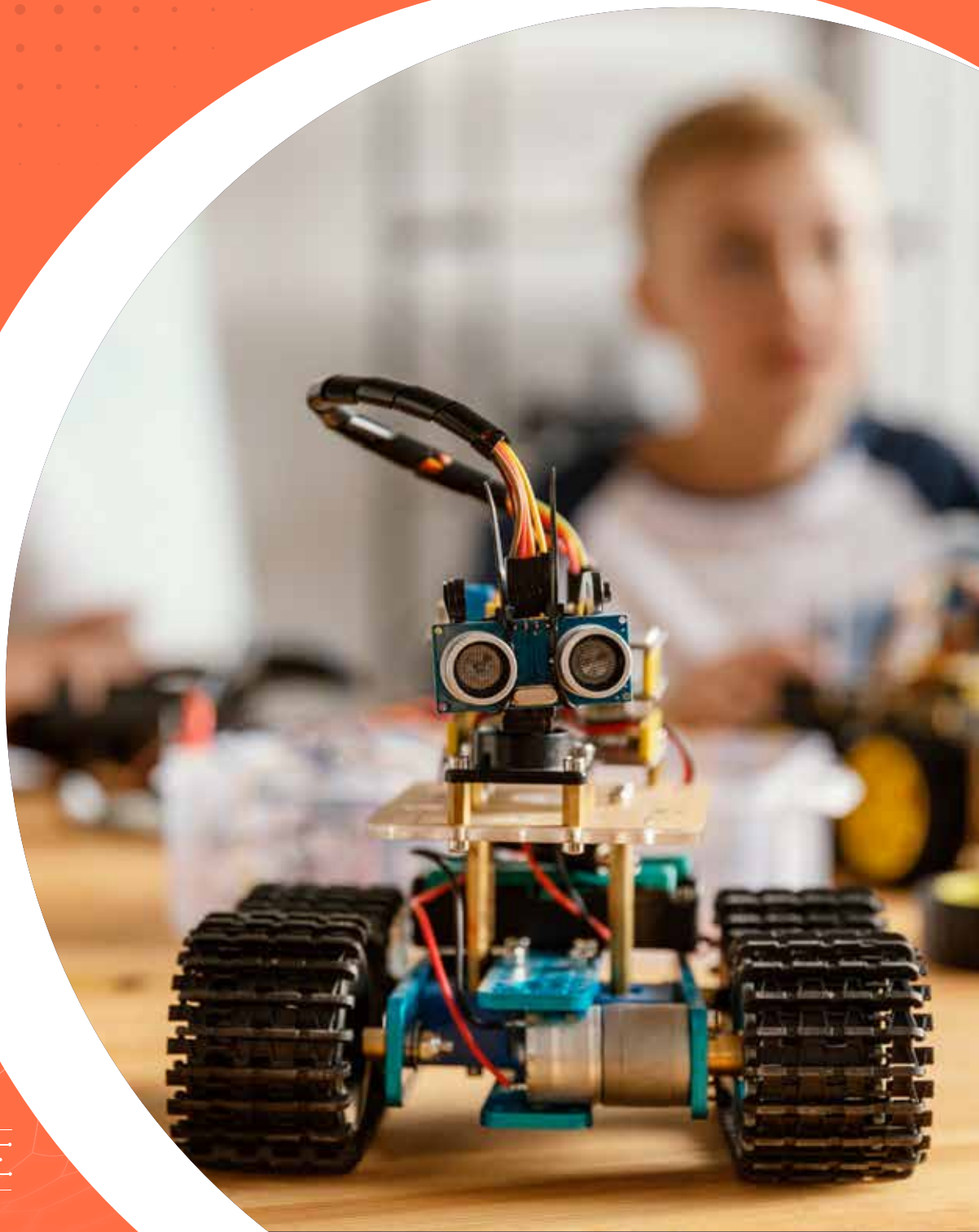
İcathane Eğitim Müfredatı

TEMEL DÜZEY

Maker | Tasarım | Yazılım | Elektronik | Devre Tasarımı | İHA | Robotik | Siber Güvenlik | BT Kültürü



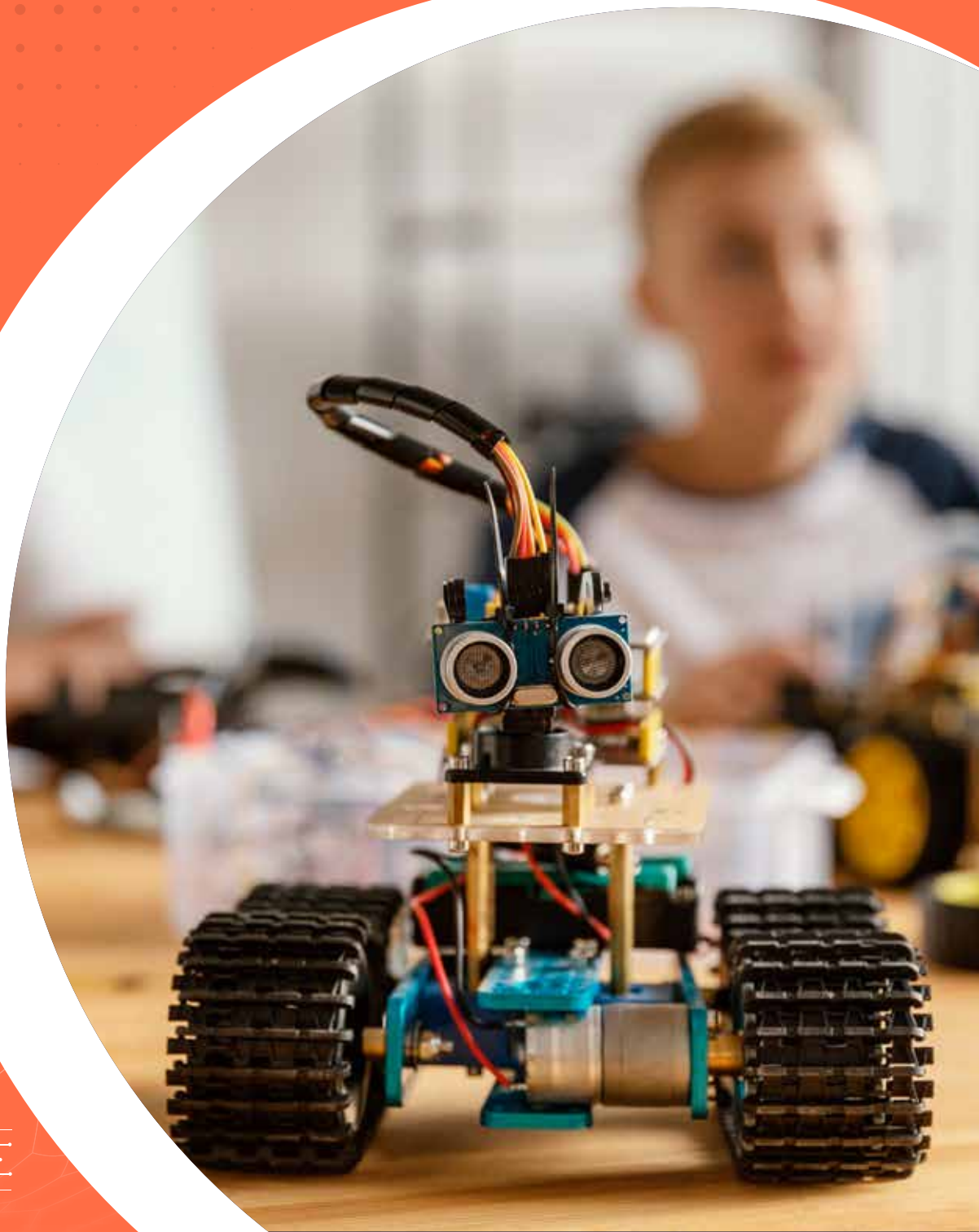
TÜGVA İCATHANE

İcathane Eğitim Müfredatı

TEMEL DÜZEY

Maker | Tasarım | Yazılım | Elektronik | Devre Tasarımı

İHA | Robotik | Siber Güvenlik | BT Kültürü



TÜGVA İCATHANE

Sürüm :1.0

İÇİNDEKİLER

1. Önsöz . . . 5
2. TEMEL DÜZEY . . . 6
2.1 TASARIM . . . 7
2.1.1 Tinkercad Müfredat Takvimi . . . 8
2.1.2 Tinkercad Eğitimi . . . 9
2.1.2.1 Tinkercad-Tanıtım ve Giriş . . . 10
2.1.2.2 3 Boyutlu Katı Oluşturma . . . 11
2.1.2.3 3 Boyutlu Uygulamalar . . . 12
2.1.3 3 Boyutlu Baskı Eğitimi . . . 14
2.1.3.1 3 Boyutlu Yazıcılar Tarihi Gelişimi ve Çeşitleri . . . 15
2.1.3.2 Filament Türleri ve Kullanım Şartları . . . 16
2.1.3.3 XDesktop Dilimleme Program Kurulumu . . . 17
2.1.3.4 XDesktop Arayüz Tanıtımı . . . 18
2.1.3.5 3B Dilimleme Teknikleri ve Yazıcı Seçimi . . . 19
2.1.3.6 3 Boyutlu Yazıcı Ayarları . . . 21
2.1.3.7 3 Boyutlu Yazıcılarda Mekanik Bakım . . . 22
2.1.3.8 Tinkercad: 3 Boyutlu Uygulamalar . . . 23
2.1.4 Fusion 360 Müfredat Takvimi . . . 24
2.1.5 Fusion 360 Eğitimi . . . 25
2.1.5.1 Fusion360: Tanıtım ve Giriş . . . 26
2.1.5.2 Sketch Çizim Komutları . . . 29
2.1.5.3 Kısıtlama (Constraints-Sketch Palette) Komutu . . . 32
2.1.5.4 Budama (Trim) Komutu . . . 34
2.1.5.5 Kavis -Yuvarlatma (Fillet) Komutu . . . 37
2.1.5.6 Aynalama (Mirror) Komutu . . . 39
2.1.5.7 Öteleyerek Çoğaltma (Offset) Komutu . . . 40
2.1.5.8 Dairesel Çoğaltma (Circular Pattern) ve Doğrusal Çoğaltma (Rectangular Pattern) Komutları . . . 41
2.1.5.9 Extrude - Katı Model Oluşturma . . . 42
2.1.5.10 Döndürerek Katı Oluşturma (Revolve) Komutu . . . 43
2.1.5.11 Yol Kullanarak Katı Oluşturma (Sweep) Komutu . . . 44
2.1.5.12 Profiller ve Yüzeyler Arasında Katı Oluşturma (Loft) Komutu . . . 46
2.1.5.13 Delik Sihirbazı (Hole) Komutu . . . 47
2.1.5.14 Web Komutu . . . 48
2.1.5.15 Uygulama . . . 49
2.2 ROBOTİK . . . 54
2.2.1 Maker Eğitim Müfredatı . . . 55
2.2.2 Temel Elektronik Kavramlar ve Tanımlar . . . 56
2.2.2.1 Temel Elektroniğe Giriş . . . 57
2.2.2.2 Elektrik Akımı - Gerilimi Bilgileri . . . 59
2.2.2.3 Temel Elektronik: AC/DC Kavramları ve Özellikleri . . . 61
2.2.2.4 Elektronik Değerleri Ölçme Aletleri . . . 62
2.2.2.5 Frekans, Periyot Kavramları . . . 63
2.2.3 Scratch Eğitimi . . . 64
2.2.3.1 Scratch: Algoritma . . . 65
2.2.3.2 Scratch: Değişkenler . . . 68
2.2.3.3 Scratch: if-else . . . 71
2.2.3.4 Scratch: Döngüler . . . 73
2.2.3.5 Scratch: Fonksiyonlar . . . 75
2.2.3.6 Scratch: Sınıflar ve Nesneler . . . 77
2.2.3.7 Selamlama Uygulaması . . . 78
2.2.3.8 Gol Atma Uygulaması . . . 79
2.2.3.9 Duvarlara Çarpan Ok Uygulaması . . . 82
2.2.3.10 Dikdörtgen Çizme Uygulaması . . . 83
2.2.3.11 Üçgen Çizme Uygulaması . . . 85
2.2.3.12 Çokgen Çizim Uygulaması . . . 87
2.2.3.13 İki Kişilik Oyun Uygulaması . . . 89
2.2.3.14 Yılan Oyunu Yapımı . . . 92
2.2.3.15 Home Sweet Home Uygulaması . . . 95
2.2.3.16 Haritalarla Denizli Uygulaması . . . 96
2.2.4 Makeblock mBot Eğitimi . . . 98
2.2.4.1 Makeblock mBot: mBot Robot Kiti . . . 99
2.2.4.2 Makeblock mBot: mBot Montajı . . . 101
2.2.4.3 Makeblock mBot: mBlock Programı . . . 103
2.2.4.4 Makeblock mBot: mBlock Kod Blokları . . . 105
2.2.4.5 Makeblock mBot: mBot ile Örnek Programlar . . . 110
2.2.5 Robotik Eğitim Müfredatı . . . 113
2.2.6 Robotik . . . 114
2.2.6.1 Temel Tanımlar ve Kullanımlar . . . 115
2.2.6.1.1 Geliştirme Kartları ve Arduino . . . 116
2.2.6.1.2 Arduino Kart Çeşitleri ve Özellikleri . . . 118
2.2.6.1.3 Arduino IDE Kurulumu ve Tanıtımı . . . 120
2.2.6.1.4 Arduino IDE Programlama Dili ve Özellikleri . . . 122

2.2.6.1.5 Breadboard Kullanımı . . . 124
2.2.6.2 Sensörler ve Uygulamaları . . . 125
2.2.6.2.1 Sensörler . . . 126
2.2.6.2.2 Blink Uygulaması . . . 127
2.2.6.2.3 Kara Şimşek Uygulaması . . . 129
2.2.6.2.4 Butondan Değer Okuma Uygulaması . . . 132
2.2.6.2.5 Potansiyometreden Değer Okuma Uygulaması . . . 134
2.2.6.2.6 LDR Değer Okuma Uygulaması . . . 136
2.2.6.2.7 Kara Şimşek Uygulaması - For Döngüsü . . . 138
2.2.6.2.8 LDR ile Led Yakma Uygulaması (if-else) . . . 140
2.2.6.2.9 Potansiyometre ile Led Yakma Uygulaması . . . 143
2.2.6.2.10 Park Sensörü Uygulaması . . . 145
2.2.6.2.11 Ekran Kullanım Uygulaması . . . 146
2.2.6.2.12 Servo Motor Uygulaması . . . 147
2.2.6.2.13 DC Motor Uygulaması . . . 149
2.2.6.3 Proje Geliştirme . . . 151
2.2.6.3.1 Engelden Kaçan Robot Yapımı . . . 152
2.2.6.3.2 Robot Kol Yapımı . . . 155
2.2.6.3.3 Bluetooth İle Araç Kontrolü . . . 156
2.2.6.3.4 Akıllı Ev Uygulaması . . . 157
2.2.7 Elektronik Eğitim Müfredatı . . . 158
2.2.8 Elektronik . . . 159
2.2.8.1 Elektronik Komponent ve Özellikleri . . . 160
2.2.8.1.1 Temel Elektronik: Direnç . . . 161
2.2.8.1.2 Temel Elektronik: Kondansatör . . . 164
2.2.8.1.3 Temel Elektronik: Bobin . . . 166
2.2.8.1.4 Temel Elektronik: Transformatör (Trafo) . . . 167
2.2.8.1.5 Temel Elektronik: Diyot . . . 169
2.2.8.1.6 Temel Elektronik: Transistör . . . 171
2.2.8.1.7 Temel Elektronik: Tristör . . . 173
2.2.8.1.8 Temel Elektronik: Optokuplör . . . 175
2.2.8.1.9 Temel Elektronik: Triyak . . . 176
2.2.8.2 Elektronik Devre Uygulamaları . . . 178
2.2.8.2.1 Yarım Dalga Doğrultucu Devresi . . . 179
2.2.8.2.2 Tam Dalga Doğrultucu Devresi . . . 181
2.2.8.2.3 Buton İle Led Kontrol Uygulaması . . . 184
2.2.8.2.4 NPN Transistörlü Anahtarlama Devresi . . . 186
2.2.8.2.5 Flip Flop Devresi . . . 188
2.2.8.2.6 Tristörlü Uygulama Devresi . . . 190
2.2.8.2.7 Kondansatörlü Şarj Deşarj Devresi . . . 192
2.2.8.2.8 Transistörlü Hırsız Alarm Devresi . . . 194
2.3 YAZILIM . . . 196
2.3.1 Hackercan Eğitim Müfredatı . . . 197
2.3.2 Hacker Can Eğitimi . . . 198
2.3.2.1 Hacker Can: Arayüz Tanıtımı . . . 199
2.3.2.2 Hacker Can: Algoritma . . . 201
2.3.2.3 Hacker Can: Söz Dizimi . . . 203
2.3.2.4 Hacker Can: Objeler . . . 205
2.3.2.5 Hacker Can: Karakterler . . . 207
2.3.2.6 Hacker Can: Kere Döngüsü . . . 209
2.3.2.7 Hacker Can: İç İçe Kere Döngüsü . . . 211
2.3.2.8 Hacker Can: Değişkenler . . . 213
2.3.2.9 Hacker Can: Bitinceye Kadar Döngüsü . . . 215
2.3.2.10 Hacker Can: Fonksiyonlar . . . 217
2.3.2.11 Hacker Can: Parametreli Fonksiyonlar . . . 219
2.3.2.12 Hacker Can: Hazır Fonksiyonlar . . . 221
2.3.2.13 Hacker Can: if-else Kontrol Yapıları . . . 223
2.3.3 MIT APP INVENTOR Eğitim Müfredatı . . . 225
2.3.4 MIT App Inventor Eğitimi . . . 226
2.3.4.1 MIT App Inventor: Arayüz Tanıtımı . . . 227
2.3.4.2 MIT App Inventor: User Interface Menüsü . . . 229
2.3.4.3 MIT App Inventor: Layout Menüsü . . . 231
2.3.4.4 MIT App Inventor: Media Menüsü . . . 233
2.3.4.5 MIT App Inventor: Drawing and Animations Menüsü . . . 235
2.3.4.6 MIT App Inventor: Sensors Menüsü . . . 237
2.3.4.7 MIT App Inventor: Social Menü . . . 239
2.3.4.8 MIT App Inventor: Storage Menüsü . . . 241
2.3.4.9 MIT App Inventor: Connectivity Menüsü . . . 243
2.3.4.10 MIT App Inventor: Değişkenler . . . 245
2.3.4.11 Renklendirme Uygulaması . . . 247
2.3.4.12 Boyama Uygulaması . . . 249
2.3.4.13 Pusula Uygulaması . . . 255

2.3.4.14 Alarm Uygulaması . . . 258
2.3.4.15 Web Tarayıcısı Uygulaması . . . 262
2.3.4.16 Qr Kod Okuyucu Uygulaması . . . 264
2.3.4.17 Telefonu Sallayarak Ekran Temizleme Uygulaması . . . 266
2.3.4.18 Fotoğraf Çekme Uygulaması . . . 268
2.3.4.19 Ses Tanıma Uygulaması . . . 270
2.3.4.20 Layout Bileşenleri Uygulaması . . . 272
2.3.5 Python Eğitim Müfredatı . . . 274
2.3.6 Python Eğitimi . . . 275
2.3.6.1 PYTHON: Tanıtım . . . 276
2.3.6.2 PYTHON: Kurulum . . . 277
2.3.6.3 PYTHON: print() Fonksiyonu . . . 279
2.3.6.4 PYTHON: Yorum Satırları . . . 282
2.3.6.5 PYTHON: Kaçış Dizileri . . . 284
2.3.6.6 PYTHON: String (Karakter) . . . 286
2.3.6.7 PYTHON: Integer (Sayılar) . . . 288
2.3.6.8 PYTHON: Sayı Metotları . . . 292
2.3.6.9 PYTHON: Listeler . . . 295
2.3.6.10 PYTHON: Liste Metotları . . . 300
2.3.6.11 PYTHON: Kümeler (Set) . . . 305
2.3.6.12 PYTHON: Değişkenler . . . 311
2.3.6.13 PYTHON: Karakter Dizileri . . . 313
2.3.6.14 PYTHON: Kullanıcıdan Bilgi Almak . . . 317
2.3.6.15 PYTHON: Operatörler . . . 320
2.3.6.16 PYTHON: if-else . . . 327
2.3.6.17 PYTHON: While Döngüsü . . . 330
2.3.6.18 PYTHON: For Döngüsü . . . 333
2.3.6.19 PYTHON: Hata Yakalama . . . 335
2.3.6.20 PYTHON: Fonksiyonlar . . . 339
2.3.6.21 PYTHON: Gömülü Fonksiyonlar . . . 344
2.3.6.22 PYTHON: Nesne Tabanlı Programlama (Bölüm 1) . . . 359
2.3.6.23 PYTHON: Nesne Tabanlı Programlama (Bölüm 2) . . . 364
2.3.7 HTML & CSS - JAVASCRİPT Eğitim Müfredatı . . . 369
2.3.8 Html & Css Eğitimi . . . 371
2.3.8.1 HTML: Giriş . . . 373
2.3.8.2 HTML: Etiketler . . . 375
2.3.8.3 HTML: Açıklama Satırı . . . 380
2.3.8.4 HTML: Başlık ve Paragraflar . . . 381
2.3.8.5 HTML: Yazı . . . 382
2.3.8.6 HTML: Renk Kodları . . . 385
2.3.8.7 HTML: Link Ekleme . . . 387
2.3.8.8 HTML: Resim Ekleme . . . 389
2.3.8.9 HTML: Tablo Ekleme . . . 391
2.3.8.10 HTML: Liste Ekleme . . . 393
2.3.8.11 HTML: Elementler . . . 396
2.3.8.12 HTML: Class ve id . . . 399
2.3.8.13 HTML: Form Yapısı . . . 401
2.3.8.14 CSS: Giriş . . . 405
2.3.8.15 CSS: Yazım Kuralları . . . 407
2.3.8.16 CSS: Kodları Ekleme Yöntemi . . . 412
2.3.8.17 CSS: Zemin Özellikleri . . . 416
2.3.8.18 CSS: Metin İşlemleri . . . 421
2.3.8.19 CSS: Yazı Tipi Özellikleri . . . 427
2.3.8.20 CSS: Linkler . . . 434
2.3.8.21 CSS: Listeler . . . 439
2.3.8.22 CSS: Tablolar . . . 443
2.3.8.23 CSS: Tablo Elemanları . . . 447
2.3.8.24 CSS: Kutu Modeli . . . 453
2.3.8.25 CSS: Border . . . 456
2.3.8.26 CSS: Padding . . . 460
2.3.8.27 CSS: Boyutlar . . . 463
2.3.9 Javascript Eğitimi . . . 466
2.3.9.1 JavaScript: Tanıtım ve Programlamaya Giriş . . . 467
2.3.9.2 Değişkenler . . . 468
2.3.9.3 Sabitler . . . 471
2.3.9.4 JavaScript: Operatörler . . . 473
2.3.9.5 JavaScript: Sayı Metotları (Numbers) . . . 479
2.3.9.6 JavaScript: String Metotları . . . 481
2.3.9.7 Koşul Durumları: if-else . . . 484
2.3.9.8 Koşul Durumları: Switch-Case Yapısı . . . 487
2.3.9.9 Döngüler: For Döngüsü . . . 490
2.3.9.10 Döngüler: While Döngüsü . . . 492

2.3.9.11 Döngüler: Do While Döngüsü 494
2.3.9.12 JavaScript: Array (Diziler) 496
2.3.9.13 JavaScript: Fonksiyonlar 498
2.3.9.14 JavaScript: Tarih ve Zaman Fonksiyonları 500
2.3.9.15 Nesne Yönelimli Programlama (1.Bölüm) 504
2.3.9.16 Nesne Yönelimli Programlama (2.Bölüm) 508

ÖNSÖZ

Kıymetli Arkadaşlarım,

Türkiye Gençlik Vakfı; dünya üzerinde iyi ve güzele dair ne varsa savunan, her daim hakkın ve adaletin tarafında olacak bir nesil yetiştirme amacıyla kurulmuş Türkiye'nin en büyük gençlik hareketidir.

İcathanelerimiz, genç arkadaşlarımızın gelişimine yönelik donanım ve altyapıya sahip teknoloji ve bilim merkezleridir.

35 ilimizde bulunan TÜGVA İcathanelerimizde, ortaokul ve liseli kardeşlerimize "Kodlama, Robotik, Endüstriyel Tasarım, Mobil Uygulama Geliştirme, Bilgisayar Programı Yazma, 3 Boyutlu Tasarım ve Üretim, Elektronik Devre Tasarlama ve Robotik Sistem Tasarlama" alanlarında eğitimler veriyoruz. "Seyyah İcathanemiz" ile ülkemizin doğusundan batısına tüm gençlere imkân sunmaya devam ediyoruz.

İcathanelerimizde eğitim alan birçok arkadaşımız ile birlikte TEKNOFEST Havacılık, Uzay ve Teknoloji Festivali'ne katılım sağladık. İnanıyorum ki; genç arkadaşlarımızın, önceki yıllarda olduğu gibi gelecek yıllarda da farklı kategorilerde dereceleri olacak.

Sizler de bu ülkenin teknofest kuşağı olarak, Türkiye'nin kritik teknoloji alanlarında tam bağımsızlık iddiasını sürdürebilmesi için Milli Teknoloji Hamlesi vizyonuna paralel güçlü projeler üretecek; zamanın ruhuna uygun ilim, fen ve teknolojide en üst düzeyi hedefleyeceksiniz.

Biliyorum ki; bu ülkenin geleceğini öz güveni yüksek, yenilikçi, iyi ahlaklı, başarılı ve sorumluluk sahibi gençler çizecektir. Dünyayı değiştirecek liderlerin sizlerin içerisinden çıkacağına can-ı gönülden inanıyorum.

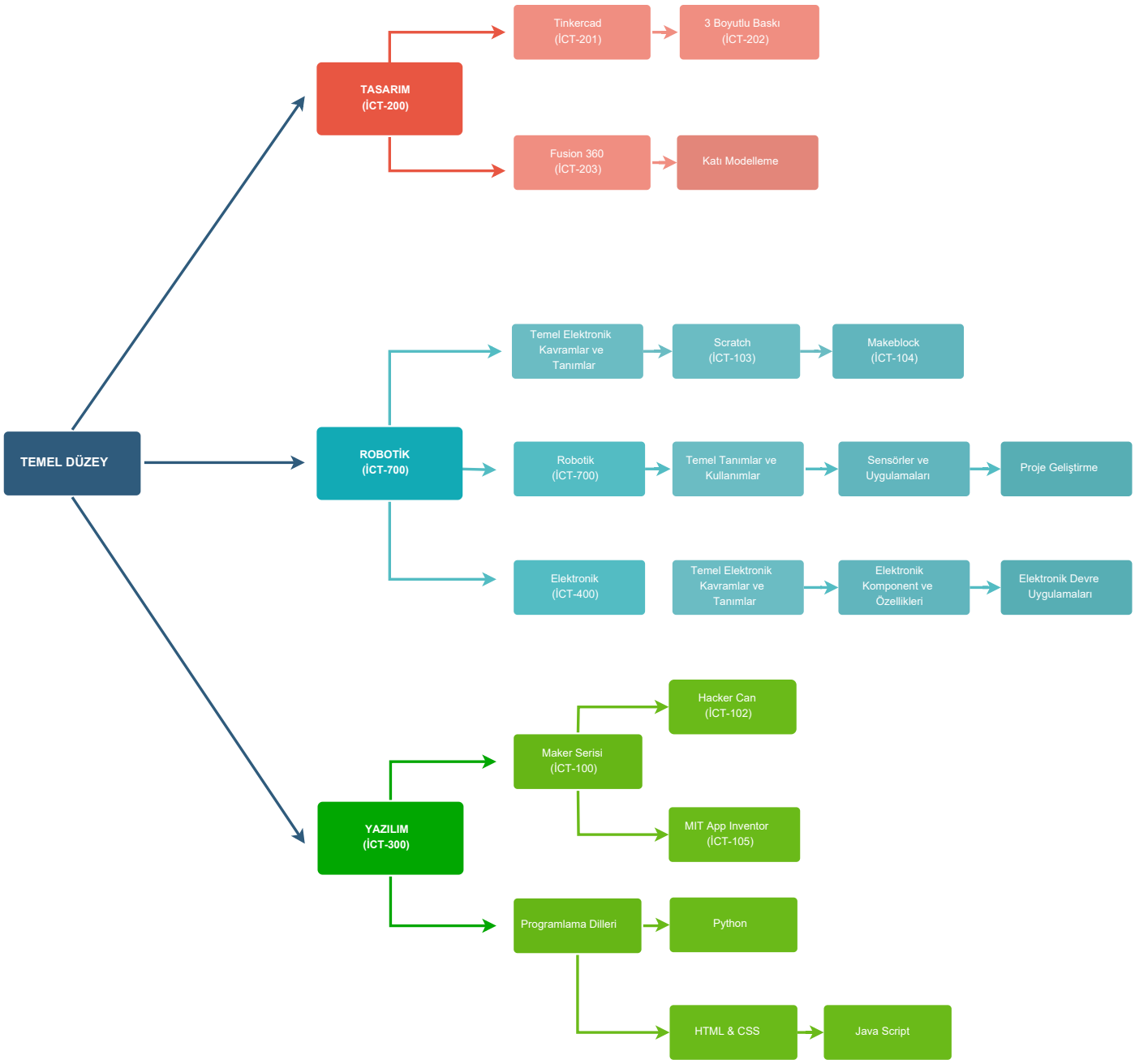
Unutmayın!

"Türkiye Yüzyılı Gençlerin Yüzyılı Olacak!"

İBRAHİM BEŞİNCİ

TÜGVA GENEL BAŞKANI

TEMEL DÜZEY


TEMEL DÜZEY
TASARIM (İCT-200)
Tinkercad (İCT-201)
3 Boyutlu Baskı (İCT-202)
Fusion 360 (İCT-203)
Katı Modelleme
ROBOTİK (İCT-700)
Temel Elektronik Kavramlar ve Tanımlar
Scratch (İCT-103)
Makeblock (İCT-104)
Robotik (İCT-700)
Temel Tanımlar ve Kullanımlar
Sensörler ve Uygulamaları
Proje Geliştirme
Elektronik (İCT-400)
Temel Elektronik Kavramlar ve Tanımlar
Elektronik Komponent ve Özellikleri
Elektronik Devre Uygulamaları
YAZILIM (İCT-300)
Maker Serisi (İCT-100)
Hacker Can (İCT-102)
MIT App Inventor (İCT-105)
Programlama Dilleri
Python
HTML & CSS
Java Script

İcathane Eğitim Müfredatı

TASARIM

Tinkercad | 3D Print | Fusion 360

TÜGVA İCATHANE

TINKERCAD MÜFREDAT TAKVIMI

TİNKERCAD (İCT-200) EĞİTİM MÜFREDATI														
EĞİTİM İÇERİĞİ / HAFTA	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.
Tinkercad														
Tinkercad : Tanıtım ve Giriş														
Tinkercad : 3 Boyutlu Katı Oluşturma														
Tinkercad : 3 Boyutlu Uygulamalar														
3 Boyutlu Baskı														
3 Boyutlu Yazıcılar ve Çeşitleri														
Filament Çeşitleri ve Kullanım Şartları														
Zaxe Program Kurulumu														
Zaxe XDesktop Arayüz Tanıtımı														
3B Baskı : Dilimleme Teknikleri ve Yazıcı Seçimi														
3B Baskı: 3 Boyutlu Yazıcı Ayarları ve Baskı														
3 Boyutlu Yazıcı Mekanik Bakımı														

TINKERCAD EĞITIMI

Müfredat Kodu	İCT-201
Ders	Tinkercad
Ders İçeriği	• Tinkercad-Tanıtım ve Giriş • 3 Boyutlu Katı Oluşturma • 3 Boyutlu Uygulamalar
Seviye	İCT-S1: BAŞLANGIÇ
Ön Koşullar	Tasarım alanında ilgili olmak
Amaç	Bu bölüm sonunda, web tabanlı 3B tasarım ve modelleme programını kullanılarak tasarım yapılabiliyor olmak amaçlanmaktadır.
Hedef	Tinkercad programına ait komutları kullanarak 3 Boyutlu tasarım yapmak hedeflenmektedir.
Gereksinimler	Bilgisayar, Ortamda İnternet Ağı, Mail adresi
Kazanım	• Kolay bir şekilde 3 boyutlu tasarımlar oluşturabilir ve bu tasarımları düzenleyebilmek. • Tinkercad web tabanlı olduğu için herhangi bir cihazdan erişebilirsiniz ve tasarımlarınızı her yerden paylaşabilmek. • Tinkercad'in birçok araç ve özelliği sayede tasarımların istenildiği gibi şekillendirilmesi.
Durum	BEKLIYOR / UYGULANIYOR / TAMAMLANDI / DEĞERLENDIRILDI
Hedeflenen Süre	990 Dakika
Eğitim Günleri	Başlama: .../.../.... Bitiş: .../.../....

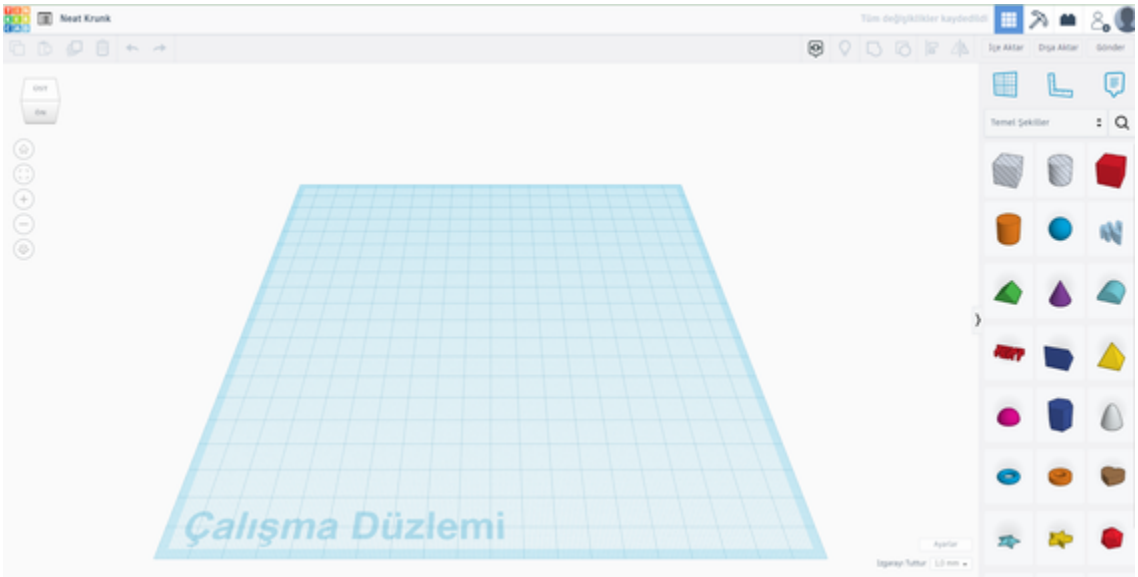
TINKERCAD-TANITIM VE GIRIŞ

Müfredat Kodu / Ünite / Konu	İCT-201-1 / Tinkercad / Tanıtım ve Giriş
Hedeflenen Süre	90 Dakika

🚀 Tinkercad Tanıtım ve Giriş

Tinkercad tarayıcı tabanlı her yaştan kişinin kolaylıkla kullanabileceği 3D tasarım, kod yazma ve modelleme aracıdır. Tinkercad internet browser (internet explorer, firefox, google, chrome gibi) üzerinden açıp kullanabilen, 3 boyutlu tasarım yapılabilen bir ortamdır.

→ https://www.tinkercad.com/ web adresinden kayıt olunarak giriş yapılmalıdır.



📘 Kazanım

Tinkercad kayıt olma ve giriş.

- Arayüz tanıma.
- 3 Boyutlu Nesneler.
- Çalışma düzlemi seçme ve ayar yapabilme.

✅ Dersin Hedefleri

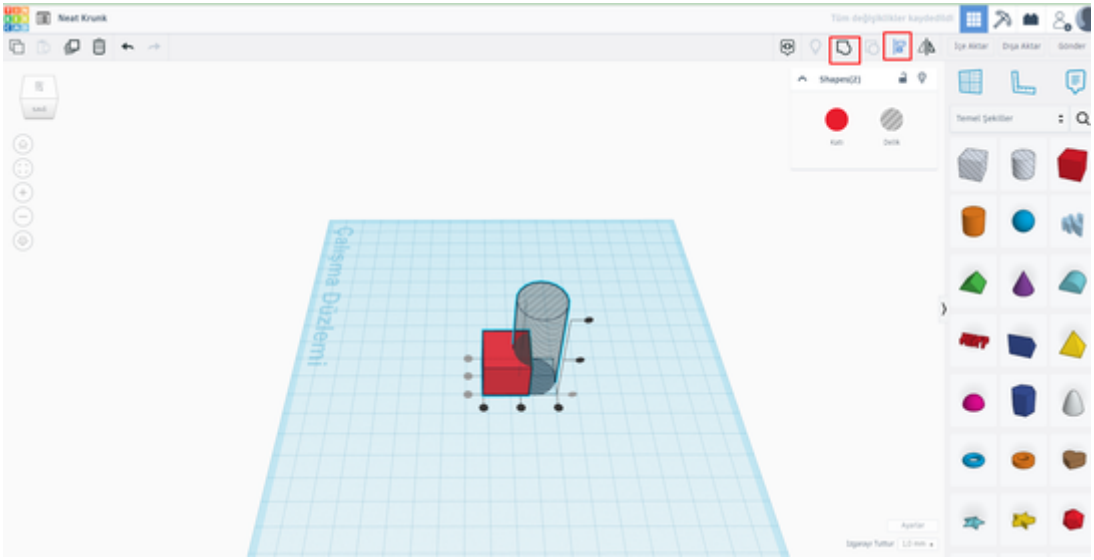
	Başlık	Hedef Kitle	Detaylar
1	☑ Tinkercad Tanıtım ve Giriş	☑ Eğitmen ☐ Öğrenciler	Eğitmenden bu bölümde; Tinkercad üye olma aşamalarını, arayüzde bulunan kavramlar (Çalışma Düzlemi, hizalama, gruplandırma, ölçülendirme) hakkında uygulamalı bilgi vermesi beklenmektedir. **Kitap: Çocuklar İçin Uygulamalarla 3 Boyutlu Tasarım -Abaküs Yayınları** kitabı referans alınabilir.
2	☑ Tinkercad Tanıtım ve Giriş	☐ Eğitmen ☑ Öğrenciler	Tinkercad program kurulumunun tamamlanması ve uygulama üzerindeki kavramların öğrenilmesi hedeflenmektedir.

3 BOYUTLU KATI OLUŞTURMA

Müfredat Kodu / Ünite / Konu	İCT-201-2 / Tinkercad / 3 Boyutlu Katı Oluşturma
Hedeflenen Süre	180 Dakika

🚀 3 Boyutlu Katı Oluşturma

Bu bölümde 3 boyutlu katı model oluşturma, delik oluşturma, hizalama ve gruplandırma çalışmaları yapılmalıdır.



📘 Kazanım

- 3 Boyut Katı Model oluşturma.
- Delik oluşturma.
- Hizalama.
- Cetvel kullanımı.
- Nesne ölçülendirme.
- Nesne aynalama.
- Nesne kopyalama.

✅ Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ 3 Boyutlu Katı Oluşturma	☑ Eğitmen ☐ Öğrenciler	Katı model oluşturulması, delik oluşturulmalı, nesnelerle gruplandırma ve hizalama işlemleri yapılmalıdır. Bu bölümde eğitmenden farklı kaynaklardan modelleri öğrencilere yaptırması hedeflenmektedir.
2	☑ 3 Boyutlu Katı Oluşturma	☐ Eğitmen ☑ Öğrenciler	Katı model oluşturmak için gerekli araçların öğrenilmesi ve tasarım yapabiliyor olmak hedeflenmektedir.

3 BOYUTLU UYGULAMALAR

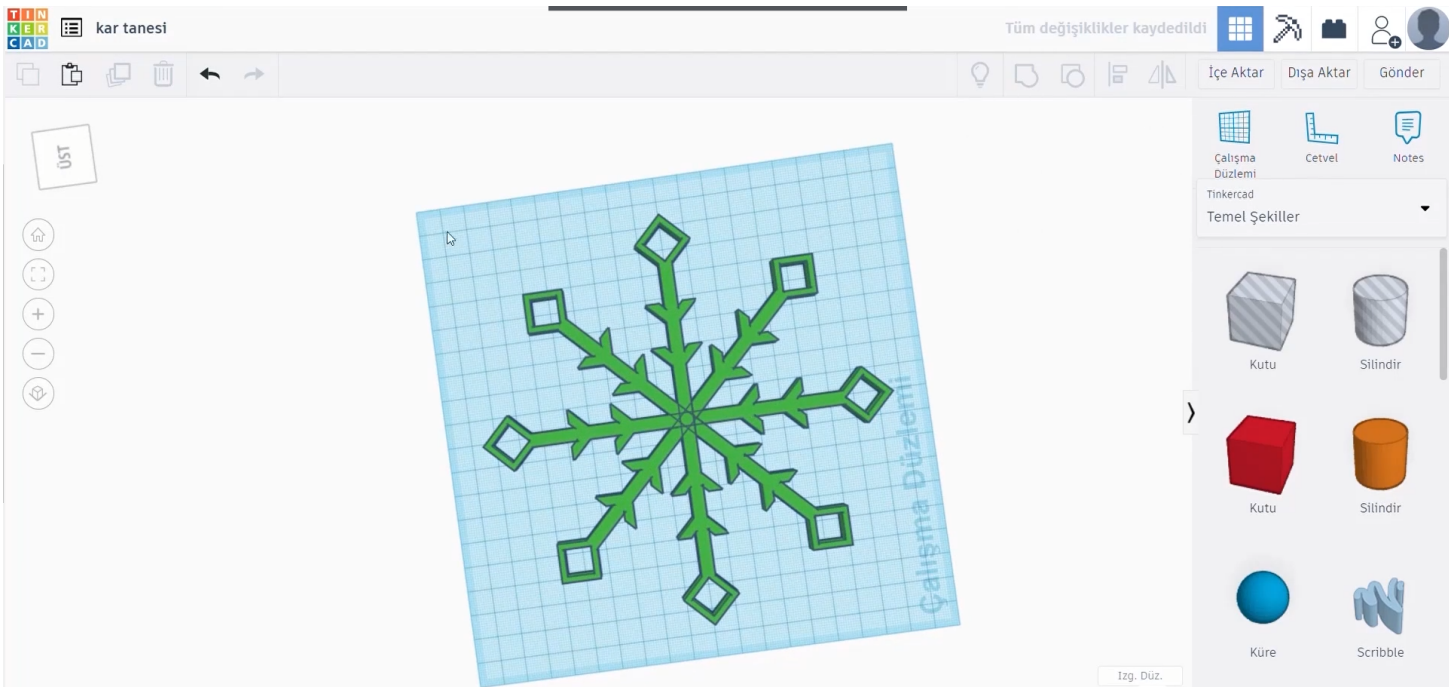
Müfredat Kodu/Ünite/Konu	İCT-201-3 /Tinkercad/ 3 Boyutlu Uygulamalar
Hedeflenen Süre	240 Dakika

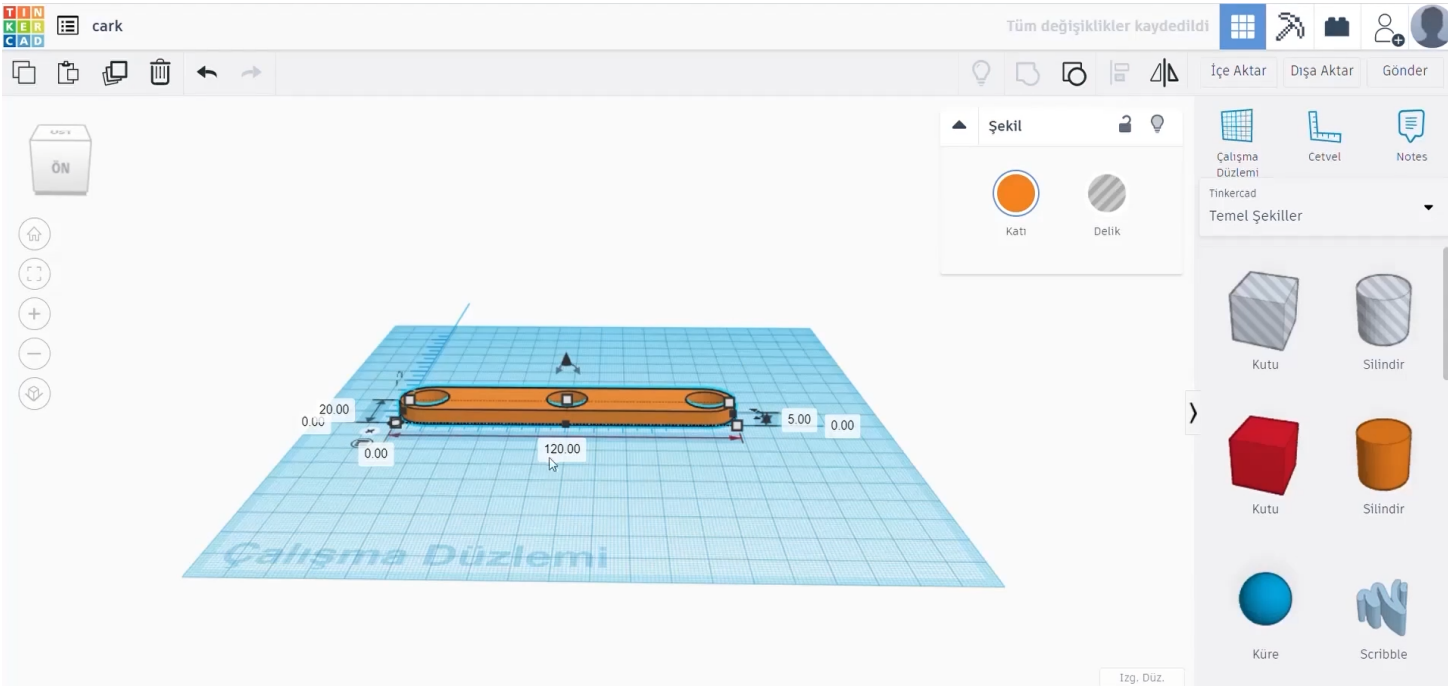
🚀 3 Boyutlu Uygulamalar

Bu bölümde aşağıda belirtilen uygulamaların yapılabiliyor olması amaçlanmaktadır.

🧰 Uygulamalar

- Köpük çubuğu.
- Kar tanesi.
- Stres çarkı.
- Yüzük yapımı.
- Bardak kupa yapımı.
- Robot kol yapımı.





Kazanım

- Tasarım becerisinin kazanılması amaçlanmaktadır.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Uygulamalar	☑ Eğitmen ☐ Öğrenciler	Eğitmen bu bölümde uygulamaları yapılırken öğrencilere yol göstererek çizim komutlarını doğru kullanıldığına dikkat etmelidir. Uygulamaların ölçülere uygun bir şekilde yapılabiliyor olması hedeflenmektedir. Tasarımlarda Referans ve Dokümantasyon bölümündeki kaynaklar dikkate alınmalıdır. NOT: Bu bölüm sonunda 3 Boyutlu yazıcılar bölümüne devam edilmelidir. Daha sonrasında 3 boyutlu yazıcılardan baskı alınacak şekilde uygulamalara devam edilmelidir.
2	☑ Uygulamalar	☐ Eğitmen ☑ Öğrenciler	Öğrenci eğitmenin önerdiği tasarımları yapabiliyor olmalıdır. Minimum Köpük Çubuğu, Kar Tanesi, Stres Çarkı, Yüzük Yapımı, Bardak Kupa Yapımı, Robot Kol Yapımı uygulamalarını yapılabilmesi hedeflenmektedir.

3 BOYUTLU BASKI EĞITIMI

Müfredat Kodu	İCT-202
Ders	3 Boyutlu Baskı
Ders İçeriği	• 3 Boyutlu Yazıcılar Tarihi Gelişimi ve Çeşitleri • Filament Türleri ve Kullanım Şartları • XDesktop Dilimleme Program Kurulumu • XDesktop Arayüz Tanıtımı • 3B Dilimleme Teknikleri ve Yazıcı Seçimi • 3 Boyutlu Yazıcı Ayarları • 3 Boyutlu Yazıcılarda Mekanik Bakım • Tinkercad: 3 Boyutlu Uygulamalar
Seviye	İCT-S2: ORTA
Ön Koşullar	Minimum düzeyde tasarım geçmişinin olması, İCT-201 -Thinkercad eğitim serisini tamamlamış olmak.
Amaç	Gelişen günümüz teknolojisinde bir çok alanda bir çok amaçla kullanılan 3 boyutlu yazıcılarla tanışmak ve kullanımlarının öğrenilmesi amaçlanmaktadır.
Hedef	3 Boyutlu yazıcıdan baskı alabilmek, 3 Boyutlu yazıcıların bakımlarını ve kalibrasyon ayarlarını yapabiliyor olmak hedeflenmektedir.
Gereksinimler	Dilimleme Programı, 3 Boyutlu Yazıcı, USB veya SD kart okuyucu.
Kazanım	• Modellenen tasarımları üretiyor olmak. • 3 Boyutlu yazıcı kullanıyor olmak. • 3 Boyutlu yazıcı bakım ve kalibrasyonlarını yapabiliyor olmak. • Dilimleme programını kullanabiliyor olmak.
Referanslar & Dokümantasyon	• https://tr.wikipedia.org/wiki/%C3%9C%C3%A7_boyutlu_bask%C4%B1 • https://arti90.com/medikal-modeller/ • https://www.robocombo.com/blog/icerik/filament-nedir-filament-cesitleri • https://serhatakademi.net/login/egitimler/courses/3dprinter.php • https://zaxe.com/
Durum	BEKLİYOR / UYGULANIYOR / TAMAMLANDI / DEĞERLENDİRİLDİ
Hedeflenen Süre	270 Dakika
Eğitim Günleri	Başlama: .../.../.... Bitiş: .../.../....

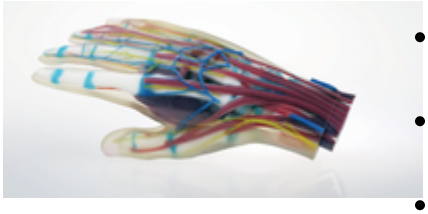
3 BOYUTLU YAZICILAR TARIHI GELIŞIMI VE ÇEŞITLERI

Müfredat Kodu / Ünite / Konu	İCT-202-1 / 3 Boyutlu Baskı / 3 Boyutlu Yazıcılar ve Çeşitleri
Hedeflenen Süre	20 Dakika

🚀 3 Boyutlu Yazıcılar ve Çeşitleri

Bu bölümde 3 Boyutlu yazıcıların tarihi, yazı modelleri ve kullanımı hakkında bilgilendirme yapılmalıdır. Örnek 3 boyutlu baskı çeşitlerinden, ülkelerdeki kullanılan 3 boyutlu yazıcılardan üretilmiş yapılara ve çeşitlerine değinilmelidir. Eğitmen bu bölümü destekleyici görseller sunmalıdır.

🧰 Bilgi



- **FDM (Fused Deposition Modeling) yazıcılar:** Bu yazıcılar, plastik filamentleri kullanarak nesneleri bir katman katman yapıştırarak üretirler.
- **SLA (Stereolithography) yazıcılar:** Bu yazıcılar, lazer ya da UV ışığı kullanarak, bir katman halinde daha ince malzemeleri (örneğin, fotopolimerler) kullanarak nesneleri üretirler.
- **SLS (Selective Laser Sintering) yazıcılar:** Bu yazıcılar, lazer kullanarak malzeme parçacıklarını eriterek ve bu parçacıkları bir araya getirerek nesneleri üretirler.
- **DLP (Digital Light Processing) yazıcılar:** Bu yazıcılar, bir projektörü kullanarak, malzeme parçacıklarını bir arada tutarak nesneleri üretirler.

📘 Kazanım

- 3 Boyutlu yazıcılar hakkında bilgi sahibi olmak.
- 3 Boyutlu yazıcıdan üretilen ürünler, yapılar ve prototip çalışmaları hakkında bilgi sahibi olmak.
- 3 Boyutlu Yazıcı çeşitlerinin ve özelliklerinin öğrenilmesi.

✅ Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ 3 Boyutlu Yazıcılar ve Çeşitlerinin tanıtılması	☑ Eğitmen ☐ Öğrenciler	Eğitmenden bu bölümde; 3 Boyutlu yazıların tarihi, çeşitleri ve yapılan çalışmalar hakkında bilgi vermesi beklenmektedir. Bilgilendirme sunum niteliğinde olmalı ve görsellerle desteklenmelidir. Eğitmen bu bölümde detaylı bilgileri edinerek öğrenciye aktarmalıdır.
2	☑ 3 Boyutlu Yazıcılar ve Çeşitlerinin tanıtılması	☐ Eğitmen ☑ Öğrenciler	3 Boyutlu yazıcılar ve bu cihazlarla neler yapılabildiği hakkında bilgi sahibi olunması hedeflenmektedir.

FILAMENT TÜRLERI VE KULLANIM ŞARTLARI

Müfredat Kodu / Ünite / Konu	İCT-202-2 / 3 Boyutlu Baskı / Filament Çeşitleri ve Kullanım Şartları
Hedeflenen Süre	20 Dakika

🚀 Filament Çeşitleri ve Kullanım Şartları

Bu bölümde 3 boyutlu yazıcılarda kullanılan yapı malzemesi olan Filament çeşitleri ve kullanım koşulları anlatılmalıdır.

🧰 Bilgi

Filament	Özellik	Baskı Sıcaklığı
PLA	Organik	180-230
ABS	Dayanıklı	210 - 250
PETG	Esnek	220-235
Naylon	Esnek	220-260
TPE	Lastik	225 - 235
Karbon Fiber	Sert	195 - 220
Ahşap	Ahşap	195 - 220
Polikarbonat	Esnek	270 - 310
İletken	İletken	215 - 230
Manyetik	Manyetik	195 - 220

📘 Kazanım

- Filament ve çeşitlerinin öğrenilmesi.
- Saklama koşulları ve kullanımları hakkında bilgi sahibi olmak.

