt;&gt;&gt; alan = uzun_kenar * kısa_kenar  50</pre>
2	Dairenin alanını değişken tanımlayarak hesaplayalım.	<pre>&gt;&gt;&gt; cap = "10" &gt;&gt;&gt; yaricap = cap/2 &gt;&gt;&gt; pi = "3,14" &gt;&gt;&gt; alan= pi * (yaricap*yaricap)  78,5</pre>

## Temel veri tipleri

Python kullanıcının girdiği değerleri otomatik olarak algılamaktadır. Diğer programlama dillerinde değişken tanımlarken değişkene atanacak değerın veri tipinin belirlenmesi gerekir. Örneğin pythonda `adı="Ahmet"` şeklinde bir değişken tanımladığımız zaman Ahmet kelimesinin karakter dizi olduğunu program otomatik olarak algılamaktadır. `Yas="45"` şeklinde bir değişken tanımladığımızı zaman python bu değişkenin tamsayı (integer) değişken olduğunu otomatik olarak algılayacaktır. Pythonda kullanılan veri tipleri aşağıdaki gibidir.

- İnteger:** Tam sayı veri tipini ifade eder (1, 2, 25000, 65000000 vb.)
- Float:** Ondalıklı sayıları ifade eden veri tipidir (0,5-1,785 vb.)
- Boolean:** Mantıksal türdeki verileri ifade eder (doğru-yanlış vb.)
- Strings: Karakter** dizilerini ifade eder.
- Listeler:** Köşeli parantez içinde farklı elemanları peş peşe saklayan türde veri tipidir (liste[1, 2, 3, 4, 5], liste1 ["ocak", "şubat", "mart"] vb.)

## Uygulama 2. Hafta Ders Notları

### Print fonksiyonu

Print ( ) fonksiyonu ekrana çıktı vermemizi sağlayan bir fonksiyondur. Python kodlarını etkileşimli kabukta yazarken direkt olarak bize çıktı vermekteydi. Fakat bir program



dosyası oluşturduğumuz zaman ekrana print ( ) fonksiyonu olmadan çıktı vermemiz mümkün değildir. Print ( ) fonksiyonu aşağıdaki gibi kullanılmaktadır.

```
>>> print("Kodlama Öğreniyorum")
```

*Kodlama Öğreniyorum* (Ekran çıktısı)

```
>>> print("Python Öğreniyorum")
```

*Python Öğreniyorum* (Ekran çıktısı)

*Birden fazla parametre kullanabiliriz.*

```
>>> print("Merhaba", "Arkadaşlar")
```

*Merhaba Arkadaşlar*

*Parametreler arasında boşluk olmasını istemiyorsak + operatörünü kullanabiliriz.*

```
>>> print("Merhaba" + "Arkadaşlar")
```

*MerhabaArkadaşlar*

*Farklı veri tiplerini ekrana yazdırabiliriz.*

```
>>> print("Hüseyin", 2017)
```

*Hüseyin 2017*

*Değişken tanımlayarak print ( ) fonksiyonunu kullanabiliriz.*

```
>>> ad = " Hüseyin"
```

```
>>>print(ad)
```

*Hüseyin*

Pythonda karakter dizilerini ekrana yazdırırken tek tırnak, çift tırnak ve üç tırnak kullanılabilir. Fakat üç tırnak genelde uzun karakter dizileri için kullanılır. Aşağıdaki örnekleri incelediğimizde farklı tırnak işaretlerinin kullanımını daha iyi anlayacağız.

```
>>> print ("Python programlama dilinin adı "piton" yılından gelmez")
```

*Yukarıdaki örnekte karakter dizisi için çift tırnak kullanıldığı gibi çift tırnak içinde "piton" kelimesi de bulunmaktadır. Dolayısıyla python karakter dizisini tanımayacak ve hata mesajı verecektir. Bu örneği şu şekilde yazarsak kodumuz doğru olarak çalışacaktır.*

```
>>> print ("""Python programlama dilinin adı "python" yılanından gelmez""")
```

*Python programlama dilinin adı "python" yılanından gelmez*

```
>>> print ('Python programlama dilinin adı "python" yılanından gelmez')
```

*Python programlama dilinin adı "python" yılanından gelmez*

```
>>> print("""İstanbul'un 5 günlük hava durumu tahmini""")
```

*İstanbul'un 5 günlük hava durumu tahmini*

Pythonda print ( ) fonksiyonu üç tırnakla başladıktan sonra kapatılana kadar karakter dizisi yazma işlemi devam eder. Üç tırnak kapatıldıktan sonra karakter dizisi sonlanır.

```
>>> print("""Merhaba
```

```
... Arkadaşlar
```

```
...Python Öğrenmeye Başlıyoruz""")
```

*Merhaba*

*Arkadaşlar*

*Python Öğrenmeye Başlıyoruz*

## **Print fonksiyonunun parametreleri**

### **Sep parametresi**

Print() fonksiyonu ile ekrana basılacak karakter dizileri arasına konulacak işaretin belirlenmesinde sep parametresi kullanılır. Sep parametresinin ön tanımlı değeri bir boşluk karakteridir. Aşağıdaki örnekte sep parametresi gözükmemektedir fakat arka planda çalışarak karakter dizileri arasına boşluk karakterini yerleştirmektedir.

```
>>> print("Merhaba", "Arkadaşlar")
```

*Merhaba Arkadaşlar*

Aşağıdaki örnekte görüldüğü gibi karakter dizileri arasına boşluk karakteri dışında başka bir karakter koymak istediğimizde sep kullanımını görülmektedir.

```
>>> print("Merhaba", "Arkadaşlar", sep="*")
```

*Merhaba\*Arkadaşlar*

Aşağıdaki örnekte de bir web adresi karakter dizileri arasına nokta konularak ekrana yazdırılmıştır.

```
>>> print("http://", "www", "kodlamaogreniyorum", "net", sep=".")
```

*http://www.kodlamaogreniyorum.net*

Aşağıdaki örnekte de ise karakter dizileri arasına bir kelime yerleştirilmiştir.

```
>>> print("bir", "iki", "üç", "dört", "ondört", sep="mumdur")
```

*bir mumdur iki mumdur üç mumdur dört mumdur ondört mumdur*

### **End parametresi**

End fonksiyonu print fonksiyonunun birleştirerek ekrana yazdırdığı karakter dizilerinin sonunda ne olacağını belirler. Print fonksiyonu ön tanımlı olarak satır başı karakterini ekler. Aşağıdaki örneği inceleyelim.

```
>>> print("Kodlama Öğreniyorum")
Kodlama Öğreniyorum
>>> |
```

Yukarıdaki kodu yazıp Enter tuşuna bastığımızda öncelikle ekrana karakter dizisini yazdırır ve sonrasında bir alt satıra geçer. Ön tanımlı değer satır başıdır.

Aşağıdaki örnekte print fonksiyonu öncelikle ekrana karakter dizisini yazdırır ve sonrasında bir alt satıra geçer. \n

```
>>> print("Kodlamayı Seviyorum", end="\n")
Kodlamayı Seviyorum
>>> |
```

Not: End ve sep parametrelerinin değerleri ya karakter dizisidir ya da none dir.

### **File parametresi**

Print fonksiyonuna verilen parametrelerin nereye yazılacağını belirleyen bir parametredir. Ön tanımlı olarak bu değerler sys.stdout (standart çıktı konumu) kaydedilir. Yani print fonksiyonu ön tanımlı olarak yazılan kodları ekrana yazdırır. Eğer ekrana değil de bir dosyaya yazdırmak istiyorsak file parametresine yazılacak dosya belirtilmelidir. Aşağıdaki örneği inceleyelim.

```
>>> dosya=open("kodlamaogreniyorum.txt", "w")
>>> print("Bu cümle kodlamaogreniyorum dosyasına kaydedilecek", file=dosya)
>>> |
```

Örnekte görüldüğü gibi print fonksiyonu ekrana bir şey yazmadı. Karakter dizisini **kodlamaogreniyorum.txt** dosyasına yazdı.

```
>>> bilgi=open("kisiselbilgiler.txt", "w")
>>> print("Adı:Ahmet", file=bilgi)
>>> print("Soyadı=Python", file=bilgi)
>>> print("Doğum Yeri=Malatya", file=bilgi)
>>> bilgi.close()
>>> |
```

Yukarıdaki örneği incelediğimiz zaman, öncelikle kisilerbilgiler.txt dosyası oluşturulmuştur. Daha sonra print fonksiyonu file parametresi ile birlikte kullanılarak bilgiler dosyaya yazılmıştır. Son olarak da dosya kapatılmıştır.

### **Flush parametresi**

Print fonksiyonu ile karakter dizilerini dosyaya yazdığımızda python dosya kapatılmayana kadar bilgileri tampon bellekte tutar. Dosya kapatıldıktan sonra tampondaki bilgiler dosyaya yazılır. Eğer anında yazılmasını istiyorsanız flush parametresini kullanabilirsiniz.

```
>>> print("Bilgileri tamponda tutmadan dosyaya yazalım", file=bilgi, flush=True) |
```

Yukarıdaki bir kod yazarsak bilgiler tamponda beklemeden dosyaya yazılırlar.

## **Uygulama 3. Hafta Ders Notları**

### **Kontrol Yapıları**

Pythonda belirli koşullu yapılar bulunmaktadır. Bu koşullu yapılarla işlemler yapabilmek için kontrol yapıları kullanılmaktadır. Python ve diğer programlama dillerinde yaygın olarak kullanılan aşağıda belirtilen kontrol yapılarını anlatacağız. Bunlar;

- if
- if else
- if elif else şeklindedir.

## If

Tüm programlama dillerinde en çok kullanılan kontrol yapılarından biri olan if aşağıda belirtildiği gibi bir yapıda kullanılır.

*if koşul:*

*bunu yap*

*şunu yap*

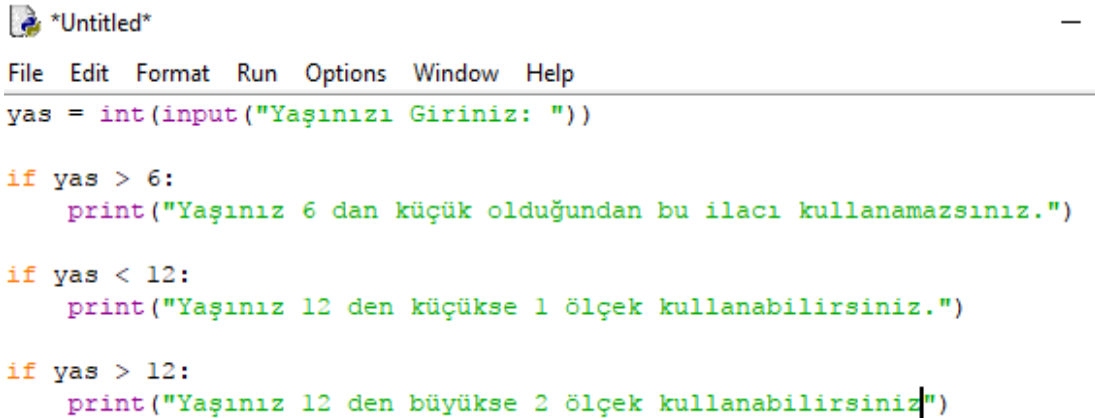
*şunu da yap*

Yukarıda görüldüğü gibi eğer if bloğundaki koşul sağlanıyorsa yani mantıksal olarak doğruysa program “bunu yap” ile başlayan bloğu takip eder. Eğer koşul sağlanmıyorsa program bloğu atlayarak çalışmaya devam eder.

```
>>> a=8
>>> if a>5:
        print("a sayısı 5 yayısından büyüktür")

a sayısı 5 yayısından büyüktür
>>> |
```

Yukarıdaki örneği incelediğimiz zaman; a değişkeni tanımlanarak if bloğu içinde a değişkeninin 5 den büyük olup olmadığı kontrol edilmiştir. Kontrol sonrasında ise a sayısının 5 den büyük olduğu ekrana yazdırılmıştır.



```
*Untitled*
File Edit Format Run Options Window Help
yas = int(input("Yaşınızı Giriniz: "))

if yas > 6:
    print("Yaşınız 6 dan küçük olduğundan bu ilacı kullanamazsınız.")

if yas < 12:
    print("Yaşınız 12 den küçükse 1 ölçek kullanabilirsiniz.")

if yas > 12:
    print("Yaşınız 12 den büyükse 2 ölçek kullanabilirsiniz|")
```

Yukarıdaki programımızı incelediğimiz zaman yaş değişkeni input komutu ile kullanıcıdan alınmıştır. Kullanıcının yaş değeri if bloklarıyla kontrol edilerek ilacı nasıl kullanacağına dair ekrana bilgi yazdırılmıştır. Bir önceki derslerden de hatırlarsanız input

ile kullanıcıdan alından veri string olarak alınmaktadır. Bundan dolayı veri tipi int (integer) dönüştürülerek alınmıştır.

### If - Elif

Aşağıdaki programları inceleyelim ve ondan sonra if ve elif arasındaki farkı ortaya koyalım.

<pre>yaş = int(input("Yaşınız: "))  if yaş == 18:     print("18 iyidir!")  elif yaş &lt; 0:     print("Yok canım, daha neler!...")  elif yaş &lt; 18:     print("Genç bir kardeşimizsin!")  elif yaş &gt; 18:     print("Eh, artık yaş yavaş yavaş kemale eriyor!")</pre>	<pre>yaş = int(input("Yaşınız: "))  if yaş == 18:     print("18 iyidir!")  if yaş &lt; 0:     print("Yok canım, daha neler!...")  if yaş &lt; 18:     print("Genç bir kardeşimizsin!")  if yaş &gt; 18:     print("Eh, artık yaş yavaş yavaş kemale eriyor!")</pre>
<p>Yukarıdaki elif kullanılan örnekte yaş değeri 18 girildiği zaman sadece ekrana "18 iyidir!" mesajı yazdırılacaktır. Yani 18 değerini sağlayan diğer yani "Eh, artık yaş yavaş yavaş kemale eriyor!" ekrana yazdırılmayacaktır.</p>	<p>Yukarıdaki örnekte ise yaş değeri 18 girildiği zaman aşağıdaki çıktılar verilecektir.</p> <pre>&gt;&gt;&gt; 18 iyidir!  &gt;&gt;&gt; Eh, artık yaş yavaş yavaş kemale eriyor!</pre>

Örnekte de görüldüğü gibi elif kontrol yapısı sadece istenen değeri ekrana yazdırmaktadır. Aynı işlemi if ile yaptığımızda ise koşulu sağlayan tüm değerler ekrana yazdırılmaktadır. Yani eğer if deyimlerini art arda sıralayacak olursak, Python doğru olan bütün sonuçları listeleyecektir. Ama eğer if deyiminden sonra elif deyimini kullanırsak, Python doğru olan ilk sonucu listelemekle yetinecektir.

### If – Else

Eğer sadece if koşulu sağlanmadığı takdirde yapılması gereken bir takım işlemler varsa, bu kez else-if ikilisi kullanılır.

*if koşul:*

*True (doğru) blok işlemi*

*True (doğru) blok işlemi2*

*else:*

*False (yanlış) blok işlemi1*

*True (yanlış) blok işlemi2*

Aşağıdaki örneği incelediğimizde kullanıcıdan programdan çıkabilmesi için “e” harfine basması istenmiştir. Kullanıcı istenen harfe bastığında ekrana “Güle güle !” mesajı yazdırılmış, farklı bir tuşa bastığında ise “Peki, biraz daha sohbet edelim!” mesajı yazdırılmıştır. Yani doğru tuşa if bloğu, yanlış tuşa ise else bloğu devreye girmiştir.

```
soru = input("Programdan çıkmak istediğinize emin misiniz? \
Eminseniz 'e' harfine basın : ")

if soru == "e":
    print("Güle güle!")

else:
    print("Peki, biraz daha sohbet edelim!")
```

Aşağıdaki örnekte ise iki tane if bloğu kullanılmıştır. Kullanıcı e harfine bastığında 1. İf bloğu devreye girmiş, b tuşuna bastığında 2. İf bloğu devreye girmiş ve son olarak farklı bir tuşa bastığında ise else bloğu devreye girmiştir.

```
soru = input("Programdan çıkmak istediğinize emin misiniz? \
Eminseniz 'e' harfine basın : ")

if soru == "e":
    print("Güle güle!")

if soru == "b":
    print("Kararsız kaldım şimdi!")

else:
    print("Peki, biraz daha sohbet edelim!")
```

**Not:** Birden sonra if bloğu kullanılıyorsa else bloğu altına yazıldığı if bloğuna bağlı olarak çalışır. Yani tüm if bloklarıyla bağlantılı değildir.

### **If – Elif - Else**

Eğer birden fazla if bloğu kullanılacaksa, 1. İf bloğundan sonra diğer if blokları elif bloğuyla oluşturulmalıdır. Neden elif bloğunu kullanmamız gerektiğini yukarıda açıklamıştır. En sonda ise else bloğunu kullanarak kontrolümüzü bitirmiş olacağız.