✅ Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Filament Çeşitleri ve Kullanım Şartları	☑ Eğitmen ☐ Öğrenciler	Eğitmen bu bölümde 3 Boyutlu yazıcı kullanım malzemesi olan filamentleri ve çeşitlerini özellikleri ile açıklamalıdır. Açıklamanın sunum, görsel ve videolar ile desteklenmesi beklenmektedir.
2	☑ Filament Çeşitleri ve Kullanım Şartları	☐ Eğitmen ☑ Öğrenciler	Filament çeşitlerinin ve kullanım koşullarının pekiştirilmesi hedeflenmektedir.

XDESKTOP DILIMLEME PROGRAM KURULUMU

Müfredat Kodu / Ünite / Konu	İCT-202-3 / 3 Boyutlu Baskı /Zaxe XDesktop Dilimleme Kurulumu
Hedeflenen Süre	116 Dakika

🚀 Zaxe Program Kurulumu

Yapılan tasarımların 3 boyutlu yazıcılardan baskı alınabilmesi için kullanılan makine türüne göre dilimleme yapılması (.gcode formatına getirilmesi) gerekmektedir. Bu yüzden bir önceki eğitimde tamamlanan (Müfredat kodu İCT-201) tasarımların baskıya verilmesi için Zaxe XDesktop programının kurulması gerekmektedir.

Kurulum Linki : https://zaxe.com/tr/softwares/xdesktop

📘 Kazanım

- Zaxe XDesktop Dilimleme programının kurulması.

✅ Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Zaxe XDesktop Dilimleme Program Kurulumu	☑ Eğitmen ☐ Öğrenciler	Zaxe XDesktop Dilimleme Program kurulumunun yapılması beklenmektedir.
2	☑ Zaxe XDesktop Dilimleme Program Kurulumu	☐ Eğitmen ☑ Öğrenciler	Zaxe XDesktop Dilimleme Program kurulumunu yapabiliyor olmak hedeflenmektedir.

XDESKTOP ARAYÜZ TANITIMI

Müfredat Kodu / Ünite / Konu	İCT-202-4 / 3 Boyutlu Baskı / Zaxe XDesktop Arayüz Tanıtımı
Hedeflenen Süre	30 Dakika

🚀 Arayüz Tanıtımı

Bu bölümde Zaxe dilimleme programının arayüz tanıtımı yapılmalıdır.

🌪️ Zaxe 3 Boyutlu Yazıcı Modelleri

X1, X1+, X2, X3, XLite, Z1, Z1+, Z2, Z3, Z3+ Eğitmen İcathane içerisinde bulunan model markaya göre anlatım yapmalıdır.

🎗️ Bilgi

Bu bölümde;

- Tasarımın dilimleme düzlemine nasıl ve hangi formatta aktarılması gerektiği.
- Tasarımın boyutlarının değiştirilmesi, aynalama, yansıtma.
- 3 Boyutlu Yazıcı ayarları.
- Önizleme süreçleri ve 3 boyutlu yazıcıya aktarmak için çıktı süreçleri hakkında bilgiler verilmelidir.

📘 Kazanım

- Zaxe Programını kullanabiliyor olmak.
- 3 Boyutlu Yazıcı ekleyebiliyor olmak.
- 3 Boyutlu yazıcı için dilimleme ayarları yapabiliyor olma kazanımı hedeflenmektedir.

✅ Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Zaxe XDesktop Arayüz Tanıtımı	☑ Eğitmen ☐ Öğrenciler	Zaxe XDesktop Arayüz ve komutların tanıtılması beklenmektedir.
2	☑ Zaxe XDesktop Arayüz Tanıtımı	☐ Eğitmen ☑ Öğrenciler	Zaxe XDesktop programını arayüzü hakkında bilgi sahibi olunması hedeflenmektedir.

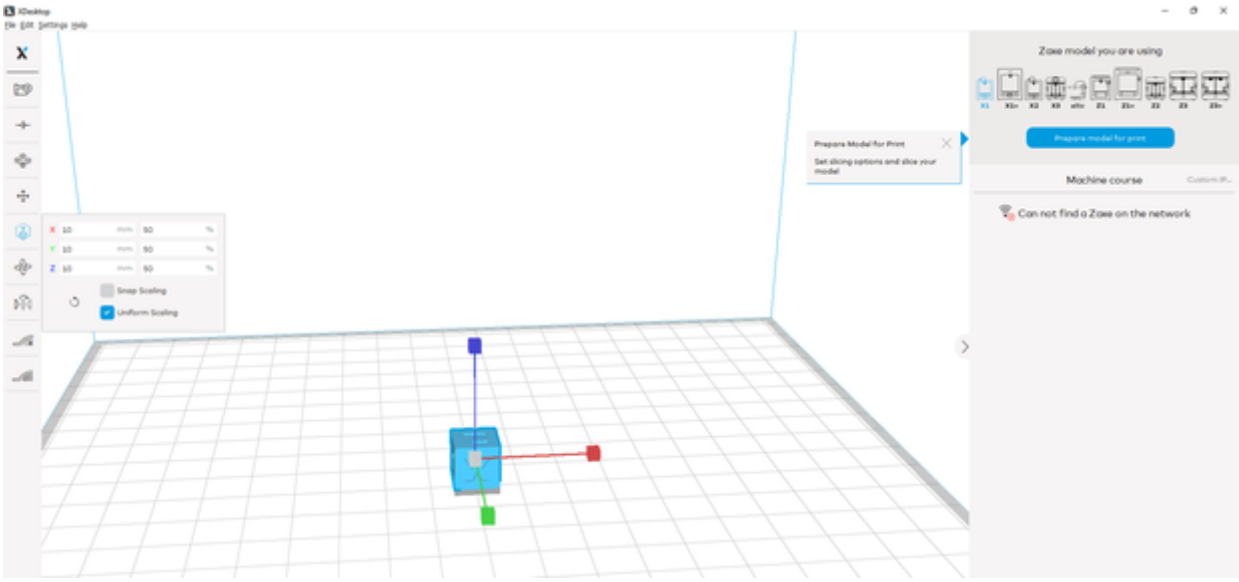
3B DILIMLEME TEKNIKLERI VE YAZICI SEÇIMI

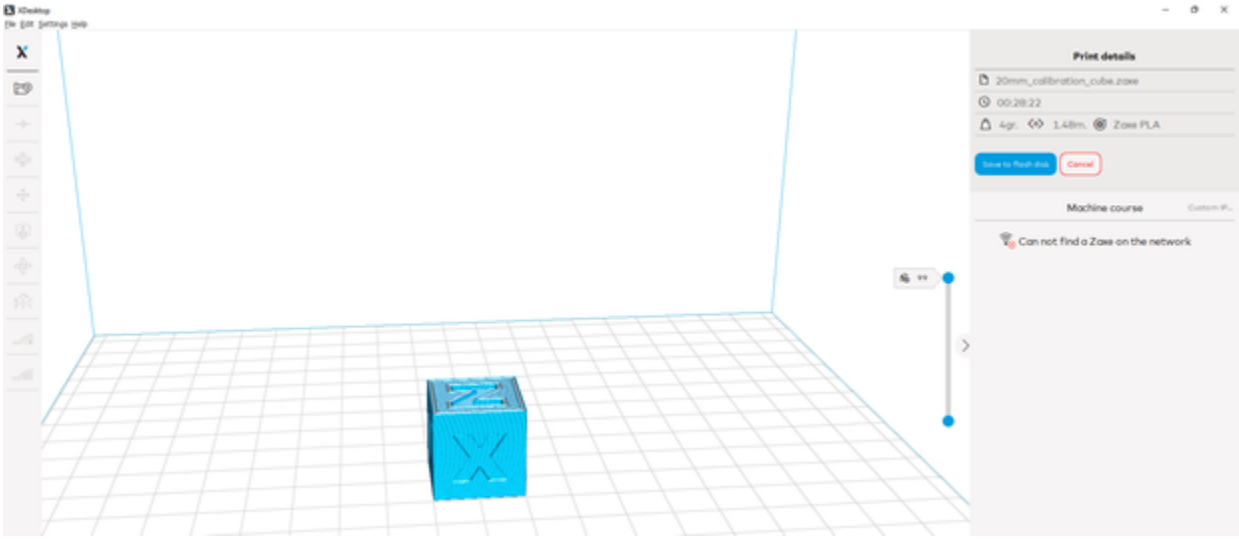
Müfredat Kodu / Ünite / Konu	İCT-202-5 / 3 Boyutlu Baskı / Dilimleme Teknikleri ve Yazıcı Seçimi
Hedeflenen Süre	38 Dakika

🚀 Dilimleme Teknikleri ve Yazıcı Seçimi

Bu bölümde bir adet 10*10*10 boyutlarında küp Cura içerisine .stl formatında alınmalıdır. Küpün boyutları 20*20*20 olarak XDesktop üzerinden boyutlandırılmalıdır. Küp baskı için önce filament türü seçilerek, 0.2 katman yüksekliğinde ve %20 dolulukla baskıya ayarlanmalı ve dilimlenmelidir.

Baskıya hazır hale getirilen tasarım USB'ye yüklenerek veya Wifi özelliği aynı ağa bağlı 3 Boyutlu Yazıcıya baskı için gönderilmelidir.





Kazanım

- Zaxe XDesktop programına .stl formatında tasarım yüklüyor olabilme.
- Baskı için ayarlamaların yapılabiliyor olması kazanımı beklenmektedir.

✅ Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Dilimleme Teknikleri ve Yazıcı Seçimi	☑ Eğitmen ☐ Öğrenciler	.stl Formatındaki tasarımın dosyasının XDesktop programı üzerinden dilimlenerek yazıcıya aktarılma süreçlerinin anlatılması beklenmektedir.
2	☑ Dilimleme Teknikleri ve Yazıcı Seçimi	☐ Eğitmen ☑ Öğrenciler	Tasarım dosyasının XDesktop programına aktarımından yazıcıya aktarılması sürecine kadar olan dilimleme tekniklerinin kullanılması hedeflenmektedir.

3 BOYUTLU YAZICI AYARLARI

Müfredat Kodu / Ünite / Konu	İCT-202-3.6 / 3 Boyutlu Baskı / 3 Boyutlu Yazıcı Ayarları ve Baskı
Hedeflenen Süre	18 Dakika

🚀 3 Boyutlu Yazıcı Ayarları ve Baskı

Bu bölümde bir önceki derste USB veya Wi-Fi ile aynı ağa bağlı 3 Boyutlu Yazıcılara yüklenen baskı yazıcıya yükleme ve baskı alma süreci hakkında bilgilendirme yapılmalıdır. Bilgilendirme uygulamalı olarak yapılmalıdır.

📘 Kazanım

- 3 Boyutlu Yazıcı kullanımının öğrenilmesi.
- Filament değiştirebilme (yükleme-çıkarma).
- 3 Boyutlu Yazıcı kalibrasyon ve tabla ayarlarını yapabilme.
- Baskı sonrası tabladan parçayı alma ve temizleme süreçleri hakkında bilgi sahibi olunması kazanım olarak hedeflenmektedir.

✅ Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ 3 Boyutlu Yazıcı Ayarları ve Baskı	☑ Eğitmen ☐ Öğrenciler	Bu bölüm kapsamında öğrenciye yazıcıya baskı öncesi ve sonrası yapılması gereken işlemler aktarılmalıdır.
2	☑ 3 Boyutlu Yazıcı Ayarları ve Baskı	☐ Eğitmen ☑ Öğrenciler	Bu bölümde yazıcının kalibrasyon ayarlarını yapılabiliyor olması ve baskı öncesi, sonrası bakımların ve işlemlerin yapılabiliyor olması beklenmektedir.

3 BOYUTLU YAZICILARDA MEKANIK BAKIM

Müfredat Kodu / Ünite / Konu	İCT-202-7 / 3 Boyutlu Baskı / 3 Boyutlu Yazıcı Mekanik Bakımı
Hedeflenen Süre	18 Dakika

🚀 3 Boyutlu Yazıcı Mekanik Bakımı

Bu bölümde kullanılan 3 Boyutlu Yazıcıların bakımları hakkında bilgi verilmesi beklenmektedir.

📕 Bilgi

- **Mil Bakımı** ➡️ Yazıcı ile gelen kutu içerisinde yer alan bakım yağı ile miller önce temizlenmeli daha sonra yağlanmalıdır.
- **Tabla Bakımı** ➡️ Baskı tablası yüzeyler önce temizlenip sonrasında ise tabla yapıştırıcısı sürülmelidir.
- **Filament Saklama Koşulları** ➡️ Filamentler oda sıcaklığında nemsiz ve rutubetsiz ortamda kapalı kutuda saklanmalıdır. Aksi takdirde filamentler de gevrekleşme olarak kırılganlıkları artabilir.
- **Nozzle Bakımı** ➡️ Nozzle tıkandığı zaman değişim yapılmalıdır. Baskılardan sonra ve filament değişim sırasında kontrolleri yapılmalıdır. Değişim esnasında sıcaklığa dikkat edilmelidir.

📘 Kazanım

- Yazıcı kalibrasyon ve bakım süreçleri hakkında bilgi sahibi olmak.

✅ Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ 3 Boyutlu Yazıcı Mekanik Bakımı	☑ Eğitmen ☐ Öğrenciler	3 Boyutlu Yazıcıların bakımları ve parça değişimleri hakkında uygulamalı bilgilendirme yapılması beklenmektedir.
2	☑ 3 Boyutlu Yazıcı Mekanik Bakımı	☐ Eğitmen ☑ Öğrenciler	3 Boyutlu Yazıcıların bakımlarının ve parça değişimlerinin yapılabiliyor olması herhangi bir arıza durumunda çözüm üretebiliyor olmak hedeflenmektedir.

TINKERCAD: 3 BOYUTLU UYGULAMALAR

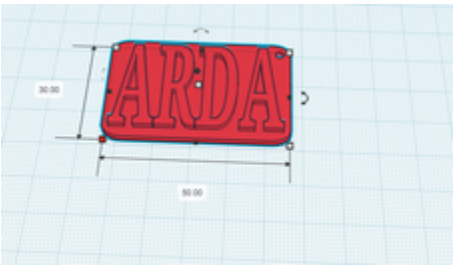
Müfredat Kodu/Ünite/Konu	İCT-201-4 /Tinkercad/Tinkercad: 3 Boyutlu Uygulamalar
Hedeflenen Süre	450 Dakika

🚀 3 Boyutlu Uygulamalar

Bu bölümde aşağıda belirtilen uygulamaların yapılabiliyor olması amaçlanmaktadır.

🧰 Uygulamalar

- İsimli anahtarlık yapımı (50mm x 30mm x 3mm)
- Kar tanesi (50mm x 30mm x 3mm)
- Stres çarkı (50mm x 30mm x 3mm)
- Thingerverse web sitesinden hazır stl dosyalarını içe aktarma yöntemi ile aktarılması ve düzenleme yapılması çıktı alınarak baskıya verilmesi uygulamasının yapılması.
- Araç yapımı
- Örnek cisim tasarımları yapılmalıdır.,



📘 Kazanım

- Tasarım becerisinin kazanılması amaçlanmaktadır.
- Tasarımdan 3 Boyutlu baskı çıktısı alabiliyor olmak.

✅ Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Uygulamalar	☑ Eğitmen ☐ Öğrenciler	Eğitmen bu bölümde uygulamaları yapılırken öğrencilere yol göstererek çizim komutlarını doğru kullanıldığına dikkat etmelidir. Uygulamaların ölçülere uygun bir şekilde yapılabiliyor olması hedeflenmektedir. Tasarımlarda Referans ve Dokümantasyon bölümündeki kaynaklar dikkate alınmalıdır.
2	☑ Uygulamalar	☐ Eğitmen ☑ Öğrenciler	Öğrenci eğitmenin önerdiği tasarımları yapabiliyor olmalıdır. Minimum Köpük Çubuğu, Kar Tanesi, Stres Çarkı, Yüzük Yapımı, Bardak Kupa Yapımı, Robot Kol Yapımı uygulamalarını yapılabilmesi hedeflenmektedir.

FUSION 360 MÜFREDAT TAKVIMI

FUSION360 (İCT-203) EĞİTİM MÜFREDATI														
EĞİTİM İÇERİĞİ / HAFTA	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.
Fusion360 : Tanıtım ve Giriş														
Fusion 360: 2 Boyutlu Tasarım ve Komutları														
Fusion360 : Sketch Çizim Komutları														
Fusion360 : Kısıtlama (Constraints-Sketch Palette) Komutu														
Fusion360 : Budama (Trim) Komutu														
Fusion360 : Kavis -Yuvarlatma (Fillet) Komutu														
Fusion360 : Aynalama (Mirror) Komutu														
Fusion360 : Öteleyerek Çoğaltma (Offset) Komutu														
Fusion360 : Dairesel Çoğaltma (Circular Pattern) ve Doğrusal Çoğaltma (Rectangular Pattern) Komutları														
Fusion 360: Katı Oluşturma ve Komutları														
Fusion 360 : Extrude - Katı Model Oluşturma														
Fusion 360 : Döndürerek Katı Oluşturma (Revolve) Komutu														
Fusion 360 : Yol Kullanarak Katı Oluşturma (Sweep) Komutu														
Fusion 360 : Profiller ve Yüzeyler Arasında Katı Oluşturma (Loft) Komutu														
Fusion 360 : Delik Sihirbazı (Hole) Komutu														
Fusion 360 : Web Komutu														
Fusion 360 : Uygulama														

FUSION 360 EĞITIMI

Müfredat Kodu	İCT-203
Ders	Fusion 360
Ders İçeriği	• Fusion360: Tanıtım ve Giriş • Sketch Çizim Komutları • Kısıtlama (Constraints-Sketch Palette) Komutu • Budama (Trim) Komutu • Kavis -Yuvarlatma (Fillet) Komutu • Aynalama (Mirror) Komutu • Öteleyerek Çoğaltma (Offset) Komutu • Dairesel Çoğaltma (Circular Pattern) ve Doğrusal Çoğaltma (Rectangular Pattern) Komutları • Extrude - Katı Model Oluşturma • Döndürerek Katı Oluşturma (Revolve) Komutu • Yol Kullanarak Katı Oluşturma (Sweep) Komutu • Profiller ve Yüzeyler Arasında Katı Oluşturma (Loft) Komutu • Delik Sihirbazı (Hole) Komutu • Web Komutu • Uygulama
Seviye	İCT-S3: ILERI
Ön Koşullar	Tasarım alanında meraklı olmak ve bilgisayar kullanabiliyor olmak.
Amaç	3 Boyutlu modelleme için gerekli alt yapıyı oluşturarak, tasarım yapabiliyor ve montaj yapabiliyor olmak amaçlanmaktadır.
Hedef	Fusion360 tasarım programı ile 3 boyutlu tasarım, montaj ve analiz yapılabiliyor olması amaçlanmaktadır.
Gereksinimler	Bilgisayar (Dizüstü veya Masaüstü), Wi-Fi
Kazanım	• Mekanik tasarım ve montaj yapabiliyor olmak • Üretim için simülasyon yapabiliyor olmak • Malzeme seçimi ve diğer tasarım işlemlerini yapabiliyor olma • Render özelliği ile gerçek ürün görüntüsü alabilme
Referanslar & Dokümantasyon	• https://knowledge.autodesk.com/support/fusion-360/getting-started/caas/simplecontent/content/autodesk-fusion-360--C3-A7al-C4-B1-C5-9Fma-alanlar-C4-B1na-genel-bak-C4-B1-C5-9F.html • https://www.autodesk.com.tr/products/fusion-360/overview?term=1-YEAR&tab=subscription • https://serhatakademi.net/login/egitimler/courses/fusion360.php • https://www.youtube.com/watch?v=ZWx3nxHPn_U&list=PL71VJFXzv0pC7JpAIicORX099_I8vVdD7&ab_channel=TeknikKariyer
Durum	BEKLİYOR / UYGULANIYOR / TAMAMLANDI / DEĞERLENDİRİLDİ
Hedeflenen Süre	1260 Dakika
Eğitim Günleri	Başlama: .../.../.... Bitiş: .../.../....

FUSION360: TANITIM VE GIRIŞ

Müfredat Kodu / Ünite	İCT-203-1 /Tanıtım ve Giriş / Tanıtım ve Giriş
Hedeflenen Süre	90 Dakika

🚀 Fusion 360 Nedir ?

Fusion360, kullanıcıların çok çeşitli nesne ve sistemlerin 3D modellerini ve simülasyonlarını oluşturmasına olanak tanıyan popüler bir bilgisayar destekli tasarım (CAD) yazılımıdır. Autodesk tarafından geliştirilen program, tasarımcıların, mühendislerin ve diğer profesyonellerin ayrıntılı, yüksek kaliteli tasarımlar oluşturmasını kolaylaştıran çok çeşitli özellikler ve araçlar sunar.

Fusion360'ın en önemli özelliklerinden biri kapsamlı tasarım içeriği kütüphanesidir. Bu kütüphane, vidalar, somunlar, cıvatalar, rulmanlar ve yaygın olarak kullanılan diğer mekanik parçalar gibi çok çeşitli önceden tasarlanmış bileşenleri içerir. Bu bileşenler bir tasarıma kolayca eklenebilir ve sıfırdan oluşturmaya kıyasla zamandan ve emekten tasarruf sağlar.

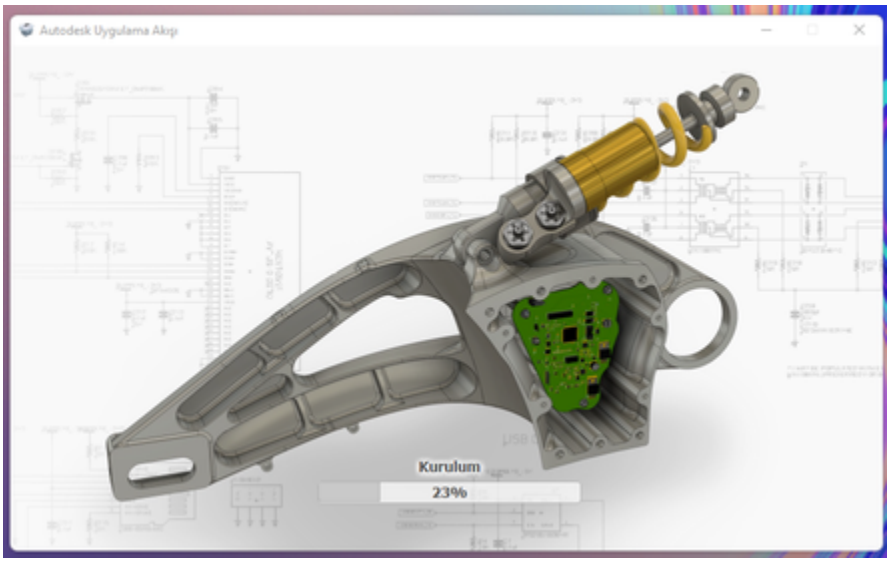
Tasarım içeriği kütüphanesi ayrıca bir dizi parametrik modelleme aracı içerir. Bu araçlar, kullanıcıların bir bileşenin boyutu, şekli ve malzemesi gibi özelliklerini tanımlamasına ve ardından bu özelliklere dayalı olarak bileşenin birden fazla varyasyonunu hızlı bir şekilde oluşturmasına olanak tanır. Bu, farklı tasarım seçeneklerini denemeyi ve belirli bir sorun için en iyi çözümü hızlı bir şekilde bulmayı kolaylaştırır.

Önceden tasarlanmış bileşenlere ve parametrik modelleme araçlarına ek olarak, Fusion360 onu güçlü bir tasarım aracı haline getiren bir dizi başka özellik de içerir. Örneğin program, kullanıcıların bir tasarımın performansını ve davranışını farklı koşullar altında test etmelerini sağlayan simülasyon araçları içerir. Bu, tasarımcıların potansiyel sorunları belirlemelerine ve fiziksel bir prototip oluşturmadan önce gerekli iyileştirmeleri yapmalarına yardımcı olabilir.

Fusion360 ayrıca ekiplerin bir tasarım projesi üzerinde birlikte çalışmasını kolaylaştıran işbirliği araçları da sunuyor. Birden fazla kullanıcı bir tasarıma aynı anda erişebilir ve düzenleyebilir ve program değişiklikleri otomatik olarak izler ve tasarım sürecinin bir geçmişini tutar. Bu, tasarımcıların işbirliği yapmasını, fikirlerini paylaşmasını ve bir projenin ilerlemesini takip etmesini kolaylaştırır.

Genel olarak, Fusion360'ın tasarım içerik kütüphanesi ve diğer özellikleri, onu yüksek kaliteli tasarımlar oluşturmak için güçlü ve çok yönlü bir araç haline getiriyor. Önceden tasarlanmış bileşenleri, parametrik modelleme araçları, simülasyon yetenekleri ve işbirliği özellikleri, onu tasarımcılar, mühendisler ve çeşitli alanlarda çalışan diğer profesyoneller için değerli bir kaynak haline getiriyor.

✍️ Örnek Modelleme



✍️ Kullanıcı Hesabı ve Kurum

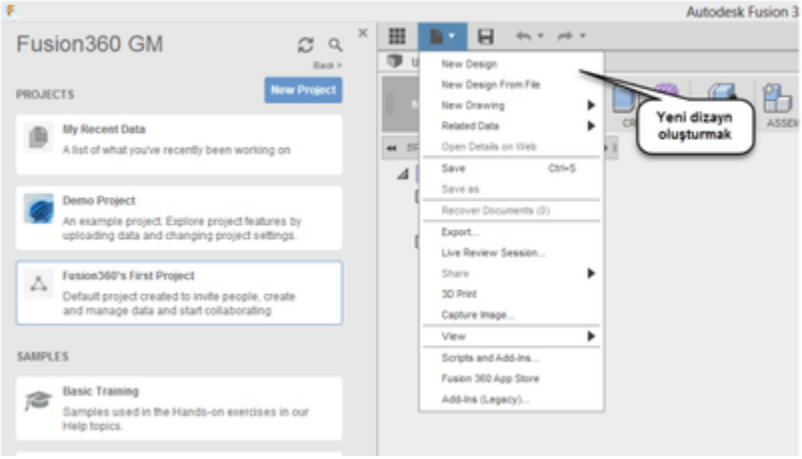
https://www.autodesk.com/education/edu-software/overview?sorting=featured&filters=individual Web adresinden öğrenci hesabı ile kayıt olarak program indirilir bilgisayara gerekli kurulum yapılmalıdır.

💻 Arayüz Tanıtımı

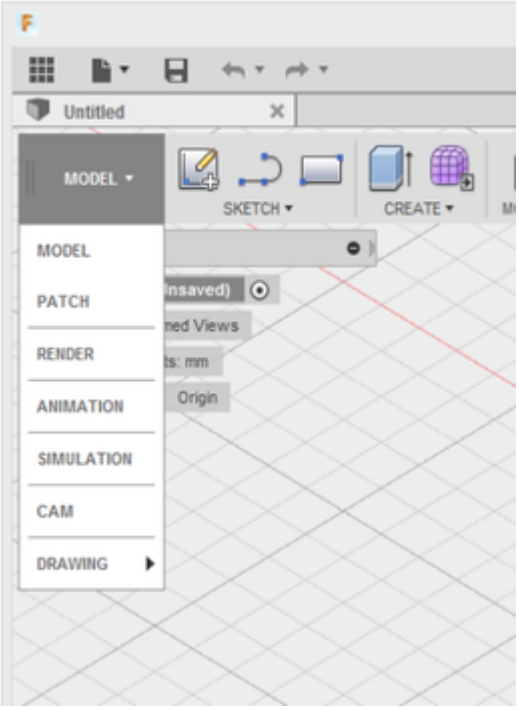
Kurulum işlemleri tamamlandıktan sonra arayüz tanıtımı yapılmalıdır.

⏳ Bilgi

- Program ara yüzünden **File** bölümünden **New Design** seçimi yapmalı; ya da önceden hazırlayıp kaydedilen bir dosyadan başlamak istenirse, **File** bölümünden **New Design From File** seçimine basılır. Ekrana "Untitled" (başlıksız / adsız) penceresi gelir.



- Yapılacak tasarıma göre çalışma modları seçilmelidir.



- **Model :** Bir parçanın ya da montajın modellenmesinde kullanılan kısımdır.
- **Patch :** Yüzey Modelleme ve yüzey düzenleme işlemlerinin yapıldığı kısımdır.
- **Render** : Bir parçanın ya da montajın fotoğraf görüntüleme işlemlerinin yapıldığı kısımdır.
- **Animator** : Bir parçanın ya da montajın görsel efektleri ve hareketli canlandırma işlemlerinin yapıldığı kısımdır.
- **Simulation :** Bir parçanın ya da montajın kuvvet analizi ve dayanım hesaplama işlemlerinin yapıldığı kısımdır.

- **Cam :** Bir parçanın üretiminde kullanılacak sayısal kod türetme ve takım yolları hesaplama işlemlerinin yapıldığı ksıımdır.
- **Drawing :** Bir parçanın ya da montajın teknik resminin oluşturulması veya animasyonunun iki boyutlu görünümlerinin çıkarılması işlemlerinin yapıldığı bölümdür.

Kazanım

- Fusion360 kurulumunun tamamlanması,
- Fusion360 arayüzünün keşfedilmesi,
- Fusion360 kullanımı hakkında önbilgiye sahip olunması.

Dersin Hedefleri

- Fusion 360 kurulum, arayüz ve kullanımı hakkında uygulamalı olarak anlatım yapılması hedeflenmektedir.

SKETCH ÇIZIM KOMUTLARI

Müfredat Kodu / Ünite / Konu	İCT-203-2.1 /2 Boyutlu Tasarım ve Komutları /Sketch Çizim Komutları
Hedeflenen Süre	180 Dakika

🚀 Sketch Çizim Komutları

Fusion 360 Sketch çizim komutları, Fusion 360'de 2 boyutlu çizimler oluşturmak için kullanılan komutlardır. Bu komutlar sayesinde, bir modelin tasarımına başlamak için gerekli temel çizimleri oluşturabilirsiniz. Fusion 360'de bir Sketch oluşturmak için aşağıdaki adımları izleyebilirsiniz:

1. Öncelikle, Fusion 360 ana ekranında "Sketch" sekmesine tıklayarak Sketch moduna geçin.
2. Daha sonra, "Create Sketch" butonuna tıklayarak bir Sketch oluşturma ekranı açın.
3. Bu ekranda, oluşturmak istediğiniz Sketch'in bulunacağı bir yüzey seçin. Bu yüzey üzerinde çizimlerinizi oluşturacaksınız.
4. Daha sonra, oluşturduğunuz Sketch'e isim verin ve "OK" butonuna tıklayarak Sketch oluşturma işlemini tamamlayın.

Bu adımları takip ederek Fusion 360'de bir Sketch oluşturabilirsiniz. Daha sonra, Sketch çizim komutlarını kullanarak çizimlerinizi oluşturmaya başlayabilirsiniz.

🧰 Bilgi

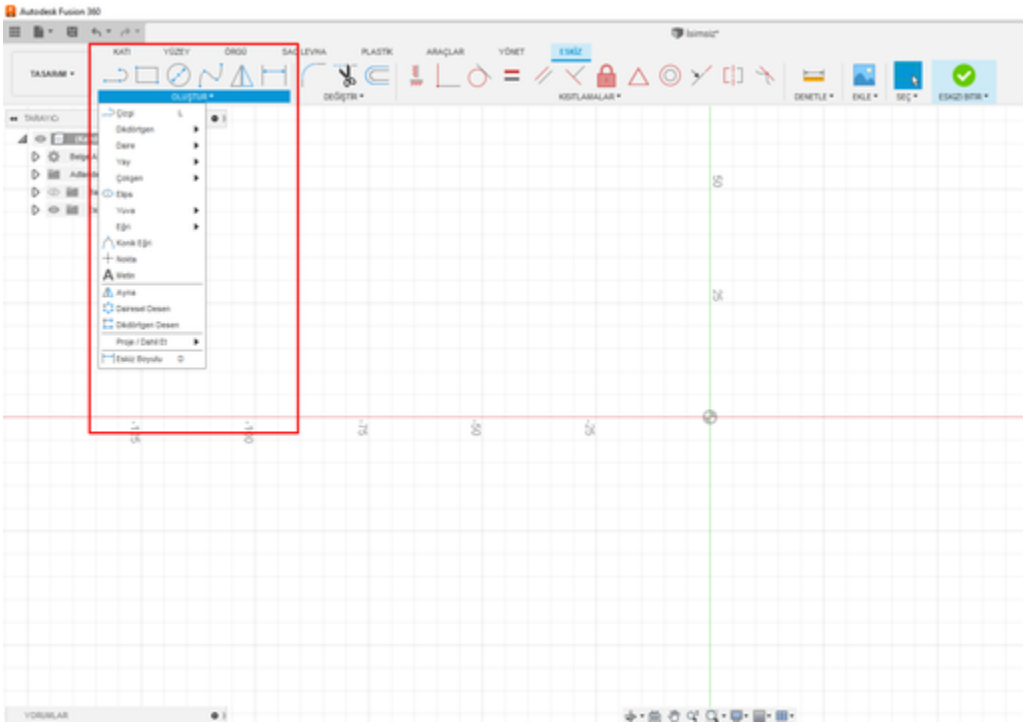
Fusion 360 Sketch Çizim komutları, Fusion 360'de 2 boyutlu çizimler oluşturmak için kullanılan komutlardır. Bu komutlar sayesinde, bir modelin tasarımına başlamak için gerekli temel çizimleri oluşturabilirsiniz. Fusion 360'de bulunan Sketch Çizim komutları şunlardır:

- **Çizgi (Line):** Bu komut sayesinde, iki nokta arasına bir çizgi oluşturabilirsiniz.
- **Daire (Circle):** Bu komut sayesinde, belirlediğiniz bir merkez noktası ve yarıçap boyutu kullanarak bir daire oluşturabilirsiniz.
  - Üç Nokta Göstererek Daire Çizmek (3-Point Circle).
  - Üç Teğet Nokta Göstererek Daire Çizmek (3-Tangent Circle).
  - İki Teğet Nokta Göstererek ve Yarıçap Ölçülendirilerek Daire Çizmek.
- **Yay (Arc)**
  - İki Uç ve Bir Orta Nokta Gösterilerek Yay Çizmek (3-Point Arc).
  - Merkez, Yarıçap ve Bitiş Noktası Göstererek Yay Çizmek (Center Point Arc).
  - Çizim Elemanın (Teğet) Ucu, Bitiş Noktası İle Yay Çizmek (Tangent Arc).
- **Dikdörtgen (Rectangle):** Bu komut sayesinde, belirlediğiniz bir merkez noktası ve boyutlar kullanarak bir dikdörtgen oluşturabilirsiniz.
  - İki Nokta İle Dikdörtgen Çizmek (2-Point Rectangle).
  - Üç Nokta ile Dikdörtgen Çizmek (3-Point Rectangle).
  - Merkez Noktası İle Dikdörtgen Çizmek (Center Rectangle).
- **Kanal (Slot / Yuva)**
  - İki Eksen Noktası ve Yarıçap İle Kanal Çizmek (Center To Center Slot).
  - Merkez, Eksen Noktası ve Yarıçap ile Kanal Çizmek (Center Point Slot).
  - İki Uç Noktası ve Yarıçap ile Kanal Çizmek (Overall Slot).
- **Poligon (Polygon):** Bu komut sayesinde, belirlediğiniz bir merkez noktası, yarıçap boyutu ve köşe sayısı kullanarak bir poligon oluşturabilirsiniz.
  - Daire Dışına Çokgen Çizmek (Circumscribed Polygon).
  - Daire İçine Çokgen Çizmek (Inscribed Polygon).
  - Kenar Boyutu Verilerek Çokgen Çizmek (Edge Polygon).
- **Elips (Ellipse):** Bu komut sayesinde, belirlediğiniz bir merkez noktası, yarıçap boyutları ve eksen açısı kullanarak bir elips oluşturabilirsiniz.

- **Parabol Eğrisi / Konik Eğri (Conic Curve)**
- **Spline**
- **Text**
- **Nokta (Point)**

Bu komutlar sadece Sketch Çizim komutlarının bazı temel özellikleridir. Fusion 360'de bulunan diğer çizim komutlarını da kullanarak daha karmaşık çizimler oluşturabilir ve modelinizi tasarlamaya devam edebilirsiniz.

Örnek Modelleme



Kazanım

- 2D ve 3D çizim yapma yeteneği.
- Ölçüleri kolaylıkla belirleme ve çizimleri ölçeklendirme imkanı.
- Çeşitli geometrik şekillerin oluşturulması ve düzenlenmesi.
- Çizimleri kolayca düzenleme ve değiştirme imkanı.
- Taslakları gerçek parçalara dönüştürme ve ürünlerin tasarımını gerçekleştirme.
- Ölçeklendirme, kırpma ve bükme gibi işlemleri kolayca yapma imkanı.
- Çizimleri 3D modeller haline dönüştürme ve gerçek zamanlı görüntüleme imkanı.
- Çizimleri diğer tasarım yazılımları ile uyumlu hale getirme ve paylaşma imkanı.
- İleri düzey tasarım ve üç boyutlu modelleme yetenekleri sağlar.
- Ölçeklendirme, çevirme, döndürme, düzleştirme ve diğer düzenleme işlemlerini kolayca gerçekleştirebilme olanağı sunar.
- Özelleştirilebilir kalıplar ve sabitler ile doğru ve hassas çizimler yapabilme imkanı sağlar.
- Birden fazla katman ve görünüm seçenekleriyle çizimlerin düzenlenmesini ve görünümünü değiştirme olanağı sunar.
- İç ve dış çizimleri birleştirme, kesişim ve çakışmaları kontrol etme gibi çoklu çizimler oluşturabilme imkanı sağlar.
- Çizimlerin 3D modellemeye dönüştürülmesi ve ürünlerin tasarımında kullanılabilmesi için gerekli araç ve özellikleri sunar.
- İnternet üzerinden paylaşım ve işbirliği yapılabilme olanağı sağlar.

✅ **Dersin Hedefleri**

- Sketch bölümünün tanıtılması ve komutların uygulamalı olarak anlatılması beklenmektedir.

NOTLAR

KISITLAMA (CONSTRAINTS-SKETCH PALETTE) KOMUTU

Müfredat Kodu / Ünite / Konu	İCT-203-2.2 /2 Boyutlu Tasarım ve Komutları / Kısıtlama (Constraints-Sketch Palette) Komutu
Hedeflenen Süre	65 Dakika

🚀 Kısıtlama (Constraints - Sketch Palette) Komutu

Fusion360 Constraints - Sketch Palette komutu, bir tasarımdaki Sketch Palette uygulamak için kullanılan bir araçtır. Kısıtlamalar, bir taslaktaki bir nesnenin boyutu, şekli ve konumu gibi farklı öğeler arasındaki ilişkiyi tanımlamak için kullanılır.

Fusion360 Constraints - Sketch Palette komutunu kullanmak için kullanıcı önce tasarım çalışma alanında bir Sketch oluşturur. Kullanıcı daha sonra programın araç çubuğundan veya menüsünden Constraints - Sketch Palette komutunu seçebilir ve Sketch öğelere kısıtlamalar uygulamak için bu komutu kullanabilir.

Constraints - Sketch Palette komutu, bir nesnenin boyutunu tanımlayan boyutsal kısıtlamalar ve bir nesnenin şeklini ve konumunu tanımlayan geometrik kısıtlamalar gibi bir dizi farklı kısıtlama içerir. Kullanıcı tek bir taslağa birden fazla kısıtlama uygulayabilir ve gerektiğinde kısıtlamaları değiştirebilir veya kaldırabilir.

Fusion360 Constraints-Sketch Palette komutu, bir tasarımın doğruluğunu ve hassasiyetini sağlamak için önemli bir araçtır. Kullanıcı, bir taslağa kısıtlamalar uygulayarak nesnelerin boyutunu, şeklini ve konumunu kontrol edebilir ve taslakta tasarımın bütünlüğünü tehlikeye atacak değişiklikleri önleyebilir.

Genel olarak, Fusion360 Constraints-Sketch Palette komutu, hassas ve doğru tasarımlar oluşturmak isteyen tasarımcılar için değerli bir araçtır. Kullanıcıların bir taslağa kısıtlamalar uygulamasına olanak tanıyarak bir tasarımdaki öğelerin düzgün bir şekilde hizalanmasını ve boyutlandırılmasını sağlamaya yardımcı olabilir ve tasarımın bütünlüğünü tehlikeye atacak değişiklikleri önleyebilir.

🧰 Bilgi

Geometrik kısıtlamalar: Bu kısıtlamalar, hizalama, yönlendirme ve simetri gibi bir nesnenin şeklini ve konumunu tanımlar.

Çakışan kısıtlamalar: Bu kısıtlamalar iki noktayı veya nesneyi aynı konumu işgal etmeye zorlar.

Kolinear kısıtlamalar: Bu kısıtlamalar iki veya daha fazla çizgiyi birbiriyle mükemmel şekilde hizalanmaya zorlar.

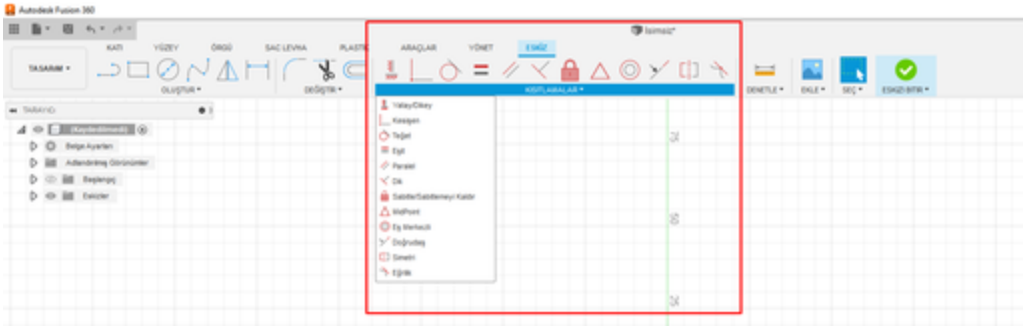
Teğet kısıtlamaları: Bu kısıtlamalar iki eğriyi belirli bir noktada birbirlerine teğet olmaya zorlar.

Dik kısıtlamalar: Bu kısıtlamalar iki çizgiyi birbirine dik olmaya zorlar.

Yatay/Dikey kısıtlamalar: Bu kısıtlamalar bir çizgi veya nesneyi tamamen yatay veya dikey olmaya zorlar.

Sabitleme kısıtlamaları: Bu kısıtlamalar bir nesnenin konumunu sabitleyerek taşınmasını veya değiştirilmesini önler.

Genel olarak, Fusion360 Constraints-Sketch Palette komutu, bir tasarımdaki Sketch 'e uygulanabilecek çok çeşitli kısıtlamalar içerir. Bu kısıtlamalar bir tasarımın doğruluğunu, hassasiyetini ve esnekliğini sağlamaya yardımcı olabilir.



Kazanım

Fusion360 Constraints-Sketch Palette komutunu kullanmanın kazanımları şunlardır:

- **Doğruluk:** Kullanıcı bir taslağa kısıtlamalar uygulayarak nesnelerin boyutunu, şeklini ve konumunu kontrol edebilir, bu da tasarımın doğru ve hassas olmasını sağlamaya yardımcı olabilir.
- **Hassasiyet:** Kısıtlamalar, bir taslakta tasarımın bütünlüğünü tehlikeye atacak değişikliklerin yapılmasını önlemeye yardımcı olabilir. Bu, özellikle küçük hataların bile önemli sonuçlar doğurabileceği mühendislik ve imalat gibi alanlarda önemli olabilir.
- **Esneklik:** Constraints-Sketch Palette komutu, çok çeşitli tasarımlar oluşturmak için birleştirilebilen ve değiştirilebilen bir dizi farklı kısıtlama içerir. Bu, kullanıcıların farklı tasarım seçeneklerini denemelerine ve belirli bir sorun için en iyi çözümü bulmalarına olanak tanır.

Genel olarak, Fusion360 Constraints-Sketch Palette komutu, tasarımcıların doğru, hassas ve esnek tasarımlar oluşturmasına yardımcı olabilecek değerli bir araçtır. Bir taslağa kısıtlamalar uygulama ve bir tasarımın bütünlüğünü tehlikeye atacak değişiklikleri önleme yeteneği, onu bir tasarımın kalitesini sağlamak için önemli bir araç haline getirir.

Dersin Hedefleri

- Kısıtlama (Constraints-Sketch Palette) Komutunun kullanımının uygulamalı anlatılması beklenmektedir.

BUDAMA (TRIM) KOMUTU

Müfredat Kodu / Ünite / Konu	İCT-203-2.3 /2 Boyutlu Tasarım ve Komutları / Budama (Trim) Komutu
Hedeflenen Süre	65 Dakika

🚀 Budama (Trim) Komutu

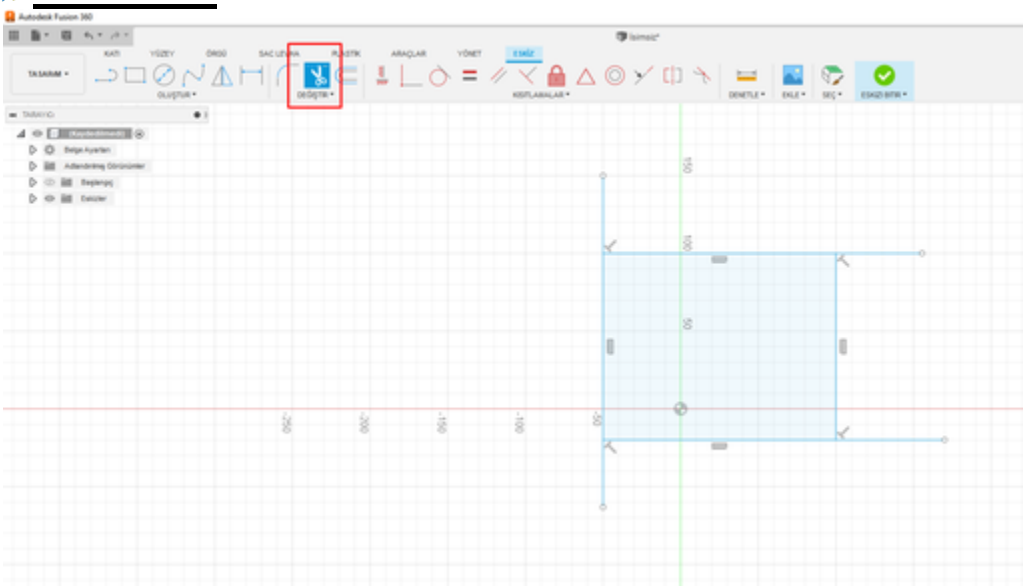
Fusion360 Dimension Trim (Boyut Kırpma) komutu, bir taslaktaki fazla boyutları kırpmak veya kaldırmak için kullanılan bir araçtır. Boyutlar, CAD tasarımlarında nesnelerin boyutunu, şeklini ve pozisyonunu Dimension Trim komutunu kullanmak için kullanıcı önce programın araç çubuğundan veya menüsünden aracı seçer. Kullanıcı daha sonra kırpmak için tasarım çalışma alanındaki bir boyuta tıklayabilir veya birden fazla boyut seçmek ve bunları aynı anda kırpmak için tıklayıp sürükleyebilir.

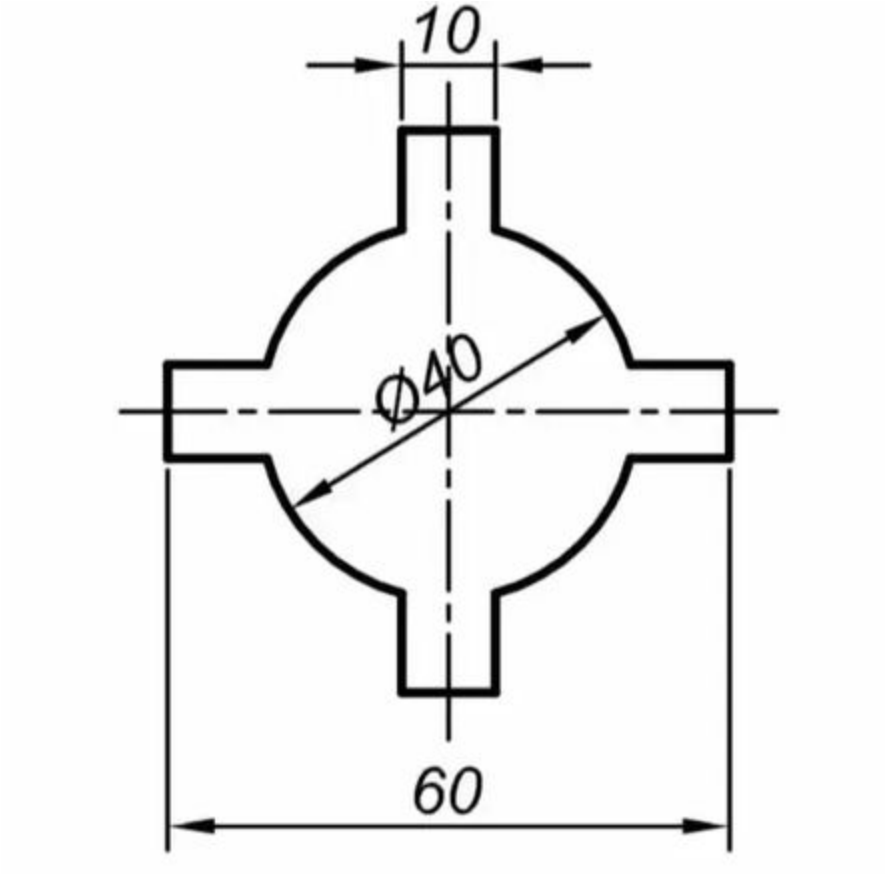
Dimension Trim komutu, bir tasarımı toparlamak ve artık ihtiyaç duyulmayan fazla boyutları kaldırmak için kullanışlıdır. Kullanıcı bu boyutları kaldırarak bir tasarımın görünümünü basitleştirebilir ve okunmasını ve anlaşılmasını kolaylaştırabilir.

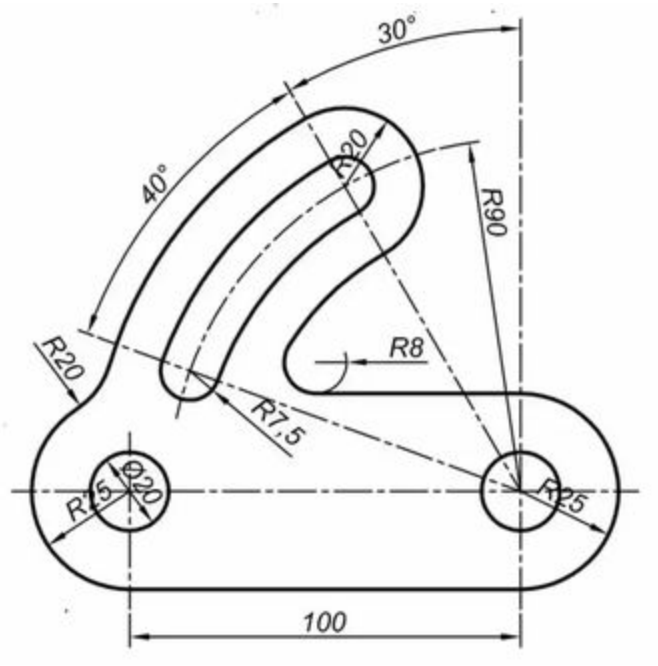
Fusion360 Dimension Trim komutu, her beceri düzeyinden tasarımcı tarafından kullanılabilecek basit ve anlaşılır bir araçtır. Bir tasarımın netliğini ve sadeliğini korumak için kullanışlı bir araçtır ve bir tasarımı daha profesyonel ve gösterişli hale getirmeye yardımcı olabilir.

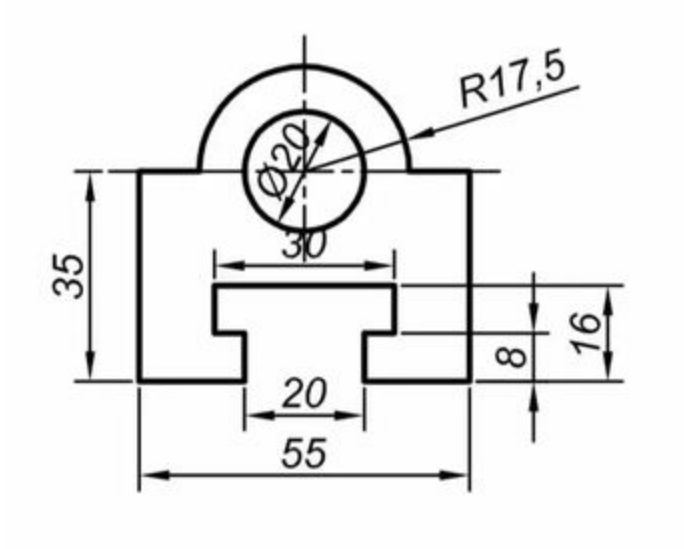
> 🎗️ Bu bölümde eğitmen Dimension Trim komutunu örneklerle uygulamalı olarak açıklamalıdır.

✍️ Örnek Modelleme




10
Ø40
60


30°
40°
R20
R90
R90
R8
R20
R7,5
R15
Ø20
R25
100



Kazanım

- **Basitlik:** Kullanıcı, bir tasarımdaki fazla boyutları kırparak tasarımı daha basit ve okunması ve anlaşılması daha kolay hale getirebilir. Bu, özellikle çok sayıda boyuta sahip tasarımlar veya teknik olmayan bir kitleye yönelik tasarımlar için yararlı olabilir.
- **Netlik:** Fazla boyutların kaldırılması bir tasarımı daha düzenli ve gezinmesi daha kolay hale getirebilir. Bu, tasarımcıların ve diğer kullanıcıların bir tasarımdaki boyutları daha kolay bulmasına ve anlamasına yardımcı olabilir.
- **Profesyonellik:** Fazla boyutların kırpılması bir tasarımın daha gösterişli ve profesyonel görünmesine yardımcı olabilir. Bu, özellikle gerçek iş hayatında müşterilerle veya diğer paydaşlarla paylaşılacak tasarımlar için önemli olabilir.
- **Verimlilik:** Fazla boyutları kırpmak, her boyutu tek tek manuel olarak silmeye kıyasla zaman ve emek tasarrufu sağlayabilir. Bu, özellikle son teslim tarihine göre çalışan veya bir tasarımda çok sayıda değişiklik yapması gereken tasarımcılar için yararlı olabilir.

Dersin Hedefleri

- Budama (Trim) Komutunun uygulamalı olarak anlatılması beklenmektedir.