*if koşul1:*

*Sonuç1*

*elif koşul2:*

*Sonuç2*

*elif koşul3:*

*Sonuç3*

*else:*

*Sonuç4*

Aşağıdaki örneği inceleyelim. Kullanıcıdan boy değişkenini girmesi istenmiş ve boy değişkeni integer veri tipine dönüştürülmüştür. Daha sonra boy değişkeni if bloğuna alınarak 170 cm den küçük olduğu durumlar için ekrana “boyunuz kısa” mesajı yazdırılmıştır. Daha sonra ise elif bloğu kullanılarak boy değişkeninin 180 cm den küçük olma ihtimali kontrol edilmiş ve ekrana “boyunuz normal” mesajı yazdırılmıştır. Son olarak ise else bloğu kullanılarak diğer değerler için ekrana “boyunuz uzun” mesajı yazdırılmıştır.

```
boy = int(input("boyunuz kaç cm?"))  
  
if boy < 170:  
    print("boyunuz kısa")  
  
elif boy < 180:  
    print("boyunuz normal")  
  
else:  
    print("boyunuz uzun")
```

#### Uygulama 4. Hafta Ders Notları

##### **Döngüler**

Python'da programların tekrar tekrar çalışmasını sağlayan yapılara döngü denir. Döngü kavramını daha iyi anlayabilmek için aşağıdaki örneği inceleyelim



```
tuttuğum_sayı = 23
bilbakalım = int(input("Aklimdan bir sayı tuttum. Bil bakalım kaç tuttum? "))
if bilbakalım == tuttuğum_sayı:
    print("Tebrikler! Bildiniz...")
else:
    print("Ne yazık ki tuttuğum sayı bu değildi...")
```

Yukarıdaki örneği aşamalı bir şekilde inceleyerek döngüleri neden kullanmamız gerektiğini anlamaya çalışalım;

- Öncelikli olarak **tuttuğum\_sayı** değişkeni 23 olarak tanımlanmıştır. Kullanıcının bu sayıyı bulması istenecektir.
- Kullanıcının verdiği değerler **bilbakalım** değişkenine atanmıştır. Input fonksiyonu ile kullanıcıdan int (integer) veri tipinde değer alınmıştır.
- Daha sonra kontrol yapıları devreye sokulmuştur. Eğer kullanıcının girdiği değer tutulan sayıya eşitse (bilbakalım==tuttuğum\_sayı) program "**Tebrikler! Bildiniz...**" uyarısını ekrana yazdırarak sonlandıracaktır.
- Eğer eşit değilse program else bloğuna geçerek "**Ne yazık ki tuttuğum sayı bu değildi...**" uyarısını vererek sonlanacaktır.

Dikkat ettiyseniz kullanıcı doğru cevapta verse yanlış cevapta verse program gerekli uyarıyı vererek sonlanmaktadır. **Eğer programın yanlış cevap verildiğinde kullanıcıdan tekrar bir sayı girmesini istiyorsak kullanmamız gereken yapı döngüdür.** Yani bu tekrarlama işlemini yapmamızı ancak döngüler sağlar. İki döngü yapımız bulunmaktadır. Bunlar for ve while dir.

### **For döngüsü**

Pythonda for ve while döngüleri benzer amaçlar için kullanılır, fakat for döngüsü while döngüsünden daha maharetlidir. Pythonda yazılan programları incelediğimiz zaman neredeyse tüm programlarda for döngüsünün kullanıldığını görmemiz mümkün. For döngüsünün kullanım şekli aşağıda belirtildiği gibidir.

*for değişken\_adi in değişkenler*

*yapılacak\_işlem*

Değişken adı, tanımlanan değişken içindeki her bir öğeyi ifade etmektedir. Şimdi for döngüsünü daha iyi anlayabilmek için aşağıdaki örneği inceleyelim.

```
ad="hüseyin"
for harfler in ad:
    print(harfler)
```

Yukarıdaki programı incelediğimizde ad isminde bir değişken tanımlanmıştır. Harfler değişkeni ise ad değişkeni içinde yer alan her harfi ifade etmektedir. Program ad değişkeninden sırayla harfleri alarak ekrana yazdırmaktadır. Programın çıktısı ise aşağıdaki gibidir.

```
>>>
= RESTART: C:/Users/apghpc/AppData/Local/Programs/Python/Python36-32/for.py =
h
ü
s
e
y
i
n
|
```

Aşağıdaki örnekleri inceleyelim.

```
sayılar=123456789
for sayı in sayılar:
    print(sayı)
|
```

Bu şekilde harfleri yazdırdığımız gibi sayıları da yazdırmaya çalışıyoruz. Fakat program hata mesajı verecektir. Çünkü döngüler karakter dizilerinde kullanılabilirken, sayılarda yani integer ifadelerde kullanılmamaktadır.

Bu programı yan taraftaki gibi yazarsak istediğimiz çıktıyı alabiliriz.

Görüldüğü gibi programda sayılar değişkeni tırnak içinde tanımlandıktan sonra, ekrana yazdırılırken int dönüştürülerek yazdırılmıştır.

```
sayılar="123456789"
for sayı in sayılar:
    print(int(sayı))
```

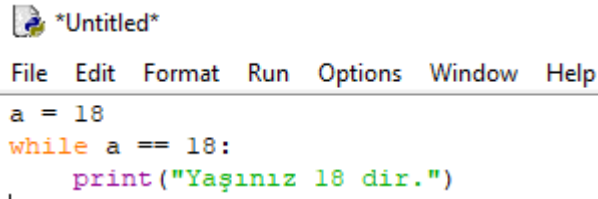
```
1
2
3
4
5
6
7
8
9
>>> |
```

<pre>sayılar = "123456789"  for i in sayılar:     if int(i) &gt; 3:         print(i)</pre> <p>Sayılar değişkeni içinde yer alan sayılar arasında if kontrol yapısı kullanılarak 3 den büyük olan sayılar i değişkenine atanarak ekrana yazdırılmıştır.</p>	<pre>4 5 6 7 8 9 &gt;&gt;&gt;</pre>
--	-------------------------------------

<pre>tr_harfler = "şçöğüıı"  parola = input("Parolanız: ")  for karakter in parola:     if karakter in tr_harfler:         print("parolada Türkçe karakter kullanılamaz")</pre> <p>Kullanıcıdan parola girmesi istenmekte ve kullanıcının girdiği parolada Türkçe karakter var ise parola kabul edilmemektedir.</p>	<pre>Parolanız: şabcde parolada Türkçe karakter kullanılamaz</pre> <p>Ekran çıktısını incelediğimizde parolada "ş" karakterinin kullanılması, parola hatasının verilmesine sebep olmuştur.</p>
---	--

### While döngüsü

While döngüsü for döngüsüne benzer işlemler yapmaktadır. Programın tekrar tekrar çalışmasını sağlamaktadır. Aşağıdaki örneği inceleyelim.



```
*Untitled*
File Edit Format Run Options Window Help
a = 18
while a == 18:
    print("Yaşınız 18 dir.")
```

Yukarıdaki örnekte görüldüğü gibi a=18 ifadesi ise a değişkenine 18 değeri atanmıştır. Burada dikkat etmeniz gereken nokta "=" operatörü değer atamak için kullanılmıştır. Daha sonra a==18 ifadesi ile a değerinin 18'e eşit olup olmadığı kontrol edilmiştir. "==" ifadesi eşittir anlamındadır. Yani a değişkeninin değeri 18 eşit olduğunda ekrana "Yaşınız 18 dir." uyarısı yazdırılmaktadır.

Aşağıdaki bir başka örneği inceleyelim.

```
*Untitled*
File Edit Format Run Options Window Help
a = 1
while a < 18:
    print("Yaşınız ehliyet için uygun değil")
```

Yukarıdaki örnekte de a değişkeni 18 den küçük olduğu sürece ekrana "Yaşınız ehliyet için uygun değil" uyarısı yazılacaktır.

Aşağıdaki örnekte is a değişkeni 1 olarak tanımlanmıştır. Program her döngüde a değeri 1 arttırılarak oluşan değeri a değişkenine atamaktadır. A değişkeni 10 değerine ulaştığında ise döngü bitmekte ve ekrana "Programın morali bozuldu" uyarısı verilmektedir.

```
*Untitled*
File Edit Format Run Options Window Help
a = 1
while a < 10:
    a += 1
    print("Programın morali bozuldu.")
```

### **Döngülerde kullanılan fonksiyonlar**

Döngülerin daha işlevsel kullanılabilmesi için belirli araçların kullanılması gerekmektedir. Şimdi bu araçları görelim.

#### **range**

Pythonda belirli aralıktaki sayıları göstermek için range() fonksiyonu kullanılır. Örnek olarak göstermek gerekirse;

```
>>> for i in range(0,10):
    print(i)
```

Yukarıdaki kod 0 dan başlayarak 10 kadar tüm değerleri ekrana alt alta yazdırır. Range() fonksiyonu bu örnekte 0-10 aralığını ifade etmektedir.

Şimdi aşağıdaki örneği aşama aşama inceleyelim.

```
while True:
    parola = input("parola belirleyin: ")

    if not parola:
        print("parola bölümü boş geçilemez!")

    elif len(parola) in range(3, 8): #eğer parolanın uzunluğu 3 ile 8 karakter
        #aralığında ise...
        print("Yeni parolanız", parola)
        break

    else:
        print("parola 8 karakterden uzun 3 karakterden kısa olmamalı")
```

- Yukarıdaki örnekte ilk olarak kullanıcıdan parola girişi istenmiştir. While döngüsü içinde bu işlem yapılmıştır.
- Daha sonra if kontrol yapıları kullanılarak parolanın boş olma durumu, parolanın 3-8 karakterden uzun olacağı kontrol edilmiştir ve gerekli uyarılar ekrana yazdırılmıştır.
- While döngüsündeki **True** ifadesi döngünün doğru parola girilinceye kadar sürmesi gerektiğini ifade etmektedir.
- If not ifadesi parolanın boş yazılma durumunu kontrol etmiştir.

## Pass fonksiyonu

Pythonda pas geç, hiçbir şey yapma anlamında kullanılmaktadır. Örnek vermek gerekirse;

```
while True:
    parola = input("parola belirleyin: ")

    if not parola:
        pass

    elif len(parola) in range(3, 8): #eğer parolanın uzunluğu 3 ile 8 karakter
        #aralığında ise...
        print("Yeni parolanız", parola)
        break

    else:
        print("parola 8 karakterden uzun 3 karakterden kısa olmamalı")
```

Görüldüğü gibi yukarıda range fonksiyonunda kullandığımız örneğin aynısı fakat if not kısmında kullanıcının parola girmemesi pass komutu ile önemsenmemiştir. Yani parola girmediği durumda tekrar parola sorulmamış program devam etmiştir.

Aşağıdaki örnekte de kullanıcıdan sayı girmesi istenmiş ve kullanıcının 0 girmesi durumundan program durdurulmuş. Sıfırdan küçük sayı girmesi durumunda ise pas geçilmiş ve girilen sayı ekrana yazdırılmamış yani hiç bir işlem yapılmamıştır.

```
while True:
    sayı = int(input("Bir sayı girin: "))

    if sayı == 0:
        break

    elif sayı < 0:
        pass

    else:
        print(sayı)
```

### break fonksiyonu

Pythonda break programı durdurmak için kullanılır. Aşağıdaki örnekte de görüldüğü gibi kullanıcı parola belirlediği zaman döngünün sonlanması için break fonksiyonu kullanılmıştır.

```
while True:
    parola = input("Lütfen bir parola belirleyiniz:")
    if len(parola) < 5:
        print("Parola 5 karakterden az olmamalı!")
    else:
        print("Parolanız belirlendi!")
        break
```

### continue fonksiyonu

Continue kendisinden sonra gelen tüm işlemler pas geçerek döngünü başına gelir. Aşağıdaki örneği inceleyelim.

---

```
while True:
    s = input("Bir sayı girin: ")
    if s == "iptal":
        break

    if len(s) <= 3:
        continue

    print("En fazla üç haneli bir sayı girebilirsiniz.")
```

Yukarıdaki programda kullanıcı iptal yazarsa program sonlanacaktır. Eğer kullanıcı 3 basamak ve üstünde bir sayı yazarsa program ekrana uyarı verecektir.

## Uygulama 5. Hafta Ders Notları

### Fonksiyon

Fonksiyon, belirli sayıda verileri kullanarak bunları işleyen ve bir sonuç üreten komut grubudur. Fonksiyonlar programlamada tekrarlı olarak kullanacağımız komutların kullanımını kolaylaştırmak için kullanılırlar. Şimdiye kadar birçok fonksiyon kullandık. Örneğin print() fonksiyonu ekrana çıktı vermek görevini üstlenen bir fonksiyon. Fakat print () fonksiyonu sadece ekrana çıktı vermekle yetinmez arka planda birçok tanımlanmış işlemi yerine getirir. Aşağıdaki örneği inceleyelim.

```
>>>print("Python", "ile", "kodlama", "öğreniyorum", sep=" ", end="\n",  
... file=sys.stdout, flush=False)
```

Yukarıdaki örnekte print() fonksiyonu aşağıda belirtilen işlemleri yerine getirmektedir;

- ✓ "Python", "ile", "kodlama", "öğreniyorum" parametrelerini ekrana yazmakta,
- ✓ sep="" parametresi ile yazdırdığı parametreler arasına boşluk eklemekte,
- ✓ end="\n" parametresi ile ekran çıktısından sonra bir alt satıra geçmekte
- ✓ file=sys.stdout ile çıktı ortamı olarak ekranı kullanmakta
- ✓ flush=False ile çıktılar ekrana gönderilmeden önce tamponda bekletilmektedir.

Eğer print () fonksiyonu olmasaydı yukarıdaki işlemleri kendimiz yerine getirmek zorunda kalacaktık.

### Fonksiyon Tanımlama

Fonksiyon tanımlamak için **def** ifadesi kullanılmaktadır. Def ifadesinden sonra tanımlanmak istenen fonksiyonun ismi yazılır. Fakat isim yazarken Türkçe karakter kullanılmaması gerekmektedir.

```
def fonsiyon_adi():
```

Şimdi bir fonksiyon tanımlayalım ve bu fonksiyon ekrana belirttiğimiz çıktıyı versin;

```
>>> def fonsiyon_tanimlama():  
    print("İlk parametre almayan fonsiyonumuzu tanımlıyoruz")
```

İlk fonsiyonumuzu yukarıda görüldüğü gibi tanımladık. İlk olarak def ifadesinden sonra fonsiyonumuzun adını yazdık ve daha sonra fonsiyonumuzun ne yapacağını tanımladık.

Biz örneğimizde belirtilen yazının ekrana yazdırılması işlemini gerçekleştirmeyi amaçladık. Şimdi sıra fonksiyonumuzu nasıl çalıştıracağımıza geldi.

```
>>> fonksiyon_tanimlama()  
İlk parametre almayan fonksiyonumuzu tanımlıyoruz
```

Yukarıda görüldüğü gibi fonksiyon adını yazdığımız zaman fonksiyonumuz tanımlanan işlemi gerçekleştirdi. Artık programımızda bu fonksiyonumuzu istediğimiz yerde kullanabiliriz.

Fonksiyonlarla ilgili olarak genel bir özet çıkaracak olursak;

- Python'da iki tip fonksiyon vardır. Bunlar gömülü ve özel fonksiyonlardır. Buradaki özel fonksiyonlar kullanıcının tanımladığı fonksiyonlardır. Gömülü fonksiyonlar ise geliştiriciler tarafından program içine yerleştirilmiş fonksiyonlardır.
- Gömülü fonksiyonlar print(), input(), open() vb. hiçbir tanımlamaya gerek kalmadan kullanılan fonksiyonlardır.
- Fonksiyon kullanımı iki aşamadan oluşur. Birinci aşama fonksiyonun tanımlandığı aşamadır. İkinci aşama ise fonksiyonun gerektiğinde çağrılıp kullanıldığı aşamadır.
- Fonksiyon tanımlamak için def ifadesi kullanılmaktadır.
- Def ifadesinden sonra fonksiyon adı Türkçe karakter kullanılmadan yazılır ve iki nokta konular ve alt satıra geçilir.
- Alt satırda girintili olarak yazılan tüm kodlar fonksiyonun gövdesini oluşturur.
- Girintinin dışına çıktığı anda fonksiyon sona erer.

## Örnekler

1. Çarpma işlemi yapan bir fonksiyon tanımlayalım;

```
>>> def carpma():  
    a=5  
    b=6  
    c=a*b  
    print("c sayısının değeri:",c)
```

Fonksiyonumuzu tanımlarken öncelikle def ile fonksiyon tanımlama işlemine başlayıp fonksiyonumuzun adını verdik. Daha sonra a ve b değişkenlerini tanımlayarak c değerinin nasıl hesaplanacağını tanımladık ve son olarak print() ile sonucu ekrana yazdırdık. Şimdi ise fonksiyonumuzu çağıralım;



```
>>> carpma ()
c sayısının deęeri: 30
```

Yukarıda görüldüğü gibi fonksiyonumuzu çağırduğımız zaman ekrana c deęerini hesaplayarak yazdırdı.

2. Kullanıcı tarafından girilen sayının karesini hesaplayan fonksiyonu yazalım,

```
>>> def karebul():
    sayi=int(input("Karesi hesaplanacak sayıyı giriniz:"))
    print("Girdiđiniz sayının karesi:",sayi**2)
```

Yukarıda görüldüğü gibi fonksiyonumuzu tanımladık. Şimdi fonksiyonumuzu çağıralım.

```
>>> karebul()
Karesi hesaplanacak sayıyı giriniz:5
Girdiđiniz sayının karesi: 25
```

Fonksiyonumuzu çağırduğımız zaman kullanıcıdan karesi hesaplanacak sayıyı girmesi istenmiş ve fonksiyon girilen sayının karesini hesaplayarak ekrana yazdırmıştır.

3. Kullanıcıdan şifre girmesini isteyen ve şifreyi tanımlanan şifreyle eşleştirip ekrana gereken uyarıyı veren fonksiyonu tanımlayalım;

```
def dogru():
    print ("Tebrikler,girdiđiniz parola dođru.")

def yanlis():
    print ("Üzgünüz,girdiđiniz parola yanliş.")

parola="123456"

giris = input("Parolayı giriniz:")

if(parola==giris):
    dogru()
else:
    yanlis()
```

Örnekte dođru() ve yanlis() ismine iki tane fonksiyon tanımlanmıştır. Programı çalıştırdığımız zaman kullanıcıdan şifre girmesi istenmiş ve şifre dođru olduđu takdirde dođru() fonksiyonu çalıştırılmıştır.

```
Parolayı giriniz:123456
Tebrikler,girdiđiniz parola dođru.
```

Şifre yanlış girildiğinde ise aşağıdaki gibi yanlış() fonksiyonu çağrılmıştır.

```
Parolayı giriniz:896523
Üzgünüz,girdiğiniz parola yanlış.
```

## Uygulama 6. Hafta Ders Notları

### Parametre Alan Fonksiyonlar

Şimdiye kadar fonksiyonlarla ilgili yaptığımız çalışmalarda parantez içine hiçbir şey yazmamıştık. Şimdi ise fonksiyon kullanımında parantez içine bazı değerler yazmamız gereken durumları inceleyeceğiz. Parantez içinde yer alan verilere programlama dillerinde parametre adı verilir.

Örneğin parametre kullanmadan iki sayının çarpımını hesaplayan bir örnek verelim.

```
>>> def carp():
    sayi1=12
    sayi2=5
    carpim=sayi1*sayi2
    print ("Çarpım Sonucu:", carpim)
```

```
>>> carp()
Çarpım Sonucu: 60
```

Örnekte görüldüğü gibi carp() fonksiyonunda parametre kullanılmadan tanımlanan değerlerin çarpımını hesaplanmıştır.

Şimdi ise parametre kullanarak fonksiyonumuzu tekrar yazalım.

```
def carp(S1,S2):
    carpim=S1*S2
    print ("Sayıların Çarpımı:",carpim)
S1=int(input("Birinci Sayıyı Giriniz:"))
S2=int(input("İkinci Sayıyı Giriniz:"))
carp(S1,S2)
```

```
Birinci Sayıyı Giriniz:10
İkinci Sayıyı Giriniz:15
Sayıların Çarpımı: 150
```

Parametre kullanarak yazdığımız fonksiyon kullanıcıdan alınan sayıların çarpımını hesaplayarak ekrana yazdırmaktadır.

## İsimli ve Sıralı Argümanlar

Fonksiyon tanımlanırken parantez içinde kullanılan değerlere parametre adı verildiğini daha önce belirtmiştik. Fonksiyon çağırma işleminde parantez içinde kullanılan verilere ise argüman adı verilir. Genellikle argümanlar içinde parametre ifadesi kullanılmaktadır.

Sıralı argümanlarda parantez içinde kullanılan verilerin sıralaması çok önemlidir. Fonksiyon tanımlanırken kullanılan sıralama, fonksiyon çağırıldığında da aynı şekilde takip edilmelidir.

```
def fonksiyon(ad,N1,N2):
    ortalama = (N1+N2)/2
    print (ad,"adındaki öğrencinin not ortalaması:",ortalama)

fonksiyon("Hüseyin",60,80)
```

Program çıktısı:

```
Hüseyin adındaki öğrencinin not ortalaması: 70.0
```

Yukarıdaki örnekte fonksiyon adında bir fonksiyon tanımlanmıştır. Fonksiyonun görevi not ortalamasını hesaplamaktadır. Alt kısımda fonksiyon çağırılırken kullanılan (ad, N1, N2) sıralaması, fonksiyon tanımlanırken kullanılan sıralamanın aynıdır.

Eğer sıralama sorunu yaşamak istemiyorsak yapacağımız işlem sıralı argümanlarımızı, isimli argümanlara dönüştürmek olacaktır. Şimdi yaptığımız ortalama örneğini isimli argümanlar kullanarak yeniden oluşturalım.

```
def fonksiyon(ad,N1,N2):
    ortalama = (N1+N2)/2
    print (ad,"adındaki öğrencinin not ortalaması:",ortalama)

fonksiyon(N1=60,ad="Hüseyin",N2=80)
```

Görüldüğü gibi fonksiyonu çağırırken argümanların sıralamalarında değişiklik yapmamıza rağmen isimli argüman kullanmamızdan dolayı programımız aşağıdaki gibi aynı çıktıyı vermiştir.

```
Hüseyin adındaki öğrencinin not ortalaması: 70.0
```

### Varsayılan Değerli Argümanlar

Fonksiyonlar tanımlanırken belirli değerler varsayılan olarak tanımlanır. Örneğin print() fonksiyonu tanımlanırken belirli değerler varsayılan olarak tanımlanır. Örneğin;

```
>>>print("Hüseyin", "Yaşar")
```

Yukarıda görüldüğü gibi biz sadece iki karakter dizisini ekrana yazdırıyoruz fakat bu fonksiyon çalışırken varsayılan değerleri arka planda otomatik olarak çalıştırır. Aslında arka planda fonksiyonumuz aşağıdaki gibidir.

```
>>>print("Hüseyin", "Yaşar", sep=" ", end="\n", file=sys.stdout, flush=false)
```

Yani biz görmesek de sep, end, file ve flush argümanları arka planda görenlerini varsayılan olarak yerine getirmektedir.

Aşağıdaki örneği inceleyelim:

```
def toplama(sayi,kacar=1):  
    toplam = 0  
    i=0  
    while(sayi>i):  
        toplam += i  
        i+=kacar  
    print ("Toplam:",toplam)
```

```
toplama(10)
```

```
Toplam: 45
```

Yukarıdaki örnekte sayı değeri birer birer artırılarak toplam değeri hesaplanmıştır. Burada dikkat edilmesi gereken nokta sayının kaçar artacağı varsayılan olarak (kacar=1) şeklinde tanımlanmıştır. Toplam değeri ise 45 olarak hesaplanmıştır. Bu örnekte fonksiyonumuzu aşağıdaki gibi çağırırsak;

```
toplama(10,3)
```

toplama değerimiz 18 olarak hesaplanacaktır. Çünkü biz varsayılan olarak tanımlanan kaçar değerinin yerine 3 değerini kullandık.

**Not:** Varsayılan değerli argümanlarda dikkat etmemiz gereken en önemli kural; varsayılan değerli parametrenin sıralamanın en sonunda olması gerektiğidir.

## Uygulama 7. Hafta Ders Notları

### Global Deyimi

Fonksiyon içinde tanımlanan değişkenler fonksiyon dışında kullanılamamaktadır.

```
>>> def fonk():  
    a="Global değişkenler"  
    print (a)
```

Yukarıdaki fonksiyonda a değişkeni tanımlanmış ve fonksiyonu çağırdığımız zaman aşağıdaki gibi çıktı verecektir.

```
>>> fonk()  
Global değişkenler
```

Şimdi ise fonksiyonu şu şekilde yazalım.

```
def fonk():  
    a="Global değişkenler"  
  
fonk()  
print (a)
```

Yukarıda görüldüğü gibi fonksiyon içinde tanımlanan a değişkenini fonksiyon dışında ekrana yazdırmaya çalıştığımız zaman program aşağıdaki hata mesajını vermektedir.

```
Traceback (most recent call last):  
  File "C:/Users/apghpc/AppData/Local/Programs/Python/Python36-32/fonk.py", line 5, in <module>  
    print (a)  
NameError: name 'a' is not defined
```

Yukarıdaki hata mesajını ortadan kaldırmak ve fonksiyon içindeki değişkeni fonksiyon dışında kullanmak için global deyimini aşağıdaki gibi kullanmamız gerekmektedir.

```
def fonk():  
    global a  
    a="Global değişkenler"  
  
fonk()  
print (a)
```

Görüldüğü gibi programımız global deyimini kullanmamızdan dolayı a değişkeni fonksiyon dışında kullanmamıza izin verecektir ve aşağıdaki gibi programımız hata vermeden çalışacaktır.

```
>>>  
= RESTART: C:/Users/apghpc/AppData/Local/Programs/Python/Python36-32/fonk.py =  
Global değişkenler
```

## Return Deyimi

Şu ana kadar yazdığımız bütün fonksiyonlarda ekrana bir sonuç bastırdık. Ama program yazarken her zaman böyle yapmayız. Yazdığımız bir fonksiyondan değer döndürmek ve bu değer ile işlemler yapmak isteyebiliriz. Yani "return" deyimi fonksiyonun geriye herhangi bir değer döndürmesini sağlar.

```
def dondur(not_):  
    if not_ > 50:  
        return "Dersten Geçtiniz."  
    else:  
        return "Dersten Kaldınız."  
  
sonuc = dondur(60)  
print (sonuc)
```

Yukarıdaki programda 60 değeri dondur() fonksiyonunda return deyimi kullanılarak döndürülmüş ve sonuç ekrana yazdırılmıştır. Sonuç aşağıdaki gibidir.

```
Dersten Geçtiniz.
```

Aşağıdaki örneği inceleyelim.

```
def dondur(sayi):  
    return sayi % 2  
  
sonuc = dondur(52)  
if sonuc==1:  
    print ("Sayı Tek.")  
else:  
    print ("Sayı Çift.")
```

Yukarıdaki örnekte return deyimi geriye bir değer döndürmektedir. Döndürülen değere göre sayının tek veya çift olduğuna karar verilmektedir.

## Pass Deyimi

Programın herhangi bir durumda hiçbir şey yapmasını istemiyorsak pass deyimi kullanılmaktadır.

```
ucret = int(input("Ücreti giriniz:"))  
  
if ucret>50:  
    pass  
else:  
    print("Bu ücrete kitap bulabilirsiniz.")  
  
print("Teşekkürler")
```

Yukarıdaki örnekte kullanıcı tarafından girilen ücret değeri 50 den büyük olduğu takdirde hiçbir işlem yapılmadan geçilmektedir. Küçük olduğu takdirde ise else bloğu çalıştırılmaktadır. Program çıktıları ise aşağıdaki gibidir.

50 den küçük değer girdiğimizde program çıktısı:

```
Ücreti giriniz:45
Bu ücrete kitap bulabilirsiniz.
Teşekkürler
```

50 den büyük değer girdiğimizde program çıktısı:

```
Ücreti giriniz:60
Teşekkürler
```

Fonksiyonlarda da bu şekilde fonksiyonu tanımlayıp içeri sonradan doldurabilirsiniz. Bu durumda "pass" deyimini bizim işimizi görür.

```
def gec():
    pass
```

Yukarıdaki fonksiyonun içi boş durumdadır. Programın hata vermesini önlemek için pass deyimini kullanılmıştır.

### İstenen Sayıda Sıralı Argüman Kullanımı

Şimdiye kadar oluşturduğumuz fonksiyonlarda parametre sayılarımızı belirliyorduk. Örneğin aşağıdaki gibi bir fonksiyon tanımladığımızda tanımladığımız parametre sayısı kadar argüman tanımlamamız gerekecektir.

```
>>> def bilgiler(ad,no,bölüm,sınıf):
```

Bu örnekte fonksiyon tanımlarken 4 parametre tanımladık. Fonksiyon çağrıldığı zaman dörtten az veya fazla argüman gönderdiğimiz zaman program hata verecektir.

İstenen sayıda argüman tanımlayabilmek için yapmamız gerekeni bir örnekle açıklayalım.

```
def kayıt(*bilgiler):
    bilgi=""
    for i in bilgiler:
        bilgi+=i
    print ("Öğrencinin bilgileri:\n",bilgi)
kayıt("Hüseyin", " Eğitim Fakültesi", " BÖTE Bölümü", " 1.Sınıf")
```

Örnekte görüldüğü gibi parametre tanımlanırken \*bilgiler ifadesi kullanılarak bir kısıtlamaya gidilmemiştir. Fonksiyon çağrılırken ise ad, fakülte, bölüm ve sınıf argümanları gönderilmiştir. Programın çıktısı aşağıdaki gibidir.

```
Öğrencinin bilgileri:  
Hüseyin Eğitim Fakültesi BÖTE Bölümü 1.Sınıf
```

Fonksiyonu çağırdığımızda aldığımız çıktı bir demet olarak saklanmaktadır. Demetler üzerinde yapılan tüm işlemleri “bilgiler “ üzerinde de yapabiliriz.

```
def demet(*bolumler):  
    for sıra,bolum in enumerate(bolumler):  
        print ("%s : %s"% (sıra,bolum))  
  
demet ("Bilgisayar", "Elektrik-Elektronik", "Makina", "İnşaat")
```

Yukarıda görüldüğü gibi bölümler bir demet olarak tanımlanmıştır. Aşağıda programın çıktısını inceleyelim.

```
0 : Bilgisayar  
1 : Elektrik-Elektronik  
2 : Makina  
3 : İnşaat
```

### Örnek:

```
def çarp(*sayılar):  
    sonuç = 1  
    for i in sayılar:  
        sonuç *= i  
    print (sonuç)  
çarp(1,2,3,4)
```

Örnekte görüldüğü gibi çarp fonksiyonu belirtilen argümanları birbiriyle çarparak sonucu 24 olarak ekrana yazdıracaktır. Burada dikkat edilmesi gereken parametre tanımlanırken bir sınırlamaya gidilmemiştir.

### İstenen Sayıda İsimli Argüman Kullanımı

İstenilen sayıda sıralı argüman tanımladığımız gibi, istenilen sayıda isimli argümanda tanımlanmaktadır.

```
def fonksiyon(**parametreler):  
    print (parametreler)  
  
fonksiyon (isim="Hüseyin", soyisim="Yaşar", meslek="Öğretmen", şehir="Adıyaman")
```



Yukarıdaki örnekte görüldüğü gibi istenilen sayıda isimli argüman tanımlanırken **\*\***parametreler ifadesi kullanılmıştır. Programın çıktısı da aşağıdaki gibidir.

```
{'isim': 'Hüseyin', 'soyisim': 'Yaşar', 'meslek': 'Öğretmen', 'şehir': 'Adıyaman'}
```

Gördüğümüz gibi, fonksiyonu tanımlarken parametremizin sol tarafına yerleştirdiğimiz **\*\*** işareti, bu fonksiyonu çağırırken yazdığımız isimli parametrelerin bize bir sözlük olarak verilmesini sağlıyor. Bu yapının bize bir sözlük verdiğini bildikten sonra, bunu sözlük veri tipinin kuralları çerçevesinde istediğimiz şekilde evirip çevirebiliriz.

### Örnek:

```
def fonksiyon(**bilgiler):
    ortalama = (bilgiler["not1"]+bilgiler["not2"])/2
    print (bilgiler["isim"],"adındaki öğrencinin not ortalaması",ortalama)

fonksiyon(not1=80,isim="Hüseyin",not2=90)
```

Gördüğümüz gibi aslında sıralı argümanlardan tek farkı fonksiyon orada argümanları "demet" olarak alıyordu burada ise "sözlük" olarak alıyor. Yani istenen sayıda isimli argümanları kullanırken sözlüklerin tüm özelliklerinden yararlanabiliriz. Sıralı argümanlarda yaptığımız gibi argümanları gönderirken **\*\*** operatöründen faydalanabiliriz. Yani bu örneğimizi şu hale getirerek de kullanabiliriz.

```
def fonksiyon(**bilgiler):
    ortalama = (bilgiler["not1"]+bilgiler["not2"])/2
    print (bilgiler["isim"],"adındaki öğrencinin not ortalaması",ortalama)

sozluk = {"not1":80,"isim":"Hüseyin","not2":90}
fonksiyon(**sozluk)
```

## Uygulama 8. Hafta Ders Notları

### Listeler

Listeler, belirli bir sırada çoklu değer tutmak için kullanılır. Listeler string veri tipine benzemekle birlikte, listelerin içine herhangi bir python nesnesi yerleştirilebilir. Listeler kullanılmaya başlamadan önce tüm değişkenler gibi tanımlanması gerekir. Listelerde değerler köşeli parantez içinde belirtilir.

```
Liste=["python", "ile", "kodlama", "eđitimi", 2017, 2018]
```

Yukarıda görüldüğü gibi tanımlanan liste içine hem strig hem de integer veriler konulmuştur.

Listelerde;

- Köşeli parantez içindeki değere indeks denmektedir.
- Lst[0] değeri ilk değerdir.
- Lst[1] değeri serideki ikinci değerdir.

### **Append() metodu**

Listelere eleman eklemek için kullanılan metottur.

```
>>> isimler=["Ahmet","Fatma","Mehmet"]
>>> print(isimler)
['Ahmet', 'Fatma', 'Mehmet']
```

Yukarıdaki listeye bir isim eklemek için append metodunu kullanalım.

```
>>> isimler=["Ahmet","Fatma","Mehmet"]
>>> isimler.append('Ayşe')
>>> print(isimler)
['Ahmet', 'Fatma', 'Mehmet', 'Ayşe']
```

Append metoduyla eklediğimiz "Ayşe" ismi listemizde yerini almıştır.

### **Remove() metodu**

Listelere eleman silmek için kullanılan metottur.

```
>>> print(isimler)
['Ahmet', 'Fatma', 'Mehmet', 'Ayşe']
>>> isimler.remove('Ahmet')
>>> print(isimler)
['Fatma', 'Mehmet', 'Ayşe']
>>>
```

Yukarıdaki örnekte görüldüğü gibi remove metodunu kullanarak "Ahmet" ismini listemizden çıkardık.

### **Del() metodu**

Listelere eleman silmek için kullanılan bir diğer yöntem ise del metodunun kullanımınıdır. Del metodunda ise parametre olarak silinecek elemanın indisinin girilmesi gereklidir.

```
>>> print(isimler)
['Fatma', 'Mehmet', 'Ayşe']
>>> del isimler[2]
>>> print(isimler)
['Fatma', 'Mehmet']
>>>
```

Del metodunu kullanarak 2 indisi "Ayşe" elemanını listemizden çıkardık.

### **Reverse() metodu**

Liste elemanlarının sırasını tersine çevirmek için kullanılır.

```
>>> dersler=["Matematik","Fizik","Kimya","Biyoloji"]
>>> print(dersler)
['Matematik', 'Fizik', 'Kimya', 'Biyoloji']
>>> dersler.reverse()
>>> print(dersler)
['Biyoloji', 'Kimya', 'Fizik', 'Matematik']
```

Liste elemanlarımız terse çevrilerek sıralanmıştır.