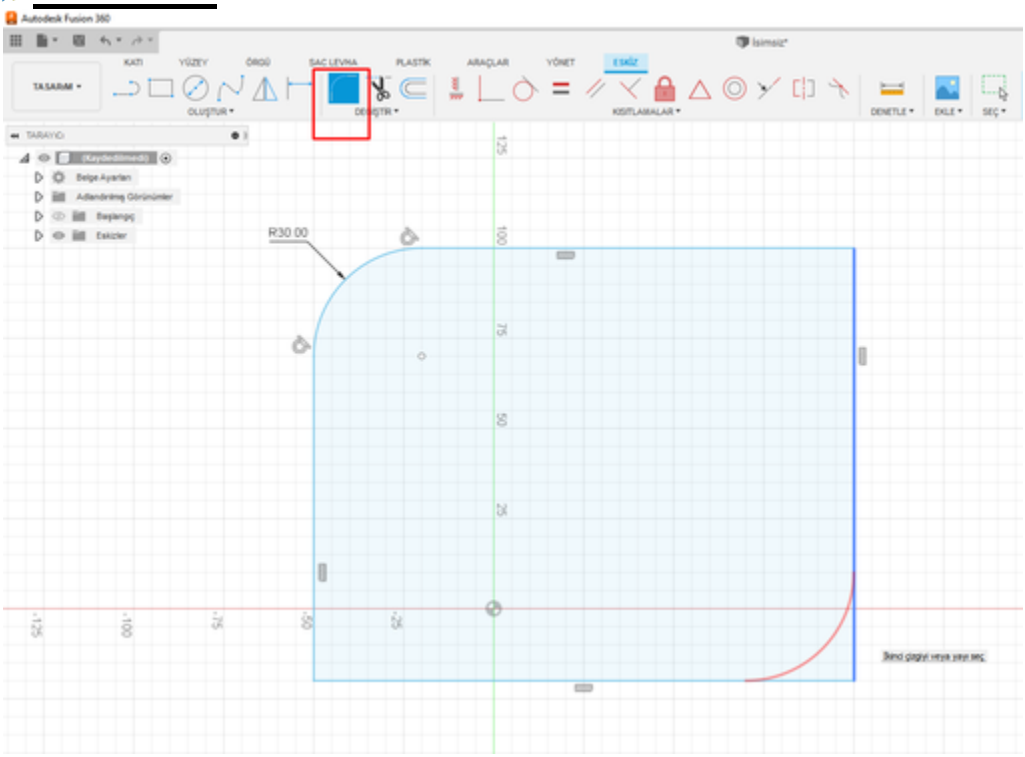
KAVIS -YUVARLATMA (FILLET) KOMUTU

Müfredat Kodu / Ünite / Konu	İCT-203-2.4 /2 Boyutlu Tasarım ve Komutları / Kavis -Yuvarlatma (Fillet) Komutu
Hedeflenen Süre	45 Dakika

🚀 Kavis -Yuvarlatma (Fillet) Komutu

Fusion360 Fillet komutu, komut sayesinde, bir nesnenin köşelerini yumuşaklaştırarak daha yumuşak bir görünüm elde edebilirsiniz. Fillet komutunu kullanarak, nesnenin köşelerini belirli bir yarıçap değeriyle yumuşaklaştırabilirsiniz. Bu sayede, nesnenizin daha estetik bir görünümü elde edebilir ve tasarımınızı daha iyi bir hale getirebilirsiniz. Fillet komutunu kullanarak, aynı zamanda nesnenizin köşelerinde meydana gelen stresleri azaltabilir ve nesnenizin daha dayanıklı bir hale gelmesini sağlayabilirsiniz.

✍️ Örnek Modelleme



📘 Kazanım

Fusion 360 Fillet komutu, bir model üzerinde yuvarlak köşeler (ya da filletler) oluşturmak için kullanılır. Bu komut sayesinde, bir modelin görünümünü daha estetik ve kullanışlı hale getirebilir ve kesme hatlarını düzgünleştirebilirsiniz. Fillet komutunun kullanarak elde edebileceğiniz bazı kazanımlar şunlardır:

- Fillet komutunu kullanarak, bir modelin görünümünü daha estetik ve güzel hale getirebilirsiniz.
- Fillet komutunu kullanarak, bir modelin kesme hatlarını düzgünleştirebilir ve daha kullanışlı hale getirebilirsiniz.
- Fillet komutunu kullanarak, bir modelin yüzeylerine daha yumuşak geçişler sağlayarak daha yüksek kaliteli bir görünüm elde edebilirsiniz.
- Fillet komutunu kullanarak, bir modelin üzerinde bulunan köşe noktalarını yuvarlayarak daha az stres oluşumuna neden olabilir ve böylece modelin daha dayanıklı hale gelmesini sağlayabilirsiniz.
- Fillet komutunu kullanarak, bir modelin üzerinde bulunan köşe noktalarını kolaylıkla değiştirebilir ve düzenleyebilirsiniz. Bu sayede, modelin görünümünü istediğiniz gibi değiştirebilir ve kişiselleştirebilirsiniz.

Bu kazanımlar sadece Fillet komutunun kullanarak elde edebileceğiniz bazı temel kazanımlarıdır.

✅ **Dersin Hedefleri**

- Kavis -Yuvarlatma (Fillet) Komutu uygulamalı olarak anlatım yapılması hedeflenmektedir.

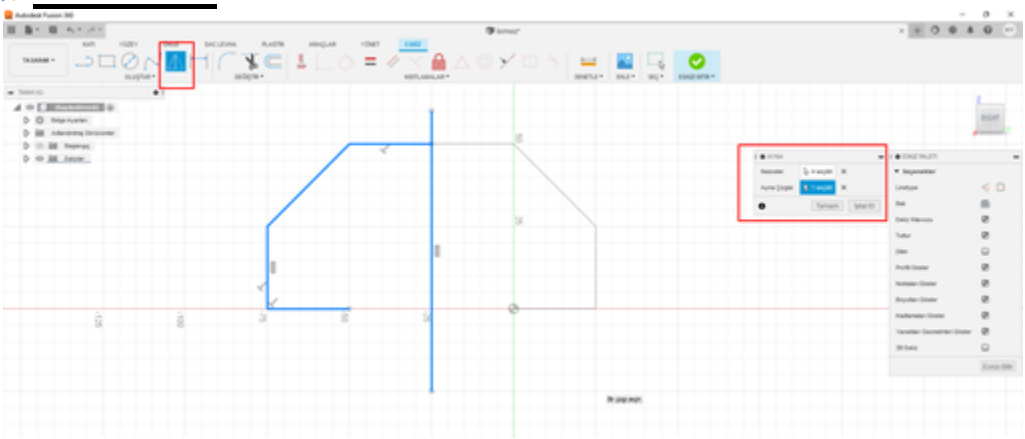
AYNALAMA (MIRROR) KOMUTU

Müfredat Kodu / Ünite / Konu	İCT-203-2.5 / 2 Boyutlu Tasarım ve Komutları / Aynalama (Mirror) Komutu
Hedeflenen Süre	45 Dakika

🚀 Aynalama (Mirror) Komutu

Fusion360 mirror komutu, bir çizimin yansımasını oluşturmaya yarayan bir araçtır. Bu komut ile bir çizim belirli bir eksen etrafında yansıtılarak bir kopyası oluşturulabilir. Yansıtma işleminin gerçekleştirileceği eksen manuel olarak belirlenebilir ya da çizimdeki bazı nesneler üzerinden otomatik olarak belirlenebilir. Bu sayede çizimlerin simetrisi kolayca sağlanabilir ve tasarım işlemleri hızlandırılabilir.

✍️ Örnek Modelleme



📘 Kazanım

- Çizimlerin simetrisini kolayca sağlar ve tasarım sürecini hızlandırmak.
- Yansıtma işlemini manuel olarak belirlenen eksenler üzerinden ya da çizimdeki nesneler üzerinden otomatik olarak gerçekleştirebilmek.
- Yansıtma işlemini tek bir nesne ya da birden fazla nesne üzerinde aynı anda gerçekleştirebilme imkanı sağlamak.
- İleri düzey tasarım ve modelleme yetenekleri ile çizimlerin daha doğru ve hassas hale getirilebilmesi olanağı sunmak.
- Çizimlerin 3D modellemeye dönüştürülmesi sırasında gerekli olan simetrik yapıların oluşturulmasını kolaylaştırmak.

✅ Dersin Hedefleri

- Aynalama (Mirror) Komutu uygulamalı olarak anlatılması beklenmektedir.

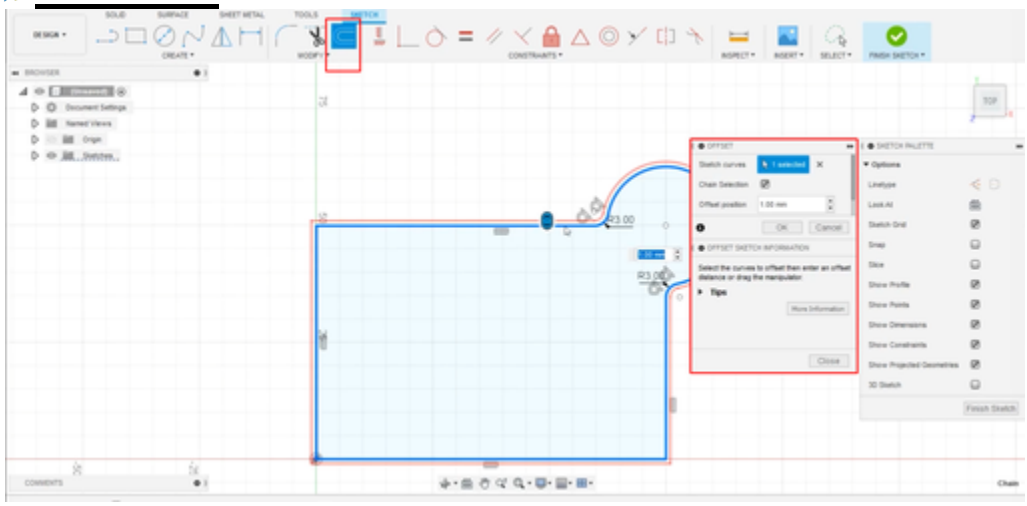
ÖTELEYEREK ÇOĞALTMA (OFFSET) KOMUTU

Müfredat Kodu / Ünite / Konu	İCT-203-2.6 /2 Boyutlu Tasarım ve Komutları / Öteleyerek Çoğaltma (Offset) Komutu
Hedeflenen Süre	70 Dakika

🚀 Öteleyerek Çoğaltma (Offset) Komutu

Fusion360 Öteleyerek Çoğaltma (Offset) komutu, bir çizimin belirli bir mesafeye çoğaltılmasını sağlayan bir araçtır. Bu komut ile bir nesne ya da bir çizimin dışındaki yüzeyler üzerinden bir kopya oluşturulabilir. Kopya oluşturulurken belirli bir mesafe veya nesne üzerinden otomatik olarak hesaplanan mesafe kullanılabilir. Bu sayede çizimlerin düzenlenmesi ve tasarım sürecinde kolaylık sağlanabilir.

✍️ Örnek Modelleme



📘 Kazanım

- Çizimlerin düzenlenmesi ve tasarım sürecinde kolaylık sağlamak.
- Öteleyerek çoğaltma işlemini belirli bir mesafe ya da nesne üzerinden otomatik olarak hesaplanan mesafe kullanarak gerçekleştirebilmek.
- Öteleyerek çoğaltılan nesneler üzerinde düzenleme ve değişiklikler yapılabilmek.
- Çizimlerin düzenlenmesi ve görünümünün değiştirilmesinde kolaylık sağlamak.

✅ Dersin Hedefleri

- Öteleyerek Çoğaltma (Offset) Komutunu kullanarak uygulamalı olarak anlatılması beklenmektedir.

DAIRESEL ÇOĞALTMA (CIRCULAR PATTERN) VE DOĞRUSAL ÇOĞALTMA (RECTANGULAR PATTERN) KOMUTLARI

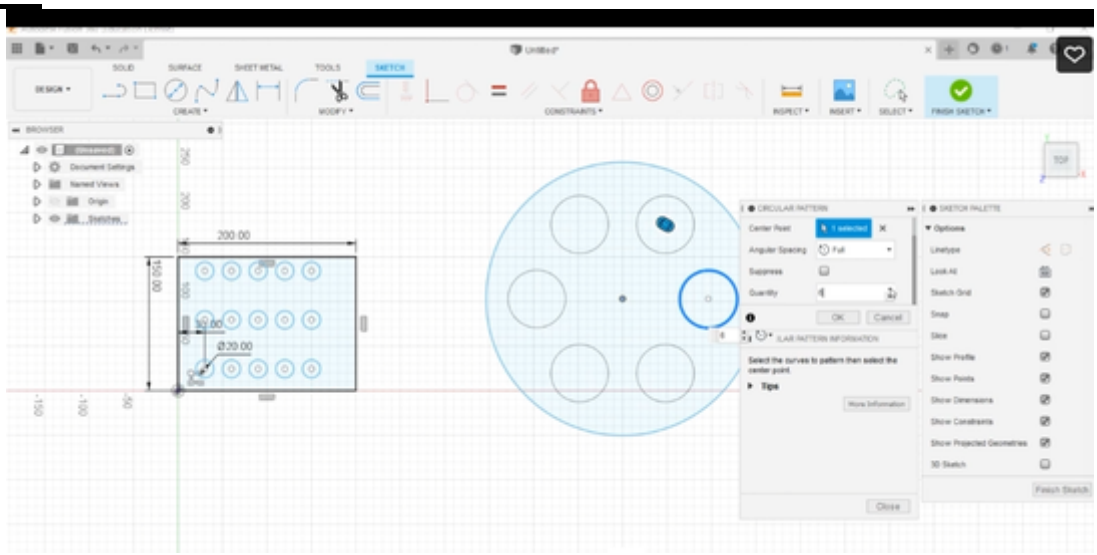
Müfredat Kodu / Ünite / Konu	İCT-203-2.7 /2 Boyutlu Tasarım ve Komutları / Dairesel Çoğaltma (Circular Pattern) ve Doğrusal Çoğaltma (Rectangular Pattern) Komutları
Hedeflenen Süre	70 Dakika

🚀 Dairesel Çoğaltma (Circular Pattern) ve Doğrusal Çoğaltma (Rectangular Pattern) Komutları

Fusion360 Dairesel Çoğaltma (Circular Pattern) komutu, bir çizimin dairesel bir şekilde çoğaltılmasını sağlayan bir araçtır. Bu komut ile belirli bir merkez noktası ve yarıçapı kullanarak belirli bir açı aralığında bir dizi kopya oluşturulabilir. Bu sayede çizimlerin düzenlenmesi ve tasarım sürecinde kolaylık sağlanabilir.

Fusion360 Doğrusal Çoğaltma (Rectangular Pattern) komutu ise bir çizimin doğrusal bir şekilde çoğaltılmasını sağlayan bir araçtır. Bu komut ile belirli bir merkez noktası ve boyutları kullanarak düzlem üzerinde belirli bir aralıkta bir dizi kopya oluşturulabilir. Bu sayede çizimlerin düzenlenmesi ve tasarım sürecinde kolaylık sağlanabilir.

✍️ Örnek Modelleme



📘 Kazanım

- Çizimlerin dairesel ya da doğrusal bir şekilde çoğaltılmasını kolayca gerçekleştirir.
- Dairesel çoğaltma komutu ile belirli bir açı aralığında, doğrusal çoğaltma komutu ile ise belirli bir düzlem aralığında çoğaltma işlemini gerçekleştirebilme imkanı sunar.
- Çoğaltma işlemini tek bir nesne ya da birden fazla nesne üzerinde aynı anda gerçekleştirebilme olanağı sağlar.
- Çoğaltma işlemini gerçekleştirdikten sonra yapılan düzenlemelerin çoğaltılmış kopyalarını da otomatik olarak güncelleyerek zaman kazandırır.
- Çizimlerin düzenlenmesi ve görünümünün değiştirilmesinde kolaylık sağlar.

✅ Dersin Hedefleri

- Dairesel Çoğaltma (Circular Pattern) ve Doğrusal Çoğaltma (Rectangular Pattern) Komutlarını uygulamalı olarak anlatılması beklenmektedir.

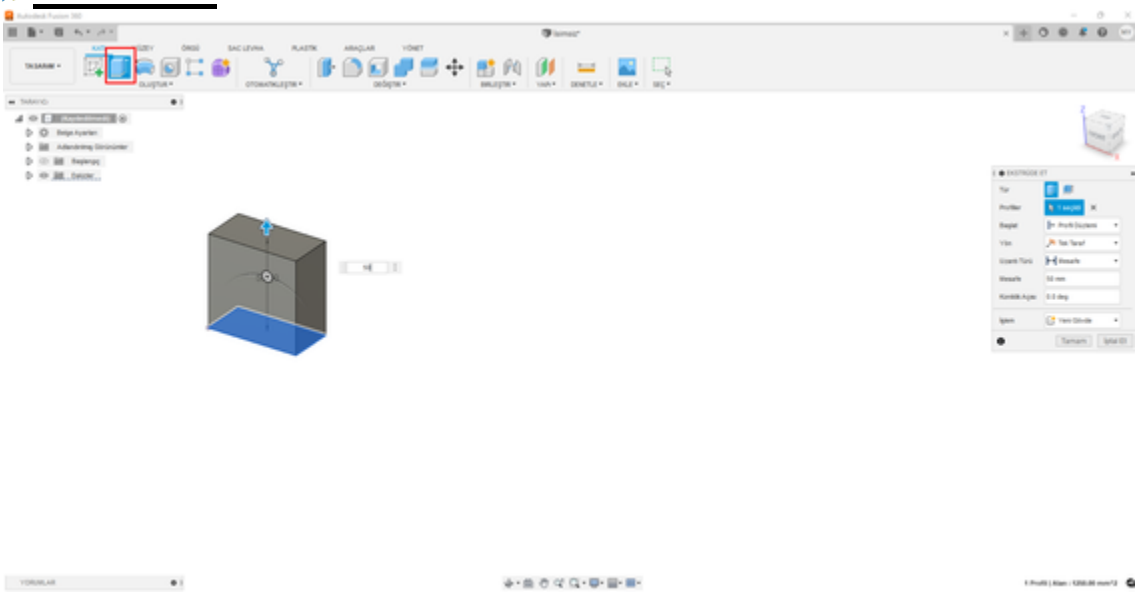
EXTRUDE - KATI MODEL OLUŞTURMA

Müfredat Kodu / Ünite / Konu	İCT-203-3.1 / Katı Model Oluşturma ve Komutları / Extrude - Katı Model Oluşturma
Hedeflenen Süre	110 Dakika

🚀 Extrude - Katı Model Oluşturma

Fusion 360 ile Extrude komutu kullanarak bir katı model oluşturabilirsiniz. Bu komut ile bir nesne üzerinde bulunan bir çizgi veya alanı belirlediğiniz yön ve miktarda açarak bir katı model oluşturabilirsiniz. Örneğin, bir düzlem üzerinde bulunan bir çizgi üzerinde Extrude komutunu kullanarak bu çizgiden bir küp oluşturabilirsiniz.

✍️ Örnek Modelleme



📘 Kazanım

- Extrude işlemi sayesinde basit geometrik şekillerden daha kompleks ve detaylı katı modellere oluşturulabilir.
- Extrude işlemini kullanarak parçaların eksenlerine göre simetrik şekiller oluşturabilir ve bu sayede daha estetik tasarımlar yapabiliriz.
- Extrude işlemini kullanarak parçaların kalınlık ve boyutlarını değiştirebilir ve bu sayede parçaların fonksiyonel özelliklerini değiştirebiliriz.
- Extrude işlemini kullanarak parçaların arkasındaki boşlukları doldurabilir ve bu sayede parçaların daha sağlam ve dayanıklı hale gelmesini sağlayabiliriz.
- Extrude işlemini kullanarak parçaların üstüne eklemeler yaparak daha karmaşık ve detaylı tasarımlar oluşturabiliriz.

✅ Dersin Hedefleri

- Extrude - Katı Model Oluşturma konusunun örneklerle açıklanması beklenmektedir.

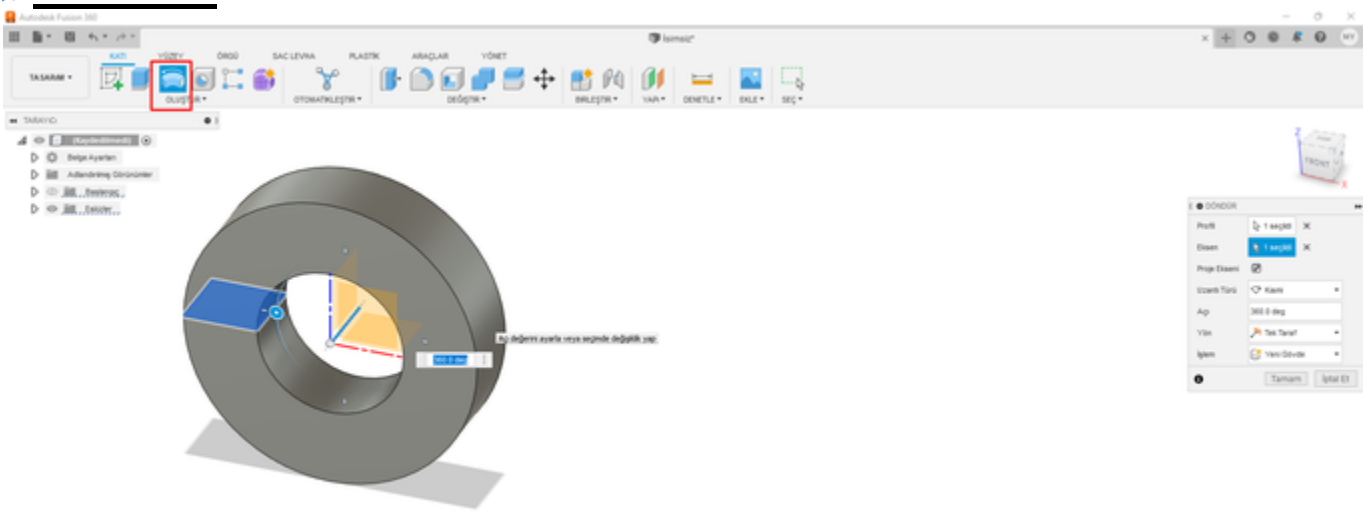
DÖNDÜREREK KATI OLUŞTURMA (REVOLVE) KOMUTU

Müfredat Kodu / Ünite / Konu	İCT-203-3.2 / Katı Model Oluşturma ve Komutları / Döndürerek Katı Oluşturma (Revolve) Komutu
Hedeflenen Süre	38 Dakika

🚀 Döndürerek Katı Oluşturma (Revolve) Komutu

Fusion 360 programında Revolve komutu, bir çizim üzerindeki bir noktayı eksen olarak belirlediğimiz bir dairesel yolda döndürme işlemini sağlar. Bu sayede, basit geometrik şekillerden daha karmaşık ve detaylı katı modeller oluşturabiliriz. Revolve komutunun kullanımı, Extrude komutuna benzer şekildedir. Öncelikle çizim yapılır ve bu çizim üzerinde Revolve komutu seçilir. Daha sonra eksen olarak belirlenen noktayı ve döndürme yönünü belirleyerek işlem tamamlanır. Revolve komutunun kullanımı sayesinde, tekerlekler, borular, dairesel parçalar gibi çok sayıda farklı tasarım oluşturulabilir.

✍️ Örnek Modelleme



📘 Kazanım

1. Revolve komutu sayesinde basit geometrik şekillerden daha karmaşık ve detaylı katı modeller oluşturulabilir.
2. Revolve komutu sayesinde parçaların dönme hareketlerine göre tasarım yapılabilir ve bu sayede fonksiyonel parçalar oluşturulabilir.
3. Revolve komutu sayesinde dairesel parçaların kalınlık ve boyutları değiştirilebilir ve bu sayede parçaların estetik görünümü değiştirilebilir.
4. Revolve komutu sayesinde parçaların iç boşlukları doldurulabilir ve bu sayede parçaların daha sağlam ve dayanıklı hale gelmesini sağlayabiliriz.
5. Revolve komutu sayesinde parçaların üstüne eklemeler yaparak daha karmaşık ve detaylı tasarımlar oluşturulabilir.

✅ Dersin Hedefleri

- Revolve komutu ile katı oluşturma hakkında bilgi verilerek örneklerle açıklama beklenmektedir.

YOL KULLANARAK KATI OLUŞTURMA (SWEEP) KOMUTU

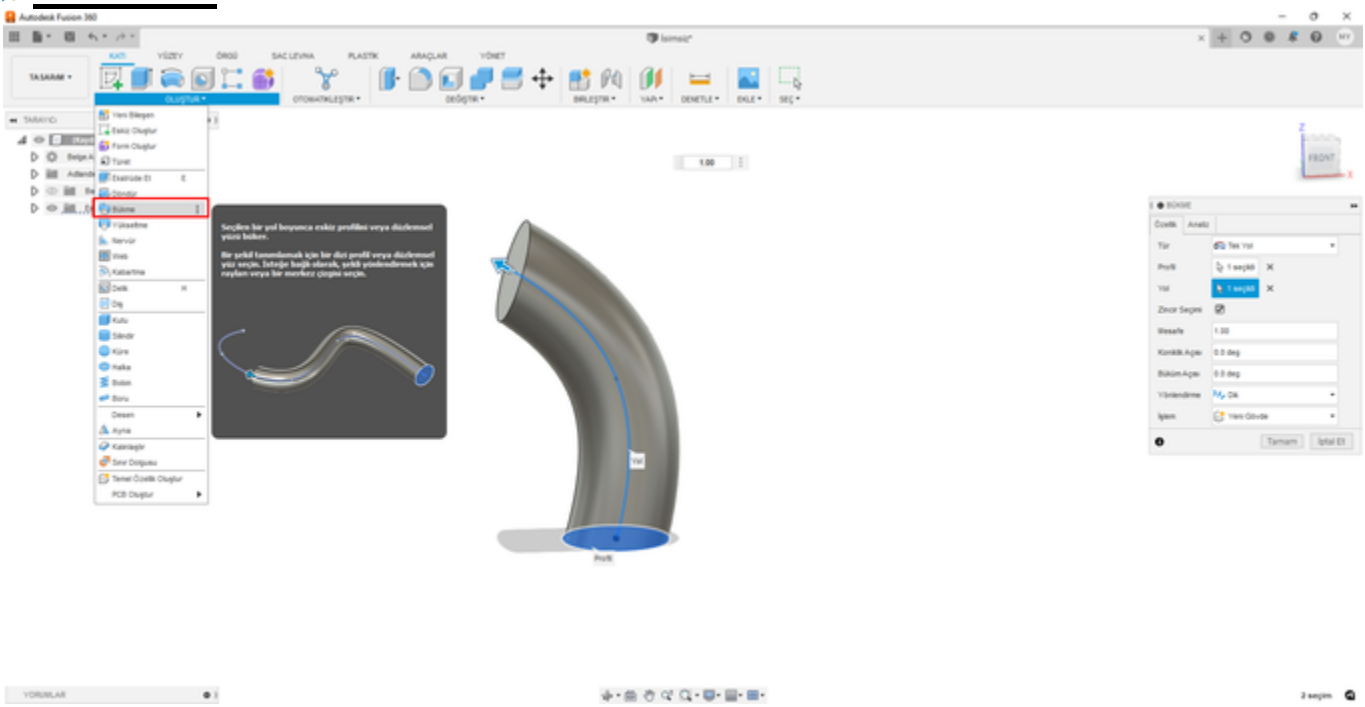
Müfredat Kodu / Ünite / Konu	İCT-203-3.3 / Katı Model Oluşturma ve Komutları / Yol Kullanarak Katı Oluşturma (Sweep) Komutu
Hedeflenen Süre	43 Dakika

🚀 Yol Kullanarak Katı Oluşturma (Sweep) Komutu

Fusion 360'ta Sweep komutu, bir yol boyunca 2D bir profili süpürerek 3D bir katı oluşturmak için kullanılır. Sweep komutunu kullanmak için aşağıdaki adımları izleyin:

Süpürmek istediğiniz bir 2B profil oluşturun. Bu, oluşturmak istediğiniz katının daire, dikdörtgen veya daha karmaşık bir şekil gibi bir kesitinin taslağı olabilir. Profilin izleyeceği bir yol oluşturun. Bu, çizgi, daire veya spline gibi bir eğrinin taslağı olabilir. Yol, 2 boyutlu bir taslak yerine 3 boyutlu bir eğri olmalıdır. 3D Modelleme araç çubuğundan Süpürme komutunu seçin veya 3D Modelleme menüsünden seçin. Sweep iletişim kutusunda, profili ve yolu seçin ve ardından kullanmak istediğiniz seçenekleri belirleyin. Örneğin, süpürmenin yönünü, profilin hizalamasını ve oluşturmak istediğiniz katının türünü belirleyebilirsiniz. Süpürülen katıyı oluşturmak için Tamam'a tıklayın. Katı, Fusion 360 tasarım çalışma alanında görünecektir ve gerektiğinde düzeltmek için diğer modelleme araçlarını kullanabilirsiniz.

✍️ Örnek Modelleme



📘 Kazanım

- Sweep komutu sayesinde basit geometrik şekillerden daha karmaşık ve detaylı katı modeller oluşturulabilir.
- Sweep komutu sayesinde parçaların hareket eksenlerine göre tasarım yapılabilir ve bu sayede fonksiyonel parçalar oluşturulabilir.
- Sweep komutu sayesinde parçaların kalınlık ve boyutları değiştirilebilir ve bu sayede parçaların estetik görünümü değiştirilebilir.
- Sweep komutu sayesinde parçaların iç boşlukları doldurulabilir ve bu sayede parçaların daha sağlam ve dayanıklı hale gelmesini sağlayabiliriz.
- Sweep komutu sayesinde parçaların üstüne eklemeler yaparak daha karmaşık ve detaylı tasarımlar oluşturulabilir.

✅ **Dersin Hedefleri**

- Sweep komutunun örneklerle uygulamalı olarak anlatılması beklenmektedir.

NOTLAR

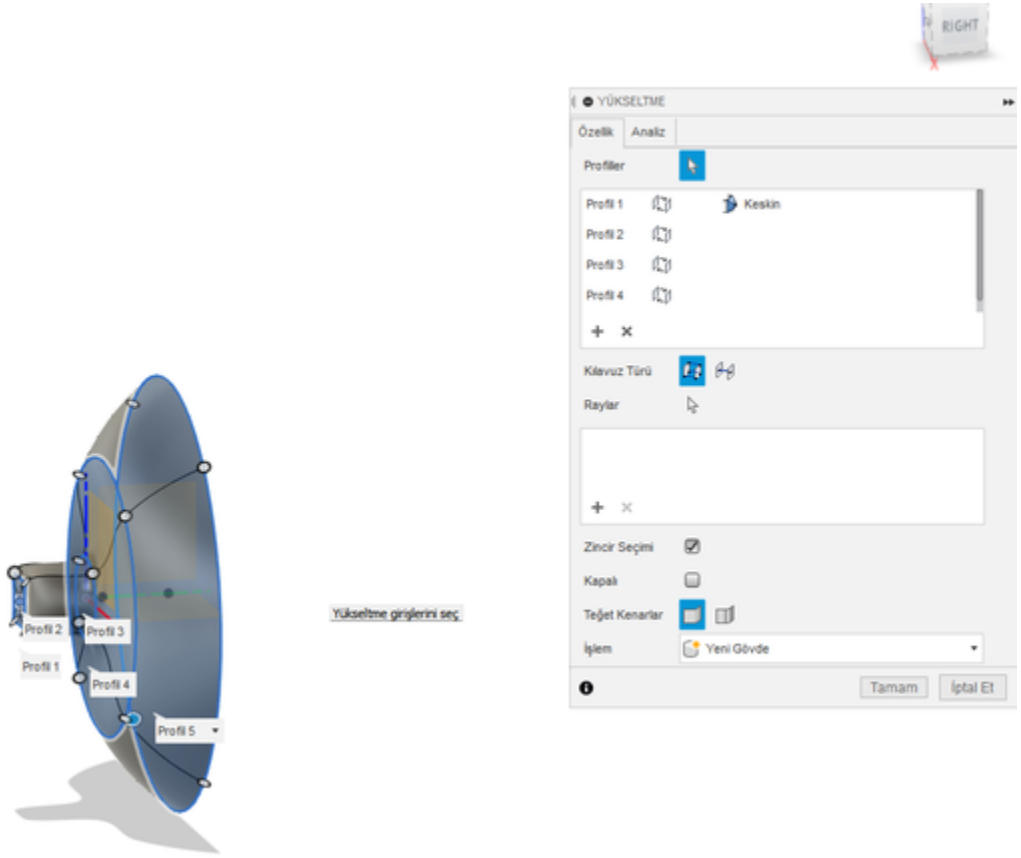
PROFILLER VE YÜZEYLER ARASINDA KATI OLUŞTURMA (LOFT) KOMUTU

Müfredat Kodu / Ünite / Konu	İCT-203-3.4 / Katı Model Oluşturma ve Komutları / Profiller ve Yüzeyler Arasında Katı Oluşturma (Loft) Komutu
Hedeflenen Süre	43 Dakika

🚀 Fusion 360 : Profiller ve Yüzeyler Arasında Katı Oluşturma (Loft) Komutu

Fusion 360 programında Loft komutu, bir çizim üzerinde iki veya daha fazla noktayı birleştirme işlemini sağlar. Bu sayede, basit geometrik şekillerden daha karmaşık ve detaylı katı modeller oluşturabiliriz. Loft komutunun kullanımı, Extrude, Revolve ve Sweep komutlarına benzer şekildedir. Öncelikle çizim yapılır ve bu çizim üzerinde Loft komutu seçilir. Daha sonra birleştirilecek olan noktalar ve birleştirme yönü belirlenir ve işlem tamamlanır. Loft komutunun kullanımı sayesinde, parçaların yüzey şekilleri değiştirilerek daha karmaşık tasarımlar oluşturulabilir.

✍️ Örnek Modelleme



Kazanım

1. Loft komutu sayesinde basit geometrik şekillerden daha karmaşık ve detaylı katı modeller oluşturulabilir.
2. Loft komutu sayesinde parçaların yüzey şekilleri değiştirilerek daha estetik tasarımlar yapılabilir.
3. Loft komutu sayesinde parçaların kalınlık ve boyutları değiştirilerek fonksiyonel özellikleri değiştirilebilir.
4. Loft komutu sayesinde parçaların iç boşlukları doldurularak daha sağlam ve dayanıklı hale getirilebilir.
5. Loft komutu sayesinde parçaların üstüne eklemeler yaparak daha karmaşık ve detaylı tasarımlar oluşturulabilir.

✅ Dersin Hedefleri

- Loft komutu hakkında uygulamalı bilgi verilerek örneklerle açıklanması beklenmektedir.

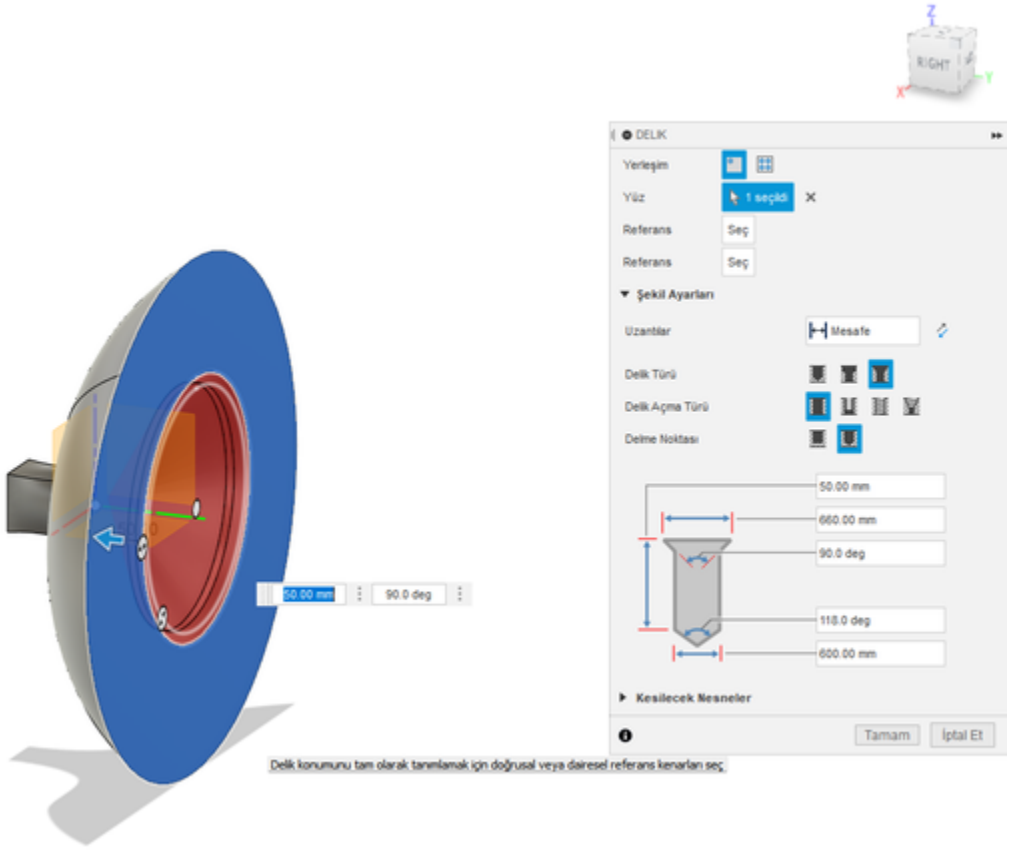
DELIK SIHIRBAZI (HOLE) KOMUTU

Müfredat Kodu / Ünite / Konu	İCT-203-3.5 / Katı Model Oluşturma ve Komutları / Delik Sihirbazı (Hole) Komutu
Hedeflenen Süre	48 Dakika

🚀 Delik Sihirbazı (Hole) Komutu

Fusion 360 programında Hole komutu, bir parçanın içine delik açma işlemini sağlar. Bu sayede, parçaların fonksiyonel özelliklerini değiştirerek daha işlevsel tasarımlar yapabiliriz. Hole komutunun kullanımı, Extrude, Revolve, Sweep ve Loft komutlarına benzer şekildedir. Öncelikle parça üzerinde bir çizim yapılır ve bu çizim üzerinde Hole komutu seçilir. Daha sonra deliğin boyutları ve konumu belirlenir ve işlem tamamlanır. Hole komutunun kullanımı sayesinde, vidalar, borular gibi farklı parçalara delik açarak daha fonksiyonel tasarımlar oluşturulabilir.

✍️ Örnek Modelleme



📘 Kazanım

1. Hole komutu sayesinde parçalara delik açarak fonksiyonel özelliklerini değiştirebilir ve bu sayede daha işlevsel tasarımlar yapabiliriz.
2. Hole komutu sayesinde deliklerin boyutları ve konumları değiştirilerek farklı tasarımlar oluşturulabilir.
3. Hole komutu sayesinde parçaların iç boşlukları doldurularak daha sağlam ve dayanıklı hale getirilebilir.
4. Hole komutu sayesinde parçaların üstüne eklemeler yaparak daha karmaşık ve detaylı tasarımlar oluşturulabilir.
5. Hole komutu sayesinde parçaların estetik görünümü değiştirilerek daha estetik tasarımlar yapılabilir.

✅ Dersin Hedefleri

- Hole komutu hakkında uygulamalı bilgi verilerek örneklerle açıklanması beklenmektedir.

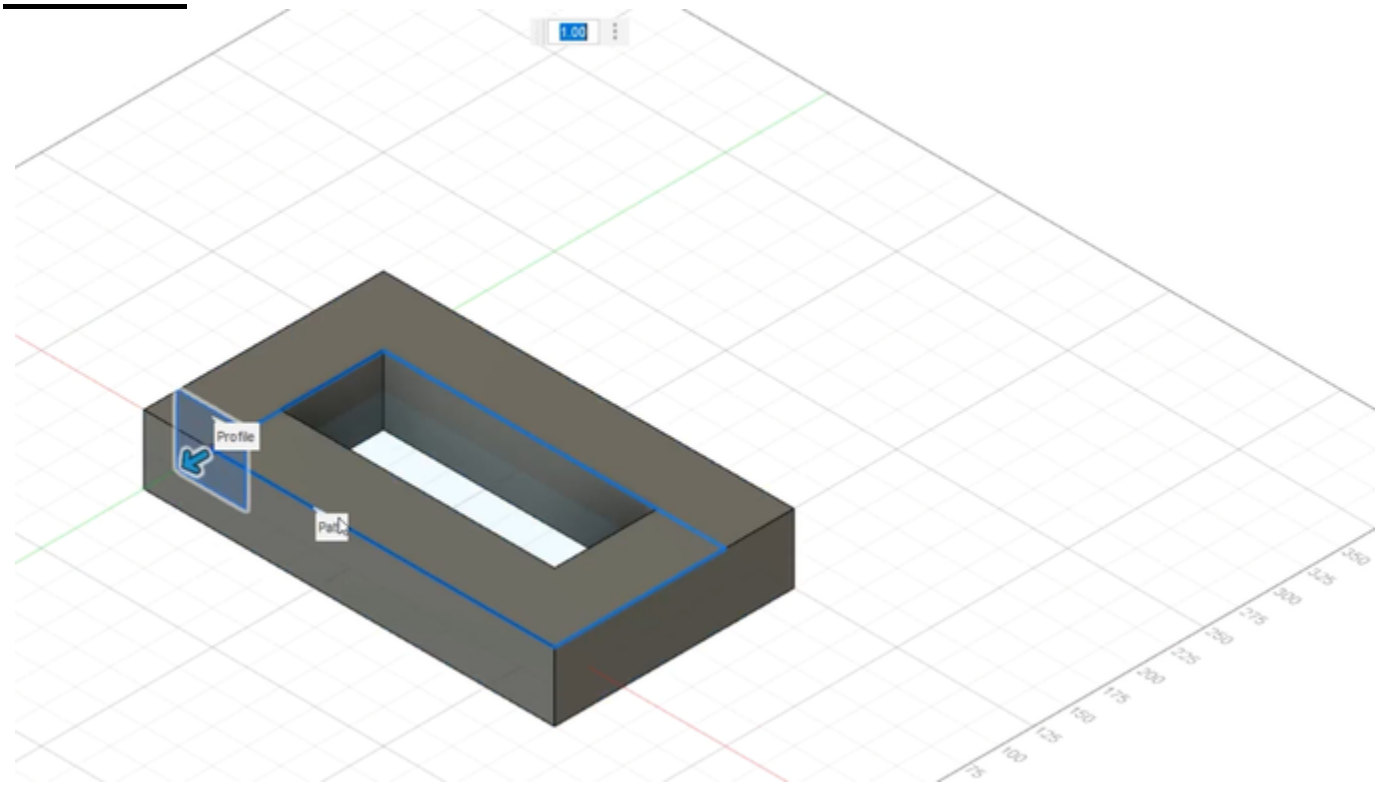
WEB KOMUTU

Müfredat Kodu / Ünite / Konu	İCT-203-3.6 / Katı Model Oluşturma ve Komutları / Web Komutu
Hedeflenen Süre	48 Dakika

🚀 Fusion 360 Web Komutu

Fusion 360 programında Web komutu, bir parçanın üzerine çizim yapma işlemini sağlar. Bu sayede, parçaların fonksiyonel özelliklerini değiştirerek daha işlevsel tasarımlar yapabiliriz. Web komutunun kullanımı, Extrude, Revolve, Sweep, Loft ve Hole komutlarına benzer şekildedir. Öncelikle parça üzerinde bir çizim yapılır ve bu çizim üzerinde Web komutu seçilir. Daha sonra çizimin kalınlığı ve konumu belirlenir ve işlem tamamlanır. Web komutunun kullanımı sayesinde, parçalara eklemeler yaparak daha karmaşık ve detaylı tasarımlar oluşturulabilir.

✍️ Örnek Modelleme



📘 Kazanım

1. Web komutu sayesinde parçalara eklemeler yaparak fonksiyonel özelliklerini değiştirebilir ve bu sayede daha işlevsel tasarımlar yapabiliriz.
2. Web komutu sayesinde eklemelerin kalınlık ve konumları değiştirilerek farklı tasarımlar oluşturulabilir.
3. Web komutu sayesinde parçaların iç boşlukları doldurularak daha sağlam ve dayanıklı hale getirilebilir.
4. Web komutu sayesinde parçaların üstüne eklemeler yaparak daha karmaşık ve detaylı tasarımlar oluşturulabilir.
5. Web komutu sayesinde parçaların estetik görünümü değiştirilerek daha estetik tasarımlar yapılabilir.

✅ Dersin Hedefleri

- Web Komutunun uygulamalı örneklerle açıklanması beklenmektedir.

UYGULAMA

✍️ **Örnek ModellemeMüfredat Kodu / Ünite / Konu**	İCT-203-3.7 / Katı Model Oluşturma ve Komutları / Uygulama
Hedeflenen Süre	300 Dakika

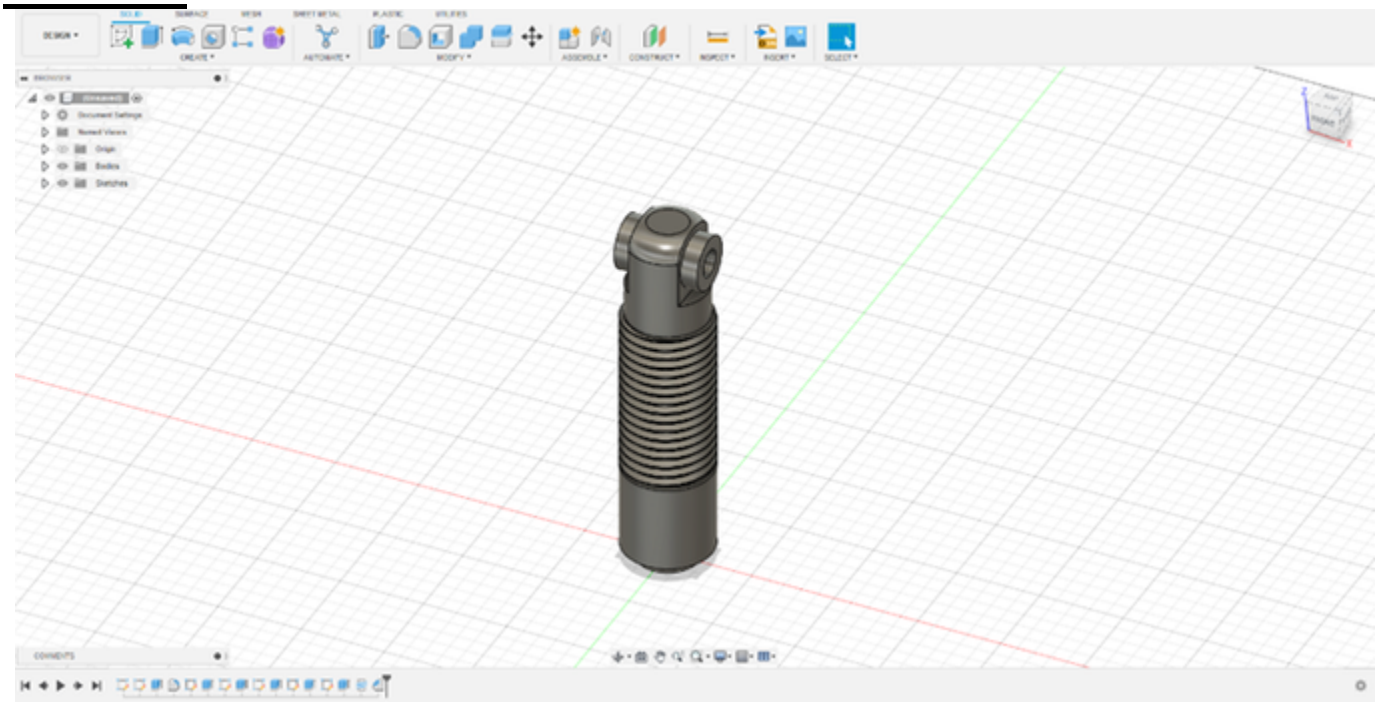
🚀 Fusion 360 Uygulaması

Bu bölümde eğitmenin öğrencilere öğrendiklerini pekiştirme adına tasarım yaptırması amaçlanmaktadır.

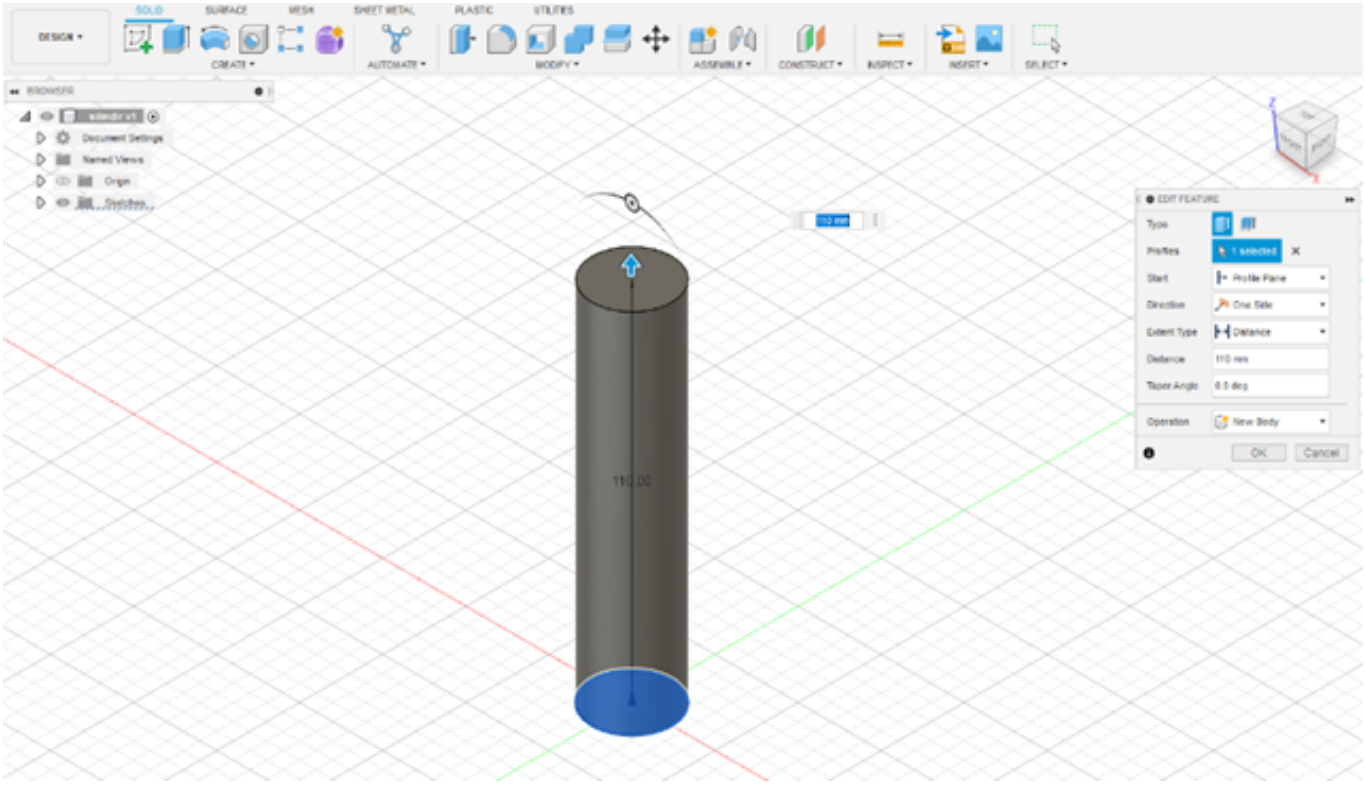
Tasarımlar;

- Endüstriyel makinelerin parçaları
- Dişli çark
- Somun ve Civa
- Yay
- Sandalye
- Masa gibi farklı alanlarda eğitmenin yönlendirmesine göre olabilir.