### **Sort() metodu**

Liste elemanlarını alfabetik sıraya koymak için kullanılır.

```
>>> print(dersler)
['Biyoloji', 'Kimya', 'Fizik', 'Matematik']
>>> dersler.sort()
>>> print(dersler)
['Biyoloji', 'Fizik', 'Kimya', 'Matematik']
>>>
```

Yukarıda görüldüğü gibi liste öğelerimiz alfabetik olarak sıraya konulmuştur.

### **Index() metodu**

Liste içerisinde arama yapmayı sağlayan methodtur. Eğer aranılan öğe listede mevcutsa, bu öğenin indisini geri döndürür.

```
>>> print(dersler)
['Biyoloji', 'Fizik', 'Kimya', 'Matematik']
>>> dersler.index('Kimya')
2
```

Görüldüğü gibi listede aradığımız "Kimya" elemanının 2 indisinde olduğunu çıktı olarak verdi.

### Count() metodu

Liste içerisinde arama yapılan değerden kaç tane olduğunu çıktı olarak verir.

```
>>> print(dersler)
['Biyoloji', 'Fizik', 'Kimya', 'Matematik']
>>> dersler.count('Matematik')
1
```

Listede "Matematik" elemanından 1 tane olduğu için çıktı olarak 1 değerini verdi.

### List() fonksiyonu ile tamsayı liste yapmak

Liste oluşturmak istenirken bazen kullanıcıdan veri almak yerine liste elemanları otomatik olarak oluşturulmak istenebilir. Bu durumda list() fonksiyonuyla birlikte range() fonksiyonu kullanılabilir. list() fonksiyonu otomatik olarak belirlenen değerlerden liste oluştururken range() fonksiyonu ise başlangıç ve bitiş değerleri arasında artım miktarına göre değerler oluşturur

#### Örnek:

```
>>> def main():
    a=list(range(0,10))
    print(a)
    a=list(range(10,-1,-1))
    print(a)
    a=list(range(0,100,10))
    print(a)
    a=list(range(-5,6))
    print(a)
```

Yukarıdaki örnekte range() fonksiyonu ile başlangıç ve bitiş noktaları belirlenmiş ve list() fonksiyonu kullanılarak listeler oluşturulmuştur. Bu işlemler main() fonksiyonu tanımlanarak yapılmıştır. Fonksiyonumuzu çalıştırdığımız zaman ekran görüntümüz aşağıdaki gibi olacaktır.

```
>>> main()
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
[10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
[0, 10, 20, 30, 40, 50, 60, 70, 80, 90]
[-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5]
```

### \* Operatörü ile liste oluşturma

Liste oluşturulurken \* operatörü solundaki liste elemanını sağındaki adet kadar çoğaltır.

#### Örnek:

```
>>> def main():
    a=[0]*10
    print(a)
    a=[3.4]*5
    print(a)
    a=3*[10,20,30]
    print(a)
    n=3
    a=n*["abc",22,8.7]
    print(a)
```

### Ekran Çıktısı:

```
>>> main()
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[3.4, 3.4, 3.4, 3.4, 3.4]
[10, 20, 30, 10, 20, 30, 10, 20, 30]
['abc', 22, 8.7, 'abc', 22, 8.7, 'abc', 22, 8.7]
```

Görüldüğü gibi \* operatörü dizileri belirtilen değer kadar çoğaltmaktadır.

### Uygulama 9. Hafta Ders Notları

#### List fonksiyonu örneği

Aşağıda verilen “alfabe” karakter dizisini listeye dönüştürelim.

```
>>> alfabe = "abcçdefgğhıijklmnoöprştuüvyz"
```

### Ekran Çıktısı:

```
>>> harfler=list(alfabe)
>>> print(harfler)
['a', 'b', 'c', 'ç', 'd', 'e', 'f', 'g', 'ğ', 'h', 'ı', 'i', 'j',
'k', 'l', 'm', 'n', 'o', 'ö', 'p', 'r', 's', 'ş', 't', 'u', 'ü',
'v', 'y', 'z']
```

Alfabe karakter dizisi harfler değişkenine list() fonksiyonu kullanılarak atanmış ve harfler değişkeni ekrana yazdırılmıştır. Alfabe karakter dizisi liste olarak ekrana yazdırılmıştır.

#### Liste öğelerine erişme örneği

Aşağıdaki örnekte iller listesi tanımlanmıştır. Döngü yapıları kullanılarak listeye ulaşılmış ve listede bulunan elamanlar ekrana yazdırılmıştır.

### Örnek:

```
>>> iller=["Ankara","İstanbul","Adıyaman","Malatya","Sakarya"]
>>> for il in iller:
    print(il)
```

```
Ankara
İstanbul
Adıyaman
Malatya
Sakarya
```

## Örnek:

```
>>> iller[1]
'İstanbul'
>>> iller[-1]
'Sakarya'
>>> iller[1:3]
['İstanbul', 'Adıyaman']
>>> len(iller)
5
```

Yandaki örneklerde de listeden farklı indislere sahip illere erişilmiştir. Ayrıca len fonksiyonu kullanılarak liste uzunluğu ekrana yazdırılmıştır.

## 2. Liste öğelerini değiştirme örneği

Renkler isiminde bir liste oluşturalım ve ekrana aşağıdaki gibi yazdıralım.

```
>>> renkler=["Kırmızı","Mavi","Yeşil","Sarı","Mor"]
>>> print(renkler)
['Kırmızı', 'Mavi', 'Yeşil', 'Sarı', 'Mor']
...
```

Oluşturduğumuz listede renklerimizde değişiklik yapalım.

```
>>> renkler[0]="Turuncu"
>>> renkler[1]="Kahverengi"
>>> renkler[2]="Pembe"
>>> print(renkler)
['Turuncu', 'Kahverengi', 'Pembe', 'Sarı', 'Mor']
```

Yukarıda görüldüğü gibi listemizdeki renklerde değişiklik yaptık. Şimdi ise listemizdeki renkleri İngilizce karşılıklarıyla değiştirelim.

```
>>> renkler[0]="Oranage"
>>> renkler[1]="Brown"
>>> renkler[2]="Pink"
>>> renkler[3]="Yellow"
>>> renkler[0]="Purple"
>>> print(renkler)
['Purple', 'Brown', 'Pink', 'Yellow', 'Mor']
>>>
```

## 3. Liste öğelerini birleştirme örneği

```
>>> derlenen_diller=["C","C++","C#","Java"]
>>> yorumlanan_diller=["Python","Perl","Ruby"]
>>> print(derlenen_diller)
['C', 'C++', 'C#', 'Java']
>>> print(yorumlanan_diller)
['Python', 'Perl', 'Ruby']
```

Yukarıdaki örneğimizde iki listemiz tanımlanmış ve ekrana yazdırılmıştır. Bu iki listemizi programlama dilleri isminde yeni listemizde birleştirelim.

```
>>> programlama_dilleri=derlenen_diller+yorumlanan_diller
>>> print(programlama_dilleri)
['C', 'C++', 'C#', 'Java', 'Python', 'Perl', 'Ruby']
```

Görüldüğü gibi iki listemizi tek liste halinde birleştirdik.

### Demetler

En başta da söylediğimiz gibi, demetlerle listeler birbirine çok benzer. Ama demetlerle listelerin birbirinden çok önemli bazı farkları da vardır. Bu iki veri tipi arasındaki en önemli fark, listelerin değiştirilebilir (*mutable*) bir veri tipi iken, demetlerin değiştirilemez (*immutable*) bir veri tipi olmasıdır. Yani tıpkı karakter dizileri gibi, demetler de bir kez tanımlandıktan sonra bunların üzerinde değişiklik yapmak mümkün değildir.

```
>>> demet=('Klavye','Fare','Monitör','Yazıcı')
>>> demet[0]
'Klavye'
```

Yukarıda bir demet tanımladık. Şimdi demette değişiklik yapmaya çalışalım.

```
>>> demet[0]='Tarayıcı'
Traceback (most recent call last):
  File "<pyshell#157>", line 1, in <module>
    demet[0]='Tarayıcı'
TypeError: 'tuple' object does not support item assignment
```

Görüldüğü gibi demetin 0. İndisinde değişikli yapmaya çalıştığımızda hata mesajı verdi. Program veri tipinin demet (tuple) olduğunu ve değişiklik yapılamayacağını belirtti.

## Uygulama 10. Hafta Ders Notları

### Kümeler

Kümeler python programlama dilinde kullanılan veri tiplerinden biridir. Matematikte kullanılan “küme” kavramıyla benzerdir. Matematikte kullanılan küme özelliklerinin tümünü taşır. Kesişim, birleşim ve fark gibi özellikler pythonda kullanılan kümeler içinde geçerlidir.

Küme oluşturmak için set() adlı özel bir fonksiyon kullanılır. Şimdi bir boş küme oluşturalım.

```
>>> boş_küme=set()
```

```
>>> küme=set(["kırmızı","sarı","yeşil"])
```

Şimdi ise renklerden oluşan bir küme tanımladık. Oluşturduğumuz kümede set fonksiyonunu içinde bir liste tanımladık. Bu kümeyi aşağıdaki gibi liste ve demet kullanarak tekrar oluşturalım.

```
>>> liste=["kırmızı","sarı","yeşil"]
```

```
>>> küme=(liste)
```

```
>>> print(küme)
```

```
['kırmızı', 'sarı', 'yeşil']
```

```
>>> demet=("kırmızı","sarı","yeşil")
```

```
>>> küme=set(demet)
```

```
>>> print(küme)
```

```
{'yeşil', 'kırmızı', 'sarı'}
```

Şimdi ise bir karakter dizisi kullanarak kümemizi oluşturalım.

```
>>> karakter_dizisi="Karakter Dizisi Kullanarak Küme Tanımlama"
```

```
>>> küme=set(karakter_dizisi)
```

```
>>> print(küme)
```

```
{'T', 's', 'i', 'l', 'a', 'z', 'u', ' ', 'r', 'ı', 'm', 't', 'e',  
'n', 'ü', 'K', 'k', 'D'}
```

Yukarıda görüldüğü gibi oluşturduğumuz kümenin elemanları karakter dizisinin harflerinden oluşmaktadır. Fakat dikkat etmemiz gereken nokta aynı harften birden fazla bulunduğu takdirde, sadece bunlardan bir tanesi küme elemanı olarak alınır.

Aynı kural listeler içinde geçerlidir. Aşağıdaki örneği inceleyelim.

```
>>> liste=["Malatya","Adıyaman","Sakarya","Bolu","Adıyaman","Bolu"]
```

```
>>> for i in set(liste):
```

```
    print(i)
```

```
Malatya
```

```
Bolu
```

```
Sakarya
```

```
Adıyaman
```

Örnekte tanımlanan listede 6 eleman bulunmaktadır fakat küme veri tipine dönüştürüldüğünde Adıyaman ve Bolu illerinin sadece birer tanesi kümeye eleman olarak dâhil edilmiştir.



## 1.1.Küme Metotları

### clear()

Küme elemanlarını temizlemek için kullanılır.

```
>>> küme=set("küme elemanlarını silme")
>>> print(küme)
{'s', 'i', 'l', 'a', ' ', 'r', 'ı', 'm', 'e', 'n', 'ü', 'k'}
```

### add()

Küme eleman eklemek için kullanılır.

```
>>> küme=set(["Python","Java","Perl"])
>>> küme.add("C#")
>>> küme
{'Java,Perl', 'C#', 'Python'}
```

### update()

Kümeye liste eklemek için kullanılır.

```
>>> küme=set(["Python","Java","Perl"])
>>> eklenecek=["C++","C#","PHP","ASP"]
>>> küme.update(eklenecek)
>>> print(küme)
{'C++', 'Java,Perl', 'ASP', 'Python', 'PHP', 'C#'}
```

### difference()

İki küme arasındaki farkı belirlemek için kullanılır.

```
>>> küme1=set(["Kırmızı","Yeşil", "Mavi","Sarı"])
>>> küme2=set(["Kırmızı","Sarı","Mor","Siyah"])
>>> küme1.difference(küme2)
{'Mavi', 'Yeşil'}
>>> küme2.difference(küme1)
{'Mor', 'Siyah'}
```

### discard()

Kümeden eleman çıkarmak için kullanılır.

```
>>> küme2
{'Mor', 'Sarı', 'Siyah', 'Kırmızı'}
>>> küme2.discard("Siyah")
>>> print(küme2)
{'Mor', 'Sarı', 'Kırmızı'}
```

## intersection()

Kümelerin kesişimini verir.

```
>>> küme1=set(["Kırmızı","Yeşil", "Mavi","Sarı"])
>>> küme2=set(["Kırmızı","Sarı","Mor","Siyah"])
>>> küme1.intersection(küme2)
{'Sarı', 'Kırmızı'}
```

Aynı işlemi & işareti ile de yapabiliriz.

```
>>> küme1&küme2
{'Sarı', 'Kırmızı'}
```

## isdisjoint()

Kümelerin kesişimlerinin boş küme olup olmadığını kontrol eder.

```
>>> küme1
{'Mavi', 'Yeşil', 'Sarı', 'Kırmızı'}
>>> küme2
{'Mor', 'Sarı', 'Siyah', 'Kırmızı'}
>>>

>>> küme1.isdisjoint(küme2)
False
```

False değeri kesişimin boş küme olmadığını gösterir.

## Issubset()

Bir kümenin başka bir kümenin alt kümesi olup olmadığını kontrol etmek için kullanılır.

```
>>> küme1
{'Mavi', 'Yeşil', 'Sarı', 'Kırmızı'}
>>> küme2
{'Mor', 'Sarı', 'Siyah', 'Kırmızı'}
>>>

>>> küme1.issubset(küme2)
False
```

Eğer alt kümesi değilse yukarıdaki gibi false değeri döndürür. Alt kümesi ise true değeri döndürür.

## İssuperset()

Bir kümenin diğerini kapsayıp kapsamadığını kontrol eden metottur.

```
>>> küme1
{'Mavi', 'Yeşil', 'Sarı', 'Kırmızı'}
>>> küme2
{'Mor', 'Sarı', 'Siyah', 'Kırmızı'}
>>> küme2.issuperset(küme1)
False
>>>
```

Eğer alt kümesi değilse yukarıdaki gibi false değeri döndürür. Alt kümesi ise true değeri döndürür.

## Union()

İki kümenin birleşimini sağlayan metottur.

```
>>> küme1
{'Mavi', 'Yeşil', 'Sarı', 'Kırmızı'}
>>> küme2
{'Mor', 'Sarı', 'Siyah', 'Kırmızı'}
>>> küme1.union(küme2)
{'Yeşil', 'Kırmızı', 'Sarı', 'Siyah', 'Mavi', 'Mor'}
```

## Symmetric\_difference()

Kümelerde ortak olmayan elemanları yazdıran metottur.

```
>>> küme1
{'Mavi', 'Yeşil', 'Sarı', 'Kırmızı'}
>>> küme2
{'Mor', 'Sarı', 'Siyah', 'Kırmızı'}
>>> küme1.symmetric_difference(küme2)
{'Mavi', 'Yeşil', 'Mor', 'Siyah'}
```

## Dondurulmuş Kümeler

Küme elemanlarında değişiklik yapılmasını istemediğimiz durumlarda dondurulmuş kümeler kullanılır. Dondurulmuş kümelerde set() yerine frozenset() fonksiyonu kullanılır.

```
>>> dondurulmus=frozenset(["bir","iki","üç","dört"])
>>> dondurulmus.add("beş")
Traceback (most recent call last):
  File "<pyshell#267>", line 1, in <module>
    dondurulmus.add("beş")
AttributeError: 'frozenset' object has no attribute 'add'
>>>
```

Örnekte görüldüğü gibi oluşturduğumuz kümeye “beş” elemanını eklediğimizde hata mesajı vermektedir. Hata mesajında dondurulmuş küme olduğu ve ekleme yapılamayacağı belirtilmiştir.

## Ek 13. OÇÖO Eğlenelim Etkinlikleri

### OÇÖO Pilot Uygulama 1. Hafta Eğlenelim Etkinliği

#### (Sürükle Bırak Etkinliği)

Bu etkinlikte amaç **Python Giriş** konusunun pekiştirilmesidir. Etkinliğin hazırlanmasında Moodle oyunlaştırma eklentilerinden **H5P Drag the Words** kullanılmıştır. Etkinlikte verilen metinde boş bırakılan alanlara sağ tarafta bulunan uygun metinlerin yerleştirilmesi gerekmektedir. Öğrenciler doğru yerleştirdikleri her metin için puan kazanmıştır.

#### Sürükle bırak etkinliği

Python programlama dilinin ilk sürümü 1989 yılında [ ] tarafından geliştirilen [ ], [ ], [ ] ve [ ] bir programlama dilidir.

İlk sürümü python 1.0 [ ] yılında yayınlandı. İlk başlarda pek tanınmayan python kısa sürede ilgi gördü.

Python 2.0 sürümü [ ] yılında yayınlandı ve 2001 yılında kar amacı gütmeyen [ ] kurularak Python 2.1 sürümü piyasaya sürüldü.

Python 3.0 sürümü ise [ ] tarihinde piyasaya sürüldü ve daha önceki sürümlerinden [ ] bir ürün olarak geliştirildi.

Kontrol et

yorumlanabilen

modüler

Guido van Rossum

Python Yazılım Vakfı

etkileşimli

3 Aralık 2008

bağımsız

1994

nesne yönelimli

2000

## OÇÖO Pilot Uygulama 2. Hafta Eğlenelim Etkinliği

### (Resmi Yorumla Etkinliği)

Bu etkinlikte amaç **Karakter Dizileri** konusunun pekiştirilmesidir. Etkinliğin hazırlanmasında Moodle oyunlaştırma eklentilerinden **H5P Flashcards** kullanılmıştır. Etkinlikte öğrencilere kod satırlarının bulunduğu 10 tane flash kart sunulmuş ve doğru ekran çıktılarını yazmaları istenmiştir. Doğru yazılan her ekran çıktısı için öğrencilere puan verilmiştir.

```
>>> ad="Kodlama"  
>>> print(ad)
```

Ekran çıktısı nedir?

Your answer

Check

## OÇÖO Uygulama 1. Hafta Eğlenelim Etkinliği

### (Hafıza Egzersizi Etkinliği)

Bu etkinlikte amaç **Değişkenler** konusunun pekiştirilmesidir. Etkinliğin hazırlanmasında Moodle oyunlaştırma eklentilerinden **H5P Memory Game** kullanılmıştır. Etkinlikte öğrencilerin resimlerde 2 adet olarak sunulan değişken türlerini bulmaları istenmiştir. Verilen süre içerisinde eşleştirmeyi tamamlayan öğrencilere puan verilmiştir.



## OÇÖO Uygulama 2. Hafta Eğlenelim Etkinliği

### (Bilgi Yarışması Etkinliği)

Bu etkinlikte amaç **Print Fonksiyonu** konusunun pekiştirilmesidir. Etkinliğin hazırlanmasında Moodle oyunlaştırma eklentilerinden **Bilgi Yarışması** kullanılmıştır. Etkinlikte öğrencilere print fonksiyonunun kullanımıyla ilgili 10 tane soru yöneltilmiştir. Öğrencilerin ise eklentide yer alan uçak ile doğru seçeneği vurmaları istenmiştir. Doğru olarak vurulana her soru için puan verilmiştir.





## OÇÖO Uygulama 3. Hafta Eğlenelim Etkinliği

### (Kod Bulmaca Etkinliği)

Bu etkinlikte amaç **Kontrol Yapıları** konusunun pekiştirilmesidir. Etkinliğin hazırlanmasında Moodle oyunlaştırma eklentilerinden **H5P Drag the Words** kullanılmıştır. Etkinlikte öğrencilere bir problem durumu ve problemin çözümüne yönelik kod satırları sunulmuştur. Öğrencilerden programın doğru olarak çalışması için gerekli kod satırlarını yerleştirmeleri istenmiştir. Programı doğru olarak çalıştıran öğrenciler puan kazanmıştır.

Kullanıcının 1 ile 7 arasında bir sayı girmesi istenmektedir. Girilen sayının haftanın hangi günü olduğunu bulan program aşağıda verilmiştir. Boş bırakılan yerlere uygun kodları yerleştiriniz.

sayi=  ("1-7 arasında bir sayı giriniz (1 ve 7 dahil)")

if  >=1  sayi<=7:

if sayi==1

print("Pazartesi")

if sayi==2:

print("  ")

if sayi==3:

print("  ")

if sayi==4:

print("Perşembe")

if sayi==5:

print("  ")

if sayi==6:

print("Cumartesi")

if sayi==7:

print("  ")

:

print("Geçersiz bir değer girdiniz")

sayi

Salı

Çarşamba

Pazar

Cuma

and

else

:

int

input

## OÇÖO Uygulama 4. Hafta Eğlenelim Etkinliği

### (Kod Asmaca Etkinliği)

Bu etkinlikte amaç **Döngüler** konusunun pekiştirilmesidir. Etkinliğin hazırlanmasında Moodle oyunlaştırma eklentilerinden **Adam Asmaca** kullanılmıştır. Etkinlikte öğrencilere verilen metinleri tahmin etmeleri istenmiştir. Doğru olarak tahmin edilen her metin için puan kazanılmıştır.

Pythonda kullanılan döngü türlerinden biridir?



Sizin 6 denemeniz var

Harfler: ABCÇDEFGĞHIİJKLMNOÖPQRSŞTUÜV**W**XYZ

## OÇÖO Uygulama 5. Hafta Eğlenelim Etkinliği

### (Şifreli Bulmaca Etkinliği)

Bu etkinlikte amaç **Fonksiyonlar** konusunun pekiştirilmesidir. Etkinliğin hazırlanmasında Moodle oyunlaştırma eklentilerinden **Şifreli Bulmaca** kullanılmıştır. Etkinlikte öğrencilere sorulan soruları cevaplayarak bulmacada işaretlemeleri istenmiştir. Soruları doğru cevaplayan ve bulmacada işaretleyen öğrenciler puan kazanmıştır.

Puan 0 %

M	P	N	K	N	K	F
Ü	A	M	E	Ü	S	O
A	R	G	Ü	M	A	N
R	A	A	M	E	P	K
G	M	R	E	N	I	S
D	E	F	E	I	K	I
E	T	K	T	A	R	Y
F	R	Y	T	R	E	O
R	E	D	E	A	N	N

Oyun bitti

Yazdır

1. Fonksiyon çağrıldığında parantez içinde kullanılan değerlere ne ad verilir?

Yanıt

2. Fonksiyon tanımlamak için hangi ifade kullanılır

Yanıt

3. Python'da tekrar tekrar kullanabildiğimiz kod kümelerine ne ad verilir?

Yanıt

4. Python'da kaç çeşit fonksiyon bulunmaktadır?

Yanıt




5. Tanımlanan fonksiyonlarda parantez içindeki değerlere ne ad verilir?

Yanıt

## OÇÖO Uygulama 6. Hafta Eğlenelim Etkinliği

### (Milyoner Etkinliği)

Bu etkinlikte amaç **Parametre Alan Fonksiyonlar** konusunun pekiştirilmesidir. Etkinliğin hazırlanmasında Moodle oyunlaştırma eklentilerinden **Milyoner** kullanılmıştır. Etkinlikte öğrencilere sorular yöneltilmiş ve öğrencilerin cevaplamaları sağlanmıştır. İlk soruyu doğru cevaplayan öğrenci puan kazanmış ve bir sonraki soruyu cevaplama hakkı kazanmıştır. Sorular tamamlanana kadar süreç bu şekilde devam etmiştir.

50:50   

Fonksiyonlarla ilgili verilen ifadelerden hangisi yanlıştır?

15	150000
14	80000
13	40000
12	20000
11	10000
10	5000
9	4000
8	2000
7	1500
6	1000
5	500
4	400
3	300
2	200
<b>1</b>	<b>100</b>

A Fonksiyon isimlerinde Türkçe karakter kullanılmamalıdır.

B Fonksiyonlar gömülü ve özel fonksiyonlar olmak üzere ikiye ayrılır.

C Fonksiyon tanımlarken alt satırda girintili yazılan kodlar fonksiyonun gövdesini oluşturur.

Ç print() fonksiyonu özel bir fonksiyondur.

## OÇÖO Uygulama 7. Hafta Eğlencim Etkinliği

### (Sudoku Etkinliği)

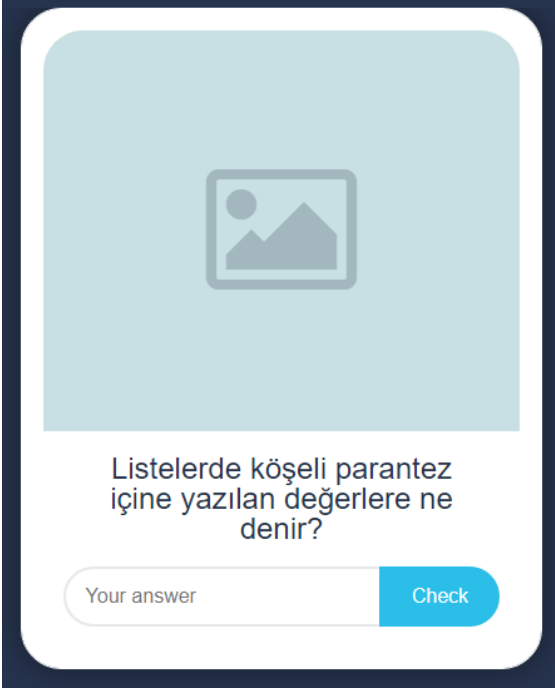
Bu etkinlikte amaç **Değer Döndüren ve Argümanlı Fonksiyonlar** konusunun pekiştirilmesidir. Etkinliğin hazırlanmasında Moodle oyunlaştırma eklentilerinden **Sudoku** kullanılmıştır. Etkinlikte öğrenciler soruları cevaplayarak sudokunun çözümüne yönelik ipuçları elde etmiş ve sudokunu tamamlayan öğrenciler puan kazanmıştır.

<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	7	<input type="text"/>	5	6	<input type="text"/>
1	2	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	5	4	2	3	<input type="text"/>	A27
5	<input type="text"/>	9	A37	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	1
<input type="text"/>	<input type="text"/>	4	2	<input type="text"/>	9	<input type="text"/>	<input type="text"/>	5
<input type="text"/>	<input type="text"/>	2	1	8	A45	<input type="text"/>	<input type="text"/>	7
9	3	<input type="text"/>	<input type="text"/>	<input type="text"/>	4	<input type="text"/>	A74	<input type="text"/>
A58	<input type="text"/>	<input type="text"/>	<input type="text"/>	2	6	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	4	9	<input type="text"/>

## OÇÖO Uygulama 8. Hafta Eğlenelim Etkinliği

### (Cevabı Sende Etkinliği)

Bu etkinlikte amaç **Liste Veri Tipi** konusunun pekiştirilmesidir. Etkinliğin hazırlanmasında Moodle oyunlaştırma eklentilerinden **H5P Flashcards** kullanılmıştır. Etkinlikte öğrencilere liste veri tipiyle ilgili soruların bulunduğu 10 tane flash kart sunulmuş ve doğru cevapları yazmaları istenmiştir. Doğru yazılan her cevap için öğrencilere puan verilmiştir.



Listelerde köşeli parantez içine yazılan değerlere ne denir?

## OÇÖO Uygulama 9. Hafta Eğlenelim Etkinliđi

### (Metot Bulmaca Etkinliđi)

Bu etkinlikte amaç **Liste Fonksiyonları** konusunun pekiştirilmesidir. Etkinliđin hazırlanmasında Moodle oyunlařtırma eklentilerinden **Adam Asmaca** kullanılmıřtır. Etkinlikte öğrencilerin liste metotlarını bulmaları istenmiřtir. Dođru olarak bulunan her metot için puan kazanılmıřtır.

Listelerden indis numarasıyla eleman silmek için hangi metot kullanılır?



Sizin 6 denemeniz var

---

Harfler: **ABCDEFGHIJKLMNOPQRSTUVWXYZ**

## OÇÖO Uygulama 10. Hafta Eğlencelim Etkinliđi

### (Küme Hafızası Etkinliđi)

Bu etkinlikte amaç **Küme Veri Tipi** konusunun pekiştirilmesidir. Etkinliđin hazırlanmasında Moodle oyunlařtırma eklentilerinden **H5P Memory Game** kullanılmıřtır. Etkinlikte öđrencilerin resimlerde 2 adet olarak sunulan küme metotlarını bulmaları istenmiřtir. Verilen süre ierisinde eřleřtirmeyi tamamlayan öđrencilere puan verilmiřtir.





## Ek 14. Kodlama Deneyimi Etkinlikleri

<b>Pilot Uygulama 1. Hafta Kodlama Deneyimi Etkinlikleri</b>	
1.	Python programlama dilinin 2.x ve 3.x olmak üzere iki sürümü bulunmaktadır. <input type="checkbox"/> Doğru <input type="checkbox"/> Yanlış
2.	Yorumlayıcı ve derleyici arasında fark bulunmamaktadır. <input type="checkbox"/> Doğru <input type="checkbox"/> Yanlış
3.	Python çok güçlü kütüphanelere sahiptir. <input type="checkbox"/> Doğru <input type="checkbox"/> Yanlış
4.	Jupyter-iPython programını kullanarak python kodlarını yazabilirsiniz. <input type="checkbox"/> Doğru <input type="checkbox"/> Yanlış
5.	Python windows işletim sisteminde çalışmaz. <input type="checkbox"/> Doğru <input type="checkbox"/> Yanlış
6.	Python programlama dilinde kodlar derlenmeden çalıştırılır. <input type="checkbox"/> Doğru <input type="checkbox"/> Yanlış

## Pilot Uygulama 2. Hafta Kodlama Deneyimi Etkinlikleri

1.	"Python ile kodlama" karakter dizisinin tipini öğrenmek için kullanılacak kodu yazınız.								
2.	'www.kodlamaogreniyorum.net' ekran çıktısını IDLE ortamında çift tırnak (") kullanarak yazan kod satırını yazınız.								
3.	Karakter(strings) dizilerinin özelliklerini yazınız. 1. 2. 3. 4. 5.								
4.	Aşağıda verilen veri tiplerini eşleştirelim. <table border="1" data-bbox="395 1377 740 1733"><tr><td>2018</td></tr><tr><td>3.14</td></tr><tr><td>True- False</td></tr><tr><td>"Python"</td></tr></table> <table border="1" data-bbox="928 1377 1310 1733"><tr><td>Float</td></tr><tr><td>Strings</td></tr><tr><td>Integer</td></tr><tr><td>Mantıksal</td></tr></table>	2018	3.14	True- False	"Python"	Float	Strings	Integer	Mantıksal
2018									
3.14									
True- False									
"Python"									
Float									
Strings									
Integer									
Mantıksal									

## Uygulama 1. Hafta Kodlama Deneyimi Etkinlikleri

1.	Silindirin hacmini <b>değişken tanımlayarak</b> hesaplayan ve ekrana yazdıran programı yazınız. (pi=3, r=4, h=5, hacim)
2.	Belirtilen ad, soyad ve dogumtarihi verilerini <b>değişken tanımlayarak</b> ekrana alt alta yazdıran programı yazınız. ad="Python" soyad="Kodlama" dogumtarihi=2018
3.	" <b>Python Kodlama</b> " ifadesinin <b>değişken uzunluğunu</b> <i>len()</i> fonksiyonunu kullanarak hesaplayınız.
4.	<pre>&gt;&gt;&gt; a=4 &gt;&gt;&gt; b=8 &gt;&gt;&gt; c="12" &gt;&gt;&gt; d="25" &gt;&gt;&gt; print (a+b+c+d) Traceback (most recent call last):   File "&lt;pyshell#5&gt;", line 1, in &lt;module&gt;     print (a+b+c+d) TypeError: unsupported operand type(s) for +: 'int' and 'str'</pre> <p>Yukarıdaki program çalıştırıldığında hata vermektedir. Programın düzgün çalışması için gerekli düzenlemeleri yaparak programı tekrar yazınız.</p>

## Uygulama 2. Hafta Kodlama Deneyimi Etkinlikleri

1.	<p>Ad, soyad, vize ve final notlarını “notlar.txt” dosyasına yazdıran programı yazınız.</p> <p>Değişkenler</p> <p>ad=Ahmet</p> <p>soyad=YAŞAR</p> <p>vize=60</p> <p>final=80</p>
2.	<p>“<i>Malatya, Ankara, Gaziantep, Adana, Bolu</i>” illerini aralarına &amp; işareti koyarak ekrana yazdıran print fonksiyonunu yazınız.</p>
3.	<p>“<i>Malatya, Ankara, Gaziantep, Adana, Bolu</i>” illerini sep parametresi kullanarak alt alta yazan kod satırını yazınız.</p>
4.	<p>“<b>Python Merhaba 2018</b>” tırnak içindeki ifadeyi print() fonksiyonunu kullanarak ekrana yazdıralım.</p>

### Uygulama 3. Hafta Kodlama Deneyimi Etkinlikleri

1.	<p>Girilen yaş değerine göre yaşı 18 ve üzeri olanlar için "Yas Uygundur", yaşı 18 den küçük olanlar için "Yas Uygun Degildir" uyarısı yazan if-else kotrol bloğunu yazınız.</p> <p>Değişken: yas</p>
2.	<p>Girilen sayı çift olduğunda ekrana "Sayi Cifttir", tek olduğunda ise "Sayi Tektir" yazan programı yazınız.</p> <p>Değişken: sayi</p>
3.	<p>Bir dersten geçme notu 50 olarak belirlenmiş olsun. Kullanıcıya notunu sorup dersten geçme-kalma durumunu ekrana yazan programı yazınız.</p> <p>Not:</p> <p>0-50 arasında ekrana Kaldi</p> <p>50-100 arasında ekrana Gecti</p> <p>0-100 arasında bir sayı girmediği zaman ekrana Yanlis Giris yazılacak</p> <p>Değişken: alinannot</p>

#### Uygulama 4. Hafta Kodlama Deneyimi Etkinlikleri

1.	<p>"ANKARA" kelimesinin harflerini alt alta ekrana yazdıran programı yazınız.</p> <p>Değişkenler: ad, harfler</p>
2.	<p>1-10 arasındaki sayıları range fonksiyonunu ve for döngüsünü kullanarak ekrana yazdıran programı yazınız.</p> <p>Değişkenler: sayı</p>
3.	<p>Kullanıcıdan 3 karakterden kısa ve 8 karakterden uzun olmayan bir parola belirlemesini isteyen programı while döngüsü ve break deyimi kullanarak yazınız.</p> <p>Not: Parola boş geçilirse uyarı verecek, uygun parola girilmezse yeniden parola girilmesi istenecek, uygun parola girilirse yeni parola oluşturulduğuna dair uyarı verilecektir.</p> <p>Değişkenler: sayı</p> <p>Ekran Çıktısı</p> <pre>1234567 Yeni Parolanız 1234567  123456789 parola 8 karakterden uzun 3 karakterden kısa olmamalı  parola bölümü boş geçilemez parola bölümü boş geçilemez</pre>

## Uygulama 5. Hafta Kodlama Deneyimi Etkinlikleri

1.	"Python, Kodlama, Programlama" kelimelerini sırasıyla ekrana alt alta yazdıran <b>fonk</b> isminde bir fonksiyon yazınız.
2.	Dairenin alanını hesaplayan ve ekrana yazdıran dairealan isminde bir fonksiyon tanımlayınız. Değişkenler: alan, r=4, pi=3
3.	Kullanıcıdan alınan kullanıcı adı ve parola bilgisini tanımlanan kullanıcı adı ve parola bilgisiyle eşleştiren giris isminde bir fonksiyon yazınız. Eğer kullanıcı adı ve parola doğruysa ekrana doğru olduğuna dair bir uyarı versin. Değilse yanlış olduğuna dair bir uyarı versin. Not: kullaniciadi="python", parola="123456"