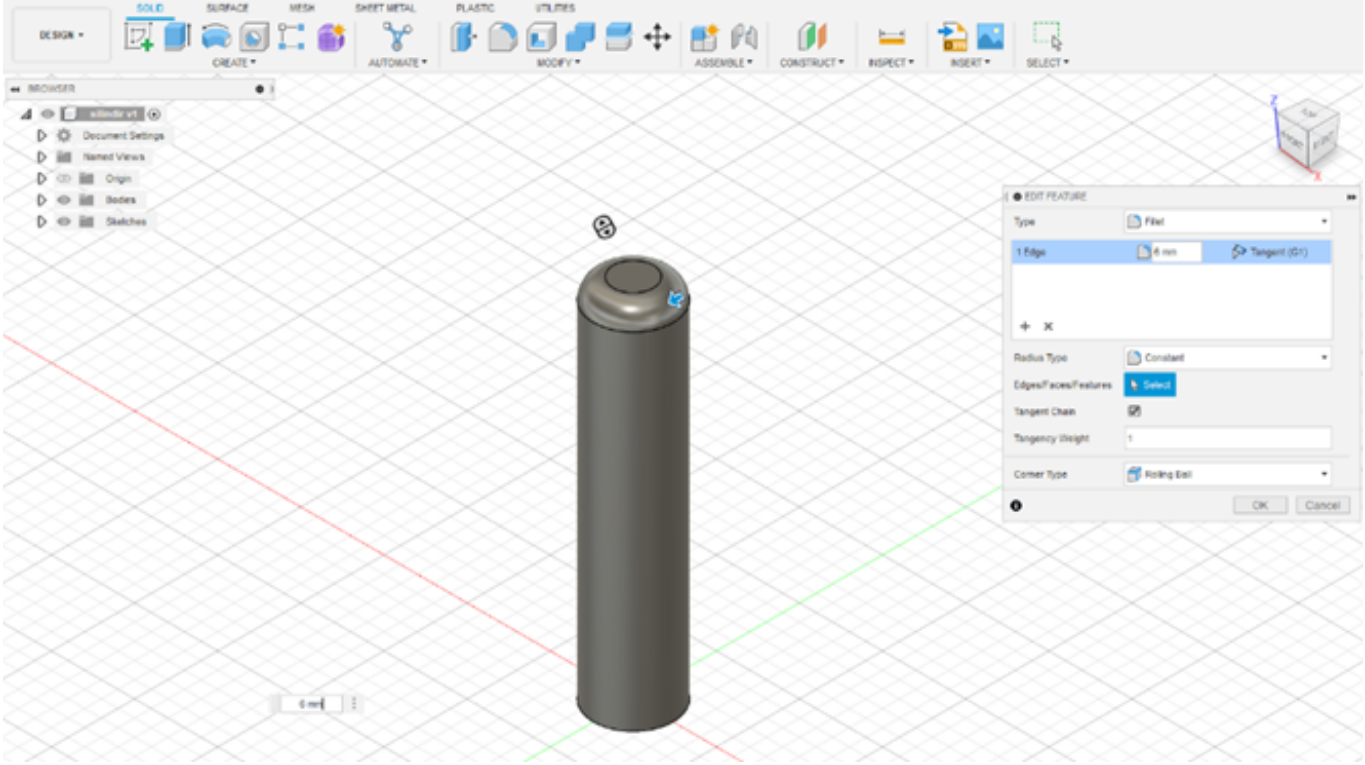
✍️ Örnek Modelleme



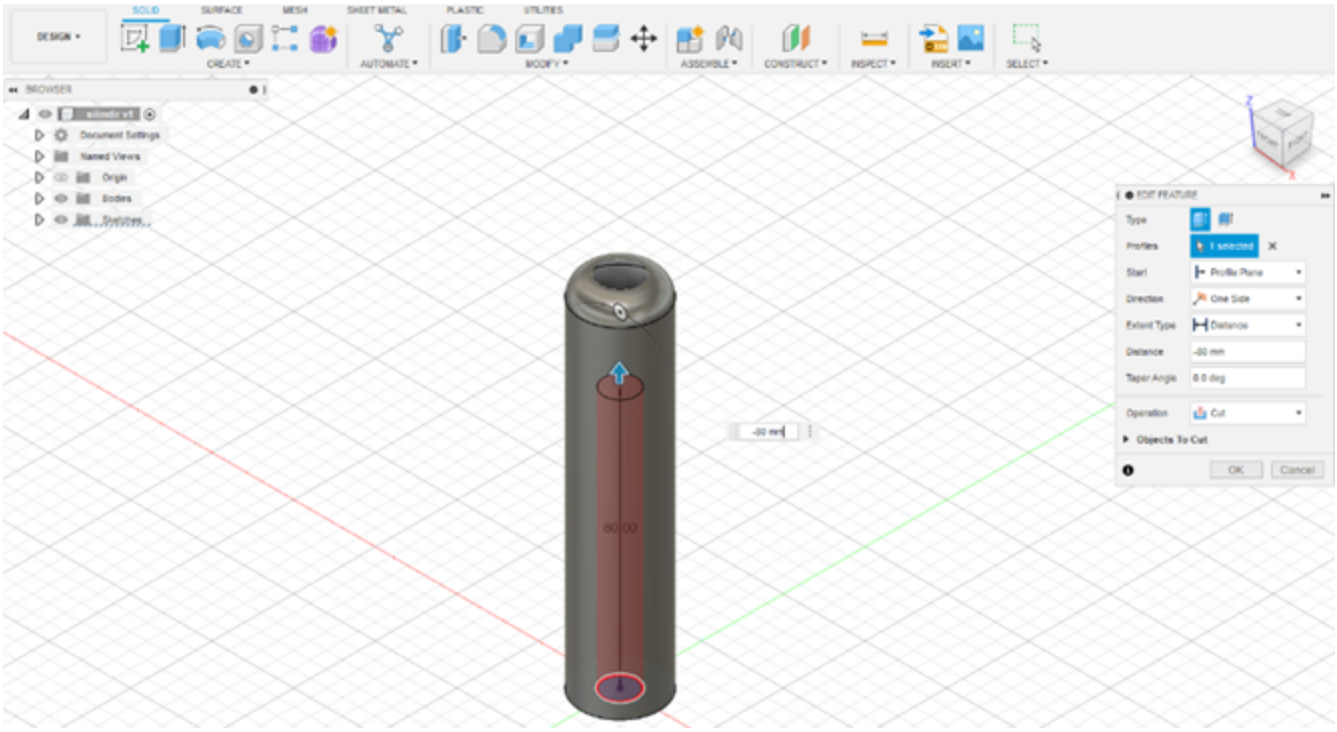
- **Öncelikle üstten görünüm şekilde düzlem seçilerek çizim açılır. Ve 24mm çapında daire çizilip 110 mm katılaştırılır.**



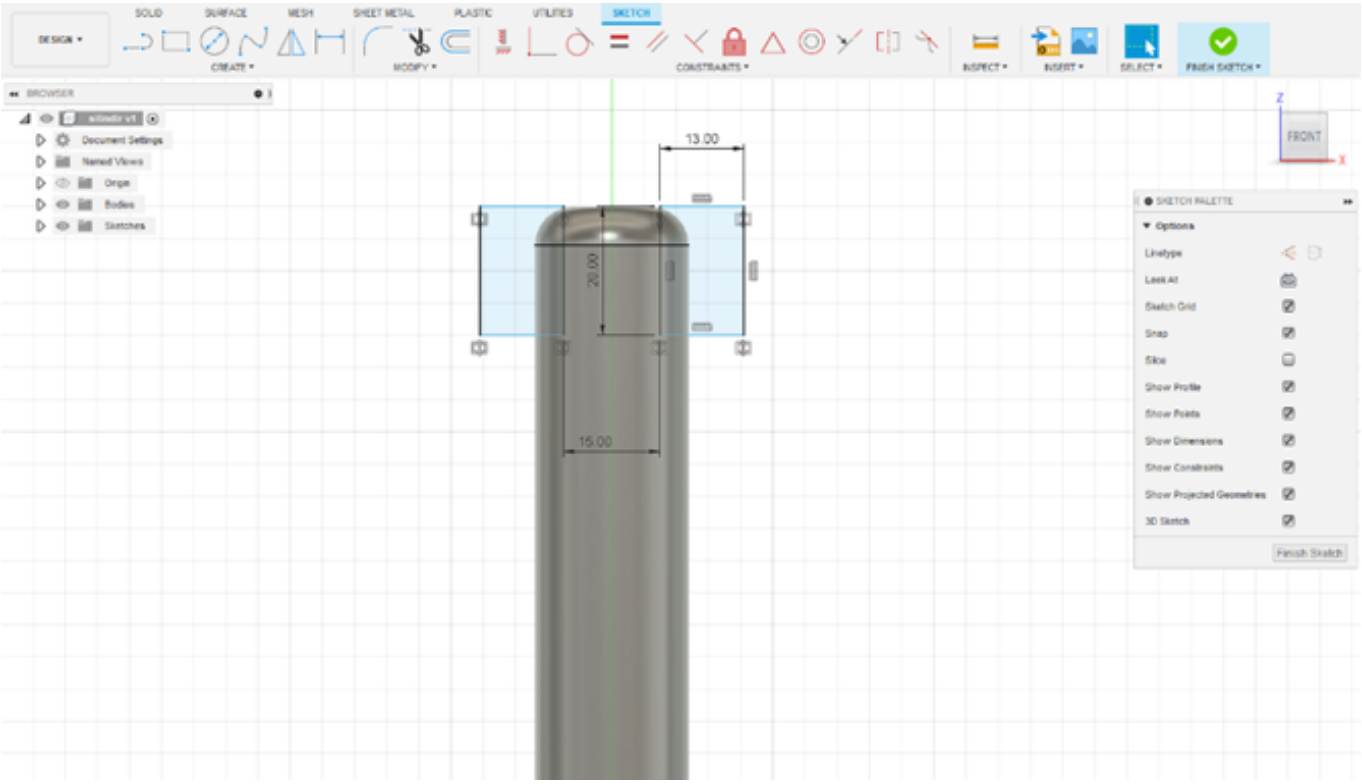
- Katı modelin üst kısmına fillet komutu ile radyus verilir.



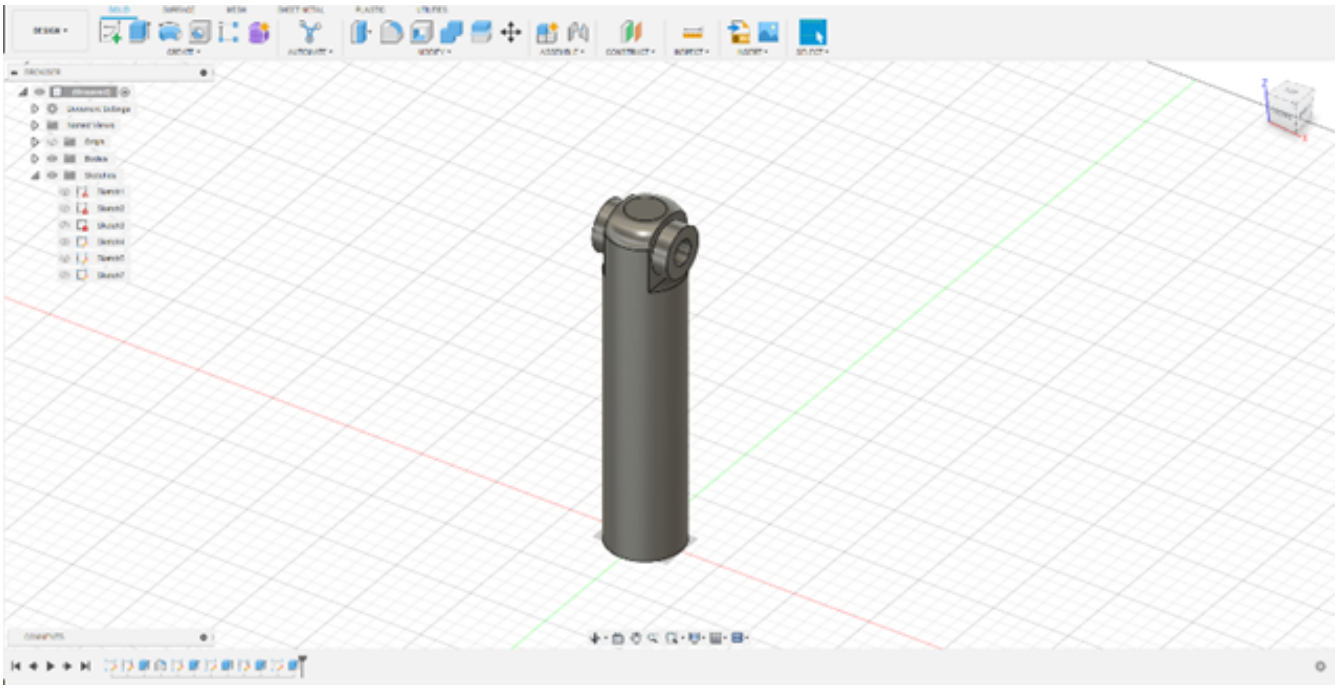
- Ardından 10mm çapında 80mm uzunluğunda bir delik açılır.



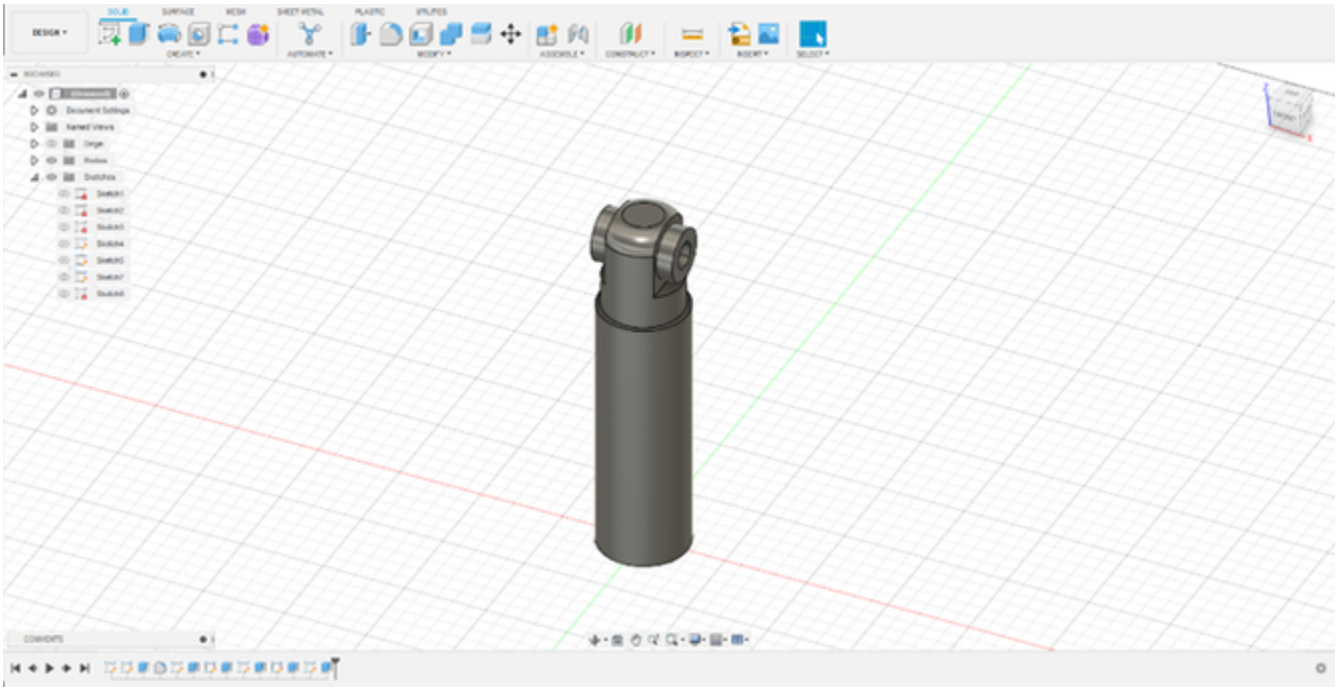
- Radyus verilen bölgenin herhangi bir konumuna sketch açılır ve 2 adet dikdörtgen çizilirek mil girişi için kenarlardan extrude cut komutu ile kesim yapılır.



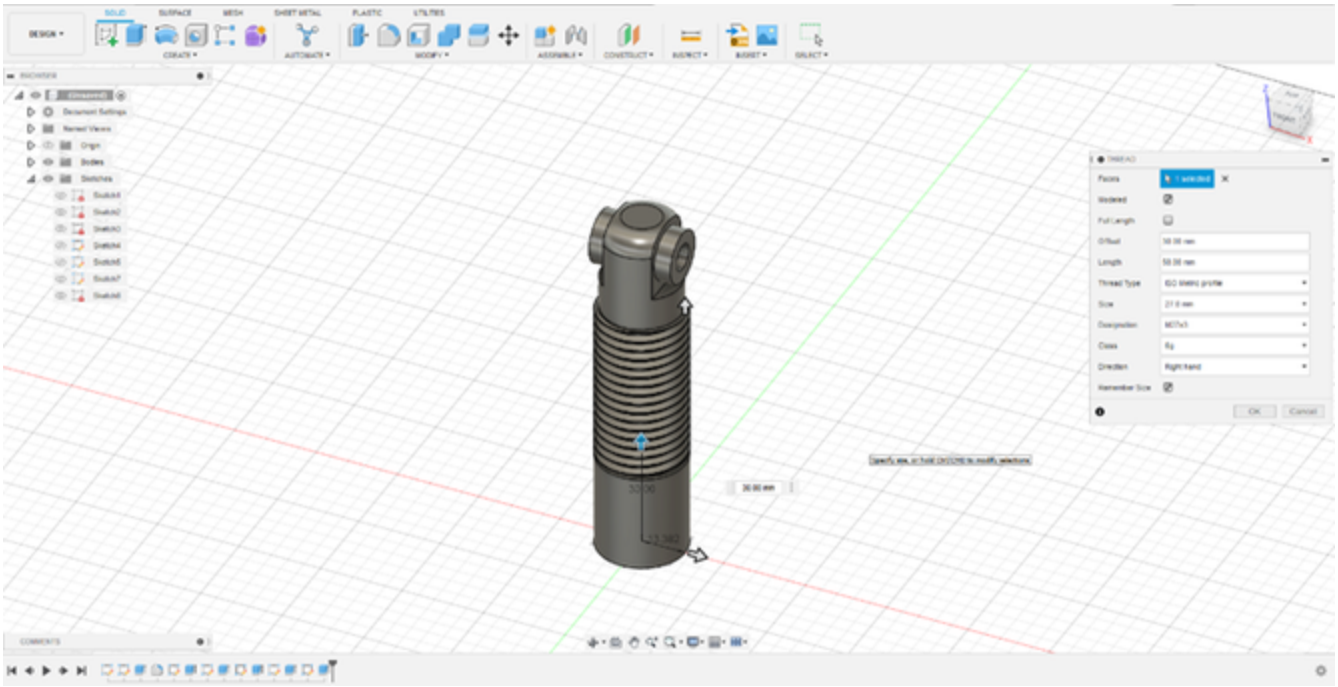
- Kesilen bölgeye yeni bir sketch açılarak 2 adet daire çizilir ve extrude komutu ile katılaştırılır.



- Parçaya diş eklemek için parça kalınlaştırılır ve diş açılır. Mil çapını genişletmek için sketch açılıp 27mm çapında daire çizilir ve 80mm uzatılır.



- Diş açmak için create bölümünden thread komutu kullanılır.



Kazanım

- Tasarım yeteneklerinin arttırılması.

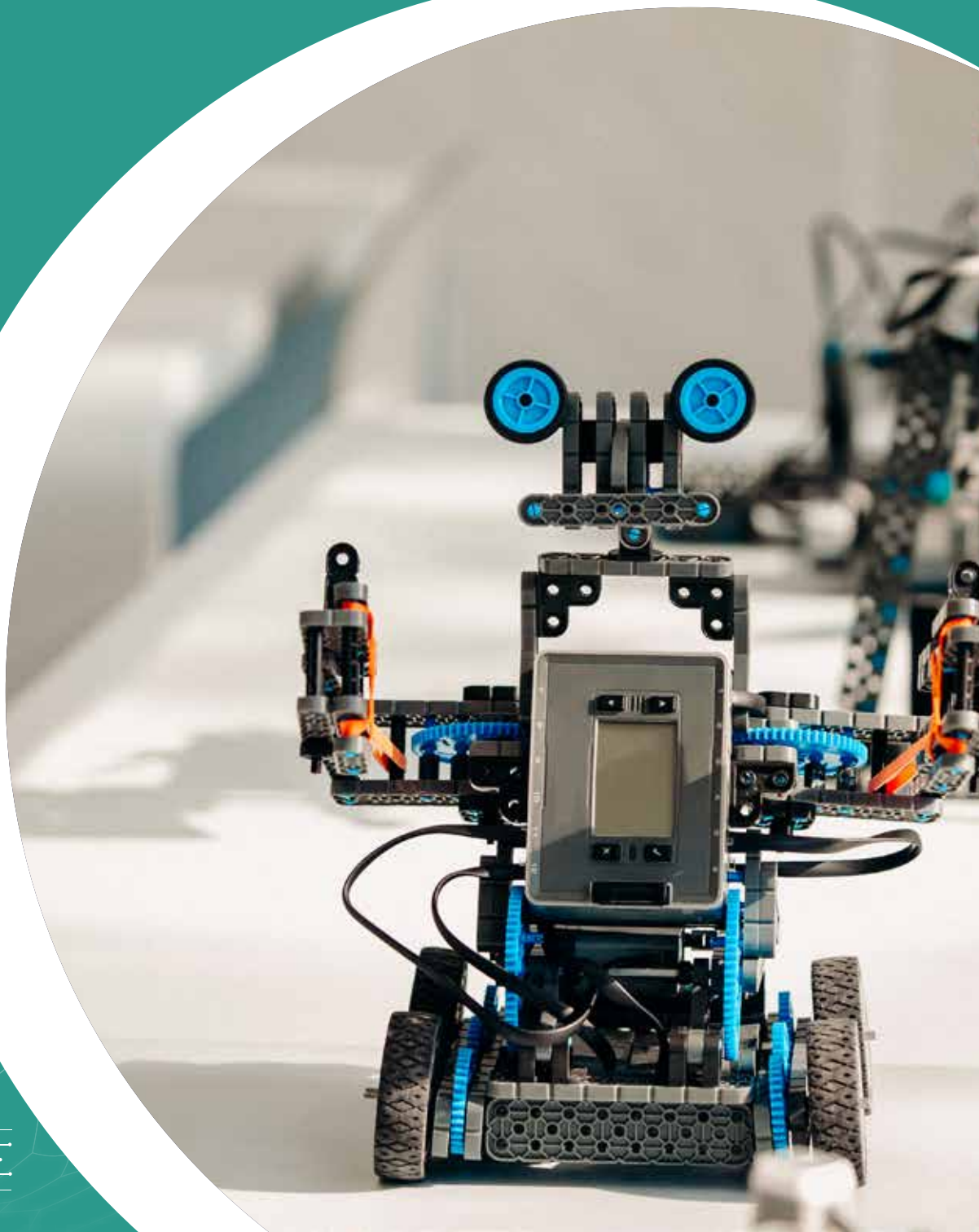
Dersin Hedefleri

- Örnek uygulamaların yapılması beklenmektedir.

İcathane Eğitim Müfredatı

ROBOTİK

Akım | Direnç | Bobin | Kondansatör | Devre | Frekans | Arduino



TÜGVA İCATHANE

MAKER EĞITIM MÜFREDATI

TEMEL SEVİYE MAKER EĞİTİM MÜFREDATI														
EĞİTİM İÇERİĞİ / HAFTA	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.
Temel Elektronik Kavramlar ve Tanımlar (İCT-400-1)														
Temel Elektroniğe Giriş														
Elektrik Akımı - Gerilimi Bilgileri														
Temel Elektronik : AC/DC Kavramları ve Özellikleri														
Elektronik Değerleri Ölçme Aletleri														
Frekans, Periyot Kavramları														
Scratch (İCT-103)														
Scratch : Algoritma														
Scratch: Değişkenler														
Scratcah: if-else														
Scratch: Döngüler														
Scratch: Fonksiyonlar														
Scrarch: Sınıflar ve Nesneler														
Scratch : Uygulamalar														
Selamlama Uygulaması														
Gol Atma Uygulaması														
Duvarlara Çarpan Ok Uygulaması														
Dikdörtgen Çizme Uygulaması														
Üçgen Çizme Uygulaması														
Çokgen Çizim Uygulaması														
İki Kişilik Oyun Uygulaması														
Yılan Oyunu Yapımı														
Home Sweet Home Uygulaması														
Haritalarla Denizli Uygulaması														
Makeblock mBot (İCT-105)														
Makeblock mBot : mBot Robot Kiti														
Makeblock mBot : mBot Montajı														
Makeblock mBot : mBlock Programı														
Makeblock mBot : mBlock Kod Blokları														
Makeblock mBot : mBot Simülatör Kod Blokları														
Makeblock mBot: mBot ile Örnek Programlar														

TEMEL ELEKTRONIK KAVRAMLAR VE TANIMLAR

Müfredat Kodu	İCT-400-1
Ders	Temel Elektronik Kavramlar ve Tanımlar
Ders İçeriği	• Temel Elektroniğe Giriş • Elektrik Akımı - Gerilimi Bilgileri • Temel Elektronik: AC/DC Kavramları ve Özellikleri • Elektronik Değerleri Ölçme Aletleri • Frekans, Periyot Kavramları
Seviye	İCT-S1: BAŞLANGIÇ
Ön Koşullar	Elektronik alanında öğrenme isteğinde olmak
Amaç	Elektronik devrelerin kurulabilmesi için kullanılacak kompanentlerin özelliklerinin ve kullanım koşullarının öğretilmesi amaçlanmaktadır.
Hedef	Elektronik bilgi seviyesinin arttırılması hedeflenmektedir.
Gereksinimler	Atölye Ortamı ve gerekli elektronik kompanentlerin olması
Kazanım	• Elektronik kavramının öğrenilmesi • Elektronik kompanentlerin kullanımı hakkında bilgi sahibi olmak • Akım-Gerilim ve Kanunlar hakkında bilgi sahibi olmak
Referanslar & Dokümantasyon	• https://www.elektrikrehberiniz.com/elektronik/elektronik-devre-elemanlari-10485/#:~:text=Bir%20elektrik%20devresinin%20olu%C5%9Fmas%C4%B1n%C4%B1%20sa%C4%9Flayan,ve%20anahtar%20birer%20devre%20eleman%C4%B1d%C4%B1r. • https://www.milliyet.com.tr/egitim/elektronik-esyalar-nelerdir-adan-zye-en-cok-kullanilan-elektronik-esyalar-6471891 • https://tr.wikipedia.org/wiki/Kirchhoff_kanunlar%C4%B1#:~:text=Kirchhoff%27un%20gerilim%20yasas%C4%B1,toplam%C4%B1%2C%20sa%C4%9Flanan%20gerilimlerin%20toplam%C4%B1na%20e%C5%9Fittir. • http://kitap.eba.gov.tr/panel/dosyalar/upload/1456/0/U_0_21_08_2020_21_08_13_91.pdf • https://320volt.com/temel-elektrik-elektronik-dersleri-basit-devre-ornekleri/ • http://elektronikderslerim.blogspot.com/2017/10/olcu-aleti-multimetre-nedir-ne-ise.html
Durum	BEKLİYOR / UYGULANIYOR / TAMAMLANDI / DEĞERLENDİRİLDİ
Hedeflenen Süre	180 Dakika
Eğitim Günleri	Başlama: .../.../.... Bitiş: .../.../....

TEMEL ELEKTRONİĞE GIRIŞ

Müfredat Kodu / Ünite / Konu	ICT-400-1.1 / Temel Elektronik Kavramlar ve Tanımlar / Temel Elektroniğe Giriş
Hedeflenen Süre	45 Dakika

🚀 Temel Elektroniğe Giriş

Bu konuda eğitmen Elektronik nedir?, Elektronik komponentler nelerdir? ve günlük hayatta kullanılan elektronik ürünler hakkında bilgi vermelidir. Bilgilendirme sunum ile desteklenmelidir. Öğrencilerin elektronik kavramını ve hangi malzemelerin elektronik sınıfında olduğunu öğrenmelerini amaçlanmaktadır.

Elektrik ve elektronik kavramları, fizik ve elektrik mühendisliği alanlarında önemli kavramlardır. Elektrik kavramı, bir elektriksel yükün hareket ettiği bir ortamda meydana gelen enerjiyi ifade eder. Örneğin, bir elektrik prizi elektrik enerjisinin dağıtımı için kullanılan bir alet olup elektrik kavramı ile ilgilidir. Elektrik kavramının temel bileşenleri, elektrik akımı, gerilim ve dirençtir. Bu bileşenler arasındaki ilişki ise Ohm Kanunu ile açıklanır.

Elektronik kavramı ise, elektriksel yüklerin hareketini kontrol eden cihazları, sistemleri ve ilgili teknolojileri ifade eder. Örneğin, bir bilgisayar elektriksel yüklerin hareketini kontrol eden bir sistem olup elektronik kavramı ile ilgilidir. Elektronik kavramının temel bileşenleri ise, elektronik devreleri, transistörler ve diyotlar gibi elemanlardır. Bu elemanlar arasındaki ilişkiyi anlamak için Kirchhoff Kanunları, Faraday Kanunu ve Lenz Kanunu gibi kanunlar kullanılır.

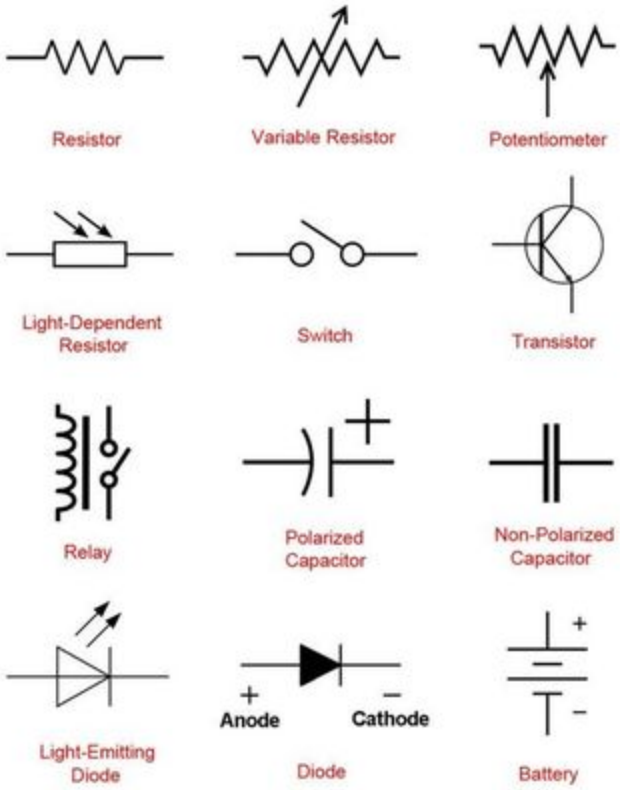
🔌 Elektronik Komponentler Nelerdir?

Bir elektrik devresinin oluşmasını sağlayan parçaların genel adıdır. **Pil, kablo, ampul, ve anahtar** birer devre elemanıdır.

Elektronik komponentler, elektronik sistemlerin çalışmasını sağlayan parçalardır. Bunlar arasında anahtarlar, dirençler, kondansatörler, transformatörler, tranzistörler, düğmeler ve diyotlar gibi birçok farklı komponent bulunmaktadır. Elektronik komponentler birbirleriyle birlikte kullanılarak, elektronik devreler oluşturulur ve bu devreler sayesinde elektronik cihazlar çalışır.

Elektronik komponentler, elektronik devrelerin çalışmasını sağlayan parçalardır. Bunlar şunlardır:

1. **Anahtar:** Elektrik akımını açıp kapatmaya yarayan parçadır.
2. **Direnç:** Elektrik akımını azaltmaya yarayan parçadır.
3. **Kondansatör:** Elektriksel enerjiyi depolamaya yarayan parçadır.
4. **Bobin:** Elektrik akımını artırmaya yarayan parçadır.
5. **Transistör:** Elektriksel sinyallerin birbirine bağlanmasını sağlayan parçadır.
6. **Dijital Çip:** Elektronik devrelerin çalışmasını koordine eden parçadır.
7. **LED:** Işık üreten parçadır.
8. **Kızılötesi Sensör:** Kızılötesi ışınları algılayarak sinyal üreten parçadır.
9. **Akıem Regülatörü:** Elektrik akımının düzenlenmesine yarayan parçadır.
10. **Pil:** Elektriksel enerji depolamaya yarayan parçadır.



Kazanım

- Temel elektronik dersi, elektronik ve elektrik kavramlarının temellerini öğrenilmesi.
- Temel elektronik dersi sayesinde, öğrenciler elektrik akımını, gerilimi ve direnci anlamaya ve bu kavramları kullanarak basit elektronik devreleri tasarlamaya başlayabilirler. Bu ders ayrıca, elektronik bileşenlerin nasıl çalıştığını ve nasıl bir araya getirilerek daha karmaşık devreler oluşturulabileceğini de öğretir.

Dersin Hedefleri

1	Başlık	Hedef Kitle	Detaylar
2	☑ Temel Elektroniğe Giriş	☑ Eğitmen ☐ Öğrenci	Temel elektronik dersine giriş ve kavramların öğrencilere tanıtılması.
3	☑ Temel Elektroniğe Giriş	☐ Eğitmen ☑ Öğrenci	Temel elektronik kavramların pekiştirilmesi hedeflenmektedir.

ELEKTRİK AKIMI - GERİLİMİ BİLGİLERİ

Müfredat Kodu / Ünite / Konu	ICT-400-1.2 / Temel Elektronik Kavramlar ve Tanımlar / Elektrik Akımı - Gerilimi Bilgileri
Hedeflenen Süre	65 Dakika

🚀 Elektrik Akımı - Gerilimi Bilgileri

Elektrik akımı, bir elektrik devresinde bulunan elektronların hareketi olarak düşünülebilir. Elektrik akımı, bir devredeki elektronların toplam hareketi olarak ölçülür ve birim olarak amper (A) kullanılır.

Gerilim, bir elektrik devresindeki elektrik akımının hareketine zorladığı kuvvet olarak düşünülebilir. Gerilim, bir devredeki elektrik akımının hareketine zorlayan kuvvet olarak ölçülür ve birim olarak volt (V) kullanılır.

Bu iki kavram birbiriyle ilişkilidir ve elektrik devrelerinde birlikte kullanılır. Örneğin, bir elektrik prizinden çıkan gerilim, bir elektrikli cihazın motorunu çalıştırmak için yeterli olabilir, ancak bu gerilimin bir elektrikli süpürgeyi çalıştırmak için yeterli olmayabileceğini unutmamalıyız.

🎗️ Bilgi

Eğitmen bu bölümde ;

- Akım ve Gerilim kavramlarından,
- Akım ve Gerilim sembollerinden,
- Akım ve Gerilim hesaplamalarından bahsetmelidir.

Elektrik akımı ve gerilim konuları fizik ve elektrik mühendisliği alanlarında önemli kavramlardır. Bu kavramların birbiriyle ilişkisi, Ohm Kanunu olarak bilinen bir kanundur. Bu kanunun yanı sıra, elektrik akımı ve gerilim ile ilgili başka önemli kanunlar da vardır, bunlar:

Ohm Kanunu, elektrik akımı ve gerilim arasındaki ilişkiyi açıklar ve şu şekildedir:

- **V = I.R**

Bu eşitlikte, I elektrik akımını (amper cinsinden), V gerilimi (volt cinsinden) ve R direnci (ohm cinsinden) gösterir. Bu eşitlik, elektrik akımının bir birim gerilimdeki bir birim dirençten geçmesiyle doğru orantılı olduğunu gösterir.

- **Kirchhoff Kanunları:** Bu kanunlar, bir elektrik devresinde elektrik akımının korunumunu ve enerjinin korunumunu açıklar.
- **Faraday Kanunu:** Bu kanun, elektriksel ve manyetik alanlar arasındaki ilişkiyi açıklar ve elektriksel enerjinin manyetik enerjiye dönüştürülmesini açıklar.
- **Lenz Kanunu:** Bu kanun, Faraday Kanunu ile bağlantılıdır ve manyetik alanların oluşumunu ve değişimini açıklar.
- **Maxwell Kanunları:** Bu kanunlar, elektromanyetik alanların davranışını açıklar ve elektromanyetik dalgaların varlığını ve özelliklerini açıklar.

📘 Kazanım

- Akım ve gerilim kavramlarını öğrenme.
- Devrede akım ve gerilim hesaplamalarının yapılabiliyor olma.
- Elektrik devrelerinin çalışma prensiplerini anlama.
- Elektrik akımının hareketini, gerilimin dağılımını ve dirençlerin etkisini anlama.
- Elektrik devrelerinin daha etkin bir şekilde tasarımı ve işletilmesi.
- Elektrik devrelerinde güvenlik sorunlarını anlama ve önleme.

Dersin Hedefleri

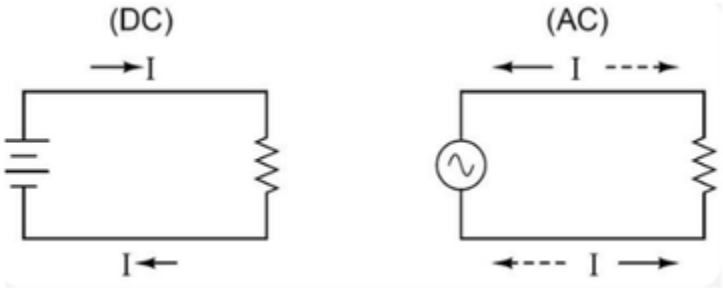
	Başlık	Hedef Kitle	Detaylar
1	☑ Elektrik Akımı - Gerilimi Bilgileri	☑ Eğitmen ☐ Öğrenci	Elektrik kanunları sayesinde elektrik devrelerinde güvenlik sorunlarını da daha iyi anlayarak önlenebileceğinin anlatılması beklenmektedir.
2	☑ Elektrik Akımı - Gerilimi Bilgileri	☐ Eğitmen ☑ Öğrenci	Elektrik kanunlarının pekiştirilmesi ve hesaplama işlemlerinin yapılabiliyor olması hedeflenmektedir.

TEMEL ELEKTRONIK: AC/DC KAVRAMLARI VE ÖZELLIKLERI

Müfredat Kodu / Ünite / Konu	İCT-400-1.3 / Temel Elektronik Kavramlar ve Tanımlar / AC/DC Kavramları ve Özellikleri
Hedeflenen Süre	30 Dakika

🚀 AC/DC Kavramları ve Özellikleri

Alternatif akım (AC) ve düz akım (DC) elektrik akımının iki farklı türüdür. AC, elektrik akımının yönünü periyodik olarak değiştiren bir akımdır. Bu, elektrik akımının bir kaynağa bağlı olmadan doğal olarak var olan bir akım türüdür. Örneğin, elektrik şebekesinde kullanılan elektrik akımı AC akımdır. DC, elektrik akımının yönünü değiştirmeyen bir akımdır. Bu tür bir akım, bir elektrik kaynağı tarafından üretilen elektrik akımıdır. Örneğin, bir pil tarafından üretilen elektrik akımı DC akımdır.



📰 Konu İçeriği

- AC/DC akımın tarihçesi (Tesla-Edison).
- AC/DC Akım kullanım alanları.

📘 Kazanım

- AC/DC Akım hakkında bilgi sahibi olmak.
  - AC, elektrik akımının yönünü periyodik olarak değiştiren bir akım türüdür.
  - DC, elektrik akımının yönünü değiştirmeyen bir akım türüdür.
  - AC, elektrik şebekesinde kullanılan elektrik akımıdır.
  - DC, bir elektrik kaynağı tarafından üretilen elektrik akımıdır.
  - AC ve DC akımlarının her ikisi de elektriksel enerjiyi taşımak için kullanılır.

✅ Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ AC/DC Kavramları ve Özellikleri	☑ Eğitmen ☐ Öğrenciler	Eğitmenden konu içeriği bölümündeki konuları iletmesi ve uygulaması beklenmektedir. Sunum, görsel ve videolarla anlatım desteklenmelidir.
2	☑ AC/DC Kavramları ve Özellikleri	☐ Eğitmen ☑ Öğrenciler	Öğrenciden ders sonunda kazanımlardaki bilgileri biliyor ve uyguluyor olması beklenmektedir.

ELEKTRONIK DEĞERLERI ÖLÇME ALETLERI

Müfredat Kodu / Ünite / Konu	ICT-400-1.4 / Temel Elektronik Kavramlar ve Tanımlar / Elektronik Değerleri Ölçme Aletleri
Hedeflenen Süre	20 Dakika

🚀 Elektronik Değerleri Ölçme Aletleri

Elektriksel ölçmede de çeşitli ölçme birimleri vardır ve bu değerleri ölçmede çeşitli ölçü aletlerinden yararlanılır. **Multimetre** elektriksel değerleri (örneğin voltaj, direnç ve akım) ölçmek için kullanılan bir alettir. Multimetre, genellikle bir voltaj ayarı, bir direnç ayarı ve bir akım ayarı içeren üç ana bölüme sahiptir. Ölçüm yapmak için, multimetre kablolarını ölçülecek cihaza veya sisteme bağlamanız gerekir. Daha sonra, multimetre ayarını ölçülecek değerin türüne göre ayarlayarak ölçüm yapabilirsiniz. Örneğin, bir cihazın voltajını ölçmek için voltaj ayarını seçebilir ve sonuçları multimetre ekranında görebilirsiniz.

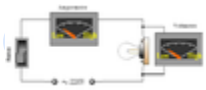
Osiloskop Elektrik, frekans ve gerilim değerlerini dalga boylarını gösteren cihazlardır.

🎗 Dijital Ölçü Aletleri

Dijital ölçü aletleri fiyatlarına ve performanslarına göre piyasada pek çok çeşidi mevcuttur. Dijital ölçü aletleri genel olarak akım, gerilim, direnç gibi ölçümler gerçekleştirmektedir. Daha çok özellikli ölçü aletlerinde ise frekans, sıcaklık, nem gibi analog ölçümler gerçekleşebilmektedir.

🎗 Analog Ölçü Aletleri

Analog ölçü aletleri , dijital ölçü aletleriyle aynı işlevi gerçekleştirmektedir. Bu ölçü aletlerinde ibre mevcuttur. İbrenin aldığı konuma göre kullanıcılar ölçüm değerlerini okurlar. Analog ölçü aletleri günümüzde pek fazla kullanılmamaktadır. Günümüzde dijital ölçü aletleri daha sık kullanılır.



🎗 Akım ve Gerilim Ölçümü

Akım: elektron hareketidir ve ampermetre ile ölçülür. Ampermetre devreye seri bağlanır.
Gerilim: iki nokta arasındaki potansiyel farktır. Gerilim voltmetre ile ölçülür. Voltmetre devreye paralel bağlanır.

📘 Kazanım

- Kompanentlerin ölçülmesinde kullanılan ölçü aletlerin tanınması.

✅ Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Elektronik Değerleri Ölçme Aletleri	☑ Eğitmen ☐ Öğrenci	Elektronik Değerleri Ölçme Aletlerinin kullanımının ve ne işe yaradıklarının öğrenilmesi beklenmektedir.
2	☑ Elektronik Değerleri Ölçme Aletleri	☐ Eğitmen ☑ Öğrenci	Devrenin ve devre elemanlarının, ölçümlerinin hangi ölçme aleti ve hangi ayarda yapılabiliyor olduğunun bilgisine sahip olunması hedeflenmektedir.

FREKANS, PERIYOT KAVRAMLARI

Müfredat Kodu / Ünite / Konu	ICT-400-1.5 / Temel Elektronik Kavramlar ve Tanımlar / Frekans, Periyot Kavramları
Hedeflenen Süre	20 Dakika

🚀 Frekans, Periyot Kavramları

Frekans, elektronikte bir sinyalin periyodik olarak tekrarlanan birim zaman diliminde kaç kez tekrarlandığını ifade eder. Bir sinyal periyodu, bir sinyalin bir tam döngüsünü tamamlaması için geçen zamanı ifade eder. Bu iki kavram birbirleriyle ilişkilidir ve birbirlerini tüm elektronik devrelerde bulunan periyodik sinyaller için tanımlar.

Frekans, Hz (hertz) cinsinden ölçülür ve sıklık olarak da adlandırılır. Örneğin, bir sinyalin frekansı 60 Hz ise, bu sinyal her saniye 60 kez tekrarlanır.

Periyot ise, bir sinyalin bir tam döngüsünü tamamlaması için geçen zamanı ifade eder. Periyot, sıklık ile ters orantılıdır ve saniye cinsinden ölçülür. Örneğin, bir sinyalin periyodu 1 saniye ise, bu sinyalin frekansı 1 Hz'dir.

Elektronik devrelerde, frekans ve periyot sıklıkla kullanılır ve birbirlerine dönüştürülerek birçok elektronik hesaplamada kullanılır. Örneğin, bir devrenin frekansını belirlemek için periyodunu ölçmek, bir sinyalin periyodunu belirlemek için de frekansını ölçmek kullanılabilir.

📘 Kazanım

- Frekans ve Periyot kavramlarının öğrenilmesi.
- Osiloskop ile frekans ve periyot ölçümünün öğrenilmesi.

✅ Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Frekans, Periyot Kavramları	☑ Eğitmen ☐ Öğrenciler	Frekans ve periyot kavramlarının öğrenilmesi ve osiloskop ile ölçüm değerlerinin açıklanması beklenmektedir.
2	☑ Frekans, Periyot Kavramları	☐ Eğitmen ☑ Öğrenciler	Frekans ve periyot kavramlarının pekiştirilmesi hedeflenmektedir.

SCRATCH EĞITIMI

Müfredat Kodu	İCT-103
Ders	Scratch
Ders İçeriği	• Scratch: Algoritma • Scratch: Değişkenler • Scratch: if-else • Scratch: Döngüler • Scratch: Fonksiyonlar • Scratch: Sınıflar ve Nesneler • Selamlama Uygulaması • Gol Atma Uygulaması • Duvarlara Çarpan Ok Uygulaması • Dikdörtgen Çizme Uygulaması • Üçgen Çizme Uygulaması • Çokgen Çizim Uygulaması • İki Kişilik Oyun Uygulaması • Yılan Oyunu Yapımı • Home Sweet Home Uygulaması • Haritalarla Denizli Uygulaması
Seviye	İCT-S1: BAŞLANGIÇ
Ön Koşullar	Tasarım alanında ilgili olmak
Amaç	Temel algoritma becerilerini edinmelerini ve kendi algoritmalarını geliştirebilecekleri düzeyde algoritma bilgisine sahip olup bu becelerini “Scratch” ortamında uygulayabilmelerini sağlamaktır.
Hedef	Kodlamayı, algoritmayı, sistematik düşünmeyi ve analitik düşünmeyi öğretmek hedeflenmiştir.
Gereksinimler	• Klavye ve mouse kullanabilmek, • Temel bilgisayar kulanım becerileri bu eğitime başlamak için yeterlidir.
Kazanım	• Kendi algoritmalarını geliştirebilir, • Geliştirdiği algoritmaları scratch ortamında uygulayabilir, • Kendi oyun ve animasyonlarını yapabilir.
Referans & Kaynak Önerisi	http://www.cankodyaz.com/egitimler/courses/scratch.html
Durum	BEKLIYOR / UYGULANIYOR / TAMAMLANDI / DEĞERLENDIRILDI
Hedeflenen Süre	540 Dakika
Eğitim Günleri	Başlama: .../.../.... Bitiş: .../.../....

SCRATCH: ALGORITMA

Müfredat Kodu / Ünite / Konu	İCT-103-1 /Scratch / Algoritma
Hedeflenen Süre	18 Dakika

🚀 Algoritma Nedir ?

Belirli bir problemi çözmek ve belli bir sonuca ulaşmak için çizilen yola algoritma denir. Algoritma, bilgisayarın adım adım ne yapması gerektiğini söyleyen bir taslaktır.

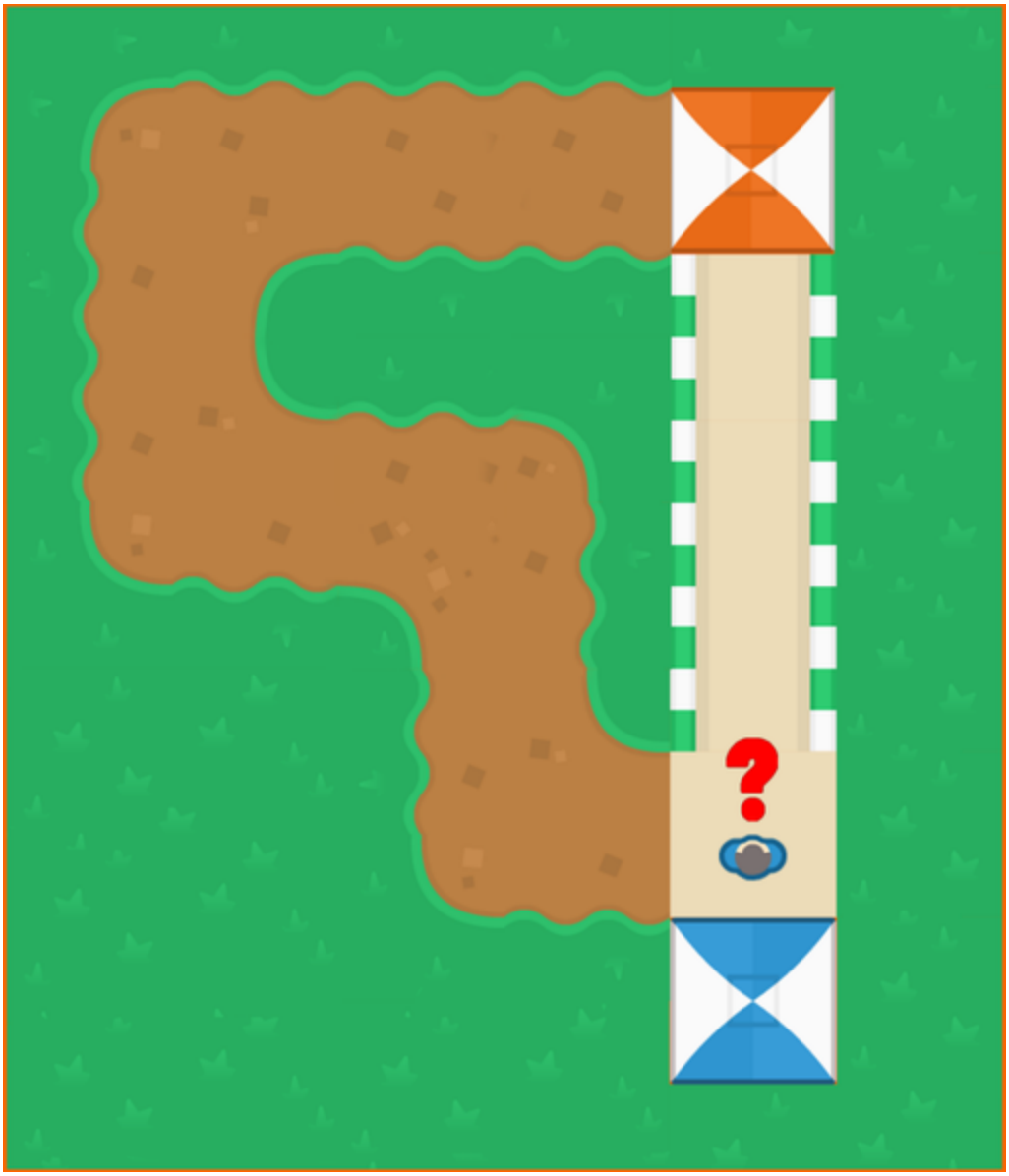
Algoritma iki kelime ile aklımızda kalabilir: **PROGRAMIN TARIFI!**

Bir Algoritmanın;

1. Başı olmalı
2. Sonu olmalı
3. Problemin çözümünü mümkün olan en az adımla, en kısa sürede gerçekleştirmeli
4. Basit olmalı

Programlamayı ilk öğrenmeye çeşitli temel konularla başlanır. Bunlar kısaca değişkenler, operatörler, şart blokları, döngüler, diziler, metotlar ya da fonksiyonlar gibi temel konulardır. Kullanıcıdan girdi almak, bu girdileri değişkenlere atayarak çeşitli işlemler gerçekleştirmek yine ilk programlamaya başladığımızda yaptığımız işlemlerdendir. Scratch ile algoritma örnekleri yaparak programlama becerilerimizi geliştirip, algoritmalar kurabilir ve bunu anlaşılır bir dille kodlayabiliriz. Böylece temel mantık kafamızda yer edinmiş olur.

Ev ile okul arasındaki yolları düşünelim: Evi A noktası, okulu da B noktası olarak tanımlayalım. A ile B arasındaki en kısa yolu tercih edilir genelde. Fakat o yolda bir sorun olduğunda, diğer alternatif yolları seçilir.



Algoritma kavramı başka bir deyişle yazılım dillerinin temelini oluşturan yapıdır.

Örneğin matematik dersinde algoritma kullanılır. Size verilen sorunun çözümü istenir ve sizler işlem, oyun, animasyon geliştirme ve sayfa önceliğine göre önce denklemi düzenler ardından gerekli işlemleri yaparak sonuca ulaşmaya çalışırsınız. Bu işlem adımlarını oluşturmak bir algoritmadır.

Farklı bir örnek ile algoritma kavramını bitirelim. Mutfak becerilerine değinelim; eğer herhangi bir yemeğin yapımı bilinir ise bu yemeğin yapılışı Türkçe, İngilizce ya da farklı bir dilde öğrenerek anlatılabilir. Değişen tek şey dillerin gramer yapıları ve telaffuzları olacaktır. Fakat herhangi bir yemeği yapımı bilinmiyorsa bilinen dil sayısının önemi olmayacaktır çünkü bilmeden o yemeğin yapılışı anlatılamaz. Bizim bu kitapta fark etmenizi istediğimiz ince çizgi budur.

Kazanım

Bu bölümde öğrenciler günlük yaşamda kullanılan programların oluşturulma mantığını kavrar.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar

1	☑ Algoritmanın tanıtımı.	☑ Eğitmen ☐ Öğrenciler	Algoritmanın öğrencilere tanıtımı. Öğrencilere algoritma mantığının anlatılması.Örnekler kullanılarak, öğrencilerin anlamasını kolaylaştırmak adına eğitmen tarafından çözümlemeler yapılması. Öğrencilere anlatılan örnekler hakkında soruların sorulup, konunun anlaşılabilirliğinin ölçülmesi. Öğrenciler için tanıtılan, algoritma örneklerinden herhangi 3 örnek ile ödevlendirmenin yapılması beklenmektedir.
2	☑ Uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen aracılığıyla tanıtılan, algoritmalara ait uygulamaların yapılması.

Referanslar & Dokümantasyon

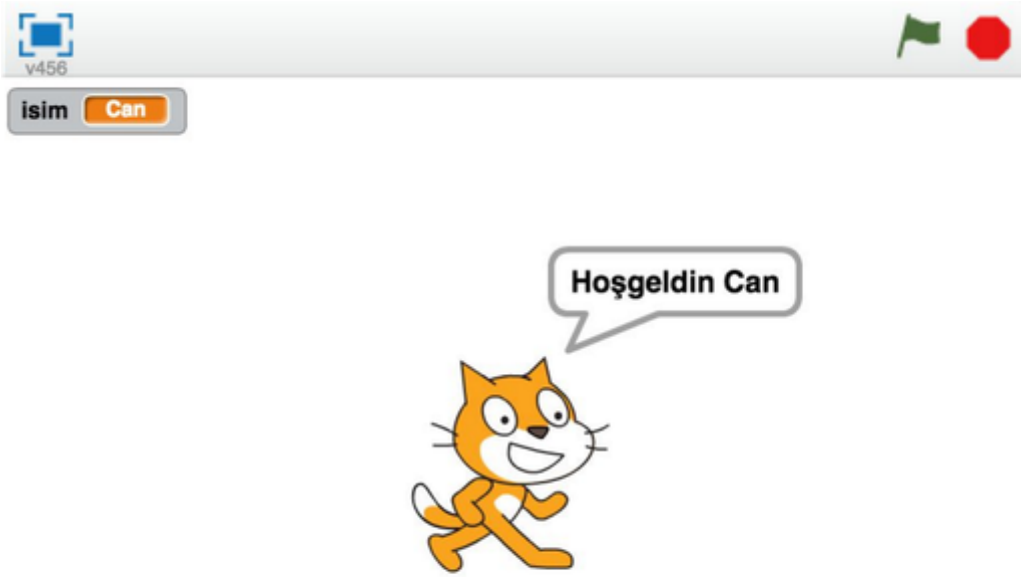
- http://www.cankodyaz.com/egitimler/courses/scratch.html

SCRATCH: DEĞİŞKENLER

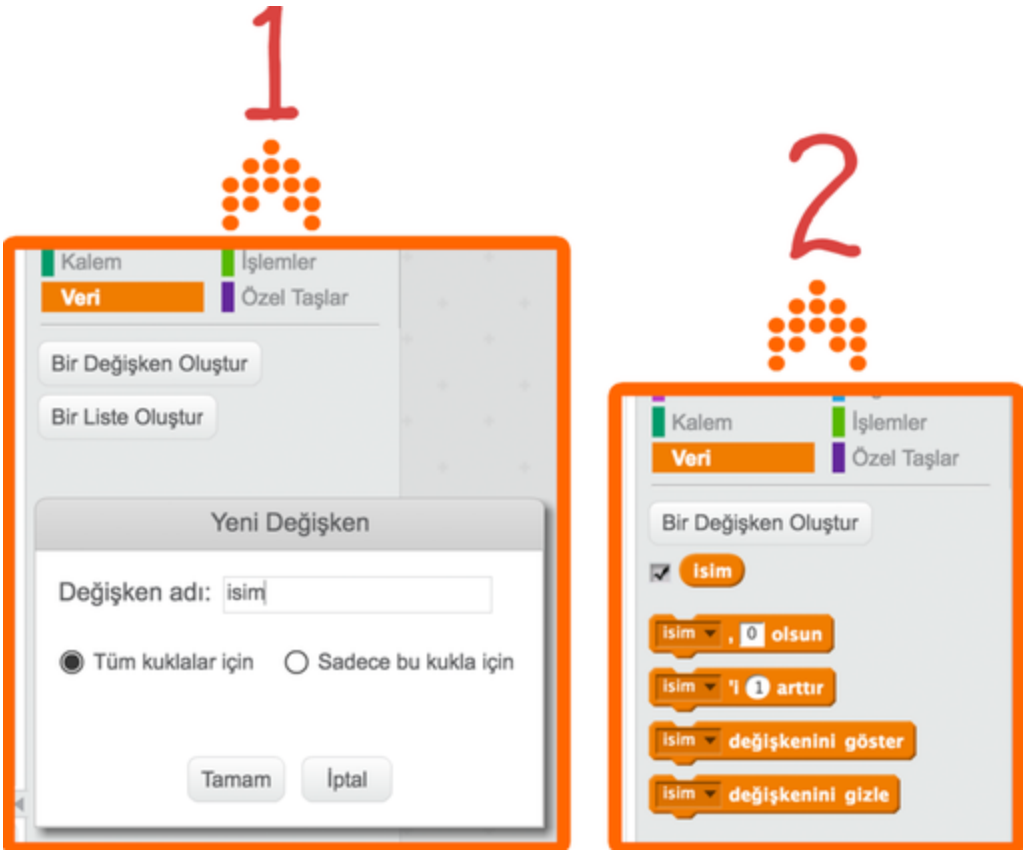
Müfredat Kodu / Ünite / Konu	İCT-103-2 / Scratch / Değişkenler
Hedeflenen Süre	18 Dakika

🚀 Değişken Nedir?

- İnternet tarayıcısını (Google Chrome, Mozilla Firefox v.b.) açıp herhangi bir site üzerinde üyelik formu doldururken "üye ol" butonuna tıklamadan tarayıcının kapandığını düşünelim. Yazılan tüm veriler (kullanıcı isminiz, e-posta adresiniz, şifreniz v.s.) silinecek ve tarayıcıyı yeniden açıldığında kayıt işlemine yeniden başlamak zorunda kalınacaktır.
- Bu senaryodaki tüm girilen veriler bilgisayarın RAM belleği denilen geçici belleğinde o an sadece o kullanıcı adı(e-posta adresi, şifre gibi) verisini tutmak için ayrılmış hafıza odacıklarına yazılmıştır.
- Bu hafıza odacıkları, yazılımda kodlama sırasında isim verilerek tanımlanır. O hafıza odacığının ismi çağırılarak içerisine veri atanır ve gerekli işlemler yapılır.
- Örneğin Selamlama uygulamasında selamlama örneği bulunmaktadır. Uygulama açıldığı zaman RAM bellekte "isim" adında bir değişken tanımlanır ve daha sonra kullanıcıdan ismi istenir. Kullanıcının adı "isim" adındaki değişkenin içerisine atanır. Bir sonraki ekranda "Hoş geldin" mesajı ile birlikte "isim" değişkeninin içerisindeki değer ekrana yazılır.



- Kısaca değişken matematik derslerinden tanınan “X” kavramıdır. Bir isim tanımlanır ve içerisine değerler atanarak işlemler yapılır.
- Değişkenlerin yazılım dillerinde bir çok çeşidi bulunmaktadır. Bu çeşitlerin oluşmasındaki temel neden değişkenlerin bellekteki kapasitelerinin farklı olmaları ve işlenecek verilerin farklı yapılarda olmasıdır.
- Farklı yapılardan kastedilen ise şudur. Bir değişken ya metinsel değişkendir ya da sayısal değişkendir.
- Örneğin yaş, sayı, gibi değerler sayısal değişken iken isim soyisim gibi değişkenler metinseldir. Bu konuda kafa karışıklığını önlemek adına şöyle bir formül bulunmaktadır;
- Eğer bir değişken toplama çıkarma gibi matematiksel işlemlere tabi tutulmayacak ise o değişken bir metinsel değişkendir.
- Örnek T.C. kimlik numarası ya da cep telefonu numarası toplama işlemi ya da diğer matematiksel işlemler içerisine alınmaz o halde bu veriler RAM’de geçici olarak metinsel değişken şeklinde tutulur.
- Scratch üzerinde değişken tanımlama işlemi için; Veri başlığı altından “Bir Değişken Oluşur” butonu tıklanır. Ekrana çıkan isimlendirme ekranına değişken ismi girilir. Ancak değişken tanımlama ve diğer kodlama içinde geçecek ifadelerde Türkçe karakterler(ö,ç,ş,i,ğ,ü,Ö,Ç,Ş,İ,Ğ,Ü) ve boşluk karakteri kullanılmaması faydalı olacaktır.



NOT: Görselde dikkat edildiğinde iki seçecek vardır; bu seçeneklerden ilki yani “Tüm kuklalar için” seçeneği yazılım dillerindeki “global” değişken kavramına denktir ve tüm program içerisinde tanımlanan değişkenin kullanılacağını belirtir. Diğer seçenek olan “Sadece bu kukla için” seçeneği ise “yerel” değişken diye ifade edilen kısıtlı alanda kullanılacak değişkenleri ifade eder.

Kazanım

- Bu bölümde öğrenciler oyun uygulamaları ile değişkenleri kavrar.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar

1	☑ Değişkenlerin tanıtımı.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere değişkenlerin anlatılması amaçlanmaktadır. Öğrencilere değişkenlerin mantığının anlatılması. Örnekler kullanılarak, öğrencilerin anlamasını kolaylaştırmak adına eğitmen tarafından çözümlemeler yapılması.Öğrencilere anlatılan örnekler hakkında soruların sorulup, konunun anlaşılabilirliğinin ölçülmesi. Konunun daha iyi kavranılması için, öğrencilerinizle birlikte uygulamalar bölümü altındaki **SELAMLAMA UYGULAMASI** 'nı yapabilirsiniz.
2	☑ Uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen aracılığıyla tanıtılan, değişkenlere ait uygulamaların yapılması amaçlanmaktadır.

SCRATCH: IF-ELSE

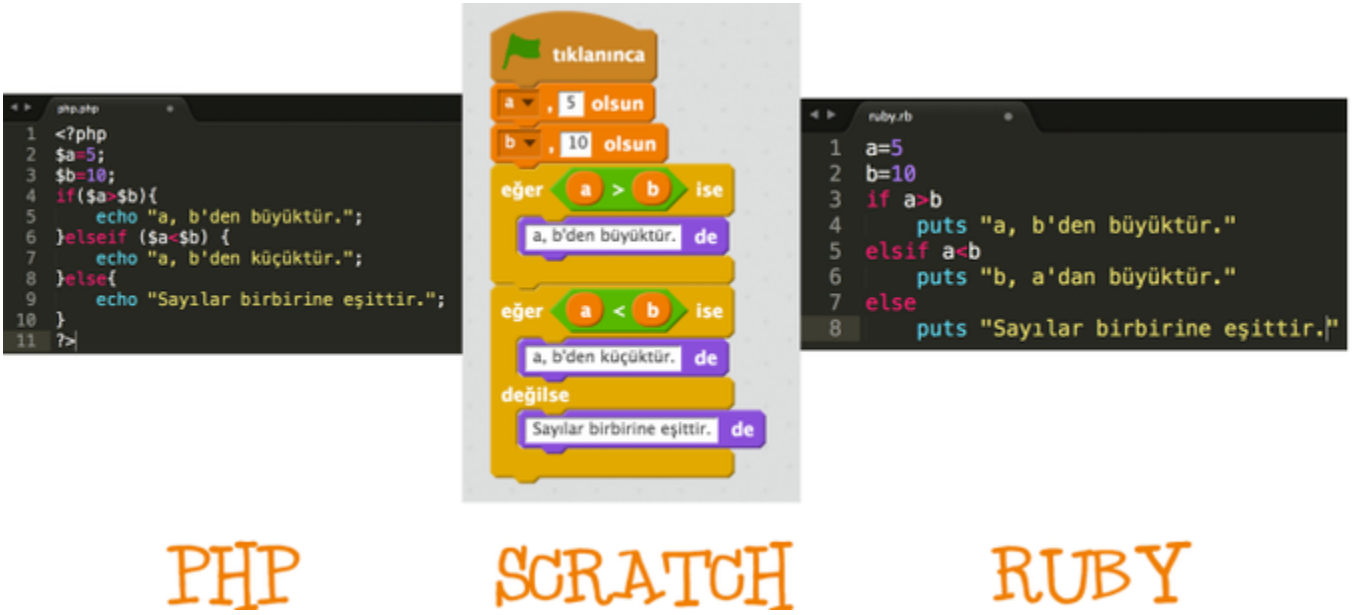
Müfredat Kodu / Ünite / Konu	İCT-103-3 / Scratch / if-else Komutları
Hedeflenen Süre	18 Dakika

🚀 if-else Komutları Nelerdir?

- Bilindiği gibi biz insanlar olarak iletişim kurmak için “dil” kavramını kullanırız. Her toplumun kendine has bir dili mevcuttur ve bu dili bilenler arasında rahatlıkla iletişim kurulur. Yazılım dünyasında ise iletişim kurmaya çalıştığımız nesne bir makinedir.
- Makine diye tanımladığımız bilgisayarlar sadece bizim yazdıklarımızı gerçekleştirirler. Evet bizim yazdıklarımızın dışında herhangi bir şey yapmazlar. Durum böyle olunca bazı belirsizlikleri biz önceden görüp ona söylemeliyiz. Yani bilgisayara hava durumuna göre yanına alacağı eşyaları söyleyecek olursak şuna benzer bir algoritma kurmalıyız.

> ***Eğer*** *hava yağmurluysa;*
>
> *Şemsiye ya da yağmurluk al.*
>
> ***Değilse Eğer*** *hava karlı ve soğuk ise;*
>
> *Mont, atkı, bere, eldiven ve bot al.*
>
> ***Değilse Eğer*** *hava rüzgarlı ama soğuk değilse;*
>
> *İnce bir mont ya da rüzgarlık al.*
>
> ***Değilse Eğer*** *hava güneşli ve sıcak ise;*
>
> *Tişört, şort ve yazlık ayakkabı al.*

- Yukarıdaki benzetmede görüldüğü gibi, bilgisayara karar verebilmesi için öngörülen tüm ihtimaller yazıldı. Artık bilgisayar hangi durumda hangi davranışları yapacağını bilmektedir.
- Kontrol deyimlerinin temel mantığı bu kadardır, değişen tek şey ise PHP, C#, C, Ruby, Python gibi güncel yazılım dillerindeki kullanım şekilleridir. Örnekleri incelendiği takdirde aynı işlemi yaptıkları fakat yazım şekillerinin farklı olduğu anlaşılacaktır.



- Scratch üzerinde if/else yada if tanımlama işlemi için; Kontrol başlığı altından if/else yada if taşları sürüklenir.
- Ekrana çıkan isimlendirme ekranına değişken ismi girilir. Ancak değişken tanımlama ve diğer kodlama içinde geçecek ifadelerde Türkçe karakterler (ö,ç,ş,i,ğ,ü,Ö,Ç,Ş,İ,Ğ,Ü) ve boşluk karakteri kullanılmaması faydalı olacaktır.

Kazanım

- Bu bölümde öğrencilere örnekler ve bazı karşılaştırmalarla if-else komutu öğretilmektedir.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ if-else komutlarının tanıtımı.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere if-else komutunun anlatılması amaçlanmaktadır. Öğrencilere if-else komutlarının anlatılması. Örnekler kullanılarak, öğrencilerin anlamasını kolaylaştırmak adına eğitmen tarafından çözümlemeler yapılması.Öğrencilere anlatılan örnekler hakkında soruların sorulup, konunun anlaşılabilirliğinin ölçülmesi. Konunun daha iyi kavranılması için, öğrencilerinizle birlikte uygulamalar bölümü altındaki **DUVARA ÇARPAN OK UYGULAMASI** 'nı yapabilirsiniz.
2	☑ Uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen aracılığıyla tanıtılan, if-else komutlarının ait uygulamaların yapılması tamamlanması.

SCRATCH: DÖNGÜLER

Müfredat Kodu / Ünite / Konu	İCT-103-4 / Scratch / Döngüler
Hedeflenen Süre	18 Dakika

🚀 Döngüler Nelerdir?

- Tekrarlanan işlemler ve kod bloklarında kullanılan yapılara döngü denir.
- Döngü bloğu içerisinde bulunan kod blokları, döngü sayısı kadar ve sırasıyla gerçekleştirilir.
- Döngü bloğu içerisinde 1 veya daha fazla kod bloğu kullanılabilir.

> *İlk okul birinci sınıfta, öğrencilere verilen ödevler arasında aynı harfin çok tekrar edilerek yazılması yer alır. Bu ödev parmak kaslarını geliştirdiği gibi o harfin güzel yazılması için de deneyim sağlar. Ancak bu ödevler öğrencilere genelde sıkıcı gelir. Yazılımcılar buna benzer tekrar edilecek işlemler olduğunda bazı döngü ya da tekrar ifadeleri denilebilecek terimler kullanırlar. Örneğin;*
>
> **100 kere tekrar edilecek;**
>
> **Yaz "A";**
>
> *Bu iki satır içerisinde neredeyse 1 saniye içerisinde 100 defa "A" harfini ekrana yazdırmak mümkündür.*

Bu örneklerden yola çıkarak yazılımda döngünün tanımını yapacak olursak; Döngü, bir işin tekrar edilmesidir. Gerekli durumlarda bir şarta bağlanabilir.

📘 Kazanım

- Bu bölümde öğrenciler döngülerin yapısını kullanarak oluşturulmuş kod bloklarının mantığını kavrar.

✅ Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar

1	☑ Döngülerin tanıtımı.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere döngülerin ve döngülere ait kavramların anlatılması amaçlanmaktadır.Öğrencilere değişken mantığının anlatılması.Örnekler kullanılarak, öğrencilerin anlamasını kolaylaştırmak adına eğitmen tarafından çözümlemeler yapılması.Öğrencilere anlatılan örnekler hakkında soruların sorulup, konunun anlaşılabilirliğinin ölçülmesi.Konunun daha iyi kavranması için, öğrencilerinizle birlikte uygulamalar bölümü altındaki **DİKDÖRTGEN ÇİZME UYGULAMASI** 'nı yapabilirsiniz.
2	☑ Uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen aracılığıyla tanıtılan, döngülere ait uygulamaların yapılması amaçlanmaktadır.

SCRATCH: FONKSIYONLAR

Müfredat Kodu / Ünite / Konu	İCT-103-5 / Scratch / Fonksiyonlar
Hedeflenen Süre	18 Dakika

🚀 Fonksiyonlar Nelerdir?

- Daha önceden tanımlanan işleri yapan kod gruplarıdır.
- Yazılımda yapmak istenilen işleri kolaylaştırmak adına o yazılım dilinde hazırlanmış fonksiyonlar vardır.
- Örneğin bir fonksiyonun görevi ona iletilen sayıya 3 eklenip ekrana yazdırması istendiğinde farklı dillerde ve scracth üzerindeki tanımlama şekli aşağıda verilmiştir.

```php
<?php
public function ekle($sayi='0')     //php üzerinde fonksiyon tanımlama işlemi...
{
    echo $sayi+3;
}                                   //bu satırda tanımala işlemi bitmektedir.

ekle(3);                            //Tanımlanan değişkene 3 değeri gönderilmiştir.
                                    //Sonuç olarak ekranda 6 değeri gösterilecektir.
?>
```

```ruby
def ekle(sayi)        #ruby üzerinde fonksiyon tanımla işlemi
    puts sayi+3
end                   #bu satırda tanımlama bitmektedir.

ekle(3)               # "ekle" isimli fonksiyona 3 değeri gönderilmiştir
                      # çalıştığı zaman ekrana 6 değeri yazacaktır.
```

Yukarıda fonksiyonların yazılım dillerinde hazır olarak geldiği söylendi ancak kendi algoritmamız içerisindeki işleri yaptırmak için biz de fonksiyonlar oluşturup kullanabiliriz.

Kazanım

- Bu bölümde öğrenciler örnekler yardımıyla fonksiyonları öğrenmektedir.

✅ Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar

1	☑ Fonksiyonların tanıtımı.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere fonksiyonların anlatılması amaçlanmaktadır. Öğrencilere fonksiyon mantığının anlatılması. Örnekler kullanılarak, öğrencilerin anlamasını kolaylaştırmak adına eğitmen tarafından çözümlemeler yapılması. Öğrencilere anlatılan örnekler hakkında soruların sorulup, konunun anlaşılabilirliğinin ölçülmesi. Öğrenciler için tanıtılan, fonksiyon örneklerinden herhangi 3 örnek ile ödevlendirmenin yapılması.
2	☑ Uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen eşliğinde tanıtılan fonksiyonlara ait uygulamaların tamamlanması amaçlanmaktadır.

SCRATCH: SINIFLAR VE NESNELER

Müfredat Kodu / Ünite / Konu	İCT-103-6 / Scratch / Sınıf ve Nesneler
Hedeflenen Süre	45 Dakika

🚀 Sınıf Ve Nesneler Nelerdir?

- Ailelerin kullandığı ya da çevrede görülen tüm motorlu taşıtların ortak bir özelliği vardır; güç üreten bir motor ve hareket etmesini sağlayan tekerlek sistemi.
- Çevredeki tüm bu otomobiller, kamyonlar, motosikletler bu iki ortak özelliğe sahiptir.
- Yazılım dünyasındaki nesne ve sınıf kavramları yukarıda anlatılan örnek ile uyuşmaktadır. Öncelikle motor ve hareket sistemi oluşturulur bu nesne tanımlamadır. Daha sonra bu temel donanımlara ek özellik eklenir örneğin hava yastığı, müzik ve ses sistemi, klima ve sonuçta ortaya çıkan bir otomobil motorlu taşıtlar nesnesinin otomobil sınıfıdır.
- Motorlu taşıtlar nesnesine eklenecek olan ağır yük taşımak için büyük bir kasa ve diğer ek donanımlar kamyon sınıfını oluşturur

📘 Kazanım

- Bu bölümde öğrenciler sınıfları ve nesneleri kavrar, uygulamalarla öğrendiklerini pekiştirir.

✅ Dersin Hedefleri

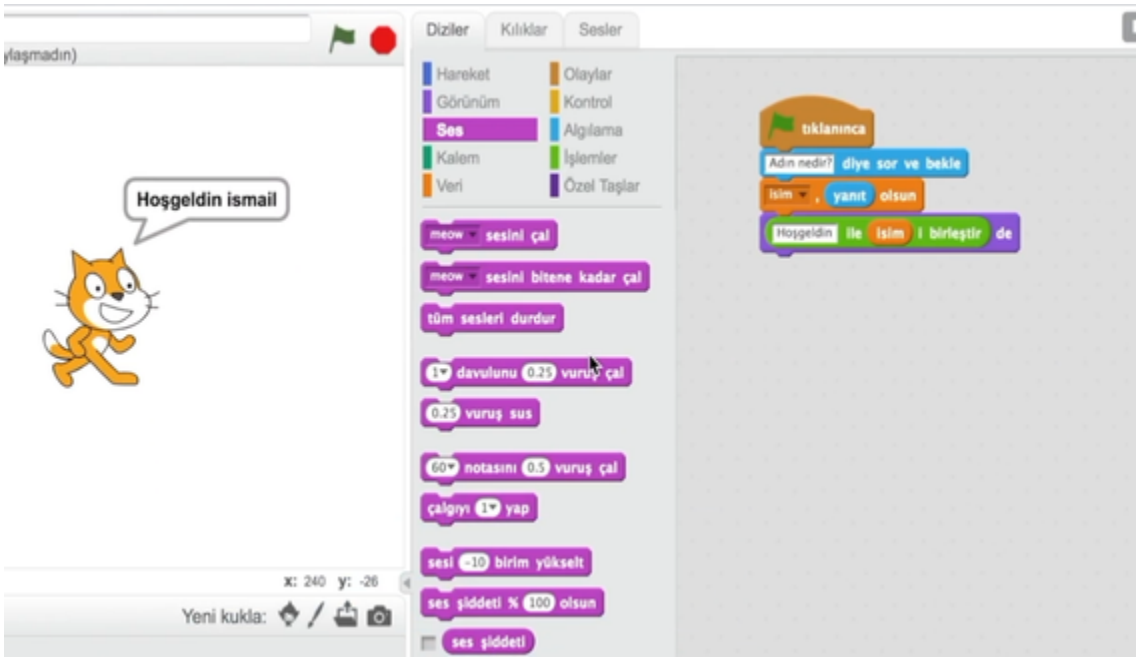
	Başlık	Hedef Kitle	Detaylar
1	☑ Sınıf ve nesnelerin tanıtımı.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere sınıf ve nesnelerin öğretilmesi amaçlanmaktadır. Öğrencilere sınıf ve nesnelerin anlatılması. Örnekler kullanılarak, öğrencilerin anlamasını kolaylaştırmak adına eğitmen tarafından çözümlemeler yapılması. Öğrencilere anlatılan örnekler hakkında soruların sorulup, konunun anlaşılabilirliğinin ölçülmesi. Öğrenciler için tanıtılan, sınıf ve nesne örneklerinden herhangi 3 örnek ile ödevlendirmenin yapılması.
2	☑ Uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen eşliğinde sınıf ve nesnelere ait uygulamaların tamamlanması amaçlanmaktadır.

SELAMLAMA UYGULAMASI

Müfredat Kodu / Ünite / Konu	İCT-103-7.1 / Uygulamalar / Selamlama Uygulaması
Hedeflenen Süre	45 Dakika

🚀 Selamlama Uygulaması

- Soru sorma ve cevap alma eğilimi için “Adınızı giriniz” taşını yapıştırıyoruz.
- Bir isim değişkeni tanımlıyoruz ve bu isim değişkenine aldığımız yanıtı atıyoruz.
- Daha sonra işlemler menüsü altından “Hello ile world” taşını yapıştırıyoruz, içene “hoş geldiniz” yazıyoruz ve isim değişkeni ekliyoruz.



Kazanım

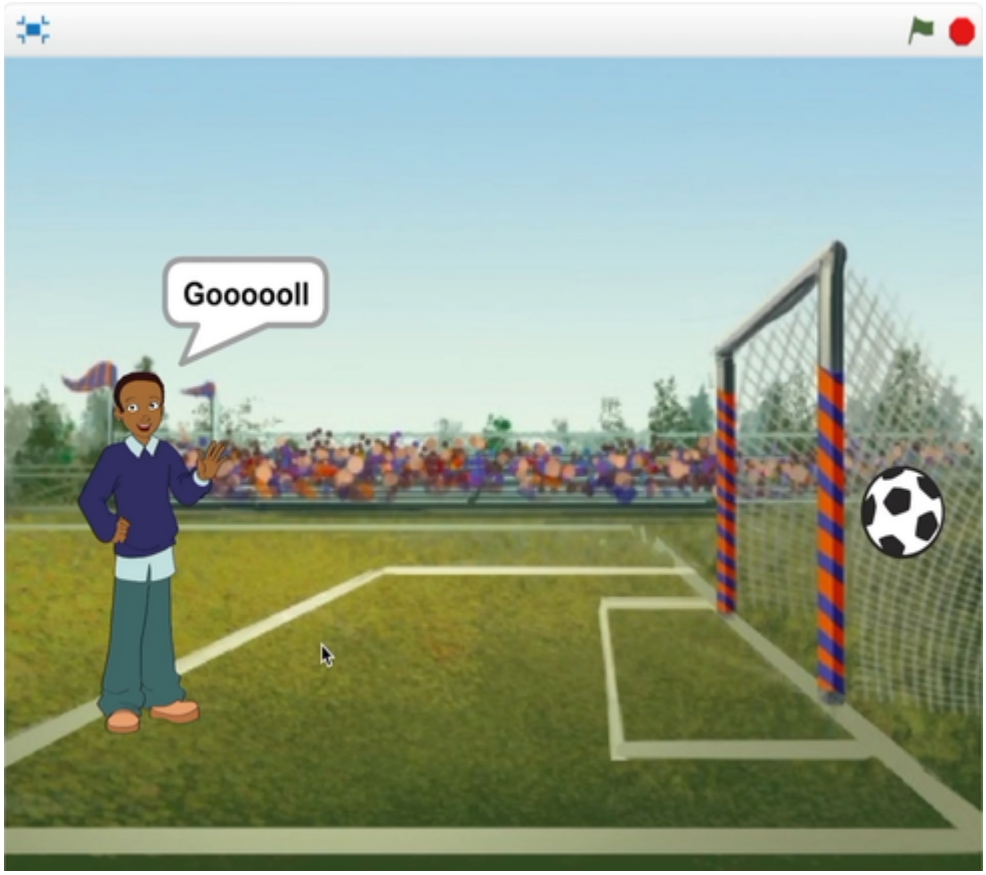
- Değişken kavramının pekiştirilmesi.
- Veri kavramının pekiştirilmesi.

GOL ATMA UYGULAMASI

Müfredat Kodu / Ünite / Konu	İCT-103-7.2 / Uygulamalar / Gol Atma Uygulaması
Hedeflenen Süre	30 Dakika

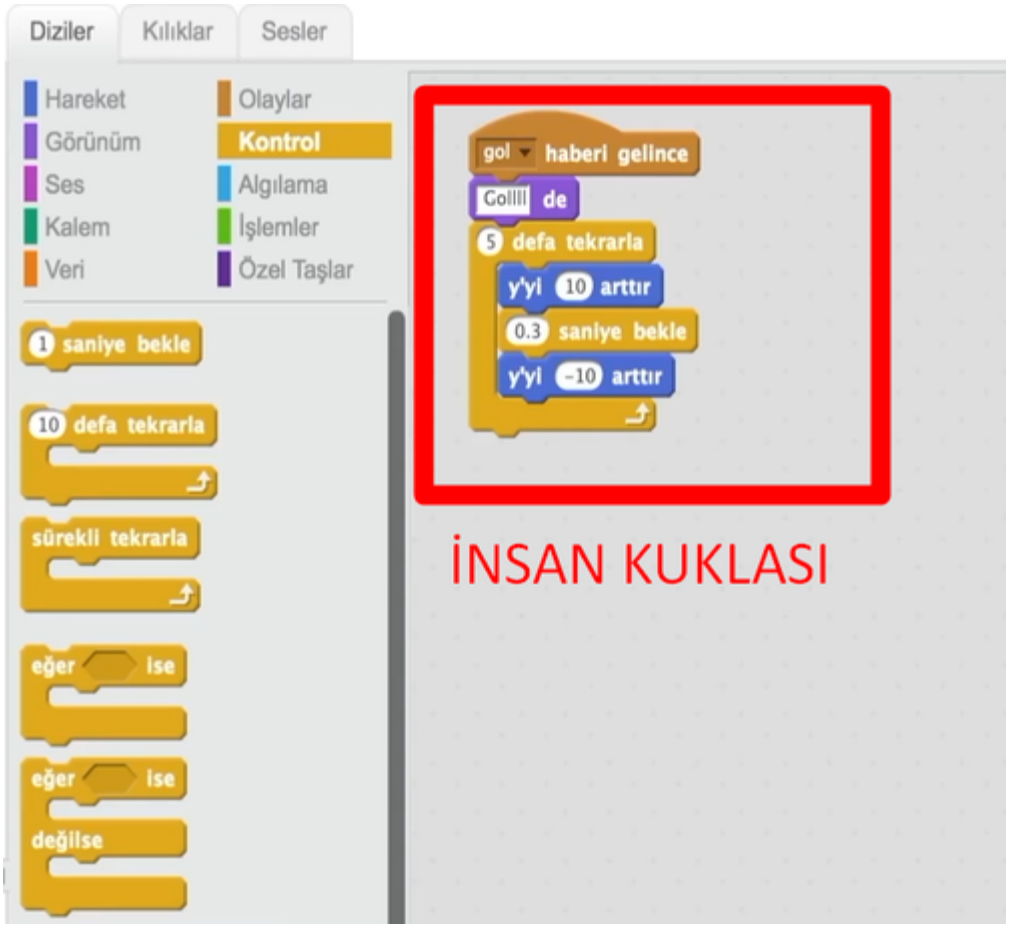
🚀 Gol Atma Uygulaması

Bu uygulamada ana karakterimiz top olacaktır. Sabit duran bir oyuncu kuklası seçilecek ve uygun sahne eklenecektir. Boşluk tuşuna basılınca top kaleye girecek ve eklediğimiz sabit oyuncumuz “gol” diyerek haber verecektir.

- 

- Öncelikle sahnemizi yani arka planımızı değiştirmek ile başlayalım.
- Sahnenin altında bulunan yeni dekor panelinden “Dekoru kütüphaneden seç” düğmesi tıklayarak dekor kütüphanesini açıyoruz ve dekor kütüphanesinden sahne için dekor seçiyoruz.
- Sonra Gol Atma Oyunumuzun ana karakterlerini, kuklalarımızı eklemek için “Kuklayı Kütüphaneden Seç” düğmesine tıklayarak karakterleri içeren resim dosyaları seçip yüklüyoruz.
- Buradan bir “Top” kuklası bir de “Oyuncu” kuklası seçiyoruz.
- Kuklaları seçtikten sonra büyültme ve küçültme işlemleri gerçekleştiriyoruz.
- Daha sonra boşluk tuşuna basılınca seçili top kuklamızın kalenin nereye gitmesini istiyorsak oraya taşıyıp hareket edecek ve “gol” diye haber verecektir.

- Yukarıda görüldüğü gibi “TOP” nesnesi seçili iken başlangıç konumu ve topun gitmesi istenilen koordinasyonlar arasındaki alması gereken aksiyonlar kod ile yazılmıştır. Amaç uygulama başlatıldığında topun başlangıç komutuna gitmesi, boşluk tuşuna basıldığında ise 2 saniyede kale içindeki konuma süzülerek gitmesidir.



- Yukarıda görüldüğü gibi top nesnesinin hareketi sağlandıktan sonra, Kuklaya bilgi gelerek Golll demesi ve 5 kere zıplaması sağlanmaktadır.
- Bu şekilde başlatma komutundan sonra boşluk tuşuna basıldığında top istenilen konuma süzülerek gider ve kukla bir sevinç gösterisi yaparak gol der ve zıplama hareketi yapmış olur.
- Bu uygulama ses komutları ve Kukla hareketleriyle daha da zenginleştirilebilir.

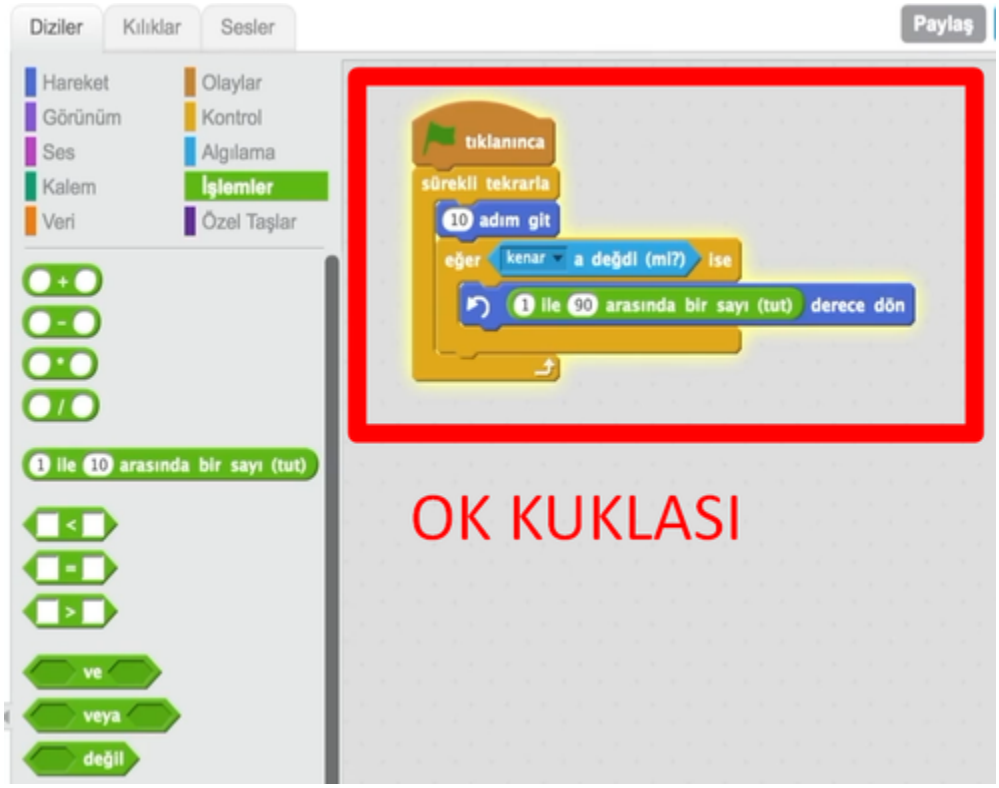
Kazanım

- Bu uygulamada öğretilmek istenilen durum, bilgisayar gibi düşünmenin ne kadar önemli olduğu, zıplama ve vurma gibi insani bir eylemin bilgisayara taşınırken ne gibi aşamalardan geçtiğini öğretmek.
- Oyun tasarlamak.
- Kodların pekiştirilmesi.

DUVARLARA ÇARPAN OK UYGULAMASI

Müfredat Kodu / Ünite / Konu	İCT-103-7.3 / Uygulamalar / Duvarlara Çarpan Ok Uygulaması
Hedeflenen Süre	30 Dakika

🚀 Duvarlara Çarpan Ok Uygulaması



- Uygulamanın amacı, sürekli hareket eden bir ok uygulaması yapmaktır.
- Kütüphane bölümünden yeni bir ok kuklası seçilir.
- Ok sürekli hareket edeceği için “Sürekli tekrarla” tuşunu ekliyoruz ve 10 adım atıyoruz.
- İf-else taşlarını ekliyoruz ve okun kenarlara çarptığı durumlarda ki kontrolünü sağlıyoruz.
- Daha sonra saat yönünün tersinde 1 ile 90° arasında dönmesi komutunu ekliyoruz.
- Ardından yeşil bayrağa tıklayarak uygulamayı çalıştırıyoruz.

Kazanım

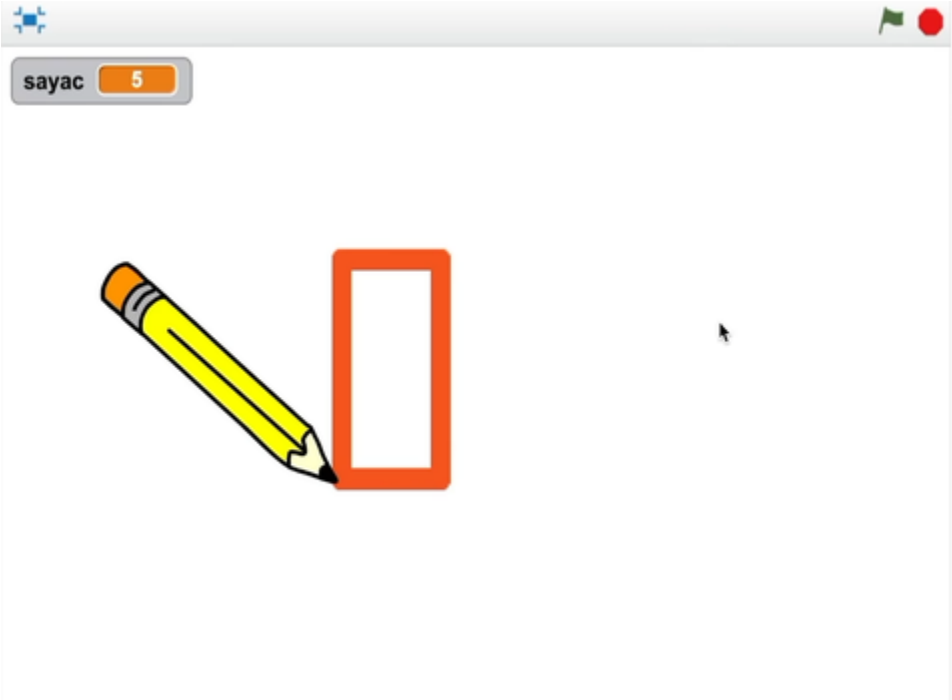
- if-else komutunun pekiştirilmesi

DIKDÖRTGEN ÇIZME UYGULAMASI

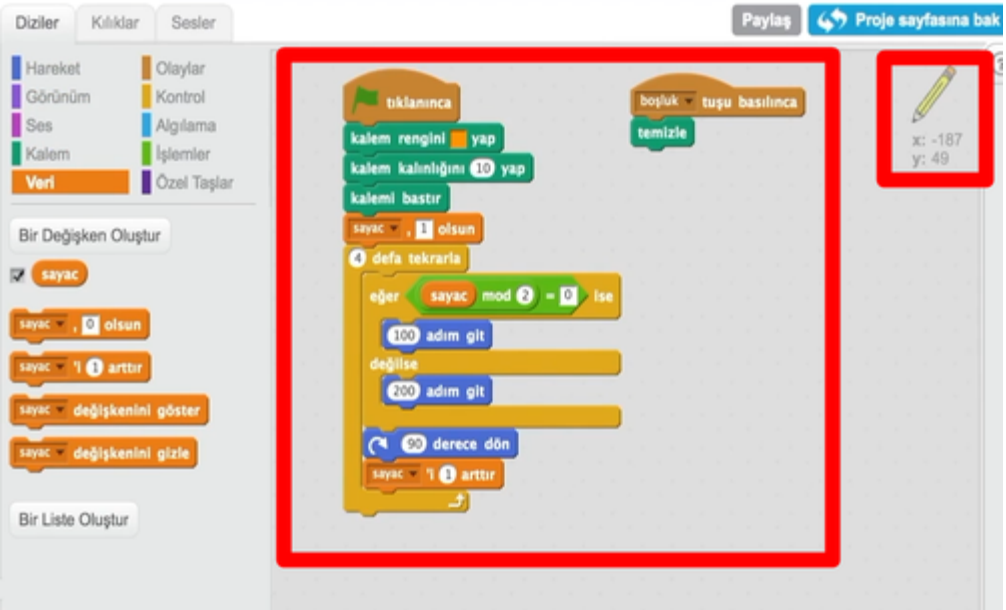
Müfredat Kodu / Ünite / Konu	İCT-103-7.4 / Uygulamalar / Dikdörtgen Çizme Uygulaması
Hedeflenen Süre	30 Dakika

🚀 Dikdörtgen Çizme Uygulaması

- Uygulamanın boşluk tuşuna basıldığında ekranda çizili olan dikdörtgenin silinmesi ve bayrak komutuna basıldığında dikdörtgen çizmesi uygulamasıdır.



- Kalem Kuklası seçilir.
- Kalem uygulaması içerisindeyken şu komutlar yazılır. Amaç kalemin yazı yazma kalınlığı, rengi ve yapacağı harekettir. Bu sayede dikdörtgen şeklinin oluşması amaçlanmaktadır.



- Dekor,kukla ve içerik geliştirilerek uygulama zenginleştirilebilir.

Kazanım

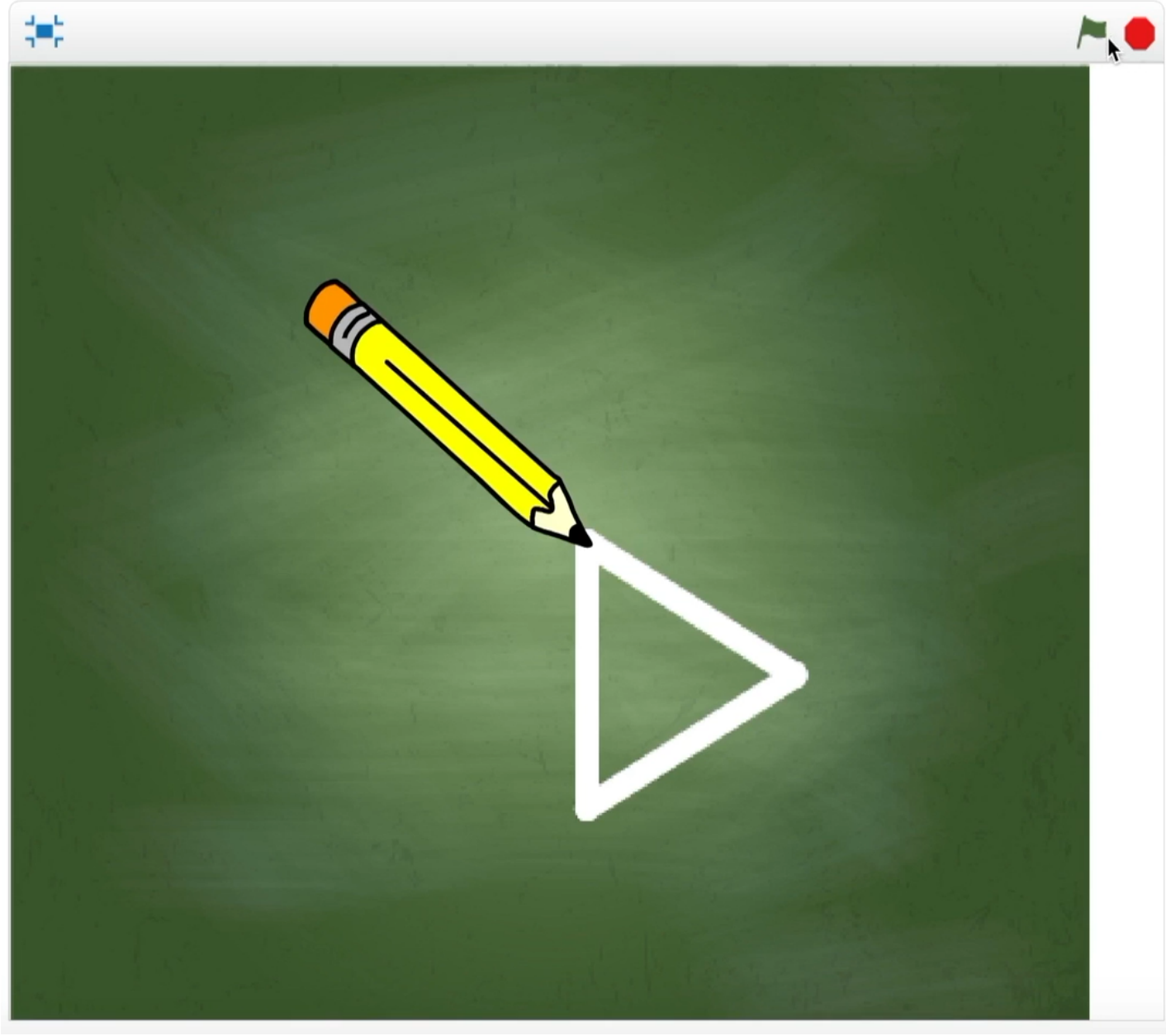
- İf-Else ve döngü yapılarının pekiştirilmesi.

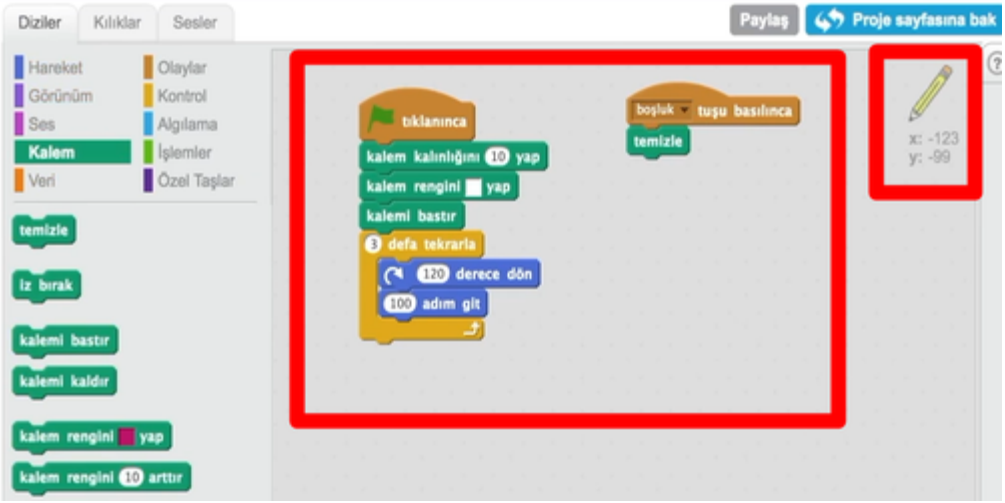
ÜÇGEN ÇIZME UYGULAMASI

Müfredat Kodu / Ünite / Konu	İCT-103-7.5 / Uygulamalar / Üçgen Çizme Uygulaması
Hedeflenen Süre	30 Dakika

🚀 Üçgen Çizme Uygulaması

- Uygulamanın amacı, kalem kuklası ile üçgen çizmektir.
- Kalem Kuklası seçilir.
- Kalem uygulaması içerisindeyken şu komutlar yazılır. Amaç kalemin yazı yazma kalınlığı, rengi ve yapacağı harekettir. Bu sayede üçgen şeklinin oluşması amaçlanmaktadır.





- Yeşil bayrağa basarak üçgeni çizdiriyoruz.
- Bu uygulama sonrasında çokgen çizimleri yapılmalıdır. Altıgeni, Sekizgen , Daire vb.

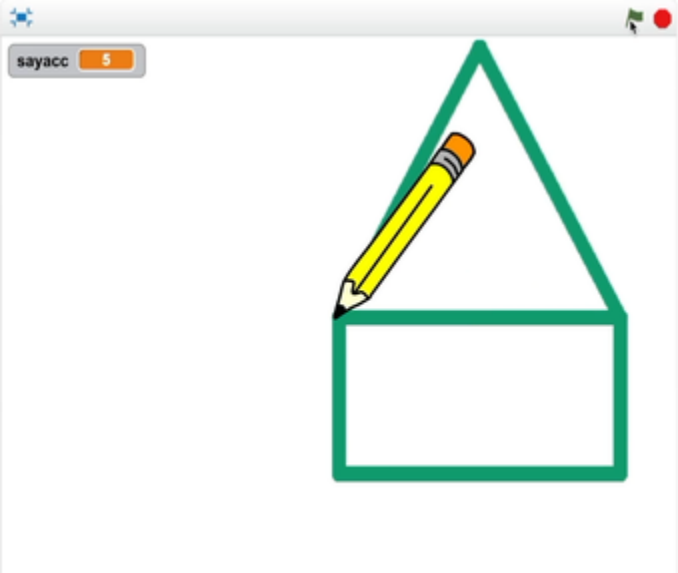
Kazanım

- Yaratıcılık, programlama ve bilgi işlemsel düşüncenin arttırılması.

ÇOKGEN ÇIZIM UYGULAMASI

Müfredat Kodu / Ünite / Konu	İCT-103-7.6 / Uygulamalar / Çokgen Çizme Uygulaması
Hedeflenen Süre	30 Dakika

🚀 Çokgen Çizme Uygulaması



Uygulamanın amacı; Kalem karakteri ile çokgen oluşturabilmektir.

Kukla olarak kalem karakteri seçilir.

Kalem karakterinin rengi ve kalınlığı ayarlanır.

Sayaç değişkeni oluşturularak çokgenin oluşması sağlanır.

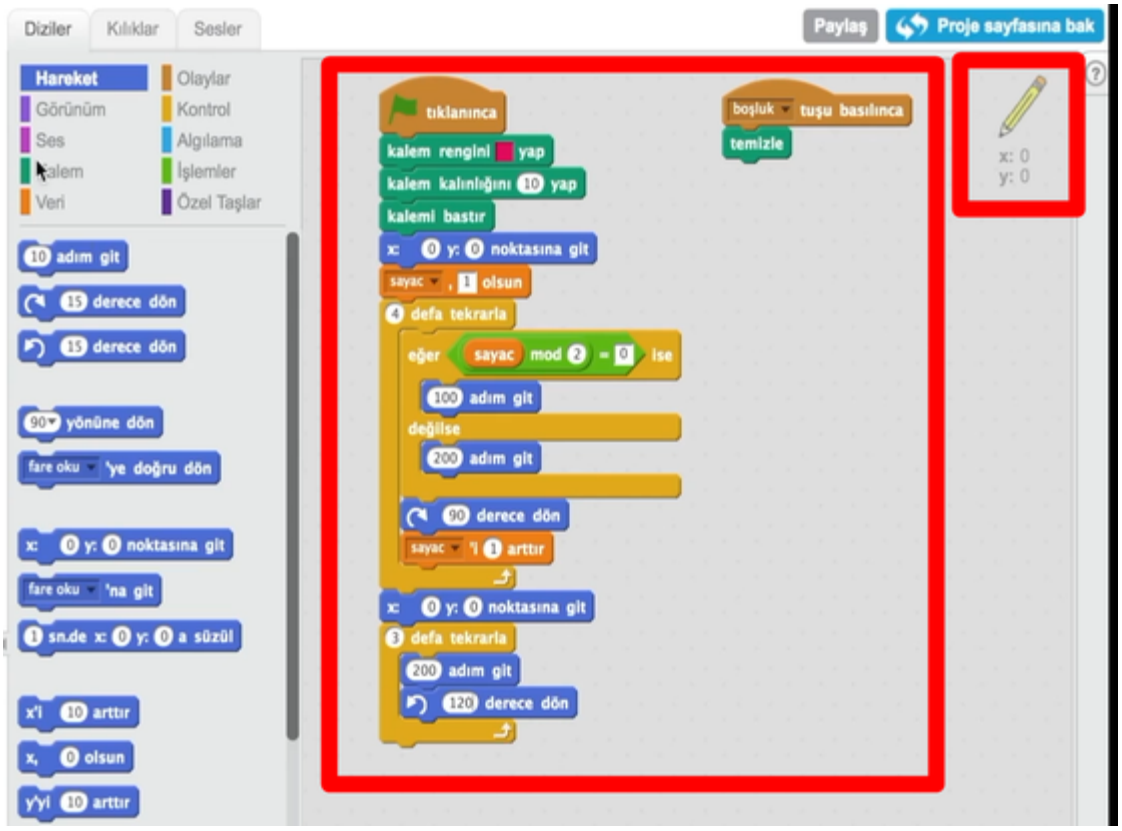
Yapılan işlemler döngüye alınır. “4 “defa tekrarla” bloğu “.

Daha sonra if-else bloğunu alarak içine sayaç eklinir.

Adım olarak kısa kenara 100 uzun kenara 200 adım girilir.

Saat yönünde 90° dönüş bloğu eklenir.

Tekrar bir sayaç ekleyip yeşil bayrağa basarak çokgen oluşturulur.



- Bu uygulama sonrasında arka dekor kukla vb. şekilde geliştirme yapılması tavsiye edilmektedir.

Kazanım

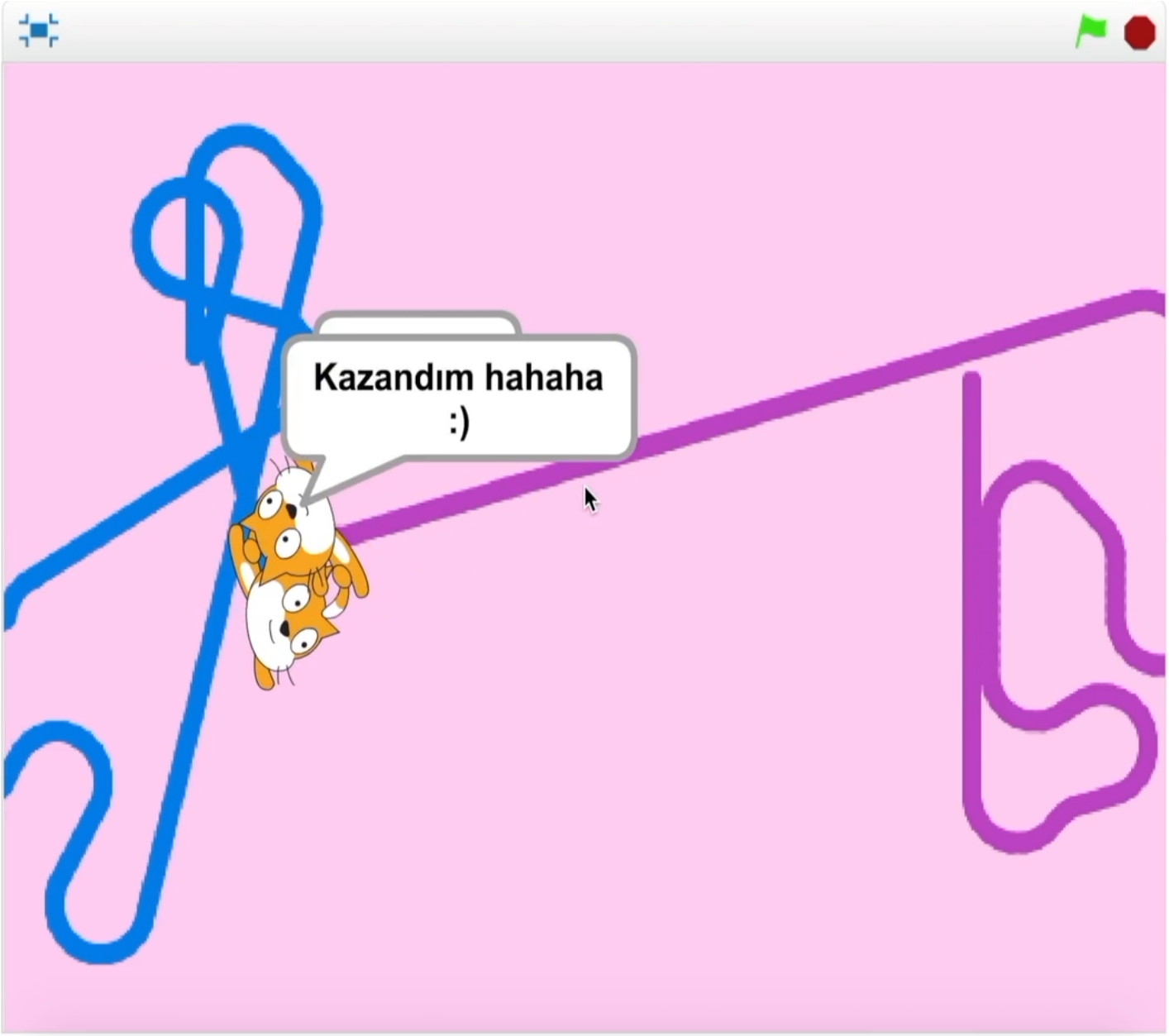
- Yaratıcılık, programlama ve bilgi işlemsel düşüncenin arttırılması ve geliştirilmesi.