## Uygulama 6. Hafta Kodlama Deneyimi Etkinlikleri

1.	<pre>def alan(uk, kk) :     alan=uk*kk     print ("Alan:", alan) alan(20,10)</pre> <p>Yukarıdaki programın ekran çıktısı nedir?</p>
2.	<pre>def durum(ad, n1, n2) :     ortalama=(n1+n2)/2     if (ortalama&gt;=50):         print (ad, ":", "Geçti")     else:         print (ad, ":", "Kaldı") durum("Mehmet", 45, 65)</pre> <p>Yukarıdaki programın ekran çıktısı nedir?</p>
3.	<pre>def durum(ad, n1, n2) :     ortalama=(n1+n2)/2     if (ortalama&gt;=50):         print (ad, ":", "Geçti")     else:         print (ad, ":", "Kaldı") durum(ad="Ayşe", n1=45, n2=50)</pre> <p>Yukarıdaki programın ekran çıktısı nedir?</p>



## Uygulama 7. Hafta Kodlama Deneyimi Etkinlikleri

1.	Fonksiyon içinde tanımlanan değişkenleri fonksiyon dışında da kullanabilmek için hangi deyim kullanılır?
2.	Fonksiyonlarda değer döndürmek için hangi deyim kullanılır?
3.	Programın herhangi bir durumda hiç bir işlem yapmasını istemiyorsak hangi deyimi kullanmamız gerekmektedir?
4.	<pre>def carpma(*degerler):     carpim=1     for i in degerler:         carpim*=i     print (carpim) carpma (3,4,5,6)</pre> <p>Yukarıdaki programın ekran çıktısı nedir?</p>

## Uygulama 8. Hafta Kodlama Deneyimi Etkinlikleri

1.	<b>Ocak, Şubat, Mart, Nisan</b> aylarından oluşan 4 elemanlı <b>aylar</b> isminde bir liste oluşturunuz?												
2.	<pre>gunler=["Pazartesi","Salı","Çarşamba","Perşembe"] gunler.append('Cuma') gunler.append('Cumartesi') gunler.append('Pazar')</pre> <p>Yukarıdaki listenin son halinin yazınız?</p>												
3.	<p>Aşağıda verilen liste metotlarını görevleriyle eşleştiriniz?</p> <table border="1"><tr><td>Del()</td><td>Listeden indis numarasıyla eleman silme</td></tr><tr><td>Remove()</td><td>Liste içerisinde arama yapma</td></tr><tr><td>Reverse()</td><td>Listeye eleman ekleme</td></tr><tr><td>Index()</td><td>Liste elemanlarını alfabetik sıralama</td></tr><tr><td>Append()</td><td>Listeden eleman silme</td></tr><tr><td>Sort()</td><td>Liste elemanlarının sırasını tersine çevirme</td></tr></table>	Del()	Listeden indis numarasıyla eleman silme	Remove()	Liste içerisinde arama yapma	Reverse()	Listeye eleman ekleme	Index()	Liste elemanlarını alfabetik sıralama	Append()	Listeden eleman silme	Sort()	Liste elemanlarının sırasını tersine çevirme
Del()	Listeden indis numarasıyla eleman silme												
Remove()	Liste içerisinde arama yapma												
Reverse()	Listeye eleman ekleme												
Index()	Liste elemanlarını alfabetik sıralama												
Append()	Listeden eleman silme												
Sort()	Liste elemanlarının sırasını tersine çevirme												

## Uygulama 9. Hafta Kodlama Deneyimi Etkinlikleri

<b>1.</b>	<pre>liste=["Sarı","Kırmızı","Mavi"]</pre> <p>Yukarıdaki listenin elemanlarını sırasıyla "Siyah", "Mor", "Beyaz" olarak değiştirecek kodu yazınız?</p>
<b>2.</b>	<pre>metotlar1=["append","sort","reverse"] metotlar=["remove","del","index"] tummetotlar=metotlar1+metotlar2 print(tummetotlar)</pre> <p>Yukarıdaki programın ekran çıktısı ne olur?</p>
<b>3.</b>	<p>Elma, Armut, Kayısı, Karpuz, Muz</p> <p>Yukarıdaki verilen meyvelerden meyve isminde bir demet tanımlayınız?</p>

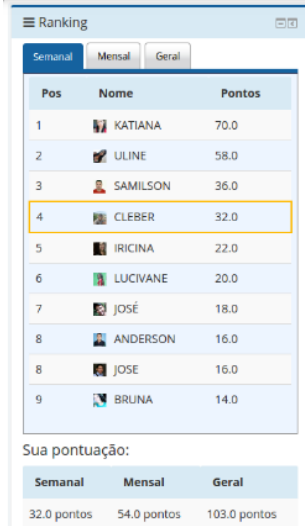
## Uygulama 10. Hafta Kodlama Deneyimi Etkinlikleri

1.	<p>Keyboard, Mouse, Printer, Scanner</p> <p>Yukarıdaki verilen donanım birimlerinden donanım adında bir küme oluşturmak için gerekli kodları yazınız?</p>												
2.	<pre>liste=["clear","add","discard","update"] metotlar=set(liste) print(metotlar)</pre> <p>Yukarıdaki verilen donanım birimlerinden donanım adında bir küme oluşturmak için gerekli kodları yazınız?</p>												
3.	<p>Aşağıda verilen küme metotlarını görevleriyle eşleştiriniz?</p> <table border="1" data-bbox="338 1149 549 1680"><tr><td>intersection()</td></tr><tr><td>clear()</td></tr><tr><td>union()</td></tr><tr><td>update()</td></tr><tr><td>add()</td></tr><tr><td>discard()</td></tr></table> <table border="1" data-bbox="775 1149 1367 1680"><tr><td>Kümeden eleman çıkarma</td></tr><tr><td>İki kümenin birleşimini sağlama</td></tr><tr><td>Kümeye eleman ekleme</td></tr><tr><td>Küme elemanlarını temizleme</td></tr><tr><td>Kümeye liste ekleme</td></tr><tr><td>Küme kesişimini bulma</td></tr></table>	intersection()	clear()	union()	update()	add()	discard()	Kümeden eleman çıkarma	İki kümenin birleşimini sağlama	Kümeye eleman ekleme	Küme elemanlarını temizleme	Kümeye liste ekleme	Küme kesişimini bulma
intersection()													
clear()													
union()													
update()													
add()													
discard()													
Kümeden eleman çıkarma													
İki kümenin birleşimini sağlama													
Kümeye eleman ekleme													
Küme elemanlarını temizleme													
Kümeye liste ekleme													
Küme kesişimini bulma													

## Ek 15. OÇÖO Oyunlaştırma Yönergesi

Merhaba arkadaşlar. Öğrenme ortamı olarak kullanacağınız oyunlaştırılmış çevrimiçi öğrenme ortamı size oyun tadında ve eğlenerek öğrenmenize olanak sağlayacaktır. Dersi bir oyun olarak düşünün ve oyunu başarıyla tamamlamak için gayret gösterin. Gayret gösterdikçe, hem kazandığının hem de öğrendiğinizin farkına varacaksınız. Sadece aşağıdaki yönergeleri dikkatlice okuyun ve uygulayın.

1. Çevrimiçi öğrenme ortamında katıldığınız her etkinlikten puan kazanacaksınız. Puan kazanacağınız etkinlikler aşağıdaki gibidir;
  - a. Ödevleri yapma
  - b. Ders izlencesini okuma
  - c. Ders notlarını inceleme
  - d. Ders videolarını izleme
  - e. Eğlenceli etkinliklere katılım
  - f. Kodlama deneyimine katılım
  - g. Forum katılımı
2. Aldığınız puanlar **LİDER TAHTASI** bloğunda yer alacaktır. Sıralama bloğu aşağıdaki gibidir.




Pos	Nome	Pontos
1	KATIANA	70.0
2	ULINE	58.0
3	SAMILSON	36.0
4	CLEBER	32.0
5	IRICINA	22.0
6	LUCIVANE	20.0
7	JOSÉ	18.0
8	ANDERSON	16.0
8	JOSE	16.0
9	BRUNA	14.0

Sua pontuação:

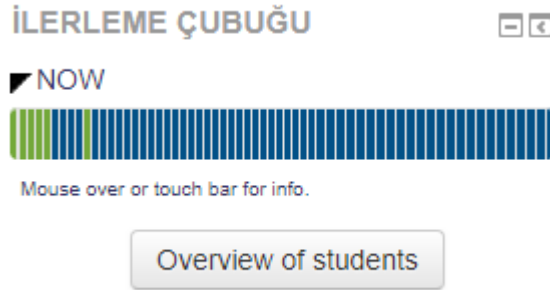
Semanal	Mensal	Geral
32.0 pontos	54.0 pontos	103.0 pontos

3. Diğer puanlama sistemi ise deneyim puanıdır. Ders sitemizi kullanım deneyiminize ve kullanım sıklığınıza göre deneyim puanları kazanacaksınız. Kazandığınız puanlara göre seviyeler, seviyelere göre ise rütbelere kazanacaksınız. Seviyelerinizi ve

rütbelerinizi **SEVİYELERİNİZ** alanında görebileceksiniz. Seviyeler ve rütbeler aşağıdaki tabloda verilmiştir.

Seviye	Deneyim Puanı (xp)	Rütbeler	
Level 1	0	-	
Level 2	500	Yeni Kodlayıcı	
Level 3	1000	Acemi Kodlayıcı	
Level 4	2000		
Level 5	2500	Kodlayıcı	
Level 6	3000		
Level 7	3500	İyi Kodlayıcı	
Level 8	4000		
Level 9	6000	Uzman Kodlayıcı	
Level 10	6500		

4. Derslerdeki ilerlemenizi **İLERLEME ÇUBUĞU** kısmında görebilirsiniz.



5. Yaptığınız bazı davranışlar size özel rozetler kazandıracaktır. Rozetlerinize **SON ROZETLERİM** kısmından ulaşabilirsiniz. Bu rozetler;



**Kodcu:** Kodlama deneyimi etkinliklerinden başarılı olan öğrencilere verilecektir.



**Düzenli:** Haftalık olarak tüm etkinliklerde başarı gösteren ve etkin katılım sağlayan öğrencilere verilecektir.



**Etkinlik:** Eğlenceli etkinliklerinde tam başarı gösteren öğrencilere verilecektir.



**İşbirlikçi:** İşbirliği ortamı forumuna katılım gösteren öğrencilere verilecektir.

6. Derslerinizi aşamalı olarak sürdürmeniz gerekmektedir. Bir aşamayı tamamlamadan diğer aşamaya geçemeyeceksiniz. Ayrıca bir haftanın etkinliklerini tamamlamadan diğer haftanın etkinliklerine geçemeyeceksiniz. Kısıtlamalar ekranda görüldüğü gibidir;

### 1.Hafta (Python Giriş)

	İşbirliği Ortamı	<input checked="" type="checkbox"/>
	1. Hafta Ödevi	<input checked="" type="checkbox"/>
	Pythona giriş dersi izlencesi	<input checked="" type="checkbox"/>
	Sürece izin verilmez: The activity 1. Hafta Ödevi is marked complete	
	Python Giriş Ders Notları	<input checked="" type="checkbox"/>
	Sürece izin verilmez: The activity Pythona giriş dersi izlencesi is marked complete	
	Pythona Giriş Videosu	<input checked="" type="checkbox"/>
	Sürece izin verilmez: The activity Python Giriş Ders Notları is marked complete	
	Eğlenceli (Sürükle-Bırak)	<input checked="" type="checkbox"/>
	Sürece izin verilmez: The activity Pythona Giriş Videosu is marked complete	
	Kodlama Deneyimi (python giriş)	<input type="checkbox"/>
	Sürece izin verilmez: The activity Eğlenceli (Sürükle-Bırak) is marked complete	

7. Videolarda **Kodlama Deneyimi** etkinliğine girebilmeniz için bir şifre bulunmaktadır. Videoyu dikkatli seyrederek şifreyi öğreniniz. Kodlama deneyimi etkinliklerinde aşağıdaki görselde gördüğünüz gibi şifre istenmektedir.

Uygulamayı başlat

Şifre

Bu sınavı uygulayabilmeniz için sınav parolasını bilmeniz gerekiyor





Sınav şifresi

8. Belirli durumlarda gerçek ödül kazanılacaktır. Bu durumlar aşağıdaki gibidir;
- Lider tahtasında her ay birinci olan öğrenciler gerçek ödül kazanacaktır.
  - Kodlama deneyimi etkinlikleri sonucu toplamda **8 kodcu rozeti** kazananlar gerçek ödül ile ödüllendirilecektir.

- c. Haftalık tüm etkinlikleri sonucu toplamda **8 düzenli rozeti** kazananlar gerçek ödül ile ödüllendirilecektir.
- d. Eğlenceli etkinlikleri sonucunda **8 etkinlik rozeti** kazananlar gerçek ödül ile ödüllendirilecektir.
- e. İşbirliği forumuna katılım gösteren **8 işbirlikçi rozeti** kazananlar gerçek ödül ile ödüllendirilecektir.



**Ek 16. OÇÖO Rozet Durumu**

Sıra No	Öğrenci Kodu	 Kodcu	 İşbirlikçi	 Düzenli	 Etkinlikçi
1	OÇÖO-1	5	6	5	9
2	OÇÖO-2	5	7	6	8
3	OÇÖO-3	3	5	4	5
4	OÇÖO-4	6	5	6	7
5	OÇÖO-5	8	5	8	9
6	OÇÖO-6	6	6	7	7
7	OÇÖO-7	8	8	8	9
8	OÇÖO-8	6	8	7	7
9	OÇÖO-9	6	6	7	8
10	OÇÖO-10	4	6	5	5
11	OÇÖO-11	8	8	8	9
12	OÇÖO-12	5	5	6	6
13	OÇÖO-13	7	9	7	9
14	OÇÖO-14	5	2	6	7
15	OÇÖO-15	5	7	5	8
16	OÇÖO-16	5	6	6	6
17	OÇÖO-17	8	8	8	9
18	OÇÖO-18	10	9	9	9
19	OÇÖO-19	4	4	5	5
20	OÇÖO-20	4	5	5	7
21	OÇÖO-21	7	7	7	7
22	OÇÖO-22	6	7	7	7
23	OÇÖO-23	6	5	7	8
24	OÇÖO-24	6	8	6	8
25	OÇÖO-25	6	8	6	9
26	OÇÖO-26	5	7	5	8
27	OÇÖO-27	7	10	6	10
28	OÇÖO-28	2	4	2	8
29	OÇÖO-29	5	7	5	8
30	OÇÖO-30	6	7	6	7
31	OÇÖO-31	6	5	7	8
32	OÇÖO-32	7	7	7	9
33	OÇÖO-33	4	4	5	5

**Ek 17. OÇÖO Seviye ve Rütbe Durumu**

Sıra No	Öğrenci Kodu	Puan	Seviye	Rütbe
1	OÇÖO-1	5790	Level 8	İyi Kodlayıcı
2	OÇÖO-2	7776	Level 10	Uzman Kodlayıcı
3	OÇÖO-3	4902	Level 8	İyi Kodlayıcı
4	OÇÖO-4	3582	Level 7	İyi Kodlayıcı
5	OÇÖO-5	4935	Level 8	İyi Kodlayıcı
6	OÇÖO-6	2745	Level 5	Kodlayıcı
7	OÇÖO-7	6846	Level 10	Uzman Kodlayıcı
8	OÇÖO-8	6486	Level 9	Uzman Kodlayıcı
9	OÇÖO-9	4173	Level 8	İyi Kodlayıcı
10	OÇÖO-10	3294	Level 6	Kodlayıcı
11	OÇÖO-11	6501	Level 10	Uzman Kodlayıcı
12	OÇÖO-12	4548	Level 8	İyi Kodlayıcı
13	OÇÖO-13	4251	Level 8	İyi Kodlayıcı
14	OÇÖO-14	2778	Level 5	Kodlayıcı
15	OÇÖO-15	3924	Level 7	İyi Kodlayıcı
16	OÇÖO-16	2793	Level 5	Kodlayıcı
17	OÇÖO-17	7194	Level 10	Uzman Kodlayıcı
18	OÇÖO-18	10005	Level 10	Uzman Kodlayıcı
19	OÇÖO-19	4767	Level 8	İyi Kodlayıcı
20	OÇÖO-20	6639	Level 10	Uzman Kodlayıcı
21	OÇÖO-21	2937	Level 5	Kodlayıcı
22	OÇÖO-22	3051	Level 6	Kodlayıcı
23	OÇÖO-23	2643	Level 5	Kodlayıcı
24	OÇÖO-24	4503	Level 8	İyi Kodlayıcı
25	OÇÖO-25	7698	Level 10	Uzman Kodlayıcı
26	OÇÖO-26	4122	Level 8	İyi Kodlayıcı
27	OÇÖO-27	8439	Level 10	Uzman Kodlayıcı
28	OÇÖO-28	4956	Level 8	İyi Kodlayıcı
29	OÇÖO-29	4743	Level 8	İyi Kodlayıcı
30	OÇÖO-30	5676	Level 8	İyi Kodlayıcı
31	OÇÖO-31	3096	Level 6	Kodlayıcı
32	OÇÖO-32	8250	Level 10	Uzman Kodlayıcı
33	OÇÖO-33	3126	Level 6	Kodlayıcı

## Ek 18. OYYÖO Eğlenelim Etkinlikleri

### OYYÖO Pilot Uygulama 1. Hafta Eğlenelim Etkinliđi

#### (Kes Yapıştır Etkinliđi)

Bu etkinlikte amaç **Küme Veri Tipi** konusunun pekiştirilmesidir. Etkinliđin hazırlanmasında Moodle oyunlaştırma eklentilerinden **H5P Memory Game** kullanılmıştır. Etkinlikte öğrencilerin resimlerde 2 adet olarak sunulan küme metotlarını bulmaları istenmiştir. Verilen süre içerisinde eşleştirmeyi tamamlayan öğrencilere puan verilmiştir.