İKI KIŞILIK OYUN UYGULAMASI

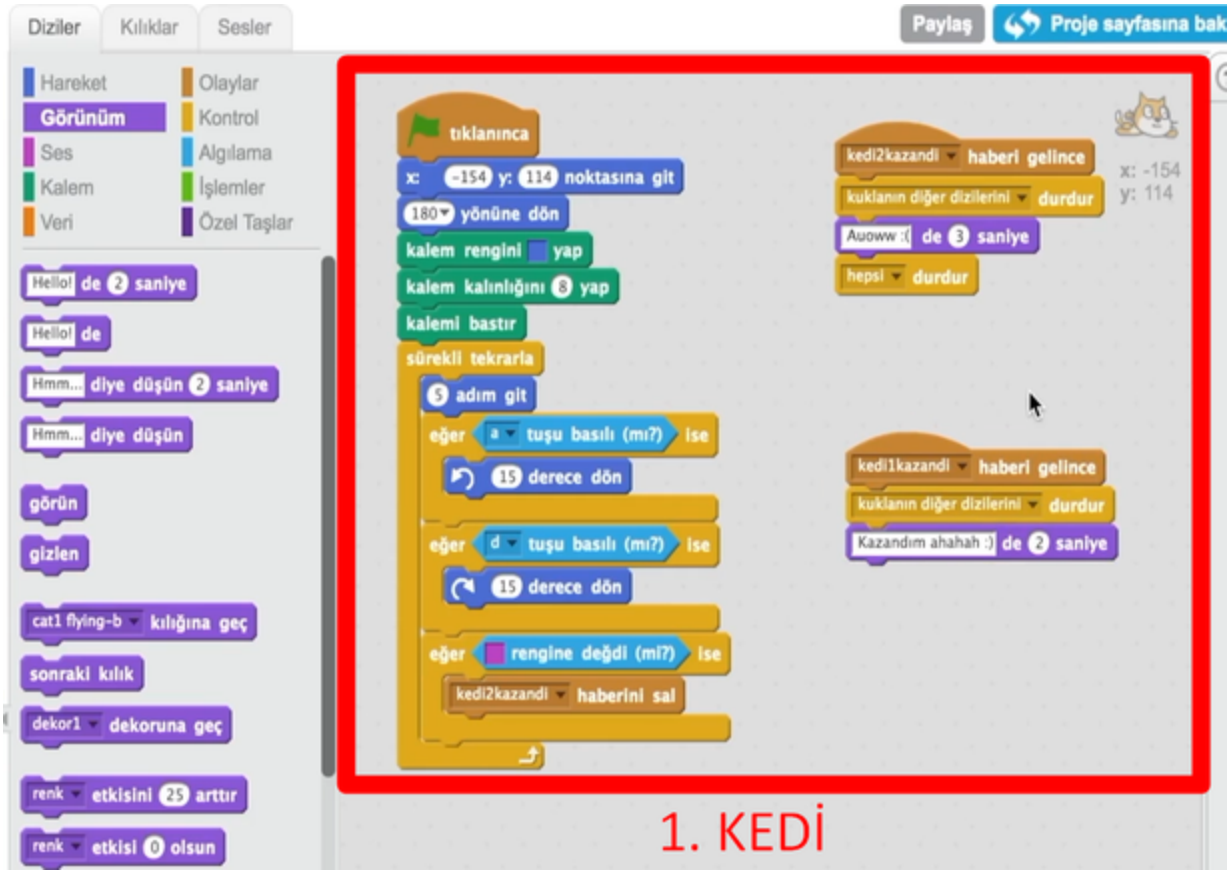
Müfredat Kodu / Ünite / Konu	İCT-103-7.7 / Uygulamalar / İki Kişilik Oyun Uygulaması
Hedeflenen Süre	30 Dakika

🚀 İki Kişilik Oyun Uygulaması

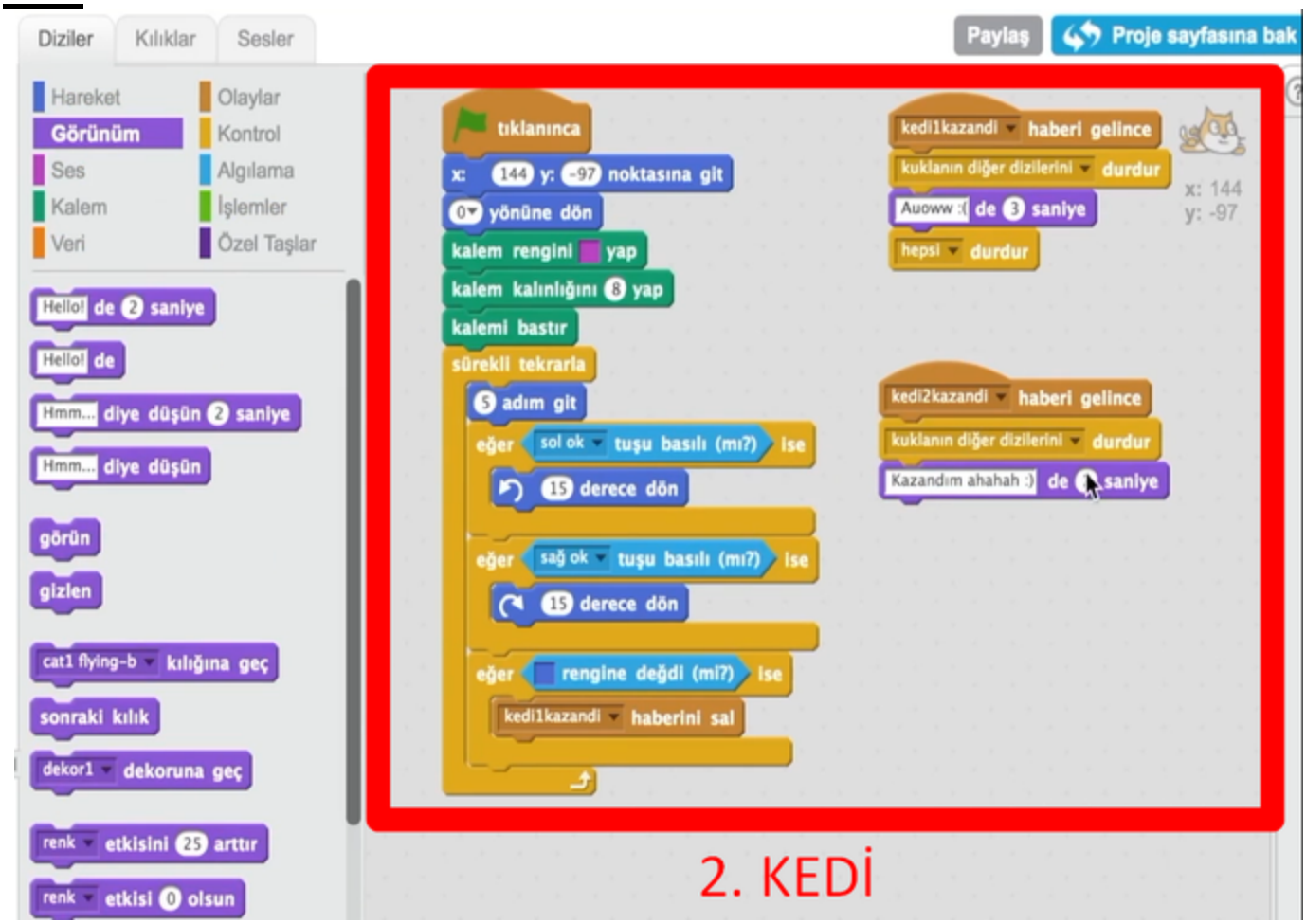
- Uygulamanın amacı; iki kişilik çizim yapan ve birbirlerinin çizgilerine değdikleri zaman puan kaybeden bir oyun geliştirmektir.



- Kütüphane bölümünden iki tane kukla seçilir.
- Kuklalara isimler verilmelidir.
- Çizgi komutunun rengi ve kalınlığı belirlenir.
- Her kukla için yapılması planlanan hareket ve diğer komut kodları her kuklanın kendi kod bölümü içine yazılır.
- **1. KEDİ:**


Diziler
Kılıklar
Sesler
Paylaş
Proje sayfasına bak
Hareket
Görünüm
Ses
Kalem
Veri
Olaylar
Kontrol
Algılama
İşlemler
Özel Taşlar
Hello! de 2 saniye
Hello! de
Hmm... diye düşün 2 saniye
Hmm... diye düşün
görün
gizlen
cat1 flying-b kılığına geç
sonraki kılık
dekor1 dekoruna geç
renk etkisini 25 arttır
renk etkisi 0 olsun
tıklanınca
x: -154 y: 114 noktasına git
180 yönüne dön
kalem rengini yap
kalem kalınlığını 8 yap
kalemi bastır
sürekli tekrarla
5 adım git
eğer a tuşu basılı (mı?) ise
15 derece dön
eğer d tuşu basılı (mı?) ise
15 derece dön
eğer rengine değdi (mi?) ise
kedi2kazandı haberini sal
kedi2kazandı haberi gelince
kuklanın diğer dizilerini durdur
Auoww :(de 3 saniye
hepsi durdur
kedi1kazandı haberi gelince
kuklanın diğer dizilerini durdur
Kazandım ahahah :) de 2 saniye
x: -154
y: 114
1. KEDİ

- 2. KEDİ:


Diziler
Kılıklar
Sesler
Paylaş
Proje sayfasına bak
Hareket
Görünüm
Ses
Kalem
Veri
Olaylar
Kontrol
Algılama
İşlemler
Özel Taşlar
Hello! de 2 saniye
Hello! de
Hmm... diye düşün 2 saniye
Hmm... diye düşün
görün
gizlen
cat1 flying-b kılığına geç
sonraki kılık
dekor1 dekoruna geç
renk etkisini 25 arttır
renk etkisi 0 olsun
tıklanınca
x: 144 y: -97 noktasına git
0 yönüne dön
kalem rengini yap
kalem kalınlığını 8 yap
kalemi bastır
sürekli tekrarla
5 adım git
eğer sol ok tuşu basılı (mı?) ise
15 derece dön
eğer sağ ok tuşu basılı (mı?) ise
15 derece dön
eğer rengine değdi (mi?) ise
kedi1kazandı haberini sal
kedi1kazandı haberi gelince
kuklanın diğer dizilerini durdur
Auoww :(de 3 saniye
hepsi durdur
kedi2kazandı haberi gelince
kuklanın diğer dizilerini durdur
Kazandım ahahah :) de saniye
x: 144
y: -97
2. KEDİ

Kazanım

- Yaratıcılık, programlama ve bilgi işlemsel düşüncenin arttırılması ve geliştirilmesi
- Oyun geliştirme kültürünün ve yeteneğinin oluşturulması

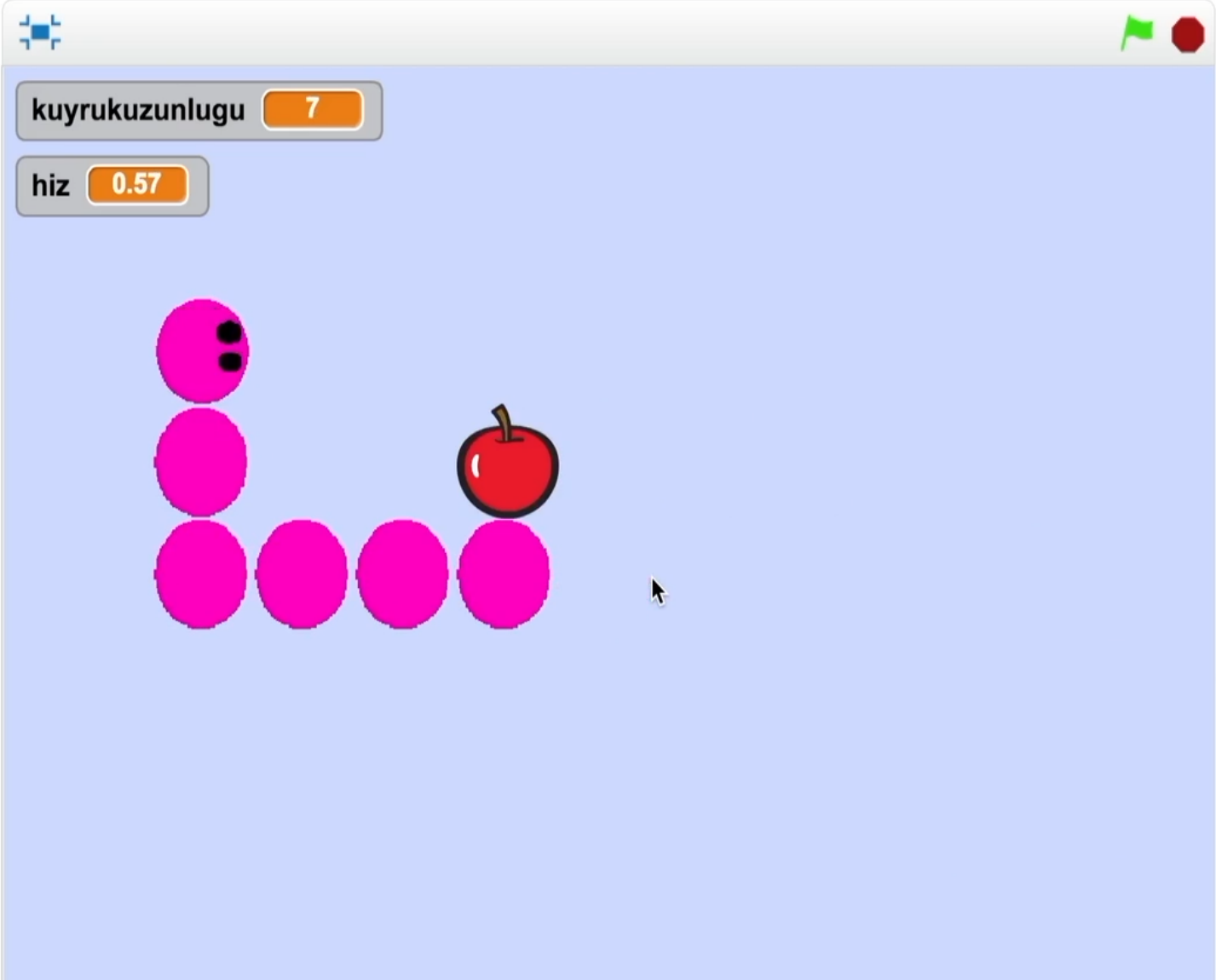
NOTLAR

YILAN OYUNU YAPIMI

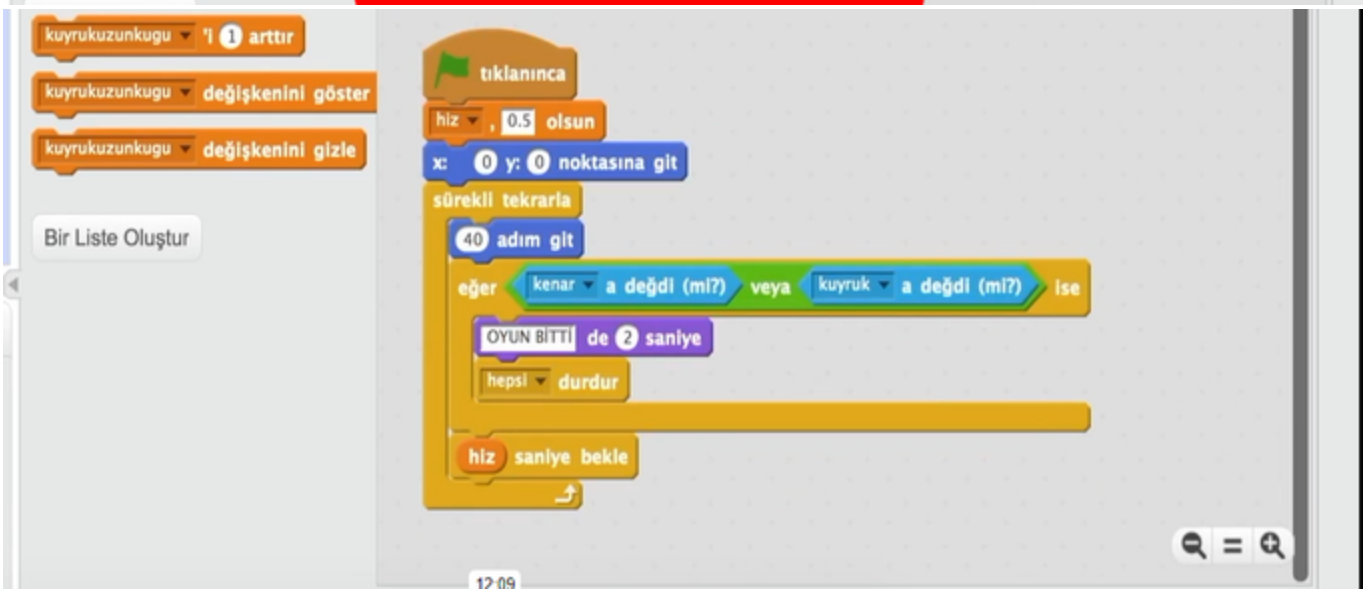
Müfredat Kodu / Ünite / Konu	İCT-103-7.8 / Uygulamalar / Yılan Oyunu Uygulaması
Hedeflenen Süre	90 Dakika

🚀 Yılan Oyunu Uygulaması

- Uygulamanın amacı, klasikleşmiş yılan oyununu yapmak. Yiyecek toplayarak yılanın büyümesi oyunu.



- Yılanın yemesi için bir tane kukla seçilir ve boyutlanması yapılır. . Örneğin elma .
- Dekor seçilir.
- Yılan için baş ve kuyruk kuklası oluşturuyoruz.
- Baş kuklası yön tuşları ile kontrol edilip sürekli hareket halinde olacak.



- Kuklamızın aşağı ve yukarı yönlerinde dönmesi için yön tuşları ekliyoruz.
- Kuyruk uzunluğu değişkeni ve hız değişkeni atıyoruz.

- Eğer kod bloğunu ekleyerek kenara değdi mi veya kuyruğa değdi mi şeklinde sorgu atıyoruz.
- Elma nesnesi için aşağıdaki kod uygulanır.

- Herhangi birine değdiği zaman ekranda 2 saniye boyunca "Oyun Bitti" yazarak oyun durdurulur.
- Yeşil bayrağa basarak oyunu başlatabiliriz.

Kazanımlar

- Bilgi işlemsel düşünce, yaratıcılık, ve programlama yeteneklerinin geliştirilmesi.

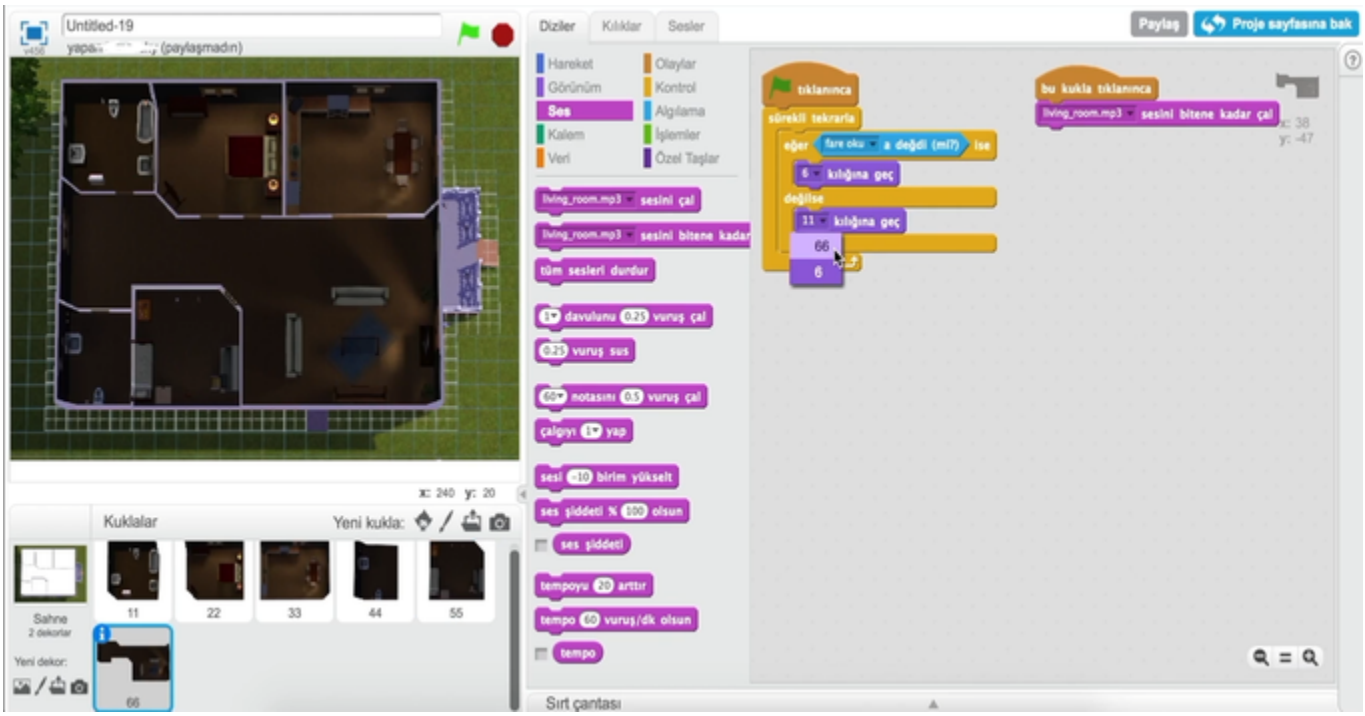
HOME SWEET HOME UYGULAMASI

Müfredat Kodu / Ünite / Konu	İCT-103-7.9 / Uygulamalar / Home Sweet Home Uygulaması
Hedeflenen Süre	45 Dakika

🚀 Home Sweet Home Uygulaması

İngilizce dersi için hazırlanmış küçük ve mantıksal bir uygulamadır. Evin bölümlerini tanıtan ve farklı kuklalar bulunuyor, her bölümün üzerine gelip tıkladığında o bölümün adını ingilizce telaffuz ediyor.

1. Kuklalar seçilir ve boyutlandırmaları yapılır.
2. Her kukla için sırası ile aşağıdaki kodlar yazılır. Kod her dekor için yazılır ve ses komutu değiştirilir.



NOT: Bu bölümde farklı dekorlarla farklı kuklalar denenerek bu uygulamanın geliştirilmesi beklenmektedir.

Kazanım

- Bilgi işlemsel düşünce, yaratıcılık ve programlama yeteneklerinin geliştirilmesi.
- Oyun geliştirme

HARITALARLA DENIZLI UYGULAMASI

Müfredat Kodu / Ünite / Konu	İCT-103-7.10 / Uygulamalar / Haritalarla Denizli Uygulaması
Hedeflenen Süre	45 Dakika

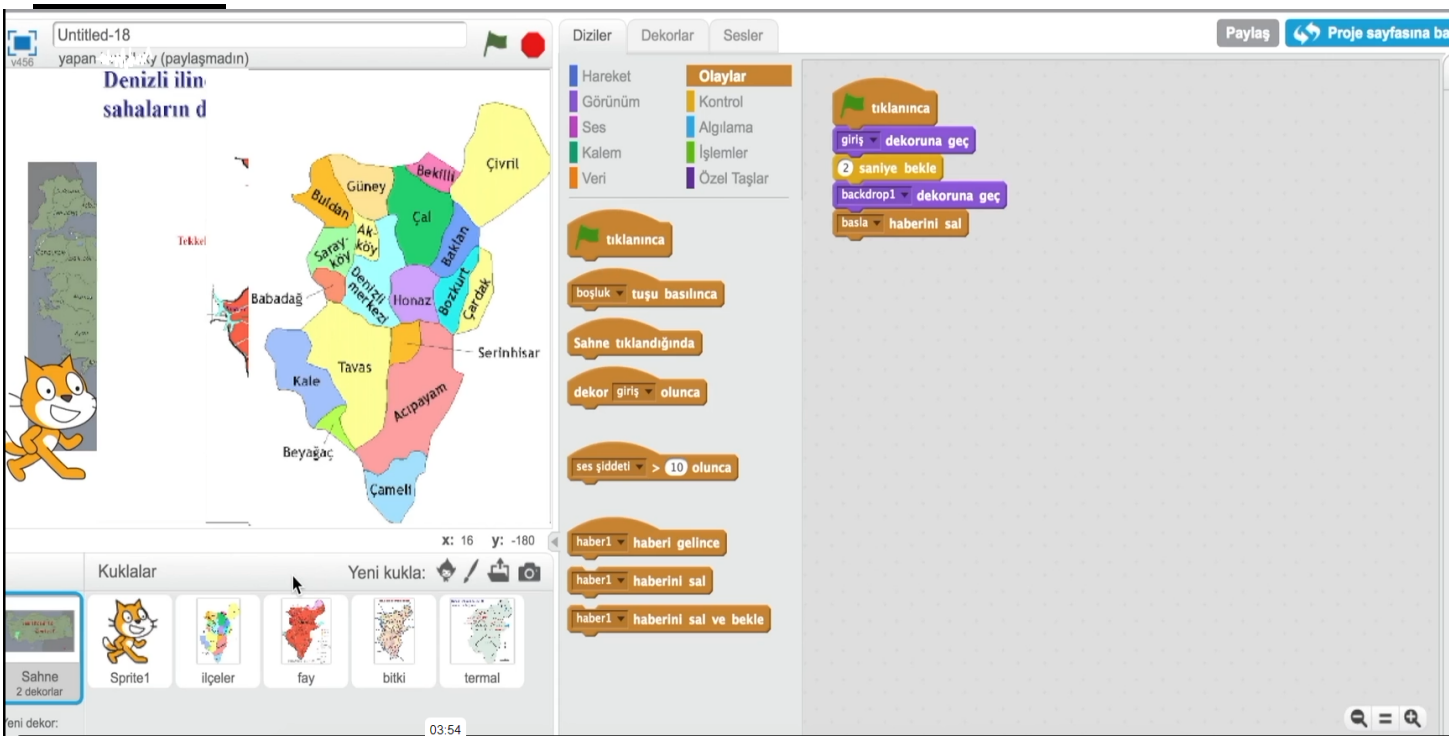
🚀 Haritalarla Denizli Uygulaması

Bu uygulamada ders içeriklerinde kullanılacak küçük ve basit yapımlı bir haritalarla Denizli uygulaması geliştirilmesi hedeflenmektedir.

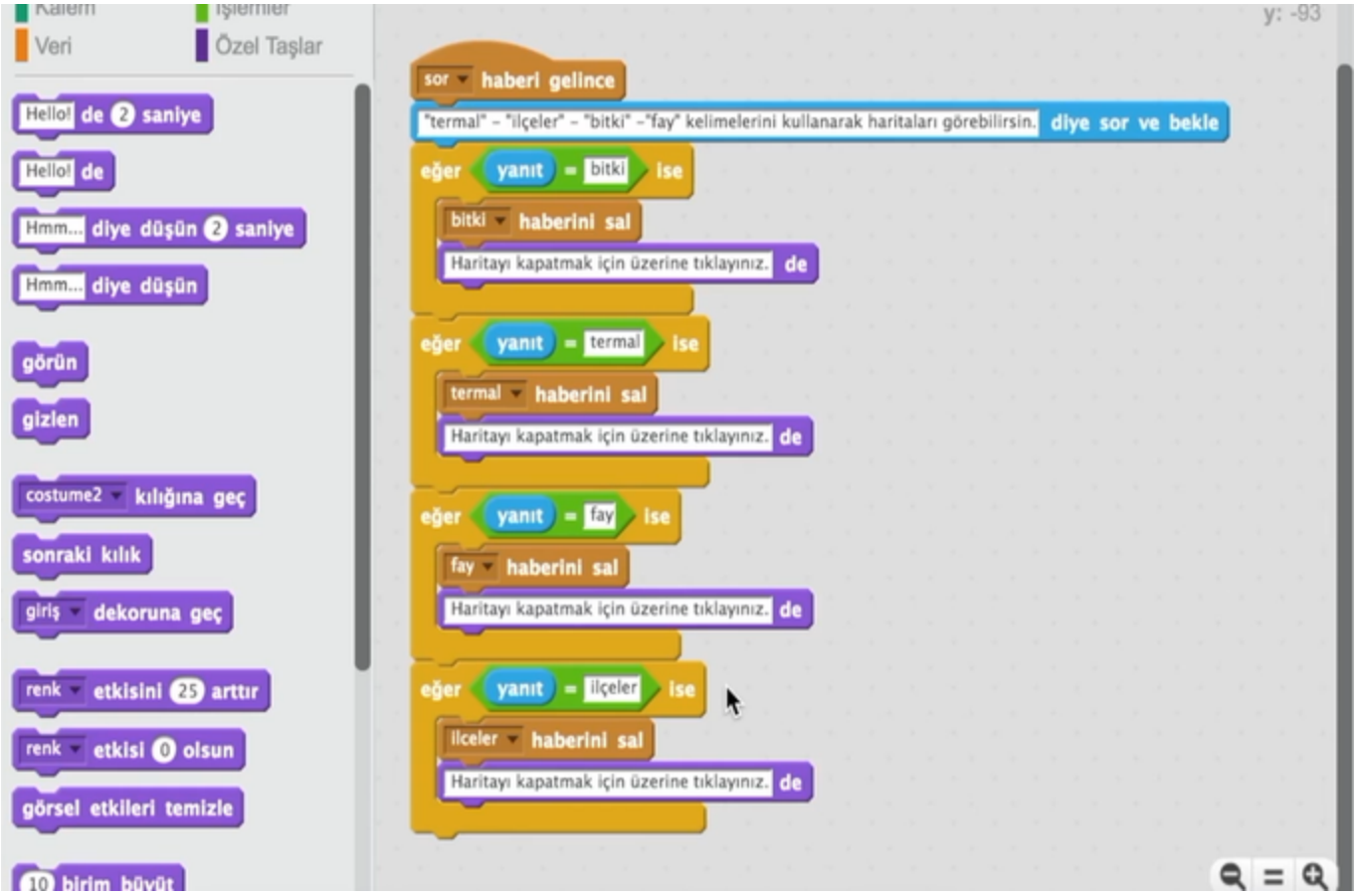
Uygulamada Denizli'ye ait “termal, ilçeler, bitki, fay hattı, ve turizm” gibi haritaları arasında geçiş yaparak kullanıcının istediği haritaya yönlendirme yapılan bir uygulamadı.

Her harita için ayrı ayrı dekor olarak yüklenmelidir.

- **Sahne Dekoru Kod:**



- **Kukla Kod:**



- Diğer haritaların her birine aşağıdaki kodlar eklenerek ana ekrana istenilen komut gönderildiğinde ilgili haritanın görünmesi ve gizlenmesi sağlanmaktadır.

NOT:

Bu bölümde uygulama istenilen başka bir şehir olabilir. Bu uygulama başka bir konu altında aynı mantıkla geliştirilebilir.

Kazanım

- Bilgi işlemsel düşünce, Yaratıcılık ve Programlama yeteneklerinin geliştirilmesi.

MAKEBLOCK MBOT EĞITIMI

Müfredat Kodu	İCT-105
Ders	ICT-105 Makeblock mBot
Ders İçeriği	• Makeblock mBot: mBot Robot Kiti • Makeblock mBot: mBot Montajı • Makeblock mBot: mBlock Programı • Makeblock mBot: mBlock Kod Blokları • Makeblock mBot: mBot ile Örnek Programlar
Seviye	İCT-S1: BAŞLANGIÇ
Ön Koşullar	mBot alanına ilgili olmak
Amaç	mBot, yeni başlayanlar için robot kodlamayı öğretmeyi ve öğrenmeyi eğlenceli hale getirmeyi amaçlar.
Hedef	Robotik alana ilgi duyan bireylerin, öğrenim alanındaki konuları kolaylıkla öğrenmesini ve uygulamasını amaçlar.
Gereksinimler	Bilgisayar, Ortamda İnternet Ağı, Mail adresi
Kazanım	• Robot programlamayı öğretmek, • Analitik düşünme yeteneğini geliştirmek, • Ekip çalışmasını öğretmek, • Odaklanma ve dikkat süresini arttırmak.
Referanslar & Dokümantasyon	• https://maker.robotistan.com/stem-nedir-mbot-ile-stem-egitimine-baslayin/ • https://www.robotakademi.org/makeblock-mbot-robotik-ve-kodlama-egitimi/ • https://tr.wikipedia.org/wiki/MBlock • https://robotikkodlama.net/mbot-montaji-ve-sarj/ • https://www.protopars.com/mblock-nedir-kimler-nasil-kullanabilir/ • http://www.dudigan.com/2017/10/mblock-ile-arayuz-tanitimi/ • https://gelecegikodlayanlar.org/cizgi-izleyen-robot
Durum	BEKLIYOR / UYGULANIYOR / TAMAMLANDI / DEĞERLENDIRILDI
Hedeflenen Süre	540 dakika
Eğitim Günleri	Başlama: .../.../... Bitiş: .../.../...

MAKEBLOCK MBOT: MBOT ROBOT KITI

Müfredat Kodu / Ünite / Konu	İCT-105-1 / Makeblock mBot / mBot Robot Kiti
Hedeflenen Süre	90 Dakika

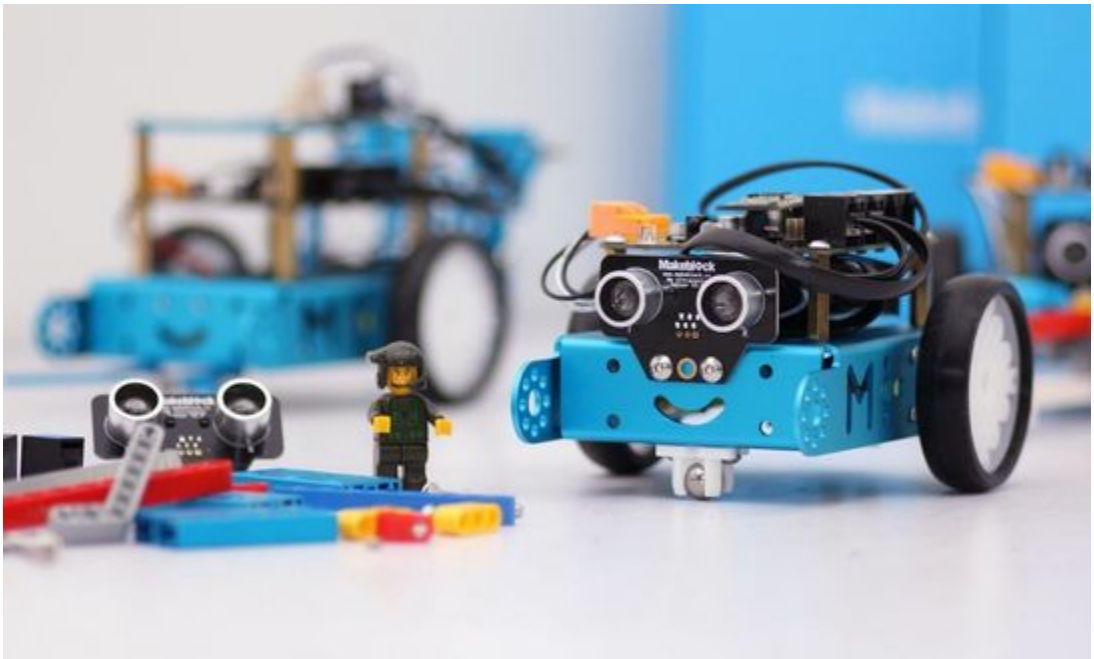
🚀 MakeBlock mBot Robot Kiti Nedir ?

mBlock, blok tabanlı yapısı sayesinde, herhangi bir programlama dilini bilmeksizin, kod bloklarını yap boz birleştirir gibi bir araya getirme ve basit oyunlar geliştirme olanağı sağlar. Özellikle 6-10 yaş arası çocuklara yönelik bir araç olan mBlock ile görerek ve sürükleyip bırakarak, robotlardan özel devre kartlarına kadar birçok şeyi programlayabilirsiniz.

🧐 mBot ile Neler Yapılabilir ?

- Duvardan kaçan bir robot inşa edilebilir,
- Çizgi izleyen robot yapabilirsiniz,
- Standart TV kumandası veya uyumlu uzaktan kumanda ile robot kontrolü yapabilirsiniz,
- Scratch ile etkileşim yaptırarak, Scratch programındaki senaryo ve oyunları uygulatabilirsiniz.
- App ile futbol oynatabilirsiniz.
- Diğer mBot'lar ile haberleşmesini sağlayarak koordine hareketler yapmasını sağlayabilirsiniz.

📸 MakeBlock mBot



Kazanım

- Scratch programlama diline hâkimiyet kazandırır.
- Proje tabanlı öğrenme yardımıyla sorun gidermeyi öğretir.
- Sonralık – öncelik kavramı geliştirme konularında bireye kazanım sağlar.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Kurulumun yapılması. ☑ Öğrencilere Robot Kitinin tanıtılması.	☑ Eğitmen ☐ Öğrenciler	Eğitmenler tarafından program kurulumlarının tamamlanması. Öğrencilere uygulama üzerindeki kavramların tanıtılması.
2	☑ Uygulamaya erişim sağlamak. ☑ Kavramların pekiştirilmesi.	☐ Eğitmen ☑ Öğrenciler	Eğitmenler tarafından kurulumu sağlanan uygulamaya erişim sağlamak. Uygulama içerisindeki kavramların pekiştirilmesi.

Eğitmen Yol Haritası

1	Öğrencilere mBot robot kitinin tanıtılması.
2	mBot robot kiti ile neler yapılabileceğinin anlatılması ve videolarla anlatımın desteklenmesi.

MAKEBLOCK MBOT: MBOT MONTAJI

Müfredat Kodu / Ünite / Konu	İCT-105-2 / Makeblock mBot / mBot Montajı
Hedeflenen Süre	120 Dakika

🚀 Mbot Montajı Nasıl Yapılır ?

mBot'un montajını aşağıdaki 10 adımı takip ederek gerçekleştirebilirsiniz. mBot ile birlikte detaylı montaj kılavuzu da satın alınacak sete ait kutuda yer almaktadır.

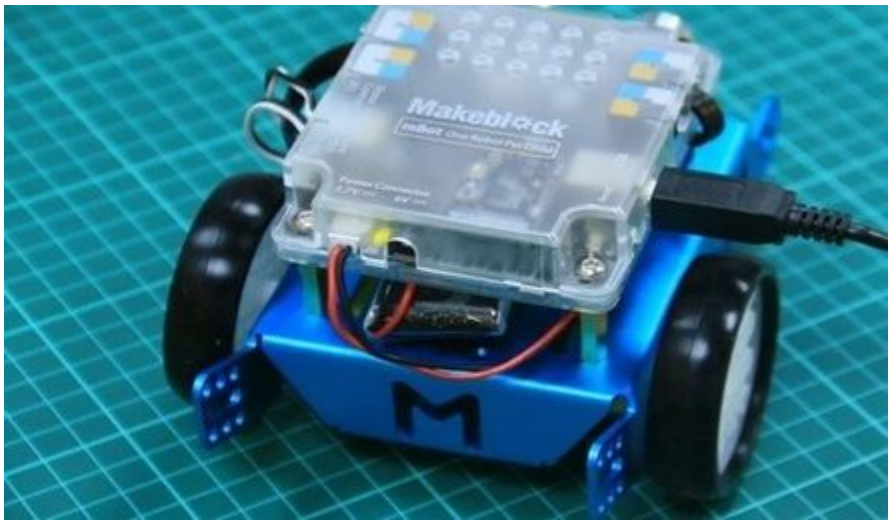
- Gövdeye motorları vida ve somunlar ile sabitleyin.
- Tekerlekleri motorlara geçirip vidalayın.
- Tekerleklere lastikleri geçirin.
- Çizgi izleme sensörünü gövdeye monte edin.
- Ultrasonik sensörü gövdeye monte edin.
- Sensörlere kabloları bağlayın ve kabloları gövdenin üst kısmına geçirin.
- Ana kartı taşıyacak yükseltici ayakları gövdeye monte edin.
- Pil yuvasını (ya da LiPo pil kullanıyorsanız LiPo'nun kendisini) gövdenin üst kısmına sabitleyin.
- 4 adet vida ile ana kartı ayaklara sabitleyin.
- Kabloları ve kablosuz haberleşme modülünü uygun konektörlere takın.

🧐 mBot Güç Bağlantısı Nasıl Yapılır ?

mBot'unuzu isterseniz 4 adet kalem pil ile, isterseniz mBot uyumlu LiPo batarya ile çalıştırmanız mümkündür.
LiPo bataryayı, aşağıda görmüş olduğunuz sokete takabilirsiniz. Konektör üzerinde yer alan çentik sayesinde ters takmanın ihtimali yoktur. LiPo batarya, mBot'unuza USB kabloyu bağladığınızda otomatik olarak şarj olacaktır. Şarj durumunu kart üzerindeki kırmızı ve yeşil renkli LED'lerden görebilirsiniz.

Şarj esnasında kırmızı renkli LED yanıyor.

Şarj tamamlanınca yeşil renkli LED yanacaktır.



Kazanım

- Bu ünite bireye donanımsal ve elektronik becerilerini geliştirmede yardımcı olmayı hedefler.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Öğrencilere mBot Montajının anlatılması.	☑ Eğitmen ☐ Öğrenciler	Eğitmen tarafından öğrencilere mBot montajının nasıl yapılacağı hakkında tanıtım yapılması.
2	☑ Montaja başlanması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen eşliğinde montaj işlemine başlanması.

Eğitmen Yol Haritası

1	Öğrencilere mBot montajında kullanılacak malzemelerin tanıtılması.
2	mBot montajının nasıl yapılacağının anlatılması ve kaynak video ile öğrencilere gösterilmesi.
3	Öğrencilerle birlikte adım adım montaj işlemine başlanması.

MAKEBLOCK MBOT: MBLOCK PROGRAMI

Müfredat Kodu / Ünite / Konu	İCT-105-3 / Makeblock mBot / mBlock Programı
Hedeflenen Süre	30 Dakika

🚀 mBlock Programı Nedir ?

mBlock, blok tabanlı kodlama ortamıdır. Çalışma mantığı Scratch gibidir. Sürükle bırak yöntemi ile kod bloklarını yan yana, alt alta, iç içe yerleştirerek büyük kod blokları oluşturup bu kod blokları ile karakteri istediğimiz gibi hareket ettirebilmemize yardımcı olur.

mBlock'un Scratch'ten farkı ise bu oluşturulan kod bloklarını robot kitlerine göndererek somut bir şekilde robotun hareket etmesini sağlamaktır.

👷 mBlock Nasıl İndirilir ?

mBlock' un 2 farklı sürümü yer almaktadır. Bunlardan ilki ve hala kullanılmakta olan mBlock 3 ve diğeri ise hala geliştirme aşamasında olan mBlock 5 Beta sürümüdür.

mBlock'u bilgisayarımıza indirebilmek için kendi sitesine giriş yapıyoruz. Site burada bize tarayıcıda çalıştırmak veya uygulama olarak indirmek için seçenekler sunuyor.

Tarayıcı üzerinden her iki sürüm için kodlama yapılabiliyor fakat mBlock 3'te Adobe Flash yüklemesi isteniyor.

mBlock 5 ise HTML 5'in gerektirdiği gibi herhangi bir ek programa gerek kalmadan tıklanıldığı gibi çalışmaya başlıyor.

Bu uygulama bizlere iki farklı çalışma ortamı sağlamaktadır. Bu çalışmalardan biri interaktif uygulama diğeri ise çevrim dışı uygulamadır.

Yazılımı çalıştırdıktan sonra etkileşim içerisinde olduğumuz kodları oluşturduğumuz kısım interaktif kısımdır. Bu mod üzerinde bir sahne, sahnenin içerisinde birçok farklı karakter de vardır. Bu karakter oluşturduğumuz kod bloklarına göre hareket etmektedir.

Diğer mod olan çevrim dışı mod da ise yazdığımız kod bloklarını arduinonun belleğine yükleriz ve böylece bilgisayardan bağımızı koparmış oluruz.

📘 Kazanım

- Bireylerin kodlama ve algoritma mantığının gelişmesine yardımcı olur.

✅ Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Program kurulumunun yapılması. ☑ Öğrencilere mBlock programının tanıtılması.	☑ Eğitmen ☐ Öğrenciler	Eğitmenler tarafından program kurulumunun yapılması ve öğrencilere mBlock programının tanıtılması.
2	☑ Uygulamaya erişim sağlanması.	☐ Eğitmen ☑ Öğrenciler	Eğitmenler tarafından kurulumu sağlanan uygulamaya erişim sağlanması ve uygulamanın öğrenci tarafından keşfedilmesi.

👩‍🏫 Eğitmen Yol Haritası

1	Öğrencilerin derse katılım sağlayacağı cihazlara mBlock programı kurulmalıdır.

2	Kurulan mBlock programı öğrencilere tanıtılmalı, eğer bulunabilirse, programın nasıl kullanılacağına dair kaynak video izletilmelidir.

NOTLAR

MAKEBLOCK MBOT: MBLOCK KOD BLOKLARI

Müfredat Kodu / Ünite / Konu	İCT-105-4 / Makeblock mBot / mBlock Kod Blokları
Hedeflenen Süre	120 Dakika

🚀 mBlock Kod Blokları

mBlock programı ile kodlama yaparken kullanacağımız temel kısım arayüzün sol tarafında bulunan Diziler bölümüdür. Bu bölüm verilerin kullanılması, değişkenler, kontrol yapıları, Arduino'nun temel fonksiyonlarıyla birlikte ekleyeceğimiz kütüphanelerin bulunduğu ve en çok kullanılacak olan kısımdır. Bu bölümde bulunan sekmeler aşağıda listelenmiştir.

Veri&Blok:

Veri&Blok sekmesi altında değişken oluşturma ve blok oluşturma seçenekleri bulunmaktadır. Değişken oluştur seçeneği ile çeşitli amaçlarla kullanacağımız verilerin kaydedilebileceği değişken adında isimlerini bizim belirlediğimiz hafıza alanları tanımlayabiliriz. Blok oluştur seçeneği ile çeşitli görevleri yerine getirecek, işlevini tamamen bizim belirlediğimiz bloklar oluşturabiliriz.

Kontrol:

Kontrol sekmesi altında bulunan işlevler, kodlamanın temelini teşkil eden kontrol yapılarının bulunduğu alandır. Bu alanda, kodlamada kullanılan belli şartlara bağlı veya herhangi bir şarta bağlı olmaksızın çalışan döngüsel yapılar ile zaman blokları bulunmaktadır.

İşlemler:

İşlemler sekmesinde bulunan bloklar, mantıksal ve aritmetik işlemlerin bulunduğu sekmedir. Burada bulunan bloklar ile çeşitli matematiksel işlemler, mantıksal koşullar oluşturabiliriz.

Robotlar:

Robotlar sekmesinde bulunan bloklar ile temel Arduino fonksiyonlarına erişebiliriz. Bu bölüm doğrudan Arduino platformuna yönelik kod bloklarını barındırmaktadır. Örneğin Arduino'dan bilgi okuma, Arduino'ya veri gönderme, Ekrana bir şeyler yazdırma gibi pek çok blok

bulunmaktadır. Ayrıca uzantılar menüsünden uzantı ekle seçeneği ile Arduino ile kullanılabilecek birçok kütüphane ile çeşitli özellikler de eklenebilmektedir.

Kazanım

- Bu bölümde bireylerin robotik kodlama alanında kullanacağı blokları tanıtmak amaçlanmıştır.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Kod bloklarının tanıtılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere program arayüzü ve kod bloklarının tanıtılması.
2	☑ Program arayüzünün keşfedilmesi.	☐ Eğitmen ☑ Öğrenciler	Öğrenciler tarafından program arayüzünün ve blokların keşfedilmesi.

Eğitmen Yol Haritası

1	Öğrencilere arayüz üzerindeki Diziler panelinde bulunan kod bloklarının tanıtılması.

2	Tanıtılan blokların kullanılarak öğrencilerin anlamasını kolaylaştırmak adına eğitmen tarafından uygulamaların yapılması.
3	Öğrencilere tanıtılan bloklar hakkında soruların sorulup, konunun anlaşılabilirliğinin ölçülmesi.
4	Öğrencilere, yukarıda tanıtılan bloklar hakkındaki uygulamaların ödevlendirme yapılması.

NOTLAR

MAKEBLOCK MBOT: MBOT ILE ÖRNEK PROGRAMLAR

Müfredat Kodu / Ünite / Konu	İCT-105-5 / Makeblock mBot / mBot ile Örnek Programlar
Hedeflenen Süre	180 Dakika

🚀 Çizgi İzleyen Robot

Çizgi izleyen robotların çok geniş kullanım alanları vardır. Gelişmiş bir çizgi izleyen robot, endüstriyel olarak sürekli bir yolda malzeme taşımak gibi amaçlar için kullanılabilmektedir. Bizim yapacağımız proje ise hobi amaçlı olan ve daha çok yarış robotları olarak kullanılan çizgi izleyen robotlara bir giriş ve robotik mantığını, algoritmasını kavramayı amaçlayan bir proje olacaktır.

✍️ Gerekli Malzemeler

- Bilgisayar
- mBot
- mBlock 3 programı
- USB kablo
- Siyah elektrik bandı

💻 Uygulama

- Öncelikle masaya ya da yere siyah elektrik bandı ile bir yol çiziyoruz. Belirli varış noktaları ya da bir labirent oluşturabiliriz. Bunların dışında mBot robot kiti içerisinde gelen çizgi izleyen yolu da kullanabiliriz.
- Aşağıda örnek bir görsel görebilirsiniz.

- Alanımızı oluştuktan sonra mBlock programını açabiliriz. Eğer program açıksa üst kısımda bulunan “Dosya” sekmesinden “Yeni” kısmına tıklayarak yeni bir proje açabiliriz.
- mBot’u USB kablo ile bilgisayara bağlıyoruz ve “Kartlar” sekmesinden “mBot(mCore)”, “Uzantılar” sekmesinden ise “Makeblock” un seçeneğinin seçili olduğundan emin oluyoruz. Daha sonra “Bağlan” sekmesinden “Seri Port” kısmına tıklıyoruz ve açılan kısımdan seri portumuzu seçiyoruz. Böylece mBot’ u kullanıma hazırlamış oluyoruz.
- Böylece kodlarımızı yazmaya başlayabiliriz.
- Öncelikle robotumuzun çizgi gördüğünde yada çizgi yokken verdiği değerleri görmek için küçük bir kod yazıyoruz. Bunun için “Görünüm” dizisinden “merhaba de 2 saniye” kod bloğunu alarak, içerisine “Robotlar” dizisinden “Kapı_ çizgi izleyen” bloğunu alarak “merhaba” yazısı üzerine bırakıyoruz. Böylece gelen değerleri ekranımızda görebiliriz.
- Gerekli kod blokları aşağıda detaylı olarak gösterilmiştir.

- Gelen değerlere göre robotumuzun dönmesi gereken yönleri belirliyoruz.
  - 0 : hem sağ hem sol

- 1 : sol
- 2 : sağ
- 3 : boşluk

- Değerleri belirledikten sonra, artık robotumuzun çizgi izlemek için gereken kodlarını yazabiliriz. Öncelikle sensörden gelen değerleri yazabileceğimiz bir değişken oluşturacağız. “Veri&Blok” dizisinde bulunan “Bir Değişken Oluştur” butonuna basarak “cizgizleyen” isminde yeni bir değişken oluşturuyoruz. Daha sonra bu değişkenin içerisine “Robotlar” dizisinden “Kapı _ çizgi izleyen” kod bloğunu alarak gelen değerleri değişkenin içerisine aktarmış oluyoruz.
- Artık gereken koşul ifadelerimizi ekleyebiliriz.
- “Kontrol” dizisinden “eğer ise _ , değilse _” kod bloğunu alarak içerisine “İşlemler” dizisinden “=” (eşittir) bloğunu alıyoruz. Bu kod bloğunun bir tarafına, “Veri&Blok” dizisinde bulunan “cizgizleyen” bloğunu, diğer tarafa ise üst kısımda belirlediğimiz dönüş yönünü yazarak koşulumuzu oluşturuyoruz. Buna göre eğer gelen değer “ 0 “ ise “Robotlar” dizisinden “ileri git hızı 0 yap” kod bloğunu alarak, hız ayarlamasını yapıyoruz ve sensörden gelen değer “0” olduğunda robotumuz ileri gitmesini sağlıyoruz. “değilse” kısmımıza ise bir diğer gelen değerimize göre yön belirlemelerini yaparak robotumuzu hazır hale getiriyoruz.
- Gerekli kod blokları aşağıda detaylı olarak gösterilmiştir.

- Kodlama kısmımızı tamamladıktan sonra usb kablosu veya wifi yardımı ile mBlock programında yazdığımız kodları mBot robotuna aktarabiliriz. Kodumuzu yükledikten sonra artık robotumuzu hazırladığımız çizgi izleyen pistine yerleştirerek denemelerini gerçekleştirebiliriz.

Kazanım

- Bu ünitede öğrenci robot programlamayla birlikte robot ve sensörlerin çalışma mantığını kavrayacaktır.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar

1	☑ Uygulamanın tanıtılması.	☑ Eğitmen ☐ Öğrenciler	Eğitmen tarafından uygulamanın ve uygulamada kullanılacak malzemelerin tanıtılması.
2	☑ Uygulama yapımına başlanması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen gözetiminde öğrencilerin uygulama yapımına başlaması.

Eğitmen Yol Haritası

1	Öğrencilere kullanılacak malzemelerin tanıtılması.
2	Uygulamaya başlamadan önce uygulamanın nasıl yapılacağının anlatılması, uygulamanın yapılışını anlatan video kaynakları ile dersin desteklenmesi.
3	Uygulamanın öğrencilerle birlikte eş zamanlı olarak yapılması.

ROBOTIK EĞITIM MÜFREDATI

ROBOTİK EĞİTİM MÜFREDATI																				
EĞİTİM İÇERİĞİ / HAFTA	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.	16.	17.	18.	19.	20.
Temel Tanımlar ve Kullanımlar (İCT-700-1)																				
Geliştirme Kartları ve Arduino																				
Arduino Kart Çeşitleri ve Özellikleri																				
Arduino IDE Kurulumu ve Tanıtımı																				
Arduino IDE Programlama Dili ve Özellikleri																				
Breadboard Kullanımı																				
Sensörler ve Uygulamaları (İCT-700-2)																				
Sensörler																				
Blink Uygulaması																				
Kara Şimşek Uygulaması																				
Butondan Değer Okuma Uygulaması																				
Potansiyometreden Değer Okuma Uygulaması																				
LDR Değer Okuma Uygulaması																				
Kara Şimşek Uygulaması - For Döngüsü																				
LDR ile Led Yakma Uygulaması (If-else)																				
Potansiyometre ile Led Yakma Uygulaması																				
Park Sensörü Uygulaması																				
Ekran Kullanım Uygulaması																				
Servo Motor Uygulaması																				
DC Motor Uygulaması																				
Proje Geliştirme (İCT-700-3)																				
Engelden Kaçan Robot Yapımı																				
Robot Kol Yapımı																				
Bluetooth İle Araç Kontrolü																				
Akıllı Ev Uygulaması																				

ROBOTIK

Müfredat Kodu	İCT-700
Ders	Robotik
Ders İçeriği	• Temel Tanımlar ve Kullanımlar • Sensörler ve Uygulamaları • Proje Geliştirme
Seviye	İCT-S2: ORTA
Ön Koşullar	Bu alanda istekli ve meraklı olmak.
Amaç	Ders içeriği ile temel sensörleri kullanarak kodlama ve günlük hayattaki kullandığımız teknolojilerin donanımsal ve programsal faaliyetlerini anlamak, yorumlamak ve uyguluyor olmak amaçlanmıştır.
Hedef	Günümüz teknolojisini kullanabiliyor , programlayabiliyor ve yorumluyor olabilmeyi hedeflemektedir.
Gereksinimler	• Arduino seti • Atölye malzemeleri. • Bilgisayar
Kazanım	• Programlama • Elektronik bağlantıların uygulanması • Devre kurabiliyor ve geliştirme kartını programlıyor olabilmek • Sorunlara çözüm üretmek • Ekip çalışması
Durum	BEKLİYOR / UYGULANIYOR / TAMAMLANDI / DEĞERLENDİRİLDİ
Hedeflenen Süre	1800 Dakika
Eğitim Günleri	Başlama: .../.../.... Bitiş: .../.../....

TEMEL TANIMLAR VE KULLANIMLAR

Müfredat Kodu	İCT-700-1
Ders	Temel Tanımlar ve Kullanımlar
Ders İçeriği	• Geliştirme Kartları ve Arduino • Arduino Kart Çeşitleri ve Özellikleri • Arduino IDE Kurulumu ve Tanıtımı • Arduino IDE Programlama Dili ve Özellikleri • Breadboard Kullanımı
Seviye	İCT-S2: ORTA
Ön Koşullar	Arduino alanında öğrenme isteği
Amaç	Günümüzde birçok alanda kullanılan geliştirme kartları hakkında bilgi sahibi olarak Arduino ve yapılan çalışmalar hakkında bilgi sahibi olunması amaçlanmaktadır.
Hedef	Arduino, çeşitleri, özellikleri ve konu ile ilgili temel kavramların öğrenilmesi hedeflenmektedir.
Gereksinimler	Arduino Seti, Programlama için Bilgisayar
Kazanım	• Arduino geliştirme kartı ve özelliklerinin öğrenilmesi • Yapılabilecek projelerin keşfedilmesi • Konu ile ilgili kavramların öğrenilmesi
Referanslar & Dokümantasyon	• Geliştirme Kartları • https://tr.wikipedia.org/wiki/Arduino • https://www.arduino.cc/
Durum	BEKLİYOR / UYGULANIYOR / TAMAMLANDI / DEĞERLENDİRİLDİ
Hedeflenen Süre	180 Dakika
Eğitim Günleri	Başlama: .../.../.... Bitiş: .../.../....

Geliştirme Kartları ve Arduino

Müfredat Kodu / Ünite / Konu	İCT-700-1.1 / Temel Kavramlar ve Kullanımlar / Geliştirme Kartları ve Arduino
Hedeflenen Süre	30 Dakika

Geliştirme Kartları Nelerdir?

Geliştirme kartları bir ürünün üretilmeden önce kolayca geliştirilmesini sağlayan portatif elektronik kartları ifade etmektedir. Geliştirme Kartları kısaca; işlemci, bellek ve depolama donanımlarından en az birine sahip devre kartlarıdır.

> *Eğitmenin bu konu başlığında geliştirme kartlarını görsel olarak anlatması tavsiye edilmektedir.*

Geliştirme Kartları Çeşitleri ve Kullanım Alanları

Kullanıcıların proje türüne ve ihtiyaçlarına göre piyasada birçok geliştirme kartı yer almaktadır. Kullanıcı projesinin niteliğine göre (Proje kapasitesi, kullanılacak program dili yetkinliği vs.) geliştirme kartları piyasadan temin edilebilmektedir.

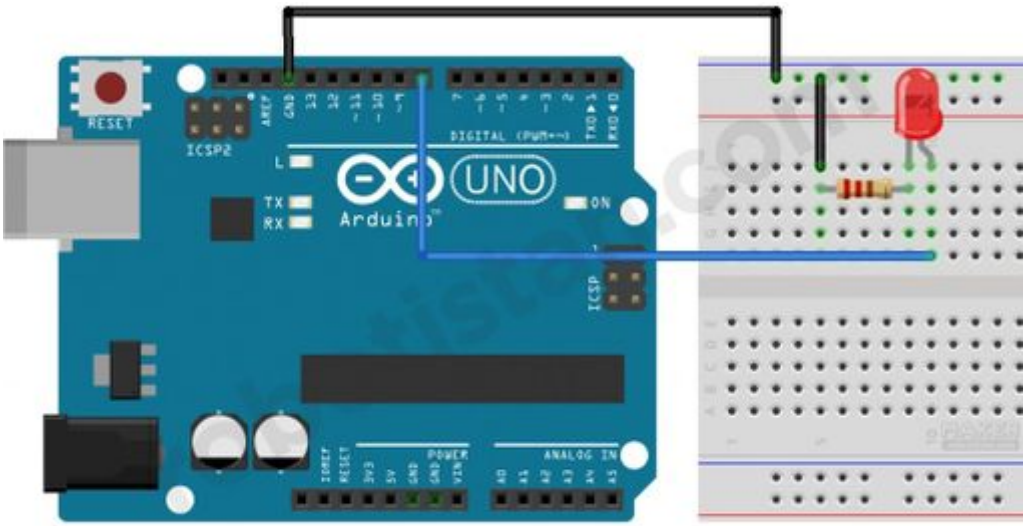
En yaygın kullanılan Geliştirme Kartları Arduino, Raspberry Pi, Deneyap Kartı, Discovery, Banana Pi, Android Geliştirme Kartı ...

> *Eğitmenin bu konu başlığında geliştirme kartlarını ve örnek projelerde kullanımlarını görsel ve video ile desteklemesi tavsiye edilmektedir.*

Arduino Nedir?

Günümüzde en çok kullanılan geliştirme kartı olarak kabul edebileceğimiz Arduino Açık kaynak donanım'dır. Donanım referans tasarımları Creative Commons Attribution Share-Alike 2.5 lisansı altında dağıtılır ve Arduino web sitesinde mevcuttur. Donanımın bazı sürümleri için düzen ve üretim dosyaları[1] da mevcuttur.

Arduino kartları bir Atmel AVR mikrodenetleyici (Eski kartlarda ATmega8 veya ATmega168, yenilerinde ATmega328 ya da ATmega4809[2]) ve programlama ve diğer devrelere bağlantı için gerekli yan elemanlardan oluşur. Her kartta en azından bir 5 voltluk lineer regülatör ve bir 16 MHz kristal osilator (bazılarında seramik rezonatör) bulunur. Mikrodenetleyiciye önceden bir bootloader programı yazılı olduğundan programlama için harici bir programlayıcıya ihtiyaç duyulmaz.



Arduino Kullanım Alanları

Arduino genellile okul projeleri , prototipleme projeleri ve endüstriyel uygulamalarda kullanılmaktadır.

Drone,3 Boyutlu Yazıcı, Robot Kol, Hırsız Alarmı, Uzaktan Kumandalı Araç, Otonom Araç Projesi ...

Eğitmenin bu konu başlığında Arduino ile yapılan projeleri video ile göstermesi tavsiye edilmektedir.

Kazanım

- Geliştirme kartları ve modellerinin öğrenilmesi.
- Geliştirme kart çeşitleri ile birlikte yapılan projelerin aktarılması.
- Arduino geliştirme kartı ve kullanım alanlarının öğrenilmesi.
- Yapılması planlanan projeler için ufuk açılması.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Geliştirme Kartları ve Arduino	☑ Eğitmen ☐ Öğrenci	Arduino dersine giriş ve kavramların öğrencilere tanıtılması beklenmektedir.
2	☑ Geliştirme Kartları ve Arduino	☐ Eğitmen ☑ Öğrenci	Arduino çeşitleri ve özellikleri hakkında bilgi sahibi olunması hedeflenmektedir.

Arduino Kart Çeşitleri ve Özellikleri

Müfredat Kodu / Ünite / Konu	İCT-700-1.2 / Temel Kavramlar ve Kullanımlar / Arduino Kart Çeşitleri ve Özellikleri
Hedeflenen Süre	30 Dakika

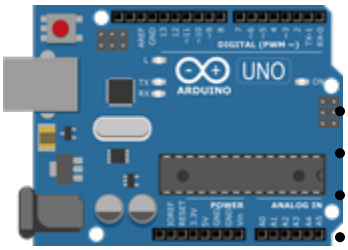
Arduino Geliştirme Kart Çeşitleri

Arduino geliştirme kartının kendi içerisinde bir çok çeşidi bulunmaktadır.

Arduino UNO, Arduino MEGA, Arduino NANO, Arduino LİLYPAD, Arduino MİNİ, Arduino LEONARDO, Arduino ETHERNET

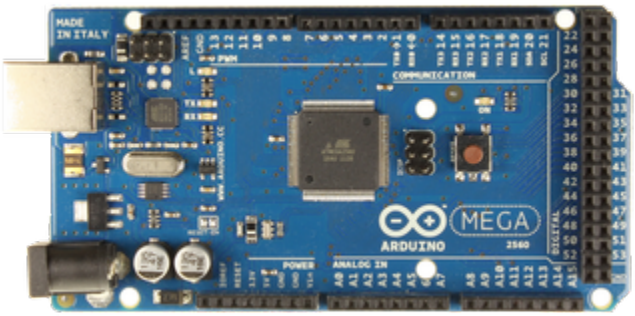
> *Eğitmenin bu bölümü görsel ögelerle desteklemesi tavsiye edilmektedir.*

Arduino Geliştirme Kart Özellikleri



Arduino UNO

- **MikroDenetleyici :** ATmega328
- **Besleme Voltajı :** 7-12 Volt
- **Çalışma Voltajı :** 5 Volt
- **Pin Sayısı :** 14 adet dijital pin(6 adet PWM çıkış), 6 adet analog pin
- **Boyutları :** Uzunluk 68.6 mm, genişlik 53.4 mm



Arduino MEGA

- **MikroDenetleyici :** ATmega2560
- **Besleme Voltajı :** 7-12 Volt
- **Çalışma Voltajı :** 5 Volt
- **Pin Sayısı :** 54 adet dijital pin(15 adet PWM çıkış), 16 adet analog pin
- **Boyutları :** Uzunluk 101,6 mm, genişlik 53.4 mm

Arduino NANO

- **MikroDenetleyici :** ATmega328
- **Besleme Voltajı :** 7-12 Volt
- **Çalışma Voltajı :** 5 Volt
- **Pin Sayısı :** 14 adet dijital pin(6 adet PWM çıkış), 6 adet analog pin
- **Boyutları :** Uzunluk 45 mm, genişlik 18 mm

> *Eğitmen bu bölümde Arduino kart çeşitlerini yukarıda verilen örnek şeklinde açıklamalıdır. Destekleyici materyaller (Fiziki olarak sınıf ortamında sunmalı, destekleyici video ve görseller) ile sunum yapmalıdır. Kartların teknik özelliklerini incelerken Arduino.cc web sitesini referans alması tavsiye edilir.*
>
> Eğitmen bu bölümde bundan sonraki süreçte kullanılacak GND, 5V, Analog, Digital ve PWM kavramlarını anlatmalıdır.

Kazanım

- Arduino geliştirme kartı çeşitleri tekleri ve kullanım alanlarının daha detaylı öğrenilmesi.
- Digital, Analog ve PWM kavramlarının pekiştirilmesi.

Dersin Hedefleri

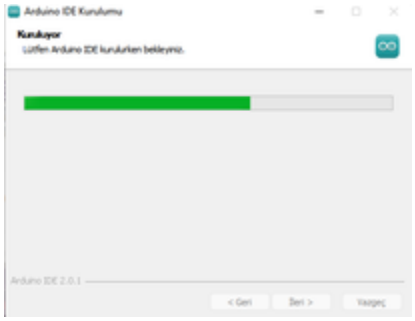
	Başlık	Hedef Kitle	Detaylar
1	☑ Arduino Geliştirme Kart Çeşitleri	☑ Eğitmen ☐ Öğrenci	Arduino çeşitleri uygulamalı ve görsel materyallerle sunulmalıdır.
2	☑ Arduino Geliştirme Kart Çeşitleri	☐ Eğitmen ☑ Öğrenci	İlerleyen konularda kullanacağımız ünitenin temel yapı taşı olan geliştirme kartı Arduino ve çeşitlerinin öğrenilmesi hedeflenmektedir.

Arduino IDE Kurulumu ve Tanıtımı

Müfredat Kodu / Ünite / Konu	İCT-700-1.3 / Temel Kavramlar ve Kullanımlar / Arduino IDE Kurulumu ve Tanıtımı
Hedeflenen Süre	30 Dakika

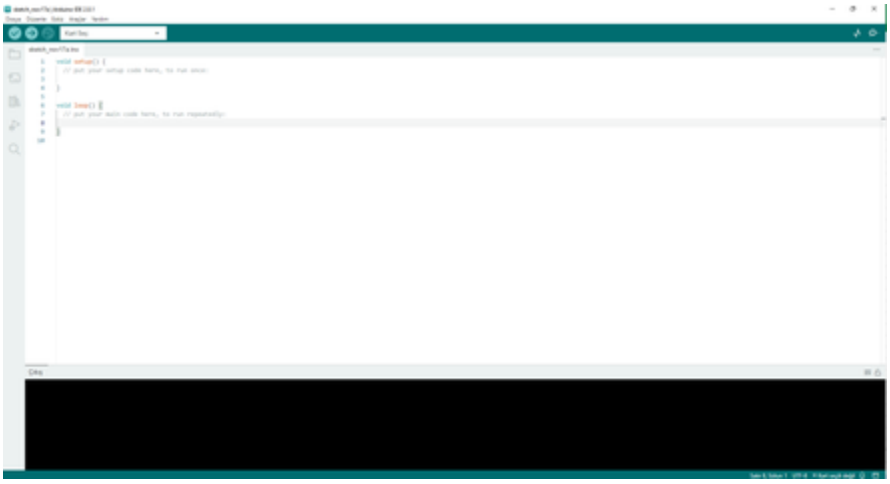
Arduino IDE Kurulumu

Arduino IDE programının kurulumu için https://www.arduino.cc/en/software linkinden bilgisayarınızın işletim sistemi özelliklerine göre programı indirebilirsiniz.



IDE programı üzerinden arduino kartlara programı atmak için bazen CH340 seriport sürücüsüne ihtiyaç duyulabilmektedir. Karta yükleme sorunu yaşandığından eğitmenin bu sürücüyü ilgili bilgisayara kurması tavsiye edilir.

IDE Arayüz Tanıtımı



Eğitmen bu bölümde Arduino IDE programının arayüz ayarları , kütüphane, araçlar bölümünü anlatması beklenmektedir.

Seriport Tanıtımı ve Kullanımı

Eğitmenin bu bölümde Seriport Ekranının nasıl çalıştığından, baund genişliği kavramından ve seri port ekran ayarlarından bahsetmesi tavsiye edilmektedir.

Kazanım

- Arduino IDE programının kurulması ayarlarının yapılabiliyor olması.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Arduino IDE Kurulumu ve Tanıtımı	☑ Eğitmen ☐ Öğrenciler	Arduino IDE kurulumu ve tanıtımı uygulamalı olarak anlatılması hedeflenmektedir. • Arduino
2	☑ Arduino IDE Kurulumu ve Tanıtımı	☐ Eğitmen ☑ Öğrenciler	Arduino IDE kurulumunun sorunsuz yapılması hedeflenmektedir.

Arduino IDE Programlama Dili ve Özellikleri

Müfredat Kodu / Ünite / Konu	İCT-700-1.4 / Temel Kavramlar ve Kullanımlar / Arduino IDE Programlama Dili ve Özellikleri
Hedeflenen Süre	45 Dakika

Arduino IDE Programlama Dili Nedir?

C++, C ve Java programlama dilleri temelli olan Arduino programlama dili üç ana bölüme ayrılabilir: fonksiyonlar, değerler (değişkenler ve sabitler) ve yapı.

> Eğitmenin bu bölümde programlama dillerini C++, C ve Java kısaca bahsetmesi beklenmektedir.

Fonksiyonlar (Functions)

Arduino kartını kontrol etmek ve hesaplamalar yapmak için fonksiyonlar kullanılmaktadır. Tüm programlama dillerinde olduğu gibi Arduino programlama dilinde de yazılımcının işini kolaylaştırmak için fonksiyonlar kullanılmaktadır.

> Eğitmen bu bölümde Arduino Reference kısmını referans almalıdır.
>
> Eğitmenden bu kısımda Digital I/O (digitalRead(), digitalWrite(), pinMode()) , Analog I/O (analogRead(), analogWrite()) , Time (delay (), delayMicroseconds(), millis(),) , Math (map(), max(), min()) fonksiyonları ve diğer temel fonksiyonları anlatması beklenmektedir. Amaç ileride kulanacağımız fonksiyon çeşitlerinin öğrenilmesini sağlamaktır.

Değişkenler (Variables)

Arduino programlama esnasında kullanılan değişkenler bu bölümde anlatılmalıdır.

> Eğitmen bu bölümde Arduino Reference kısmını referans alarak anlatmalıdır.
>
> Değişkenler konu başlığı altında Sabitler (Constants), Constants (Dönüştürmek), Data Types (Veri tipleri), Variable Scope & Qualifiers (Değişken Kapsam ve Niteleyiciler), ve Utilities (Araçlar) anlatılmalıdır.

Yapılar (Structure)

Arduino programlama esnasında kullanılan yapılar bu bölümde anlatılmalıdır.

> Eğitmen bu bölümde kullanılan döngüleri (loop(), setup()), Kontrol Yapılarını, Kullanılan Dizileri, Aritmetik operatörleri, Karşılaştırma Operatörleri, Boole Operatörleri, İşaretçi Erişim Operatörleri, Bileşik Operatörleri açıklamalıdır.

```
int buttonPin = 3;

// setup initializes serial and the button pin
void setup() {
  Serial.begin(9600);
  pinMode(buttonPin, INPUT);
}

// loop checks the button pin each time,
// and will send serial if it is pressed
void loop() {
  if (digitalRead(buttonPin) == HIGH) {
    Serial.write('H');
  }
  else {
    Serial.write('L');
  }

  delay(1000);
}
```

Kazanım

- Arduino programlama dilinde kullanılan Değişkenler-Fonksiyonlar-Yapı kavramlarının pekiştirilmesi.
- Kod yazma süreçlerinin keşfedilmiş olması.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Arduino IDE Programlama Dili ve Özellikleri	☑ Eğitmen ☐ Öğrenciler	Arduino IDE Programlama Dili ve Özellikleri örneklerle açıklanmalıdır. • Arduino
2	☑ Arduino IDE Programlama Dili ve Özellikleri	☐ Eğitmen ☑ Öğrenciler	Arduino IDE Programlama Dili ve Özellikleri pekiştirilmelidir.

Breadboard Kullanımı

Müfredat Kodu / Ünite / Konu	ICT-700-1.5 / Temel Kavramlar ve Kullanımlar / Breadboard Kullanımı
Hedeflenen Süre	45 Dakika

Breadboard Kullanımı

Arduino ile sensörleri ve diğer elektronik kompanentlerin bağlantısını yapmak için kullanılmaktadır. Breadboard kullanılarak yetersiz pinlerin çoğaltılması ve bağlantı yapılması sağlanmaktadır.

Eğitmenin bu bölümde breadboard kullanımında güç pinlerinin (Gnd, +5V) çoğaltılma şeklinden, diğer sensörleirn nasıl bağlanması gerektiği hakkında bilgi vermelidir.

Kazanım

- Breadboard kullanımının sağlanması.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Breadboard Kullanımı	☑ Eğitmen ☐ Öğrenciler	Breadboard Kullanımı uygulamalı olarak anlatılmalıdır.
2	☑ Breadboard Kullanımı	☐ Eğitmen ☑ Öğrenciler	Breadboard Kullanımı pekiştirilmelidir.

SENSÖRLER VE UYGULAMALARI

Müfredat Kodu	İCT-700-2
Ders	Sensörler ve Uygulamaları
Ders İçeriği	• Sensörler • Blink Uygulaması • Kara Şimşek Uygulaması • Butondan Değer Okuma Uygulaması • Potansiyometreden Değer Okuma Uygulaması • LDR Değer Okuma Uygulaması • Kara Şimşek Uygulaması - For Döngüsü • LDR ile Led Yakma Uygulaması (if-else) • Potansiyometre ile Led Yakma Uygulaması • Park Sensörü Uygulaması • Ekran Kullanım Uygulaması • Servo Motor Uygulaması • DC Motor Uygulaması
Seviye	İCT-S2: ORTA
Ön Koşullar	Arduino geliştirme kartı için gerekli olan temel kavramları biliyor olmak.
Amaç	Sensör çeşitlerininin, arduino geliştirme kartlı ile programlanması ve kullanılması amaçlanmaktadır.
Hedef	Arduino geliştirme kartının programlanması ve sensörlerin kullanımı hedeflenmektedir.
Gereksinimler	Arduino Seti, Bilgisayar, El Aletleri
Kazanım	• Arduino programlama temellerinin öğrenilmesi • Sensörlerin çalışma prensiplerinin kavranması • Programla, elektronik bağlantı arasındaki bağın kurulması
Durum	BEKLİYOR / UYGULANIYOR / TAMAMLANDI / DEĞERLENDİRİLDİ
Hedeflenen Süre	810 Dakika
Eğitim Günleri	Başlama: .../.../.... Bitiş: .../.../....

Sensörler

Müfredat Kodu / Ünite / Konu	İCT-700-2.1 / Sensörler ve Uygulamaları / Sensörler
Hedeflenen Süre	45 Dakika

Sensör Nedir?

En geniş tanımıyla **sensör** amacı ortamındaki olayları veya değişiklikleri tespit etmek ve bilgileri diğer elektronik cihazlara genellikle bir bilgisayar işlemcisine göndermek olan bir cihaz, modül, makine veya alt sistemdir. https://tr.wikipedia.org/wiki/Sens%C3%B6r

> Eğitmen sensörün tanımını ve çeşitlerini anlatırken https://tr.wikipedia.org/wiki/Sens%C3%B6r referans alarak görsel öğelerle ve videolarla desteklemesi tavsiye edilir.

Sensör Çeşitleri

Günümüz teknolojisinde günlük yaşantımızda bir çok sensör çeşiti ile karşılaşılmaktadır.

- Gaz Sensörü
- Renk Sensörü
- Kızılötesi Algılama Sensörü (IR)
- Basınç Sensörü
- Fotoelektrik Sensörü
- Nem Sensörü
- Titreşim Sensörü
- Hareket Sensörü
- Ultrasonik Mesafe Sensörü

> Eğitmen bu bölümde günlük hayatta karşılaşılan sesnör çeşitlerinden, uygulamalarından günlük hayattaki faydalarından ve endüstrideki kullanımından bahsetmeli, öğrencilerin yorumlarını almalıdır.

> ***Kazanım***
>
> - Günlük hayatta kullanılan sensörler ve kullanımları.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Sensör Nedir?	☑ Eğitmen ☐ Öğrenciler	Günümüz teknolojisinde günlük yaşantımızda karşımıza çıkan sensörlerin ve kullanım amaçlarının örneklerle ve görsel destekleyici sunumlarla anlatılması beklenmektedir.
2	☑ Sensör Nedir?	☐ Eğitmen ☑ Öğrenciler	Sensör kavramının pekiştirilmesi, kullanım alanlarının öğrenilmesi hedeflenmektedir.

Blink Uygulaması

Müfredat Kodu / Ünite / Konu	İCT-700-2.2.1 / Arduino Uygulamaları / Blink Uygulaması
Hedeflenen Süre	45 Dakika

Uygulama Amacı

Arduino kullanıcısının ilk defa Arduino ile tanışma uygulaması denilebilir. Bu uygulama ile kullanıcıya kod derleme , kart seçme, port seçme , örnek uygulama seçme karta kod yükleme gösterilmesi amaçlanmaktadır.

Kullanıcı bu uygulamada

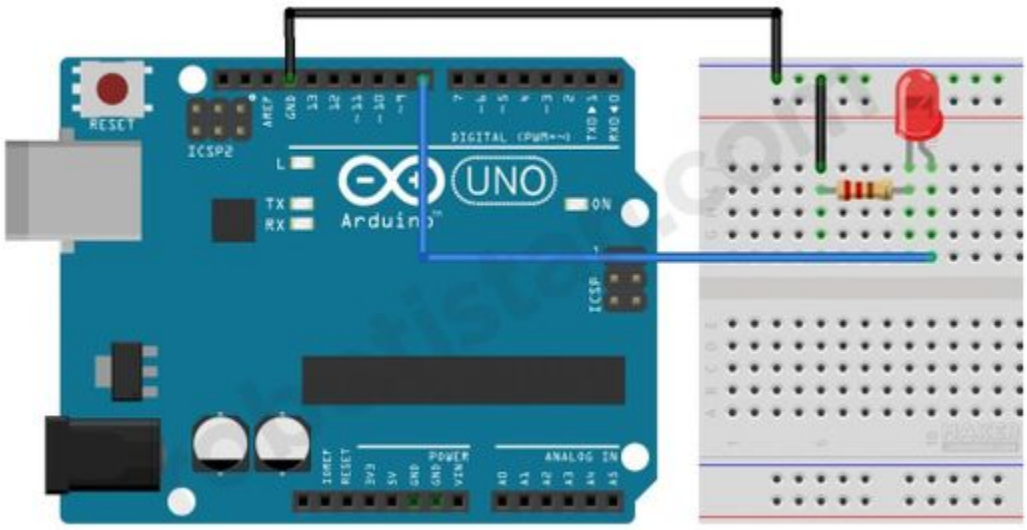
1. Arduino'yu USB kablosunu bilgisayarına takmalıdır.
2. Arduino IDE → Dosya → Örnekler → Basics → Blink uygulamasını açmalıdır.
3. Araçlar Bölümünden Kart (Arduino UNO) ve Port seçilmedilir.
4. Kod derlenip yüklenmelidir.
5. Arduino üzeindeki ledin 1 saniye yanıp sönmesi gerekmektedir.

Bu uygulamadan sonra aşağıdaki bağlantıya göre bir led takılıp örnekten bağımsız kod yazılarak uygulama tamamlanmalıdır.

Gereksinimler

- Arduino Uno ve USB Kablosu
- 1 Adet Led
- Jumper Kablo (Erkek-Erkek)
- 220 R Direnç

Bağlantı



Kod Bloğu

```
Blink Uygulaması
int led=8;
void setup() {
   pinMode(led, OUTPUT);
  }
void loop() {
  digitalWrite(led, HIGH);   // LED'i aç (HIGH voltaj seviyesidir)
  delay(1000);               // delay belirlediğimiz süre boyunca döngüyü durdurur. Burada bir saniye
bekle diyoroz.
  digitalWrite(led, LOW);    // bu kodla LED'i söndürüyoruz.
  delay(1000);               //Bir saniye bekle
}
```

Kazanım

- Arduino programlama (Kod Derleme-Yükleme).
- digitalWrite fonksiyonun pekiştirilmesi.
- Bredboard kullanımının sağlanması.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Blink Uygulaması	☑ Eğitmen ☐ Öğrenciler	Arduino giriş uygulaması olarak kabul edilen Blink uygulaması ile öğrencilere kod derleme, port seçimi, kod yükleme ve setup, void fonksiyonları hakkında bilgilendirme yapılmalıdır.
2	☑ Blink Uygulaması	☐ Eğitmen ☑ Öğrenciler	Arduino IDE kod düzenleme, derleme ve yükleme süreçlerinin pekiştirilmesi hedeflenmektedir.

Kara Şimşek Uygulaması

Müfredat Kodu / Ünite / Konu	İCT-700-2.2.2 / Arduino Uygulamaları / Kara Şimşek Uygulaması
Hedeflenen Süre	90 Dakika

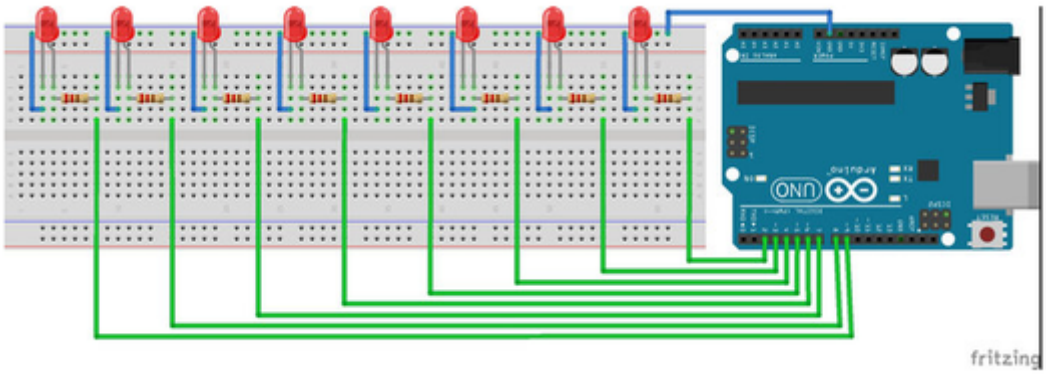
🚀 *Uygulama Amacı*

Kara Şimşek uygulaması ile kullanıcıların bir önceki konuda öğrendiklerini pekiştirmeleri amaçlanmaktadır.

🎗️ *Gereksinimler*

- Arduino UNO ve USB Kablosu
- 8 Adet LED
- 8 Adet 220 R Direnç
- Breadboard

💻 *Bağlantı*



⌨️ *Kod Bloğu*

```
//Kara Şimşek Uygulaması
void setup() {
pinMode(2,OUTPUT);
pinMode(3,OUTPUT);
pinMode(4,OUTPUT);
pinMode(5,OUTPUT);
pinMode(6,OUTPUT);
pinMode(7,OUTPUT);
pinMode(8,OUTPUT);
pinMode(9,OUTPUT);
}
void loop() {
digitalWrite(2,HIGH);
delay(100);
digitalWrite(2, LOW);
delay(100);
digitalWrite(3,HIGH);
delay(100);
digitalWrite(3, LOW);
delay(100);
digitalWrite(4,HIGH);
delay(100);
digitalWrite(4, LOW);
delay(100);
digitalWrite(5,HIGH);
delay(100);
digitalWrite(5, LOW);
delay(100);
digitalWrite(6,HIGH);
delay(100);
digitalWrite(6, LOW);
delay(100);
digitalWrite(7,HIGH);
delay(100);
digitalWrite(7, LOW);
delay(100);
digitalWrite(8,HIGH);
delay(100);
digitalWrite(8, LOW);
delay(100);
digitalWrite(9,HIGH);
delay(100);
digitalWrite(9, LOW);
delay(100);
 }
```

Kazanım

- Arduino programlama (Kod Derleme-Yükleme).
- digitalWrite fonksiyonun pekiştirilmesi.
- Breadboard kullanımının sağlanması.
- Setup () ve Loop() fonksiyonları ve delay () yapısının öğrenilmesi.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Kara Şimşek Uygulaması	☑ Eğitmen ☐ Öğrenciler	Kara şimşek uygulaması ile öğrencilere algoritma oluşturmanın öğretilmesi hedeflenmektedir. Bu bölümde kara şimşek ile öğrenciler bir algoritma oluşturmalı, delay komutunun kullanımının anlatılması beklenmektedir.
2	☑ Kara Şimşek Uygulaması	☐ Eğitmen ☑ Öğrenciler	Uygulama ile breadboard kullanımının pekiştirilmesi, algroritma oluşturma yeteneğinin arttırılması beklenmektedir.

Referanslar & Dokümantasyon

Kara Şimşek Uygulaması

Butondan Değer Okuma Uygulaması

Müfredat Kodu / Ünite / Konu	İCT-700-2.2.3 / Arduino Uygulamaları / Butondan Değer Okuma Uygulaması
Hedeflenen Süre	45 Dakika

Uygulama Amacı

Butondn değer okuma uygulaması ile kullanıcının seriporttan digitalRead fonksiyonu ile değer okuyabilmesi amaçlanmaktadır.

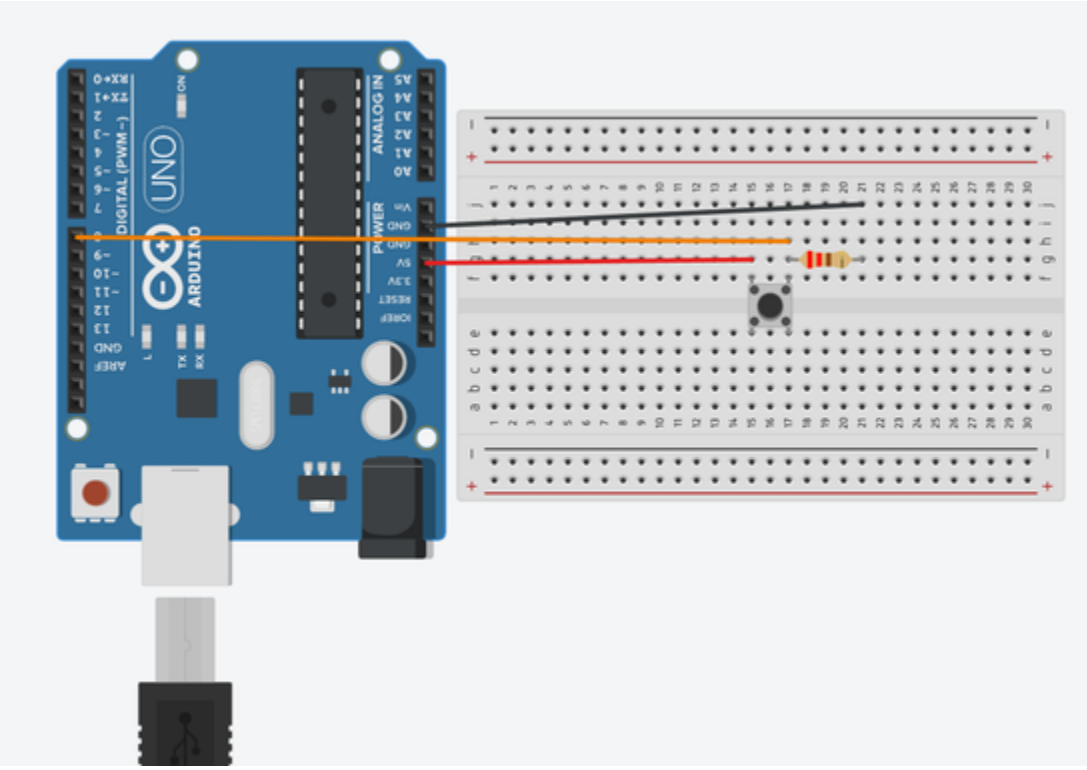
Bu bölümde dikkat edilmesi gereken nokta butonları bağlarken kullanılan dirençtir. Bu direncin ismi pull-down direncidir. Peki pull-down direnci ne işe yarar?Pull-down direnci, dijital pinleri giriş olarak kullandığımızda sinyalin bozulmamasını sağlar.
Bu projemizde buton basılı değilken dijital pinden okunan değer 0V yani lojik LOW seviyesidir.
Pull-down direnci, buton basılıp değer HIGH'a çekilmediği sürece bu pindeki gerilimin 0V'ta sabit kalmasını sağlar.

Direnç ile yapılan uygulamadan sonra dirençsiz PULLUP_INPUT komutu da kullanılmalıdır.

Gereksinimler

- Arduino UNO ve USB Kablosu
- Buton
- 10K Direnç
- Breadboard

Bağlantı



Kod Bloğu

```
int buttonApin = 9;
void setup()
{
  pinMode(buttonApin, INPUT);
   Serial.begin(9600);
}
void loop()
{
  int deger=digitalRead(buttonApin);
  Serial.println(deger);
  delay(500);
  }
```

Kazanım

- digitalRead fonksiyonun kullanılması.
- PULLUP_INPUT komutunun kullanılması.
- Seri Porttan değer okuma.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Butondan Değer Okuma Uygulaması	☑ Eğitmen ☐ Öğrenciler	Buton uygulaması ile seriport ekranından değer okuma süreçleri ve digitalRead komutunun anlatılması beklenmektedir.
2	☑ Butondan Değer Okuma Uygulaması	☐ Eğitmen ☑ Öğrenciler	Seriport ekranının kullanımı ve digitalRead komutunun kullanımının pekiştirilmesi hedeflenmektedir.

Potansiyometreden Değer Okuma Uygulaması

Müfredat Kodu	İCT-700-2.2.4 / Arduino Uygulamaları / Potansiyometreden Değer Okuma Uygulaması
Hedeflenen Süre	45 Dakika

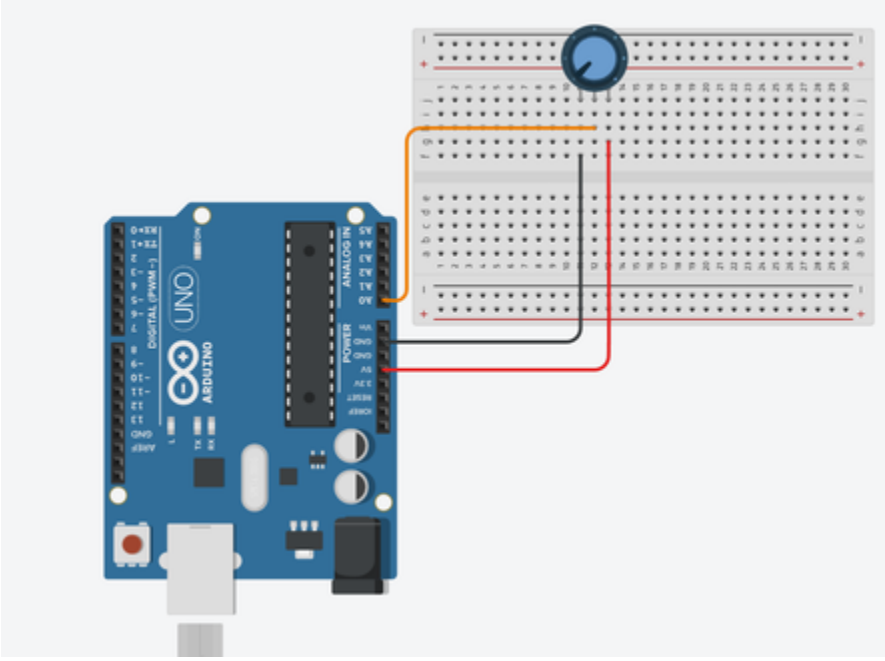
Uygulama Amacı

Potansiyometreden Değer okuma uygulaması ile analogRead fonksiyonun kullanılması ve seri port ekranından değer okuma amaçlanmaktadır.

Gereksinimler

- Arduino Uno ve USB Kablosu
- Potansiyometre
- Jumper Kablo (Erkek-Erkek)
- BreadBoard

Bağlantı



Kod Bloğu

```
Blink Uygulaması
int pot=A0;
int deger;
void setup() {
   pinMode(pot, INPUT);
   Serial.begin(9600);
 }
void loop() {
  deger=analogRead(pot);
  Serial.println(deger);
}
```

Kazanım

- analogRead fonksiyonun pekiştirilmesi.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Potansiyometreden Değer Okuma Uygulaması	☑ Eğitmen ☐ Öğrenciler	Potansiyometre uygulaması ile seriport ekranından değer okuma süreçleri ve analogRead komutunun anlatılması beklenmektedir.
2	☑ Potansiyometreden Değer Okuma Uygulaması	☐ Eğitmen ☑ Öğrenciler	Seriport ekranının kullanımı ve analogRead komutunun kullanımının pekiştirilmesi hedeflenmektedir.

LDR Değer Okuma Uygulaması

Müfredat Kodu / Ünite / Konu	İCT-700-2.2.5 / Arduino Uygulamaları / LDR Değer Okuma Uygulaması
Hedeflenen Süre	45 Dakika

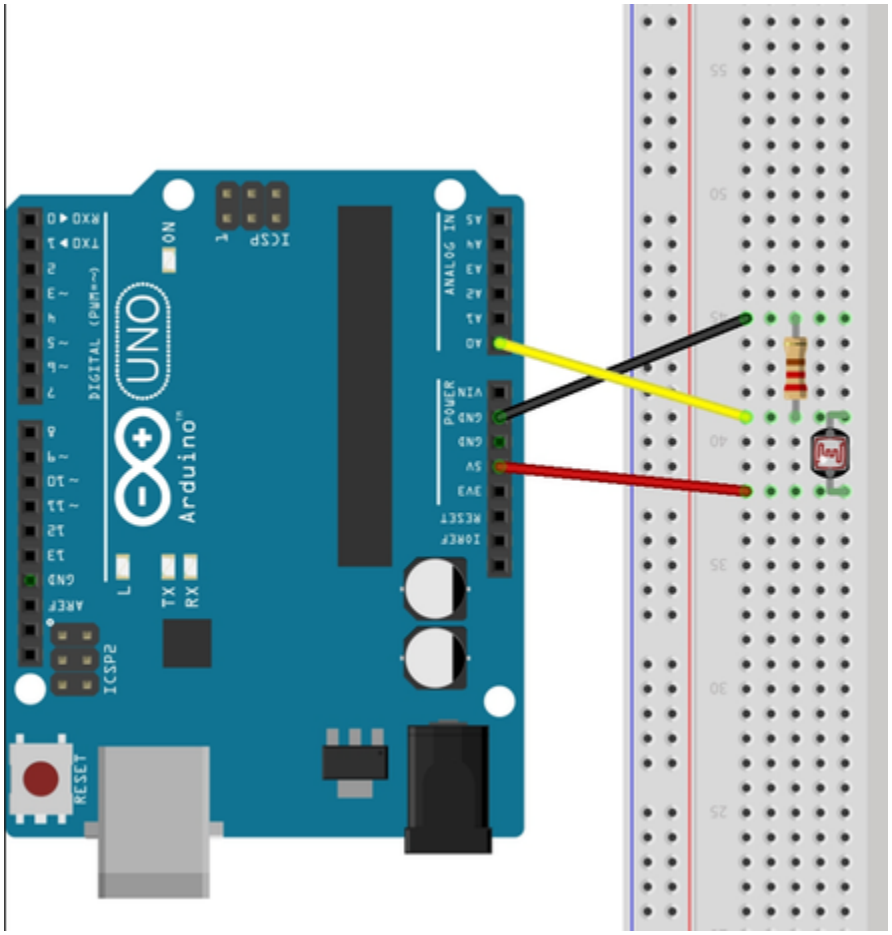
🚀 *Uygulama Amacı*

LDR ışık sensörü kullanarak değer okuma uygulaması ile analogRead fonksiyonun kullanılması ve seri port ekranından değer okuma amaçlanmaktadır.

🎗️ *Gereksinimler*

- Arduino Uno ve USB Kablosu
- LDR
- 220 R Direnç
- Jumper Kablo (Erkek-Erkek)
- BreadBoard

💻 *Bağlantı*



⌨️ *Kod Bloğu*

```
//Blink Uygulaması
int ldr=A0;
int deger;
void setup() {
   pinMode(ldr, INPUT);
   Serial.begin(9600);
  }
void loop() {
  deger=analogRead(ldr);
  Serial.println(deger);
}
```

Kazanım

- analogRead fonksiyonun pekiştirilmesi.
- LDR kullanımının öğrenilmesi.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ LDR Değer Okuma Uygulaması	☑ Eğitmen ☐ Öğrenciler	LDR değer okuma uygulaması ile LDR sensörünün pekiştirilmesi beklenmektedir.
2	☑ LDR Değer Okuma Uygulaması	☐ Eğitmen ☑ Öğrenciler	SLDR değer okuma uygulaması ile LDR sensörünün kullanımının pekiştirilmesi hedeflenmektedir.

Kara Şimşek Uygulaması - For Döngüsü

Müfredat Kodu / Ünite / Konu	İCT-700-2.2.6 / Arduino Uygulamaları / Kara Şimşek Uygulaması - For Döngüsü
Hedeflenen Süre	90 Dakika

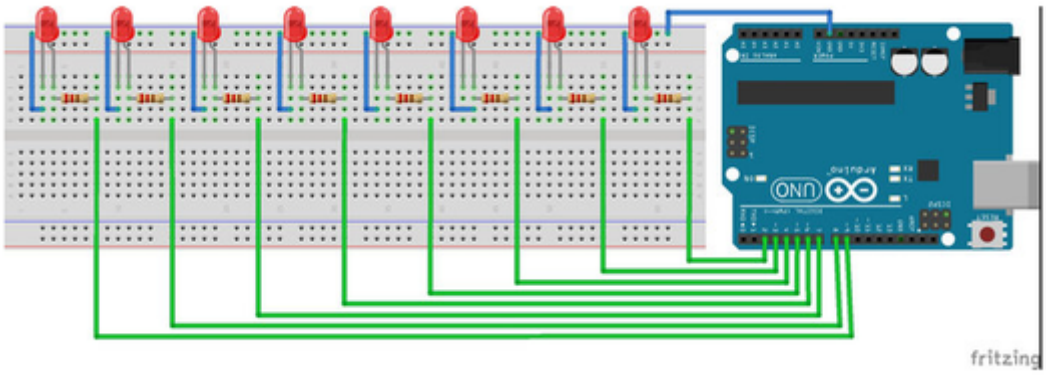
For Döngüsü Nedir?

Bu bölümde for döngüsü anlatılmalıdır. For döngüsü anlatılırken Kara Şimşek uygulaması referans alınmalıdır. Öğrencilere uzun uzadıya 8 led için karaşimşek uygulaması yaptırılmalı daha sonrasında ise for döngüsünden bahsedilmeli ve for döngüsü ile uygulama tekrardan yapılmalıdır.

Gereksinimler

- Arduino UNO ve USB Kablosu
- 8 Adet LED
- 8 Adet 220 R Direnç
- Breadboard

Bağlantı



Kod Bloğu

```
void setup() {
  for(int i=2;i<=9;i++){
    pinMode(i,OUTPUT);
  }
}
void loop() {
  // put your main code here, to run repeatedly:
for(int i=2;i<=9;i++){
   digitalWrite(i,HIGH);
  delay(100);
  digitalWrite(i,low);
  delay(100);
  }
  for(int i=9;i>=2;i--){
  digitalWrite(i,HIGH);
  delay(100);
  digitalWrite(i,low);
  delay(100);
  }
```

Bu bölümün daha iyi kavranması için

- Led Parlaklığı for döngüsü ile artıp azaltılabilir.
- Seriport Ekranına Saydırma işlemi yapılabilir.
- Servomotor kontrolü kısmında buraya atıfta bulunulabilir.

Kazanım

- For Döngüsünün kullanımı.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Kara Şimşek Uygulaması - For Döngüsü	☑ Eğitmen ☐ Öğrenciler	For döngüsünün uygulamalı olarak anlatılması ve sağladığı kolaylık uygulamalı olarak anlatılması beklenmektedir.
2	☑ Kara Şimşek Uygulaması - For Döngüsü	☐ Eğitmen ☑ Öğrenciler	For döngüsünün kullanımının pekiştirilmesi hedeflenmektedir.

LDR ile Led Yakma Uygulaması (if-else)

Müfredat Kodu / Ünite / Konu	İCT-700-2.2.7 / Arduino Uygulamaları / LDR ile Led Yakma Uygulaması (if-else)
Hedeflenen Süre	45 Dakika

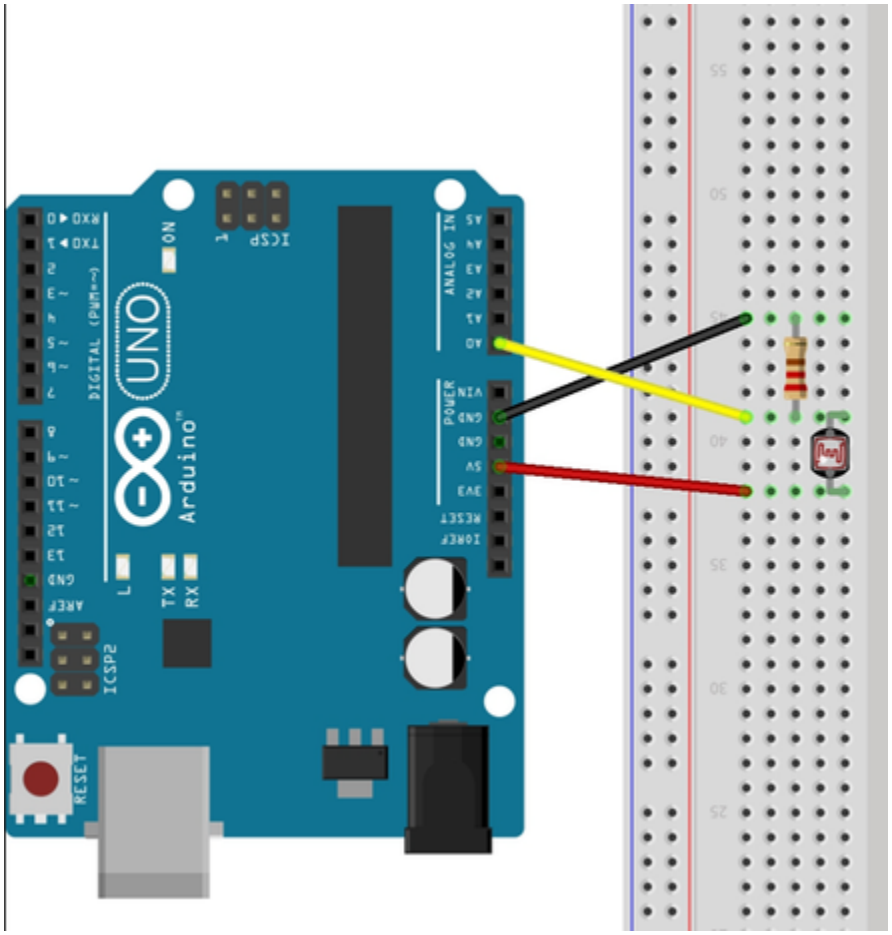
Uygulama Amacı

LDR ile led uygulaması ile İf-else yapılarının öğretilmesi amaçlanmaktadır.

Gereksinimler

- Arduino Uno ve USB Kablosu
- LDR
- 220 R Direnç
- Led
- Jumper Kablo (Erkek-Erkek)
- BreadBoard

Bağlantı



Kod Bloğu

```
int ldr=A0;
int deger;
İnt led=9;
void setup() {
    pinMode(ldr, INPUT);
    pinMode(led, OUTPUT);
    Serial.begin(9600);
  }
void loop() {
   deger=analogRead(ldr);
   Serial.println(deger);
   if(deger<=50){
   digitalWrite(led,HIGH);
   }
   else{
   digitalWrite(led,LOW);
   }
}
```

Pekiştirme Uygulaması

LM35 Sıcaklık sensörü ile belirli bir sıcaklık altında Yeşil Led ,sıcaklık üstünde ise Kırmızı Led yakma uygulaması yapılmalıdır.

Gereksinimler

- Arduino Uno
- LM35 Sıcaklık Sensörü
- Jumper Kablo
- Breadboard
- Yeşil Led
- Kırmızı Led
- 2 Adet 220R Direnç

Buton ile led yakıp söndürme uygulaması yapılmalıdır.

Gereksinimler

- Arduino Uno
- Buton
- Jumper Kablo
- Breadboard
- Led
- 220R Direnç
- 10K Direnç

Kazanım

- if-else yapısının öğrenilmesi.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ LDR ile Led Yakma Uygulaması (if-else)	☑ Eğitmen ☐ Öğrenciler	if-else yapısının uygulamalı olarak anlatılması anlatılması beklenmektedir.
2	☑ LDR ile Led Yakma Uygulaması (if-else)	☐ Eğitmen ☑ Öğrenciler	if-else yapısının kullanımının pekiştirilmesi hedeflenmektedir.

Potansiyometre ile Led Yakma Uygulaması

Müfredat Kodu / Ünite / Konu	İCT-700-2.2.8 / Arduino Uygulamaları / Potansiyometre ile Led Yakma Uygulaması
Hedeflenen Süre	45 Dakika

Uygulama Amacı

Potansiyometre değer aralıklarına göre 3 adet Led Yakma uygulaması ile İf,else-if ,else komutlarının, Karşılaştırma Operatörlerinin ve Boole Operatörlerinin kullanılması amaçlanmaktadır.

Gereksinimler

- Arduino Uno ve USB Kablosu
- Potansiyometre
- 3 adet 220 R Direnç
- 3 adet Led
- Jumper Kablo (Erkek-Erkek)
- BreadBoard

Pekiştirme Uygulaması

1. Potansiyometreden okunan değer 300'den küçük ise 1. led, 300 ile 600 arasında ise 2. led diğer durumlar için 3. led yanmalıdır.
2. Potansiyometreden değer okurken **map** komutu kullanımı da anlatılmalıdır.

Kazanım

- if, else-if, else yapılarının kullanılması.
- Karşılaştırma Operatörlerinin kullanılması.

  != (eşit değil)

  < (küçüktür)

  <= (küçüktür veya eşittir)

  == (eşittir)

  > (büyüktür)

  >= (büyüktür veya eşittir)
- Boole Operatörlerinin kullanılması.

  ! (mantıklı değil)

  && (mantıklı ve)

  || (mantıksal veya)

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Potansiyometre ile Led Yakma Uygulaması	☑ Eğitmen ☐ Öğrenciler	Karşılaştırma ve Boole Operatörlerinin if-else komutu ile uygulamalı olarak anlatılması beklenmektedir.

2	☑ Potansiyometre ile Led Yakma Uygulaması	☐ Eğitmen ☑ Öğrenciler	Karşılaştırma ve Boole Operatörlerini kullanımı ve if-else komutlarının pekiştirilmesi hedeflenmektedir.