## OYYÖO Pilot Uygulama 2. Hafta Eğlenelim Etkinliği

### (Çek Kodunu Söyle Çıktını Etkinliği)

Bu etkinlikte amaç **Karakter Dizileri** konusunun pekiştirilmesidir. Etkinlikte öncelikli olarak karakter dizileri ve aritmetiksel operatörlerle ilgili python kod kartları hazırlanmıştır. Daha sonra bu kod kartları bir kutunun içine konularak ders esnasında öğrencilerin birer tane çekmesi istenmiştir. Kartını çeken öğrenci kartta yer alan kodun ekran çıktısını söylemiş ve doğru söyleyen öğrenciler puan kazanmıştır. Etkinlikte kullanılan örnek python kodları aşağıda sunulmuştur.

```
>>> ad="Kodlama"  
>>> print(ad)
```

```
>>> ad="Python"  
>>> soyad="Kodlama"  
>>> print(ad,soyad)
```

```
>>> 125//10  
12
```

```
>>> 3**4
```

```
>>> "36"*5
```

```
>>> 136%13
```

```
>>> "Anadolu"+"Lisesi"
```

```
>>> yas=25  
>>> type(yas)
```

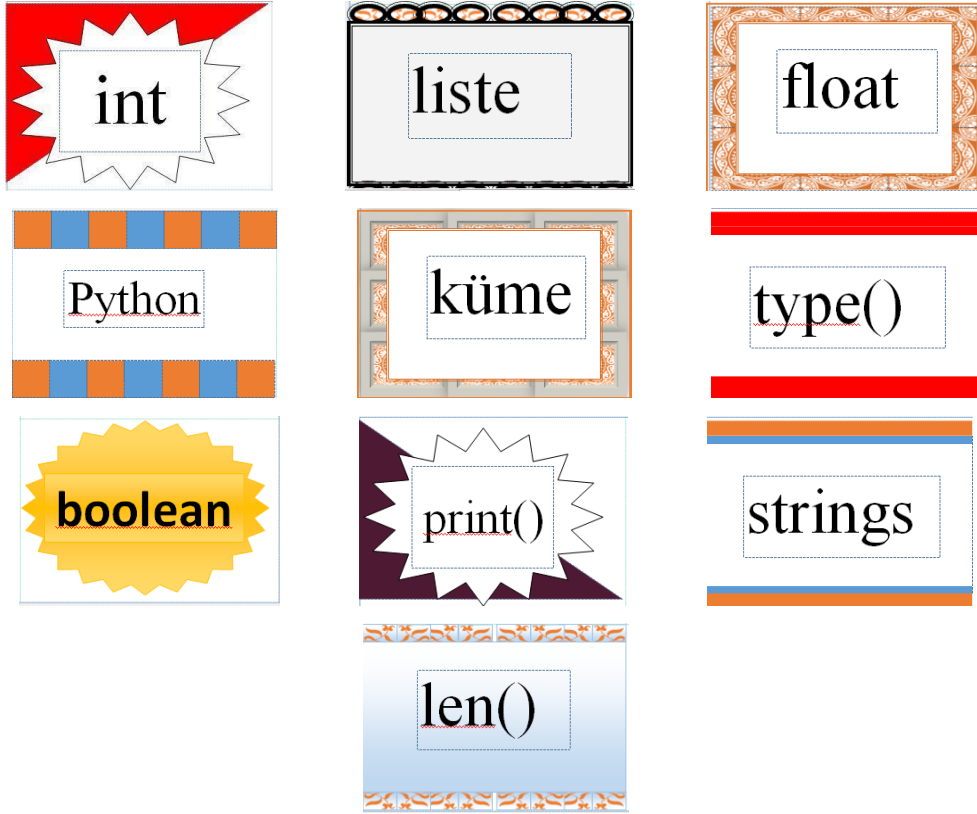
```
>>> type("Bilgisayar Bilimi")
```

```
>>> print("kodlama\n öğrenemek\n istiyorum")
```

## OYYÖO Uygulama 1. Hafta Eğlencim Etkinliđi

### (Kartları Eşleřtirme Etkinliđi)

Bu etkinlikte amaç **Deđişkenler** konusunun pekiřtirilmesidir. Etkinliđin hazırlanmasında deđişken türlerinin ve pythonla ilgili terimlerin yazılı olduđu flash kartlar hazırlanmıřtır. Hazırlanan kartlardan ikiřer tane hazırlanmıř ve öđrencilerin kartları eşleřtirmeleri istenmiřtir. Belirlenen sürede kartları eşleřtirmeyi bařaran öđrenciler puan kazanmıřtır. Hazırlanan kartlar ařađıda sunulmuřtur.



## OYYÖO Uygulama 2. Hafta Eğlenelim Etkinliği

### (Bilgi Yarışması Etkinliği)

Bu etkinlikte amaç **Print Fonksiyonu** konusunun pekiştirilmesidir. Bu etkinlikte öğrenciler 5-6 kişilik kümelere ayrılmıştır. Her gruptan birer sözcü seçilmiş ve sorulan soruları küme adına o cevaplamıştır. Sorular etkileşimli tahtadan sunulmuş ve belirtilen sürenin bitiminde küme adına sözcü tarafından kâğıda yazılarak havaya kaldırılmıştır. Yarışma sonunda küme elemanlarının tamamı küme puanı kadar puan kazanmıştır. Örnek sorular aşağıda sunulmuştur.

1. Ekranı çıktısı vermeyi sağlayan python fonksiyonu hangisidir?
2. Print() fonksiyonu ile dosyaya kayıt yapmak için hangi parametre kullanılır?
3. Veri türünü öğrenmek için hangi fonksiyon kullanılır?
4. Karakter uzunluğunu belirlemek için hangi fonksiyon kullanılır?
5. print("Kodlama") ifadesinin ekran çıktısı nedir?
6. print("Python", 2018) ifadesinin ekran çıktısı nedir?
7. print('Python programlama dili') ifadesinin ekran çıktısı nedir?

8.

```
ad="Karakter dizisi"
```

```
len(ad)
```

ifadesinin ekran çıktısı nedir?

9. **type(3.14)** ifadesinin ekran çıktısı nedir?

10.

```
numara="123456789"
```

```
int(numara)
```

```
type(numara)
```

ifadesinin ekran çıktısı nedir?

### **OYYÖO Uygulama 3. Hafta Eğlenelim Etkinliđi**

#### **(Program alıřtırma Etkinliđi)**

Bu etkinlikte ama **Kontrol Yapıları** konusunun pekiřtirilmesidir. Etkinlikte öđrenciler kümelere ayrıldı ve bir problem durumu verildi. Öđrenciler küme olarak problem durumunu kodlayarak programı alıřır hale getirmeye alıřtılar. Verilen süre ierisinde programı dođru bir řekilde alıřtıran tüm küme üyeleri puan kazandı. Problem durumu ařađıda sunulmuřtur.

**Problem Durumu:** Kullanıcının 1 ile 7 arasında bir sayı girmesi istenmektedir. Girilen sayının haftanın hangi günü olduđunu bulan programı yazınız.

## OYYÖO Uygulama 4. Hafta Eğlenelim Etkinliği

### (Kahoot Etkinliği)

Bu etkinlikte amaç **Tekrarlı Yapılar** konusunun pekiştirilmesidir. Etkinliğin hazırlanmasın Kahoot uygulaması kullanılmıştır. Tüm öğrenciler bilgisayar veya tablet kullanarak etkinliğe katılım sağlamıştır. Toplam 10 sorudan oluşan kahoot etkinlikte öğrenciler doğru cevapladıkları soru sayısına göre puan kazanmıştır. Etkinlikte kullanılan sorular aşağıda sunulmuştur.

Question	Type	Correct/incorrect
1 Pythonda programların tekrar tekrar çalışmasını sağlayan yapılara ne denir?		100%
2 Pythonda kullanılan döngü türlerinden biridir?		100%
3 Ekrana çıktı vermemizi sağlayan komut nedir?		100%
4 Pythonda ondalıklı sayılar hangi veri türündedir?		100%
5 Pythonda kullanılan döngü türlerinden biridir?		100%
6 Pythonda belirli aralıktaki sayıları göstermek için hangi ifade kullanılır?		80%
7 Tekrarlı yapılarda hiçbir şey yapma anlamındaki ifade nedir?		60%
8 Pythonda programı durdurmak için hangi ifade kullanılır?		80%
9 Pythonda tüm işlemleri pas geçerek döngünün başına dönmemizi sağlayan ifade nedir?		80%
10 Tam sayı veri tipi pythonda hangi ifade ile adlandırılır?		40%

Pythonda programların tekrar tekrar çalışmasını sağlayan yapılara ne denir?

▲ Döngü



◆ Koşullu ifadeler



● Fonksiyon



■ Liste





## OYYÖO Uygulama 5. Hafta Eğlenelim Etkinliği

### (Şifreli Bulmaca Etkinliği)

Bu etkinlikte amaç **Fonksiyonlar** konusunun pekiştirilmesidir. Etkinlikte öğrencilerin fonksiyonlarla ilgili sorulan soruları cevaplayarak bulmacada işaretlemeleri istenmiştir. Etkinlikte öğrenciler doğru cevapladıkları ve işaretledikleri soru sayısı kadar puan kazanmıştır. Etkinlikte kullanılan bulmaca ve sorular aşağıda sunulmuştur.

### ŞİFRELİ BULMACA

Soruların cevaplarını bulmacada bularak boyayınız?

O	Ü	M	A	G	P	A	R	A	M	E	T	R	E
F	I	Y	R	Ü	I	Y	K	R	G	Ü	F	Ü	M
R	A	R	A	M	A	M	D	G	K	I	K	E	N
M	A	R	P	M	A	R	E	Ü	R	A	K	K	K
D	Y	A	R	R	O	N	Y	M	A	R	Y	S	G
E	N	R	I	A	R	A	N	A	G	N	K	M	Ü
F	O	N	K	S	I	Y	O	N	S	I	N	I	Y
R	E	F	I	R	A	M	A	R	E	A	G	Ü	S

1. Fonksiyon çağrıldığında parantez içinde kullanılan değerlere ne ad verilir?
2. Fonksiyon tanımlamak için hangi ifade kullanılır.
3. Python'da tekrar tekrar kullanabildiğimiz kod kümelerine ne ad verilir?
4. Python'da kaç çeşit fonksiyon bulunmaktadır?
5. Tanımlanan fonksiyonlarda parantez içindeki değerlere ne ad verilir?

## OYYÖO Uygulama 6. Hafta Eğlenelim Etkinliği

### (Plickers Etkinliği)

Bu etkinlikte amaç **Parametre Alan Fonksiyonlar** konusunun pekiştirilmesidir. Etkinliğin hazırlanmasında plickers uygulaması kullanılmıştır. Öğrencilere etkileşimli tahtadan sorular yöneltilmiş ve öğrenciler plickers kartlarıyla soruları cevaplamışlardır. Öğrencilerin kartları plickers mobil uygulama vasıtasıyla taranmıştır. Doğru cevap sayısına göre öğrenciler puan kazanmıştır. Etkinlikle ilgili görseller aşağıda sunulmuştur.

Aşağıda kullanılmak üzere çağrılan fonksiyonlardan hangisi isimli argümanlı fonksiyonlara örnektir?

- A ortalama()
- B ortalama(not1=45,not2=65,not3=75)
- C ortalama("45";"65";"75")
- D ortalama(45,65,75)



## OYYÖO Uygulama 7. Hafta Eğlencim Etkinliği

### (Sudoku Etkinliği)

Bu etkinlikte amaç **Değer Döndüren ve Argümanlı Fonksiyonlar** konusunun pekiştirilmesidir. Etkinlikte öğrenciler kümelere ayrılmış ve her kümeye bir sudoku kartı verilmiştir. Kümelere sorular yöneltilmiş ve doğru cevap veren kümelerin sudoku kartında bir boşluk doldurulmuştur. Bu şekilde sudokunu başarıyla çözen küme üyeleri etkinlik sonunda tam puan kazanmıştır. Etkinlikte kullanılan sudoku kartı ve sorular aşağıda sunulmuştur.

				7		5	6	
1	2							
			5	4	2	3		
5		9						1
		4	2		9			5
		2	1	8				7
9	3				4			
				2	6			
						4	9	

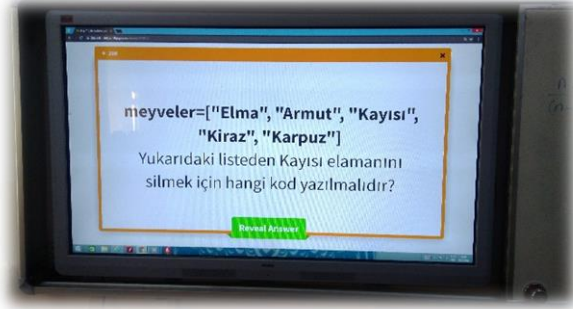
### Sorular

1. Programın herhangi bir durumda hiçbir şey yapmasını istemiyorsak hangi deyim kullanmamız gerekmektedir?
2. **def adres(cadde, sokak, apartman, binano, daireno):**  
Yukarıda verilen fonksiyonun *istenen sayıda sıralı argümanlı fonksiyon* haline dönüşmüş halini nedir?
3. Fonksiyonun geriye değer döndürmesini sağlayan deyim hangisidir?
4. Fonksiyon içinde tanımlanan değişkeni fonksiyon dışında da kullanabilmek için hangi deyim kullanılır?
5. **def bilgiler(ad, soyad, okulno, yazlizi1, yazili2,sozlu):**  
Yukarıda verilen fonksiyonun *istenen sayıda isimli argümanlı fonksiyon* haline dönüşmüş hali nedir?

## OYYÖO Uygulama 8. Hafta Eğlenelim Etkinliği

### (Flipquiz Etkinliği)

Bu etkinlikte amaç **Liste Veri Tipi** konusunun pekiştirilmesidir. Flipquiz uygulaması kullanılmıştır. Bu uygulama ile öğrenciler farklı puanda ve zorluk derecesinde olan sorulardan seçmiş ve doğru cevaplayan öğrenciler puan kazanmıştır. Etkinlik görselleri ekte sunulmuştur.



1. Listelerde köşeli parantez içine yazılan değerlere ne denir?
2. Listelerde ilk indis değeri nedir?
3. `Meyveler=["Elma", "Armut", "Kayısı", "Kiraz", "Karpuz"]` listeyi tersine çevirmek için hangi kod yazılmalıdır?
4. Listelere eleman eklemek için hangi metot kullanılır?
5. Listelerde eleman silmek için hangi metot kullanılır?
6. Listelerde indis kullanılarak eleman silmek için hangi metot kullanılır?
7. Liste elemanlarının sırasını tersine çevirmek için hangi metot kullanılır?
8. Liste elemanlarını alfabetik sıraya koymak için hangi metot kullanılır?
9. Liste içerisinde arama yapmamızı sağlayan metot hangisidir?
10. Listelerde aranan değerden kaç tane olduğunu veren metot hangisidir?

## **OYYÖO Uygulama 9. Hafta Eğlenelim Etkinliđi**

### **(Metot Bulmaca Etkinliđi)**

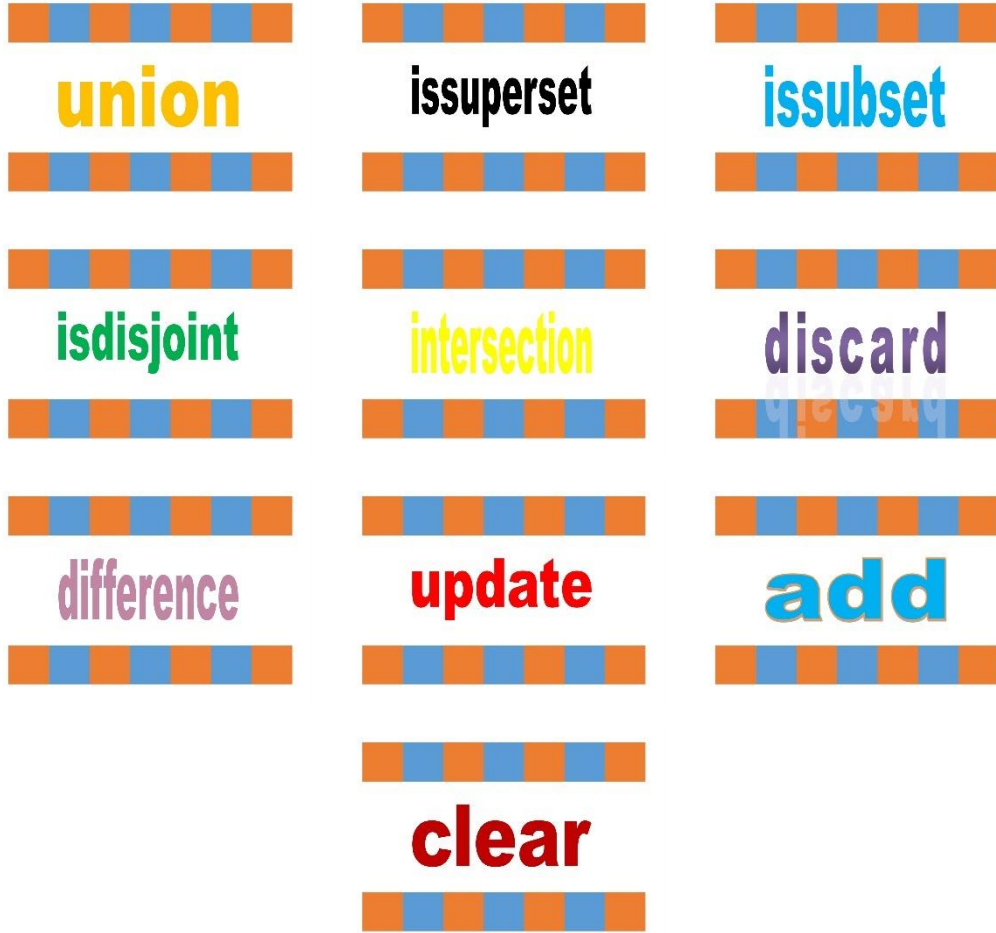
Bu etkinlikte amaç **Liste Fonksiyonları** konusunun pekiştirilmesidir. Etkinlikte öğrencilere 5 boş kart verilmiş ve okul numaralarını yazmaları sağlanmıştır. Daha sonra öğrencilere sorular sırasıyla etkileşimli tahtadan gösterilmiş ve öğrencilerin her sorunun cevabını bir karta yazmaları istenmiştir. Sorular bitince tüm kartlar toplanarak değerlendirilmiştir. Cevapları yazım yanlışı olmadan doğru olarak yazan öğrenciler puan kazanmıştır. Etkinlikte kullanılan sorular aşağıda sunulmuştur.

1. Liste elemanlarını alfabetik sıraya koymak için hangi metot kullanılır?
2. Liste içerisinde arama yapılan değerden kaç tane oluşunu çıktı olarak veren metot nedir?
3. Listelerde eleman eklemek için hangi metot kullanılır?
4. Listelerde ilk liste elemanının indis numarası kaçtır?
5. Listelerden indis numarasıyla eleman silmek için hangi metot kullanılır?

## OYYÖO Uygulama 10. Hafta Eğlenelim Etkinliği






### (Metot Çek Kodla Etkinliği)

Bu etkinlikte amaç **Küme Veri Tipi** konusunun pekiştirilmesidir. Etkinlikte küme metotlarıyla ilgili kartlar hazırlanmış ve küçük bir torbanın içine yerleştirilmiştir. Öğrencilere sırasıyla birer kart çektirilmiş ve çektikleri kartta yer alan metotlarla ilgili sorular sorulmuştur. Doğru cevap veren öğrenciler puan kazanmıştır.





yükselebileceksiniz. Seviyeleriniz ve kazanılması gereken puanlar aşağıdaki tabloda verilmiştir.

Seviyeler	Puan	Seviyeler	Puan
	0		180
	60		240
	120		

4. Yaptığınız bazı davranışlar size özel rozetler kazandıracaktır. Kazanacağının rozetler ve kazanma şartları aşağıdaki gibidir.



**Kodcu:** Kodlama deneyimi etkinliklerinden başarılı olan öğrencilere verilecektir.



**Düzenli:** Haftalık olarak tüm etkinliklerde başarı gösteren ve etkin katılım sağlayan öğrencilere verilecektir.



**Etkinlik:** Eğlenceli etkinliklerinde tam başarı gösteren öğrencilere verilecektir.









**İşbirlikçi:** Arkadaşlarıyla kurduğu iş birliği çalışmalarına göre verilecektir.

5. Belirli durumlarda gerçek ödül kazanılacaktır. Bu durumlar aşağıdaki gibidir;
  - a. Lider tahtasında her ay birinci olan öğrenciler gerçek ödül kazanacaktır.
  - b. Kodlama deneyimi etkinlikleri sonucu toplamda **5 kodcu rozeti** kazananlar gerçek ödül ile ödüllendirilecektir.
  - c. Haftalık tüm etkinlikleri sonucu toplamda **5 düzenli rozeti** kazananlar gerçek ödül ile ödüllendirilecektir.

**Ek 20. OYYÖO Rozet Durumu**

Sıra No	Öğrenci Kodu	 Kodcu	 İşbirlikçi	 Düzenli	 Etkinlikçi
1	OYYÖO-1	6	5	8	4
2	OYYÖO -2	5	6	8	5
3	OYYÖO -3	6	5	8	8
4	OYYÖO -4	6	7	7	4
5	OYYÖO -5	4	6	7	6
6	OYYÖO -6	8	8	6	8
7	OYYÖO -7	4	6	8	4
8	OYYÖO -8	4	5	7	4
9	OYYÖO -9	6	8	8	6
10	OYYÖO -10	7	6	7	4
11	OYYÖO -11	7	5	9	4
12	OYYÖO -12	3	5	7	4
13	OYYÖO -13	4	5	9	4
14	OYYÖO -14	5	5	8	4
15	OYYÖO -15	6	6	8	8
16	OYYÖO -16	6	8	6	4
17	OYYÖO -17	3	5	9	4
18	OYYÖO -18	8	6	9	5
19	OYYÖO -19	6	6	9	7
20	OYYÖO -20	5	5	8	4
21	OYYÖO -21	7	8	6	4
22	OYYÖO -22	8	5	8	4
23	OYYÖO -23	5	4	9	8
24	OYYÖO -24	3	8	9	4
25	OYYÖO -25	4	9	7	4
26	OYYÖO -26	7	6	8	4
27	OYYÖO -27	8	6	8	4
28	OYYÖO -28	7	7	9	5
29	OYYÖO -29	7	7	8	4
30	OYYÖO -30	6	6	8	4
31	OYYÖO -31	8	7	7	4
32	OYYÖO -32	6	6	10	5
33	OYYÖO -33	7	6	7	4

**Ek 21. OYYÖO Seviye ve Rütbe Durumu**

Sıra No	Öğrenci Kodu	Puan	Seviye	Rütbe
1	OYYÖO-1	198,5	Level 3	Kodlayıcı
2	OYYÖO -2	221,5	Level 5	Uzman Kodlayıcı
3	OYYÖO -3	214,5	Level 4	İyi Kodlayıcı
4	OYYÖO -4	212	Level 4	İyi Kodlayıcı
5	OYYÖO -5	172,5	Level 3	Kodlayıcı
6	OYYÖO -6	240	Level 5	Uzman Kodlayıcı
7	OYYÖO -7	209,5	Level 4	İyi Kodlayıcı
8	OYYÖO -8	211,5	Level 4	İyi Kodlayıcı
9	OYYÖO -9	217,5	Level 4	İyi Kodlayıcı
10	OYYÖO -10	205,5	Level 4	İyi Kodlayıcı
11	OYYÖO -11	218	Level 4	İyi Kodlayıcı
12	OYYÖO -12	187,5	Level 3	Kodlayıcı
13	OYYÖO -13	191	Level 3	Kodlayıcı
14	OYYÖO -14	207,5	Level 4	İyi Kodlayıcı
15	OYYÖO -15	238	Level 5	Uzman Kodlayıcı
16	OYYÖO -16	221	Level 5	Uzman Kodlayıcı
17	OYYÖO -17	195,5	Level 3	Kodlayıcı
18	OYYÖO -18	232	Level 5	Uzman Kodlayıcı
19	OYYÖO -19	231	Level 5	Uzman Kodlayıcı
20	OYYÖO -20	213	Level 4	İyi Kodlayıcı
21	OYYÖO -21	218	Level 4	İyi Kodlayıcı
22	OYYÖO -22	220,5	Level 5	Uzman Kodlayıcı
23	OYYÖO -23	190,5	Level 3	Kodlayıcı
24	OYYÖO -24	212,5	Level 4	İyi Kodlayıcı
25	OYYÖO -25	214,5	Level 4	İyi Kodlayıcı
26	OYYÖO -26	218	Level 4	İyi Kodlayıcı
27	OYYÖO -27	221	Level 5	Uzman Kodlayıcı
28	OYYÖO -28	227,5	Level 5	Uzman Kodlayıcı
29	OYYÖO -29	221	Level 5	Uzman Kodlayıcı
30	OYYÖO -30	213,5	Level 4	İyi Kodlayıcı
31	OYYÖO -31	219,5	Level 4	İyi Kodlayıcı
32	OYYÖO -32	202,5	Level 4	İyi Kodlayıcı
33	OYYÖO -33	209	Level 4	İyi Kodlayıcı

**Ek 22. ABT, MÖ ve PDYTÖ Ön Test Ölçümlerinde Elde Ettikleri Çarpıklık ve Basıklık Puanlarına İlişkin Değerler**

Ölçek	Grup	Ölçüm	N	$\bar{X}$	Ss	Ort.	Mod	Çarpıklık	Basıklık
ABT	Deney 1	Ön Test	33	28.78	7.08	28.56	24.99	.85	-.05
		Son Test	33	74.54	6.10	74.97	71.40	.39	-1.28
		İzleme	33	72.70	5.49	74.97	74.97	-.14	.06
	Deney 2	Ön Test	33	32.46	6.21	32.13	28.56*	1.40	1.07
		Son Test	33	80.27	7.09	81.11	82.11	-.85	-.27
		İzleme	33	74.97	7.36	74.97	74.97	1.34	-.21
	Deney 3	Ön Test	33	29.64	9.08	32.13	28.56	-.91	.14
		Son Test	33	72.16	6.42	71.40	78.54	-1.08	-.73
		İzleme	33	70.00	4.55	71.40	67.83*	-1.33	.66
	Kontrol	Ön Test	33	32.24	8.63	32.13	35.70	.32	-1.01
		Son Test	33	65.34	8.35	64.26	60.69	.27	.69
		İzleme	33	66.96	6.68	67.83	71.40	.77	-.66
MÖ	Deney 1	Ön Test	33	4.10	.37	4.10	4.50	-.35	.55
		Son Test	33	4.95	.59	5.05	4.95*	-1.44	-.32
		İzleme	33	4.89	.29	4.85	4.70	.55	-.10
	Deney 2	Ön Test	33	4.26	.94	4.30	3.30	-.35	-.37
		Son Test	33	5.08	.72	5.25	5.90	-1.70	-.17
		İzleme	33	4.95	.36	4.95	4.50	.07	-1.33
	Deney 3	Ön Test	33	4.34	.81	4.45	3.70*	.05	-1.19
		Son Test	33	4.59	.54	4.50	4.55	1.88	1.01
		İzleme	33	4.37	.31	4.40	4.65	1.25	-.10
	Kontrol	Ön Test	33	3.96	.63	4.00	4.20	.08	-.76
		Son Test	33	4.26	.51	4.35	3.70*	-.99	-1.05
		İzleme	33	4.28	.27	4.25	3.90*	.22	-1.70
PDYTÖ	Deney 1	Ön Test	33	2.70	.34	2.77	2.90*	-.57	.23
		Son Test	33	4.23	.22	4.25	4.25	-1.42	.91
		İzleme	33	4.16	.19	4.17	4.28	-1.47	.58
	Deney 2	Ön Test	33	2.83	.36	2.85	2.90*	-.56	-.91
		Son Test	33	4.26	.29	4.22	4.08*	.18	-1.27
		İzleme	33	4.21	.21	4.25	4.25	-1.22	-.56
	Deney 3	Ön Test	33	2.86	.48	2.80	2.28*	1.89	1.41
		Son Test	33	3.61	.23	3.60	3.40*	1.50	.61
		İzleme	33	3.72	.22	3.67	3.60	.80	-.60
	Kontrol	Ön Test	33	2.73	.37	2.78	2.50*	.08	-.83
		Son Test	33	3.32	.29	3.32	3.50	-1.19	-.17
		İzleme	33	3.45	.14	3.45	3.55	1.76	.40

\*Çoklu modlar nedeniyle en küçük değer

**Ek 23. Deney 1, Deney 2, Deney 3 ve Kontrol Gruplarının ABT Ön Test, Son Test ve İzleme Testi Puan Ortalamalarına İlişkin İkili Karşılaştırma (Bonferonni Uyumlu) Testi Sonuçları**

Bonferonni Testi Sonuçları													
		Deney 1			Deney 2			Deney 3			Kontrol		
		1	2	3	1	2	3	1	2	3	1	2	3
		Ortala ma Fark <sup>(I- J)</sup>	Ortala ma Fark <sup>(I- J)</sup>	Ortala ma Fark <sup>(I- J)</sup>	Ortala ma Fark <sup>(I- J)</sup>	Ortala ma Fark <sup>(I- J)</sup>	Ortala ma Fark <sup>(I- J)</sup>	Ortala ma Fark <sup>(I- J)</sup>	Ortala ma Fark <sup>(I- J)</sup>	Ortala ma Fark <sup>(I- J)</sup>	Ortala ma Fark <sup>(I- J)</sup>	Ortala ma Fark <sup>(I- J)</sup>	Ortala ma Fark <sup>(I- J)</sup>
Deney 1	1		- 45.761 <sup>*</sup>	- 43.922 <sup>*</sup>									
	2	45.761 <sup>*</sup>		1.839		- 5.734 <sup>*</sup>			2.380			9.195 <sup>°</sup>	
	3	43.922 <sup>*</sup>	-1.839				-2.272			2.705			5.734 <sup>*</sup>
Deney 2	1					- 47.816 <sup>*</sup>	- 42.515 <sup>*</sup>						
	2		5.734 <sup>*</sup>		47.816 <sup>*</sup>		5.301 <sup>*</sup>		8.114 <sup>*</sup>			14.929 <sup>*</sup>	
	3			2.272	42.515 <sup>*</sup>	- 5.301 <sup>*</sup>				4.976 <sup>*</sup>			8.005 <sup>*</sup>
Deney 3	1								- 42.515 <sup>*</sup>	- 40.352 <sup>*</sup>			
	2		-2.380			- 8.114 <sup>*</sup>		42.515 <sup>*</sup>		2.164		6.815 <sup>°</sup>	
	3			-2.705			- 4.976 <sup>*</sup>	40.352 <sup>*</sup>	-2.164				3.029
Kontrol	1											- 33.104 <sup>*</sup>	- 34.726 <sup>*</sup>
	2		- 9.195 <sup>°</sup>			- 14.929 <sup>*</sup>			- 6.815 <sup>*</sup>		33.104 <sup>*</sup>		-1.623
	3			- 5.734 <sup>*</sup>			- 8.005 <sup>*</sup>			-3.029	34.726 <sup>*</sup>	1.623	

Not: 1=Ön test, 2=Son test, 3= İzleme Testi

**Ek 24. Deney 1, Deney 2, Deney 3 ve Kontrol Gruplarının MÖ Ön Test, Son Test ve İzleme Testi Puan Ortalamalarına İlişkin İkili Karşılaştırma (Bonferonni Uyumlu) Testi Sonuçları**

Bonferonni Testi Sonuçları													
		Deney 1			Deney 2			Deney 3			Kontrol		
		1	2	3	1	2	3	1	2	3	1	2	3
		Ortala ma Fark <sup>(I- J)</sup>	Ortala ma Fark <sup>(I- J)</sup>	Ortala ma Fark <sup>(I- J)</sup>	Ortala ma Fark <sup>(I- J)</sup>	Ortala ma Fark <sup>(I- J)</sup>	Ortala ma Fark <sup>(I- J)</sup>	Ortala ma Fark <sup>(I- J)</sup>	Ortala ma Fark <sup>(I- J)</sup>	Ortala ma Fark <sup>(I- J)</sup>	Ortala ma Fark <sup>(I- J)</sup>	Ortala ma Fark <sup>(I- J)</sup>	Ortala ma Fark <sup>(I- J)</sup>
Deney 1	1		-.845*	-.786*									
	2	.841*		.055		-.136			.358			.683*	
	3	.786*	-.055				-.055			.518*			.615*
Deney 2	1					-.823*	-.686*						
	2		.136		.823*		.136		.494*			.820*	
	3			.055	.686*	-.136				.573*			.670*
Deney 3	1								-.250	-.035			
	2		-.358			-.494*		.250		.215*		.326	
	3			-.518*			-.573*	.035	-.215*				.097
Kontrol	1											-.305	-.318
	2		-.683*			-.820*			.326		.305		-.014
	3			-.615*			-.670*			-.097	.318	.014	

Not: 1=Ön test, 2=Son test, 3= İzleme Testi

**Ek 25. Deney 1, Deney 2, Deney 3 ve Kontrol Gruplarının PDYTÖ Ön Test, Son Test ve İzleme Testi Puan Ortalamalarına İlişkin İkili Karşılaştırma (Bonferonni Uyumlu) Testi Sonuçları**

Bonferonni Testi Sonuçları													
		Deney 1			Deney 2			Deney 3			Kontrol		
		1	2	3	1	2	3	1	2	3	1	2	3
		Ortalama Fark <sup>(I-J)</sup>	Ortalama Fark <sup>(I-J)</sup>	Ortalama Fark <sup>(I-J)</sup>	Ortalama Fark <sup>(I-J)</sup>	Ortalama Fark <sup>(I-J)</sup>	Ortalama Fark <sup>(I-J)</sup>	Ortalama Fark <sup>(I-J)</sup>	Ortalama Fark <sup>(I-J)</sup>	Ortalama Fark <sup>(I-J)</sup>	Ortalama Fark <sup>(I-J)</sup>	Ortalama Fark <sup>(I-J)</sup>	Ortalama Fark <sup>(I-J)</sup>
Deney 1	1		-1.530*	-1.455*									
	2	1.530*		.074		-.024			.623*			.911*	
	3	1.455*	-.074				-.050			.443*			.712*
Deney 2	1					-1.432*	-1.383*						
	2		.024		1.432*		.048		.647*			.936*	
	3			.050	1.383*	-.048				.493*			.762*
Deney 3	1								-.755*	-.861*			
	2		-.623*			-.647*		.755*		-.105		.289*	
	3			-.443*			-.493*	.861*	.105				-.269*
Kontrol	1											-.594*	-.719*
	2		-.911*			-.936*			-.289*		.594*		-.125
	3			-.712*			-.762*			-.269*	.719*	.125	

Not: 1=Ön test, 2=Son test, 3= İzleme Testi