NOTLAR

Park Sensörü Uygulaması

Müfredat Kodu / Ünite / Konu	İCT-700-2.2.9 / Arduino Uygulamaları / Park Sensörü Uygulaması
Hedeflenen Süre	45 Dakika

🚀 *Uygulama Amacı*

Park sensörü uygulaması ile HC SR04 Ultrasonik mesafe sensörünün kullanımı amaçlanmaktadır.

🎗️ *Gereksinimler*

- Arduino Uno ve USB Kablosu
- HC-SR04 Ultrasonik mesafe sensörü
- Jumper Kablo (Erkek-Erkek)
- BreadBoard

> 1. Ultrasonik mesafe sensörü çalışma mantığı ve çalışması anlatılarak seriporttan değer okunmalıdır.

📔 *Pekiştirme Uygulaması*

→ Ultrasonik mesafe sensörü kullanılarak belli bir mesafenin altında buzzer ötmesi ve led yanması uygulaması

🎗️ *Gereksinimler*

- Arduino Uno ve USB Kablosu
- HC-SR04 Ultrasonik mesafe sensörü
- Jumper Kablo (Erkek-Erkek)
- BreadBoard
- Led
- Buzzer
- 220R Direnç

> 📘 *Kazanım*
>
> - Ultrasonik mesafe sensörü kullanımı.

✅ *Dersin Hedefleri*

	Başlık	Hedef Kitle	Detaylar
1	☑ Park Sensörü Uygulaması	☑ Eğitmen ☐ Öğrenciler	HC-SR04 Mesafe sensörünün if else komutu ile uygulamalı olarak anlatılması beklenmektedir.
2	☑ Park Sensörü Uygulaması	☐ Eğitmen ☑ Öğrenciler	HC-SR04 Mesafe sensörünün kullanımının öğrenilmesi hedeflenmektedir.

Ekran Kullanım Uygulaması

Müfredat Kodu / Ünite / Konu	İCT-700-2.2.10 / Arduino Uygulamaları / Ekran Kullanım Uygulaması
Hedeflenen Süre	45 Dakika

Uygulama Amacı

LCD ekran kullanımının öğrenilmesi hedeflenmektedir.

Gereksinimler

- Arduino Uno ve USB Kablosu
- Jumper Kablo (Erkek-Erkek)
- BreadBoard
- 2 x16 LCD Ekran

Bu uygulamada kütüphane yüklenerek lcd ekran çalıştırılması ve yazı yazdırılmalıdır.

Pekiştirme Uygulaması

→ Ultrasonik mesafe sensörü kullanılarak LCD ekrana mesafe yazdırma uygulaması

Gereksinimler

- Arduino Uno ve USB Kablosu
- HC-SR04 Ultrasonik mesafe sensörü
- Jumper Kablo (Erkek-Erkek)
- BreadBoard
- Led
- Buzzer
- 220R Direnç
- 2x16 LCD Ekran

Kazanım

- Kütüphane kullanımının öğrenilmesi.
- LCD ekran kullanımının öğrenilmesi.
- Birçok yapının, sensörün bir arada kullanılması.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Ekran Kullanım Uygulaması	☑ Eğitmen ☐ Öğrenciler	Harici LCD ekranın kullanımı ve kütüphane yüklemenin uygulamalı olarak anlatılması beklenmektedir.
2	☑ Ekran Kullanım Uygulaması	☐ Eğitmen ☑ Öğrenciler	LCD ekran kullanımının ve kütüphane eklemenin pekiştirilmesi hedeflenmektedir.

Servo Motor Uygulaması

Müfredat Kodu / Ünite / Konu	İCT-700-2.2.11 / Arduino Uygulamaları / Servo Motor Uygulaması
Hedeflenen Süre	45 Dakika

🚀 *Uygulama Amacı*

Servo Motor kullanımının öğrenilmesi hedeflenmektedir.

🎗️ *Gereksinimler*

- Arduino Uno ve USB Kablosu
- Jumper Kablo (Erkek-Erkek)
- S90 Servo Motor

📋 Örnekler kısmından Sweep uygulaması seçilerek uygulanmalıdır.

```
#include <Servo.h>

Servo myservo;  // create servo object to control a servo
// twelve servo objects can be created on most boards

int pos = 0;    // variable to store the servo position

void setup() {
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
}

void loop() {
  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
    // in steps of 1 degree
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(15);                       // waits 15 ms for the servo to reach the position
  }
  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(15);                       // waits 15 ms for the servo to reach the position
  }
}
```

📋 *Pekiştirme Uygulaması*

→ Örnekler kısmından Knob uygulaması seçilerek uygulanmalıdır.

🎗️ *Gereksinimler*

- Arduino Uno ve USB Kablosu
- Jumper Kablo (Erkek-Erkek)

- S90 Servo Motoru
- Potansiyometre

```
#include <Servo.h>
Servo myservo;  // create servo object to control a servo
int potpin = A0;  // analog pin used to connect the potentiometer
int val;    // variable to read the value from the analog pin
void setup() {
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
}
void loop() {
  val = analogRead(potpin);           // reads the value of the potentiometer (value between 0 and
1023)
  val = map(val, 0, 1023, 0, 180);     // scale it for use with the servo (value between 0 and 180)
  myservo.write(val);                  // sets the servo position according to the scaled value
  delay(15);                           // waits for the servo to get there
}
```

Kazanım

- Servo Motor kullanımının öğrenilmesi.
- Potansiyometre ile servo motor kullanımı.
- Map komutunun kullanımı.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Servo Motor Uygulaması	☑ Eğitmen ☐ Öğrenciler	Servo motorun uygulamalı olarak anlatılması beklenmektedir.
2	☑ Servo Motor Uygulaması	☐ Eğitmen ☑ Öğrenciler	Servo motorun kullanımının öğrenilmesi hedeflenmektedir.

DC Motor Uygulaması

Müfredat Kodu / Ünite / Konu	İCT-700-2.2.12 / Arduino Uygulamaları / DC Motor Uygulaması
Hedeflenen Süre	180 Dakika

Uygulama Amacı

250 RPM DC Motor kullanımının öğrenilmesi hedeflenmektedir.

Gereksinimler

- Arduino Uno ve USB Kablosu
- Jumper Kablo (Erkek-Erkek)
- 4 Adet DC motor
- L298N Voltaj Regulatörlü Çift Motor Sürücü Kartı
- 9V Güç Kaynağı (Pil)

Bu uygulamada Dc motorlar lehimlenmelidir.Sonrasında sürücü ile bağlantıları yapılmalıdır.

Pekiştirme Uygulaması

→ Dc motorları buton ile kontrol edilmesi (1. Butona basılınca ileri, 2. butona basılınca geri, 3. butona basınca sağa ve 4. butona basınca sola dönüş yapan bir uygulama yapılmalıdır.)

Gereksinimler

- Arduino Uno ve USB Kablosu
- Jumper Kablo (Erkek-Erkek)
- 4 Adet DC motor
- L298N Voltaj Regulatörlü Çift Motor Sürücü Kartı
- 9V Güç Kaynağı (Pil)
- 4 Adet Buton

Kazanım

- DC motorların öğrenilmesi.
- DC motor kullanılması.
- Sürücü mantığının öğrenilmesi.
- Lehimleme yapılması.
- Dc motor kullanımının pekiştirilmesi.
- Fonksiyon oluşturma (İleri , geri, sağ, sol).

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ DC Motor Uygulaması	☑ Eğitmen ☐ Öğrenciler	DC motorun uygulamalı olarak anlatılması beklenmektedir.

2	☑ DC Motor Uygulaması	☐ Eğitmen ☑ Öğrenciler	DC motorun kullanımının öğrenilmesi hedeflenmektedir.

NOTLAR

PROJE GELİŞTİRME

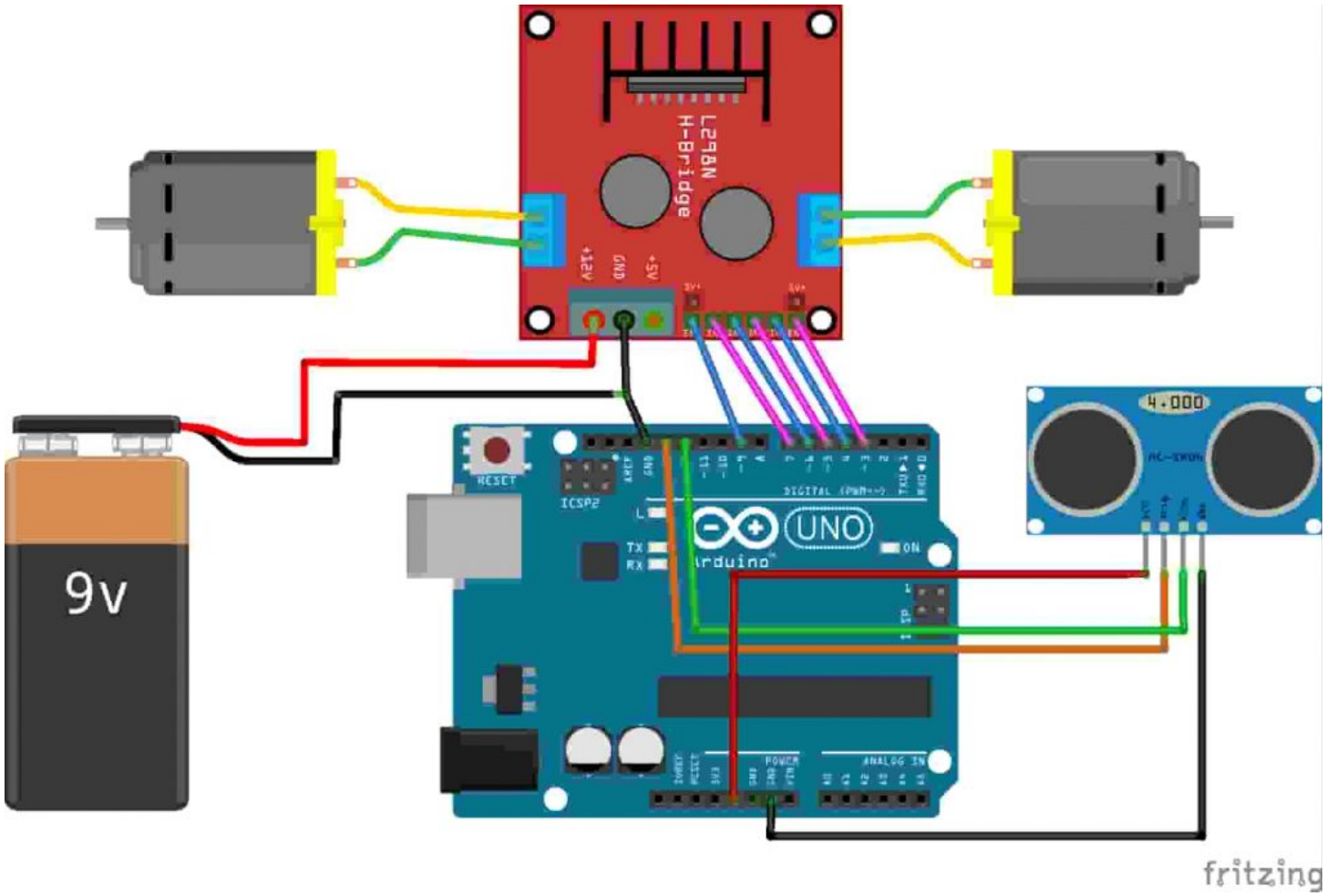
Müfredat Kodu	İCT-700-3
Ders	Proje Geliştirme
Ders İçeriği	• Engelden Kaçan Robot Yapımı • Robot Kol Yapımı • Bluetooth İle Araç Kontrolü • Akıllı Ev Uygulaması
Seviye	İCT-S2: ORTA
Ön Koşullar	Sensör kullanımı ve arduino programlama hakkında bilgi sahibi olmak.
Amaç	Bir önceki ünitede öğrenilen temel uygulamaların bir proje odaklı birleştirilmesi ve geliştirilmesi hedeflenmektedir. Bu çalışmalar yapılırken öğrencilerin elektronik bilgilerini ve yazılım bilgilerini pekiştirmeleri, proje geliştirme esnasında ortaya çıkan sorunlara çözüm üretmeleri amaçlanmaktadır.
Hedef	Projeleri amaçları doğrultusunda montajlıyor ve programlıyor olmak hedeflenmektedir.
Gereksinimler	• Gerekli olan sensörler ve arduino seti • El aletleri • Silikon • Havya takımı ve lehim • Montaj için gerekli malzemeler
Kazanım	• Problemlere çözüm üretme • Lehimleme • Elektronik bilginini artması ve kompanentlerin kullanılması • Kodlama süreçlerinin pekiştirilmesi
Durum	BEKLİYOR / UYGULANIYOR / TAMAMLANDI / DEĞERLENDİRİLDİ
Hedeflenen Süre	810 Dakika
Eğitim Günleri	Başlama: .../.../.... Bitiş: .../.../....

Engelden Kaçan Robot Yapımı

Müfredat Kodu / Ünite / Konu	İCT-700-3.1 / Proje Geliştirme / Engelden Kaçan Robot Yapımı
Hedeflenen Süre	135 Dakika

🚀 *Projenin Amacı*

Engelden kaçan araç yapımı ve kodlanması amaçlanmaktadır.



🔍 *Gereksinimler*

- Arduino UNO
- L298Nvoltaj regülatörlü çift motor sürücü kartı
- HC-SR 04 Mesafe sensörü
- 9v pil
- Pil yuvası
- Jumper kablo
- 2 adet dc motor
- 1 adet sarhoş tekerlek
- 2 adet tekerlek

- Bu projede öğrencilerden araç tasarlamaları istenmelidir. Araç tasarımı öğrencilere özgü olmalıdır.
- Gruplara ayrılarak kodlama ve elektronik bölümleri paylaşılmalıdır. Grup sayısı en fazla 3 olmalıdır.
- Öğrencilerden bu proje ile ilgili algoritma geliştirmeleri hedeflenmektedir.

```
#define echoPin 12 //Ultrasonik sensörün echo pini Arduino'nun 12.pinine
#define trigPin 13 //Ultrasonik sensörün trig pini Arduino'nun 13.pinine tanımlandı.
#define MotorR1 7
#define MotorR2 6
#define MotorRE 9  // Motor pinlerini tanımlıyoruz.
#define MotorL1 5
#define MotorL2 4
#define MotorLE 3
long sure, uzaklik; //süre ve uzaklık diye iki değişken tanımlıyoruz.
void setup() {
  // ultrasonik sensör Trig pininden ses dalgaları gönderdiği için OUTPUT (Çıkış),
  // bu dalgaları Echo pini ile geri aldığı için INPUT (Giriş) olarak tanımlanır.
  pinMode(echoPin, INPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(MotorL1, OUTPUT);
  pinMode(MotorL2, OUTPUT);
  pinMode(MotorLE, OUTPUT); //Motorlarımızı çıkış olarak tanımlıyoruz.
  pinMode(MotorR1, OUTPUT);
  pinMode(MotorR2, OUTPUT);
  pinMode(MotorRE, OUTPUT);
  Serial.begin(9600);
}
void loop() {
  digitalWrite(trigPin, LOW); //sensör pasif hale getirildi
  delayMicroseconds(5);
  digitalWrite(trigPin, HIGH); //Sensore ses dalgasının üretmesi için emir verildi
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW); //Yeni dalgaların üretilmemesi için trig pini LOW konumuna getirildi
  sure = pulseIn(echoPin, HIGH); //ses dalgasının geri dönmesi için geçen sure ölçülüyor
  uzaklik = sure / 29.1 / 2; //ölçülen süre uzaklığa çevriliyor
  Serial.println(uzaklik);
  if (uzaklik < 15) // Uzaklık 15'den küçük ise,
  {
    geri();  // 150 ms geri git
    delay(150);
    sag();  // 250 ms sağa dön
    delay(250);
  }
  else {  // değil ise,
    ileri(); // ileri git
  }
}
void ileri(){  // Robotun ileri yönde hareketi için fonksiyon tanımlıyoruz.
  digitalWrite(MotorR1, HIGH); // Sağ motorun ileri hareketi aktif
  digitalWrite(MotorR2, LOW); // Sağ motorun geri hareketi pasif
```

```
  analogWrite(MotorRE, 150); // Sağ motorun hızı 150
  digitalWrite(MotorL1, HIGH); // Sol motorun ileri hareketi aktif
  digitalWrite(MotorL2, LOW); // Sol motorun geri hareketi pasif
  analogWrite(MotorLE, 150); // Sol motorun hızı 150


}
void sag(){ // Robotun sağa dönme hareketi için fonksiyon tanımlıyoruz.
  digitalWrite(MotorR1, HIGH); // Sağ motorun ileri hareketi aktif
  digitalWrite(MotorR2, LOW); // Sağ motorun geri hareketi pasif
  analogWrite(MotorRE, 0); // Sağ motorun hızı 0 (Motor duruyor)
  digitalWrite(MotorL1, HIGH); // Sol motorun ileri hareketi aktif
  digitalWrite(MotorL2, LOW); // Sol motorun geri hareketi pasif
  analogWrite(MotorLE, 150); // Sol motorun hızı 150


}
void geri(){ // Robotun geri yönde hareketi için fonksiyon tanımlıyoruz.
  digitalWrite(MotorR1, LOW); // Sağ motorun ileri hareketi pasif
  digitalWrite(MotorR2, HIGH); // Sağ motorun geri hareketi aktif
  analogWrite(MotorRE, 150); // Sağ motorun hızı 150
  digitalWrite(MotorL1, LOW); // Sol motorun ileri hareketi pasif
  digitalWrite(MotorL2, HIGH); // Sol motorun geri hareketi aktif
  analogWrite(MotorLE, 150); // Sol motorun hızı 150
  }
```

Kazanım

- Problemlere çözüm üretme.
- Lehimleme.
- Elektronik bilginini artması ve kompanentlerin kullanılması.
- Kodlama süreçlerinin pekiştirilmesi.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar	Durum
1	☑ Engelden Kaçan Robot Yapımı	☑ Eğitmen ☐ Öğrenciler	Engelden kaçan robot yapımı ile öğrencilerin özgün bir tasarım yapmaları ve algoritma oluşturarak programlamalarının yapımına yardımcı olunması beklenmektedir. https://maker.robotistan.com/engelden-kacan-robot-yapimi/	UYGULANIYOR
2	☑ Engelden Kaçan Robot Yapımı	☐ Eğitmen ☑ Öğrenciler	Engelden kaçan robot tasarlanması ve programlanması hedeflenmektedir.	UYGULANIYOR

Robot Kol Yapımı

Müfredat Kodu	İCT-700-3.2 / Proje Geliştirme / Robot Kol Yapımı
Hedeflenen Süre	135 Dakika

🚀 *Projenin Amacı*

Günümüz teknolojisinde endüstriyel robot kolların çalışma prensiplerinin öğretilmesi amaçlanmaktadır.

🔍 *Gereksinimler*

- 4 Adet x Tower Pro SG90 RC Mini Servo Motor
- Arduino Sensor Shield
- Arduino Uno
- 12V Adaptör
- 4 Adet Potansiyometre veya 2 Adet Joystick
- Jumper Kablo

- Bu projede öğrencilerden 3 boyutlu yazıcıdan robot kol parçaları üretmeleri ve montajlamaları beklenmektedir.
- Proje yapımında ekip oluşturularak grup çalışması beklenmektedir.
- Öğrencilerden bu proje ile ilgili algoritma geliştirmeleri hedeflenmektedir.

📘 *Kazanım*

- Problemlere çözüm üretme.
- Lehimleme.
- Elektronik bilgisini artması ve kompanentlerin kullanılması.
- Kodlama süreçlerinin pekiştirilmesi.

✅ Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Robot Kol Yapımı	☑ Eğitmen ☐ Öğrenciler	Robot kol projesinin montajı ve kodlanmasında yol gösterilmesi beklenmektedir.
2	☑ Robot Kol Yapımı	☐ Eğitmen ☑ Öğrenciler	Robot kolun montajlanması ve programlanarak çalıştırılması hedeflenmektedir.

Bluetooth İle Araç Kontrolü

Müfredat Kodu	İCT-700-3.3 / Proje Geliştirme / Bluetooth İle Araç Kontrolü
Hedeflenen Süre	270 Dakika

🚀 *Projenin Amacı*

Günümüz teknolojisinde Bluetooth teknolojisinin kullanımının öğretilmesi amaçlanmaktadır.

🔍 *Gereksinimler*

- Arduino UNO
- L298N Voltaj Regülatörlü Çift Motor Sürücü Kartı
- Hc06 Bluetooth Modülü
- 9V Pil
- Pil Yuvası
- Jumper Kablo
- 4 Adet Dc Motor
- 4 Adet Tekerlek

- Bu Projede App Inventor mobil uygulama geliştirme programı kullanılarak mobil uygulama geliştirilmesi hedeflenmektedir.
- Proje yapımında ekip oluşturularak grup çalışması beklenmektedir.
- Öğrencilerden bu proje ile ilgili algoritma geliştirmeleri hedeflenmektedir.
- Bluetooth haberleşmenin öğrenilmesi hedeflenmektedir.

Kazanım

- Bluetooth haberleşme teknolojisinin öğrenilmesi.
- App Inventor uygulamasının pekiştirilmesi.

✅ *Dersin Hedefleri*

	Başlık	Hedef Kitle	Detaylar
1	☑ Bluetooth İle Araç Kontrolü	☑ Eğitmen ☐ Öğrenciler	Bluetooth'lu araç yapımında montaj ve programlama bölümlerinde yardımcı olunması ve ekip çalışmasına teşvik edilmesi beklenmektedir.
2	☑ Bluetooth İle Araç Kontrolü	☐ Eğitmen ☑ Öğrenciler	Bluetooth ’lu araç yapımı, montajı ve programlanmasının yapılabiliyor olması hedeflenmektedir.

Akıllı Ev Uygulaması

Müfredat Kodu	İCT-700-3.4 / Proje Geliştirme / Akıllı Ev Uygulaması
Hedeflenen Süre	270 Dakika

🚀 *Projenin Amacı*

Günümüz teknolojisinde evlerimizde bir çok ürün Bluetooth ve internete internet teknolojisiyle kontrol edilmektedir. Akıllı ev projesi ilegünlük hayatta yapılması mümkün olan çalışmanın minyatrünün yapılması hedeflenmektedir.

🔍 *Gereksinimler*

- Arduino UNO
- 16×2 LCD Ekran
- 1 adet 10 kOhm Potansiyometre
- 2 adet 220 Ohm Direnç
- 1 adet Buzzer
- 1 adet Push Buton
- 1 adet LM35 Sıcaklık Sensörü
- 2 adet Led
- 1 adet SG-5010 Servo Motor
- 1 adet HC-06 Bluetooth Seri Modül Kartı

- Bu Projede App Inventor mobil uygulama geliştirme programı kullanılarak mobil uygulama geliştirilmesi hedeflenmektedir.
- Proje yapımında ekip oluşturularak grup çalışması beklenmektedir.
- Öğrencilerden bu proje ile ilgili algoritma geliştirmeleri hedeflenmektedir.
- Bluetooth haberleşmenin öğrenilmesi hedeflenmektedir.
- Birden fazla sensörün çalışmasını sağlamak hedeflenmektedir.

Kazanım

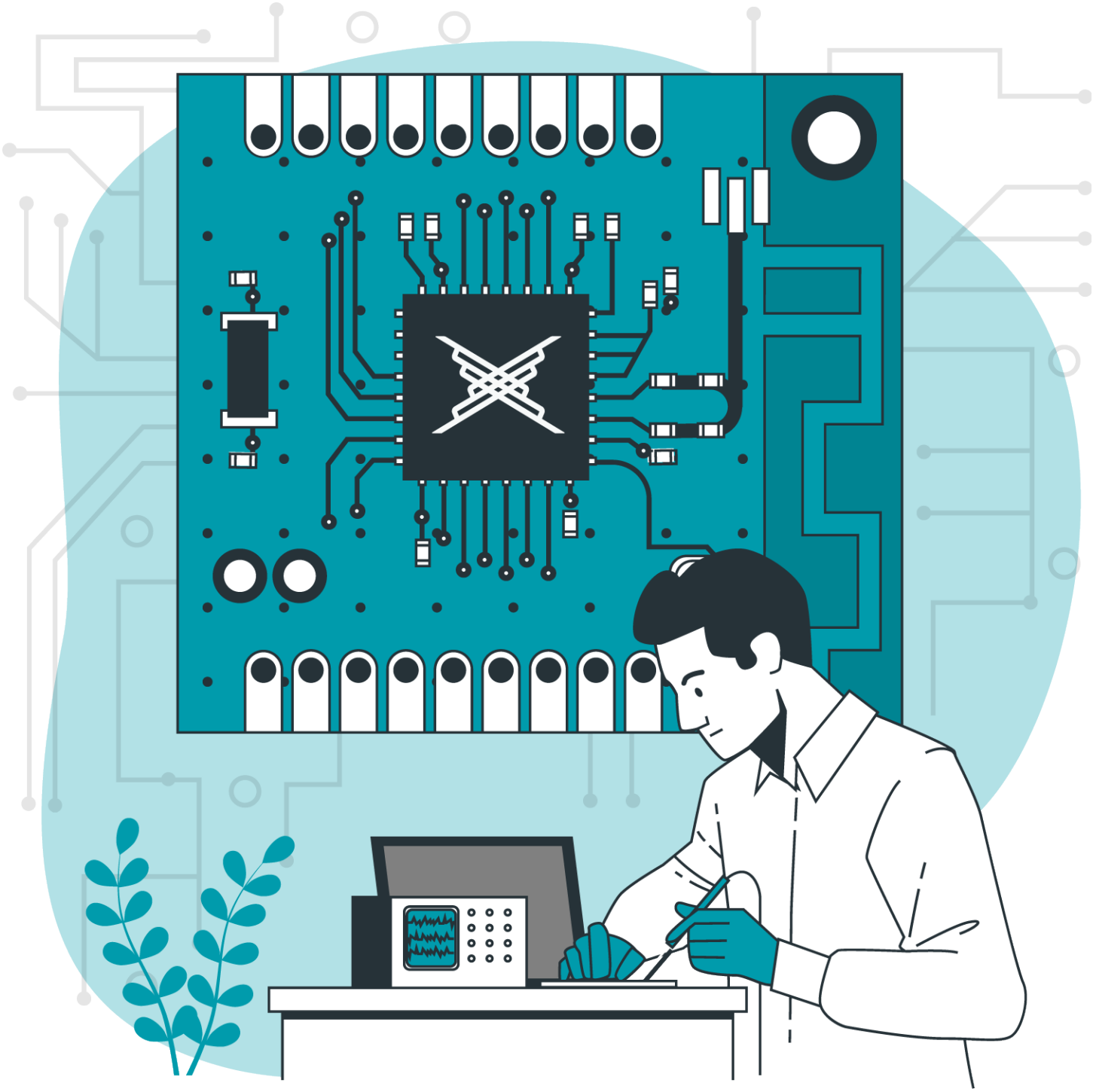
- Tasarım becerisi.
- Mobil Uygulama geliştirme becerisi.
- Kodlama becerisi.
- Ekip çalışması kültürü.

✅ *Dersin Hedefleri*

	Başlık	Hedef Kitle	Detaylar
1	☑ Akıllı Ev Uygulaması	☑ Eğitmen ☐ Öğrenciler	Akıllı Ev Uygulamasının montaj ve programlama bölümlerinde yardımcı olunması ve ekip çalışmasına teşvik edilmesi beklenmektedir.
2	☑ Akıllı Ev Uygulaması	☐ Eğitmen ☑ Öğrenciler	Akıllı ev uygulaması ile birden fazla sensörün birlikte kullanılması, programlanması ve montajının yapılabiliyor olması hedeflenmektedir.

ELEKTRONIK EĞITIM MÜFREDATI

ELEKTRONİK EĞİTİM MÜFREDATI														
EĞİTİM İÇERİĞİ / HAFTA	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.
Elektronik Komponent ve Özellikleri (İCT-400-2)														
Temel Elektronik : Direnç														
Temel Elektronik : Kondansatör														
Temel Elektronik : Bobin														
Temel Elektronik : Transformatör (Trafo)														
Temel Elektronik : Diyot														
Temel Elektronik : Transistör														
Temel Elektronik : Tristör														
Temel Elektronik : Optokuplör														
Temel Elektronik : Triyak														
Elektronik Devre Uygulamaları (İCT-400-3)														
Yarım Dalga Doğrultucu Devresi														
Tam Dalga Doğrultucu Devresi														
Buton İle Led Kontrol Uygulaması														
NPN Transistörlü Anahtarlama Devresi														
Flip Flop Devresi														
Tristörlü Uygulama Devresi														
Kondansatörlü Şarj Deşarj Devresi														
Transistörlü Hırsız Alarm Devresi														



ELEKTRONIK KOMPONENT VE ÖZELLIKLERI

Müfredat Kodu	İCT-400-2
Ders	Elektronik Komponent ve Özellikleri
Ders İçeriği	• Temel Elektronik: Direnç • Temel Elektronik: Kondansatör • Temel Elektronik: Bobin • Temel Elektronik: Transformatör (Trafo) • Temel Elektronik: Diyot • Temel Elektronik: Transistör • Temel Elektronik: Tristör • Temel Elektronik: Optokuplör • Temel Elektronik: Triyak
Seviye	İCT-S2: ORTA
Ön Koşullar	Elektronik dersini programına katılmış olmak
Amaç	Elektronik komponentleri ve özelliklerini öğrenerek elektronik devre kurabiliyor olmak amaçlanmaktadır.
Hedef	Amaca yönelik elektronik devre kurabiliyor olmak
Gereksinimler	Atölye Ortamı ve gerekli elektronik kompanentlerin olması
Kazanım	• Elektronik kompanentler hakkında bilgi sahibi olmak • Elektronik devre kurabiliyor olmak • Elektronik devrelerin çalışma prensibini anlıyor olmak
Referanslar & Dokümantasyon	• https://320volt.com/temel-elektrik-elektronik-dersleri-basit-devre-ornekleri/ • http://kitap.eba.gov.tr/panel/dosyalar/upload/1456/0/U_0_21_08_2020_21_08_13_91.pdf • https://diyot.net/direnc-renk-kodlari/ • http://elektronikderslerim.blogspot.com/2018/11/kondansatorlu-sarj-desarj-devresi.html • https://www.elektrikrehberiniz.com/elektrik-trafolari/transformator-trafo-nedir-4340/ • http://elektronikderslerim.blogspot.com/2016/12/npn-transistorun-anahtarlama-eleman.html • http://www.robotiksistem.com/tristor_nedir_cesitleri_ozellikleri.html • http://elektronikderslerim.blogspot.com/2018/12/tristorlu-uygulama-devresi.html • https://www.elektrikrehberiniz.com/elektronik/triyak-nedir-10396/
Durum	BEKLİYOR / UYGULANIYOR / TAMAMLANDI / DEĞERLENDİRİLDİ
Hedeflenen Süre	630 Dakika
Eğitim Günleri	Başlama: .../.../.... Bitiş: .../.../....

Temel Elektronik: Direnç

Müfredat Kodu / Ünite / Konu	İCT-400-2.1 / Elektronik Komponent ve Özellikleri / Direnç
Hedeflenen Süre	90 Dakika

🚀 *Direnç*

Temel elektronikte, direnç konusunda aşağıdaki başlıklara yer verilmelidir. Bu şunlardır:

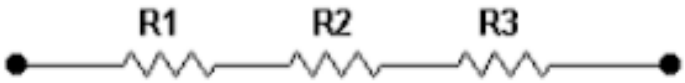
- Direnç nedir ve nasıl ölçülür?
- Direnç değeri nasıl hesaplanır ve etkileyen faktörler nelerdir?
- Seri ve paralel direnç devreleri.
- Direnç ölçme cihazları ve kullanımı.
- Direnç değerini değiştiren devre elemanları (potansiyometre, termistör vb.)
- Direnç özelliklerinin kullanımı ve uygulamaları (ısıtma elemanları, sınırlama dirençleri vb.)

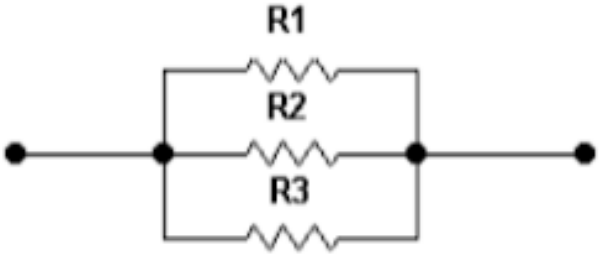
Bu başlıklar altında, direnç konusu hakkında daha detaylı bilgiler verilebilir. Örneğin, Ohm Kanunu ve karşılaştırma direnci konusunda, direnç değerinin nasıl hesaplandığı ve karşılaştırma direnci kullanımı gösterilebilir. Seri ve paralel direnç devreleri konusunda ise, dirençlerin bu devre tiplerinde nasıl birleştirildiği ve etkileşimi hakkında bilgi verilebilir.

🏆 *Direnç Kodları*

RENK	SAYI	ÇARPAN	TOLERANS
Siyah	0	$10^0 = 1$	-
Kahverengi	1	$10^1 = 10$	%1
Kırmızı	2	$10^2 = 100$	%2
Turuncu	3	$10^3 = 1000$	-
Sarı	4	$10^4 = 10.000$	-
Yeşil	5	$10^5 = 100.000$	%0,5
Mavi	6	$10^6 = 1.000.000$	%0,25
Mor	7	$10^7 = 10.000.000$	%0,1
Gri	8	$10^8 = 100.000.000$	-
Beyaz	9	$10^9 = 1.000.000.000$	-
Altın	-	0,1	%5
Gümüş	-	0,01	%10

0	1	2	3	4	5	6	7	8	9
Siyah	Kahverengi	Kırmızı	Turuncu	Sarı	Yeşil	Mavi	Mor	Gri	Beyaz





- **Termistörler (Isıya Duyarlı Elemanlar) :** Sıcaklığın değişimine maruz kaldıklarında elektriksel dirençlerini değiştirebilen elektronik araçlardır. Bu araçlar, sıcaklık artışını elektriksel bir sinyale dönüştürebilir ve bu sayede sıcaklık değişimlerini ölçebilir ve izleyebilir. PTC-NTC çeşitleri bulunmaktadır.
- **Potansiyometre:** Elektronik bir araçtır ve elektriksel bir devrenin bir noktasında gerilimin değişimini kontrol etmek için kullanılır. Potansiyometreler, sıklıkla ses ve ısı kontrolü gibi farklı uygulamalarda kullanılır.

Kazanım

Direnç konusu çerçevesinde kazanılacak bilgi ve beceriler şunlardır;

- Direnç kavramını ve direnç değerini ölçme yöntemlerini anlama.
- Direnç değerini etkileyen faktörleri belirleme ve direnç değerini hesaplama.
- Seri ve paralel direnç devrelerini anlama ve kurma.
- Direnç ölçme cihazlarını kullanarak direnç değerini doğru bir şekilde ölçme.
- Direnç özelliklerini ve uygulamalarını anlama ve kullanma.
- Elektronik devrelerde dirençlerin etkileşimini ve işlevini anlama ve devrelerin tasarımında kullanma.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Direnç	☑ Eğitmen ☐ Öğrenci	Temel elektronikte, direnç konusunun tüm detaylarıyla ; Direnç ölçümü, renk kodları, devrede kullanımı hakkında bilgi verilmesi hedeflenmektedir. Bilgilendirme esnasında öğrencilere uygulama yaptırılmalıdır. (Direnç değerleri okutulmalı vs.)
2	☑ Direnç	☐ Eğitmen ☑ Öğrenci	Temel elektronikte, direnç konusunun tüm detaylarıyla ; Direnç ölçümü, renk kodları, devrede kullanımı hakkında bilgi sahibi olunması hedeflenmektedir.

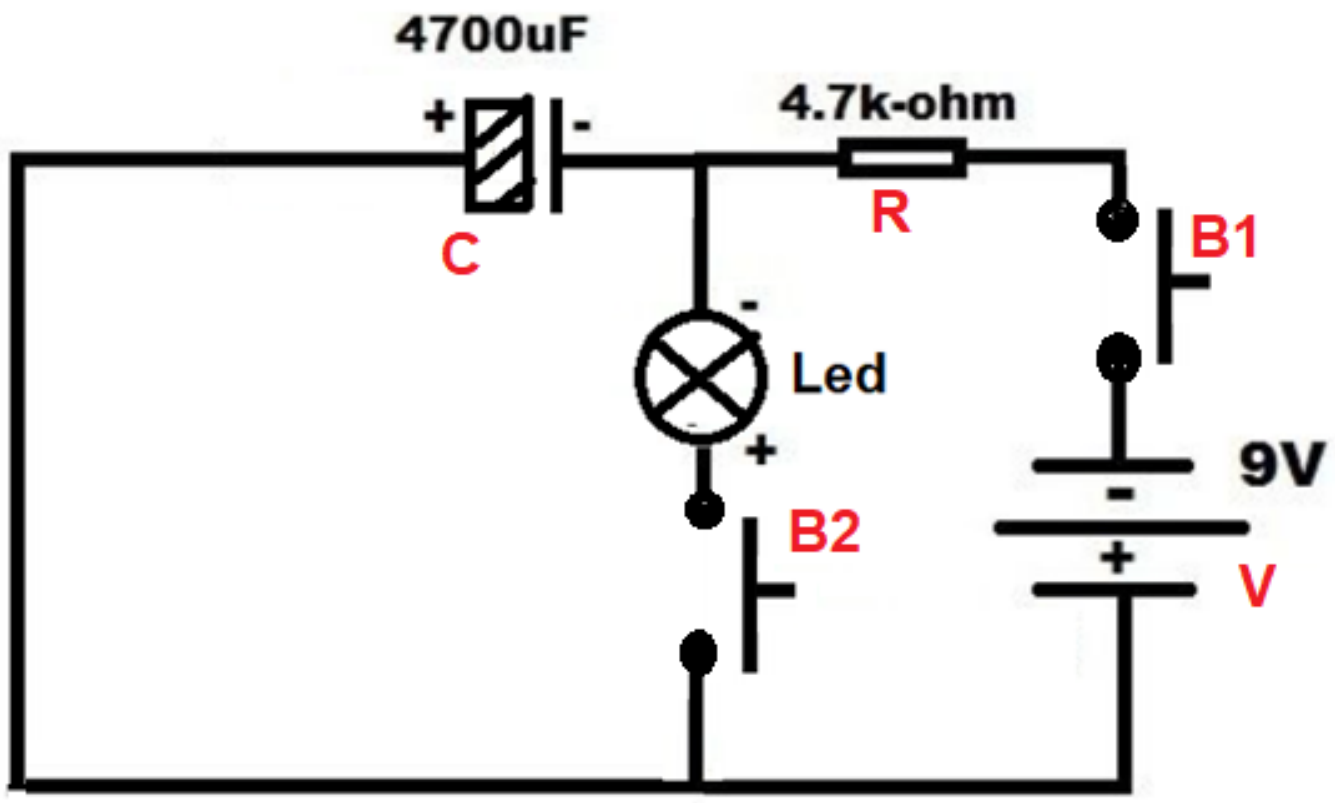
Temel Elektronik: Kondansatör

Müfredat Kodu / Ünite / Konu	İCT-400-2.2 / Elektronik Komponent ve Özellikleri / Kondansatör
Hedeflenen Süre	90 Dakika

🚀 *Kondansatör*

Kondansatörler, elektriksel enerjiyi depolayan ve akışını kontrol eden araçlardır. Bir kondansatörün işlevi, elektriksel bir devre içinde birikmiş yükleri depolamak ve bunları bir anda serbest bırakmaktır.

🏆 *Örnek Sorular*



Çalışma Prensibi : Devreye ilk enerji verildiği anda B1 butonuna basılı olmadığından dolayı devreden enerji akmaz, bundan dolayı kondansatör şarj olmaz ve ledin ışık vermesi söz konusu olmaz. İlk olarak B1 butonuna basılır. Kondansatör şarj olmaya başlar. Kondansatör şarj oluktan sonra B1 butonu açık konuma getirilir ve B2 butonuna basılır. B2 butonuna basıldığında led diyot kondansatör üzerinden enerjisini alır ve ışık vermeye başlar. Kondansatör üzerindeki enerjiyi boşaltana kadar led ışık verir. Kondansatörün üzerindeki enerji boşaldığında artık ledin ışık vermesini sağlayacak herhangi bir voltaj değeri kalmamıştır. Böylece led ışık vermez . Devrenin çalışması bu döngü içerisinde devam eder.

🛠️ *Araç Gereçler*

- Bread board.
- Led.
- Kutuplu Kondansatör.
- İki adet buton.
- 9 voltluk pil.

- Jumper Kablo.

Kazanım

- Kondansatörler hakkında bilgi sahibi olma.
  - Kondansatörler, elektrik akımının frekansını ve yönünü değiştirmek için kullanılır.
  - Kondansatörler, elektrik akımını depolamak ve dağıtmak için kullanılır.
  - Kondansatörler, elektronik devrelerdeki sinyalleri filtrelemeye yardımcı olur.
  - Kondansatörler, elektriksel sistemlerde gerilim düzenini korur.
  - Kondansatörler, elektrik akımının yüksek frekanslı bileşenlerini engelleyebilir ve bu sayede elektronik cihazların çalışmasını düzenler.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Kondansatör	☑ Eğitmen ☐ Öğrenciler	Kondansatörlerin kullanılması ve uygulanması hakkında uygulamalı olarak anlatım beklenmektedir.
2	☑ Kondansatör	☐ Eğitmen ☑ Öğrenciler	Kondansatörlerin kullanımı hakkında bilgi sahibi olmak

Temel Elektronik: Bobin

Müfredat Kodu / Ünite / Konu	İCT-400-2.3 / Elektronik Komponent ve Özellikleri / Bobin
Hedeflenen Süre	45 Dakika

🚀 *Bobin*

Bobin, birbiri üzerine sarılmış bir dizi iletken tel veya kablodan oluşan, elektrik akımının geçişini sağlamak için kullanılan bir cihazdır. Bobinler, genellikle elektromanyetik alanları üretmek veya elektriksel enerjiyi depolamak için kullanılır. Bobinler, elektrik akımının yönünü değiştirerek manyetik alanın yönünü değiştirir ve bu sayede elektrik akımını yönlendirmeye yardımcı olur.

📰 *Konu İçeriği*

- Bobinlerin çalışma mantığı ve özellikleri.
- Bobinlerin kullanım alanları.
- Bobinlerin çeşitleri.

📘 *Kazanım*

- Bobin kullanımı ve özellikleri hakkında bilgi sahibi olmak.
  - Bobinler, elektrik akımının yönünü değiştirerek magnetik alanın yönünü değiştirir.
  - Bobinler, elektriksel enerjiyi depolamak için kullanılır.
  - Bobinler, elektromanyetik alanları üretmek için kullanılır.
  - Bobinler, elektrik akımını yönlendirmeye yardımcı olur.
  - Bobinler, elektronik devrelerdeki sinyalleri filtrelemeye yardımcı olur.

✅ *Dersin Hedefleri*

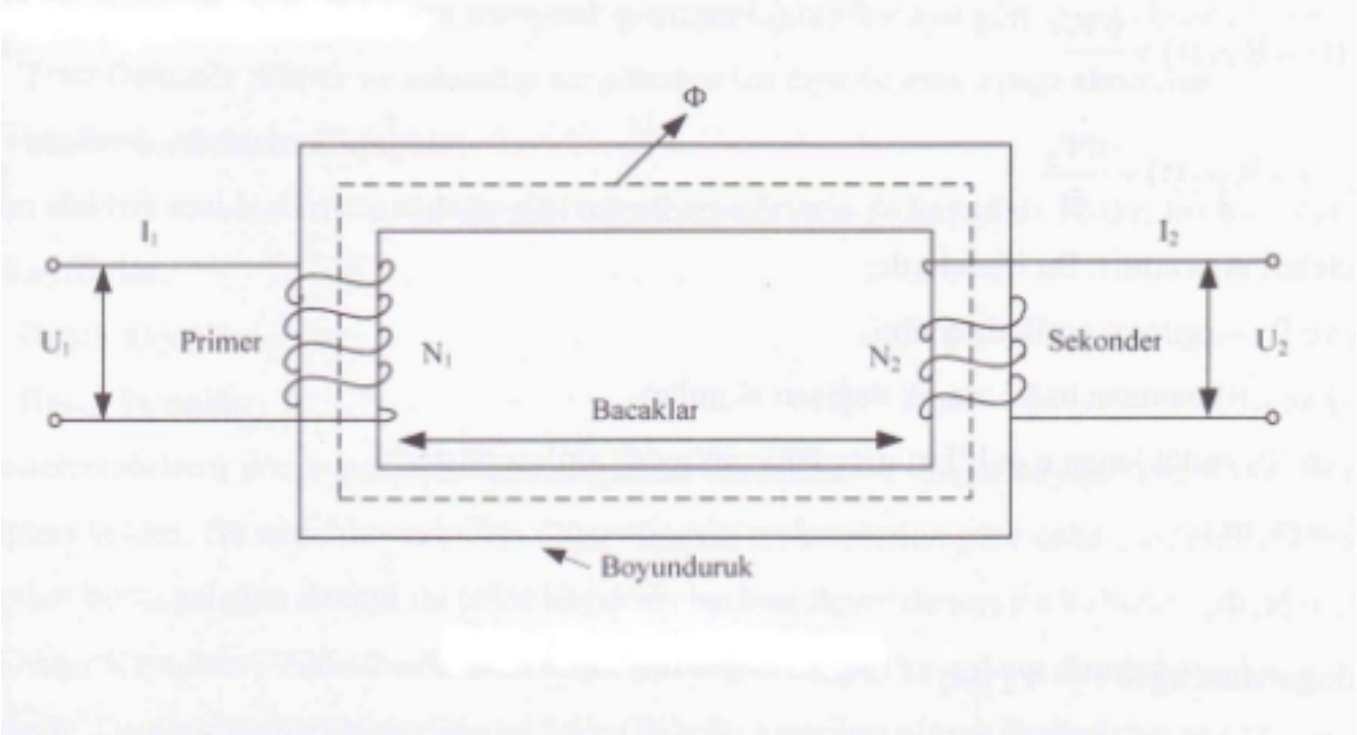
	Başlık	Hedef Kitle	Detaylar
1	☑ Bobin	☑ Eğitmen ☐ Öğrenciler	Eğitmenden bu bölümde Konu İçeriği bölümündeki konuları ele alması ve uygulanması beklenmektedir.
2	☑ Bobin	☐ Eğitmen ☑ Öğrenciler	Bobin kullanımı ve özellikleri hakkında bilgi sahibi olunması beklenmektedir.

Temel Elektronik: Transformatör (Trafo)

Müfredat Kodu / Ünite / Konu	İCT-400-2.4 / Elektronik Komponent ve Özellikleri / Transformatör (Trafo)
Hedeflenen Süre	45 Dakika

Transformatör (Trafo)

Elektronik, elektronik aygıtları çalıştırmak için küçük elektrik akımlarının nasıl kumanda edilmesi gerektiğini araştıran bilim dalıdır. Elektronik, geçtiğimiz yüzyılın başında ilk elektronik devre elemanlarının geliştirilmesiyle başlayan çok genç bir bilim dalı olmasına rağmen, günlük hayata en çok adapte olmuş bilim dalı olduğu söylenebilir. Öyle ki günümüzde elektronik cihazların olmadığı bir ortam düşünmek imkansızdır. Elektronik yüzyıllardır üzerinde çalışılan bilimler olan fizik , matematik ve kimya ile ilişkilidir. Bu bilimlere kıyasla çok genç bir bilim dalı olsa da elektronik şimdiden tüm dünyayı değiştiren sonuçlar yaratmıştır.



Konu İçeriği

- Transformatör yapısı ve özellikleri.
- Transformatörlerin kullanım alanları.
- Transformatörlerin faydaları.

Kazanım

- Transformatörler hakkında bilgi sahibi olmak.
  - Transformatörler elektrik enerjisini birbirinden farklı elektriksel potansiyellere sahip devreler arasında transfer eder.
  - Transformatörler bir girdi voltajını belirli bir oranla artırma veya azaltma işlevini gerçekleştirir.
  - Transformatörler sayesinde elektrik dağıtım sistemlerinde güç kayıpları azaltılabilir ve elektrik enerjisi kullanımı daha verimli hale getirilebilir.
  - Transformatörler farklı elektriksel gereksinimlere uygun olarak güç dağıtımı sağlar.
  - Transformatörler, elektrik makinesi olarak sınıflandırılır ve elektrik enerjisi üretim, dağıtım ve kullanımı süreçlerinde kullanılır.

✅ *Dersin Hedefleri*

	Başlık	Hedef Kitle	Detaylar
1	☑ Transformatör (Trafo)	☑ Eğitmen ☐ Öğrenciler	Eğitmenden bu bölümde, konu içeriği bölümündeki konuları uygulamalı olarak anlatması beklenmektedir. Anlatım sunum, görsel ve videolarla desteklenmelidir.
2	☑ Transformatör (Trafo)	☐ Eğitmen ☑ Öğrenciler	Öğrencinin kazanımlar bölümündeki çıktıları pekiştirmesi hedeflenmektedir.

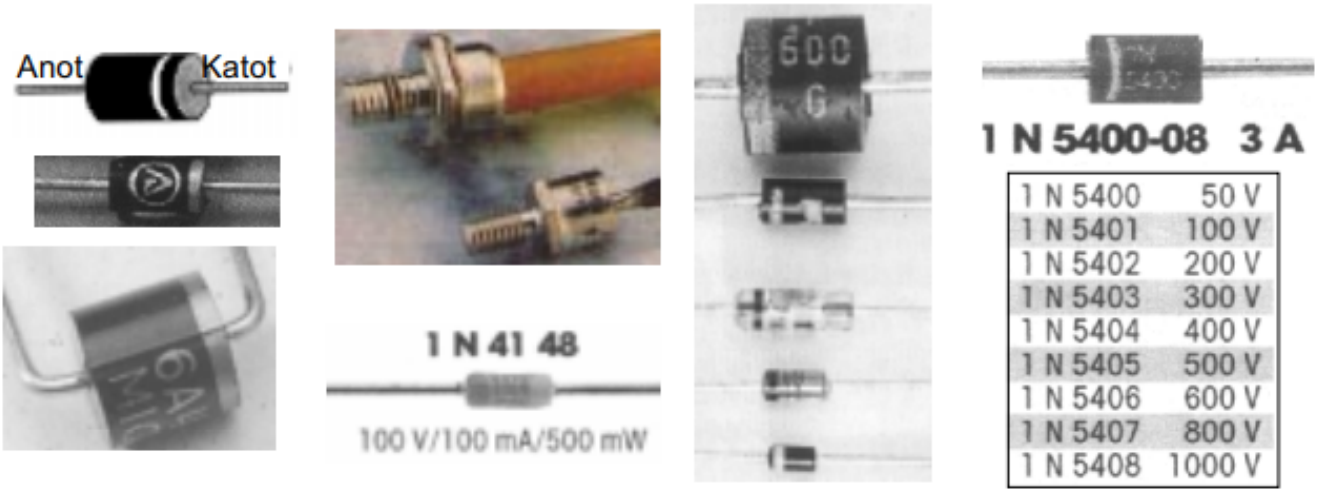
NOTLAR

Temel Elektronik: Diyot

Müfredat Kodu / Ünite / Konu	İCT-400-2.5 / Elektronik Komponent ve Özellikleri / Diyot
Hedeflenen Süre	90 Dakika

Diyot

Diyot, elektriksel bir geçiş yapabilen bir elektronik bileşenidir. Diyotlar, elektrik akımının bir yönde geçmesine izin verirken, diğer yönde geçmesini engelleyen bir katot ve anot arasından oluşur. Bu özellikleri sayesinde diyotlar, elektrik devrelerinde bir güç kaynağından bir yükün enerjisini akımın geçtiği yöne doğru yönlendirme işlevini yerine getirir. Diyotlar, genellikle elektronik devrelerde düzenleyici olarak kullanılır ve birçok elektronik cihazda bulunabilir.



Konu İçeriği

- Diyot nedir? Sembolleri nedir? Nerelerde kullanılırlar?
- Diyot Çeşitleri nelerdir, ne işe yarar?
  - **Silikon diyot:** Standart diyot çeşididir ve genellikle anot poluna silikon, katot poluna da karbon kaplıdır.
  - **Zener diyot:** Belirli bir gerilim seviyesi (zener gerilimi) üzerine çıktığında, zener diyot anot polundan katot poluna doğru elektrik akımını geçirmeye başlar. Bu özelliği nedeniyle zener diyot, gerilim düzeneği oluşturmak için kullanılır.
  - **LED (Işıklı Diod):** LED diyot, anot poluna bir metal oksit semikonduktör (MOS) tabakası ve katot poluna da bir metal kaplama ile kaplıdır. LED diyot, anot poluna uygulanan elektrik akımının enerjisini ışık olarak yayar.
  - **Lazer diyot:** Lazer diyot, küçük bir lazer çubuğu içerir ve lazer ışığını üretebilir.
  - **Schottky diyot:** Schottky diyot, silikon diyotlara göre daha hızlı çalışır ve düşük voltajlarda daha az direnç gösterir. Bu özellikleri nedeniyle Schottky diyot, hızlı anahtarlamalı devrelerde kullanılır.
  - **Varaktör diyot:** Varaktör diyot, anot polundaki metal oksit tabakasının kalınlığını değiştirerek elektrik akımını kontrol edebilir. Bu özelliği nedeniyle varaktör diyot, frekans düzeneği oluşturmak için kullanılır.
- Diyotların seri ve paralel bağlanması.
  - Seri bağlama ile 100 Volt'luk bir devre için ters dayanma gerilimi 50 Volt olan 2 adet 1N4001 diyot seri bağlanarak 100 Volt'a dayanabilen diyot yapılabilir.
  - Yüksek akımlı diyot elde etmek için paralel bağlama yapılır.
- Diyotların sağlamlık testi nasıl yapılır?
  - Ohmmetre komütatörü X1K ya da X10K kademesine alınır. Diyot bir yönde küçük direnç (300 W -3000 W), diier yönde büyük direnç (50 KW -200 KW) gösteriyorsa sağlamdır.

Kazanım

- Diyotların temel özelliklerini ve işlevlerini anlamak.
- Farklı diyot çeşitlerinin işlevlerini anlamak.
- Diyotların elektronik devrelerde nasıl kullanıldığını anlamak.

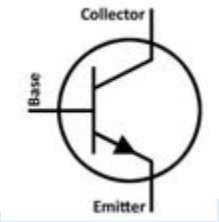
Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Diyot	☑ Eğitmen ☐ Öğrenciler	Eğitmen, Konu içeriği bölümündeki içerikleri öğrenciye uygulamalı olarak anlatmalıdır. Anlatımın sunum, video ve uygulamalı olması hedeflenmektedir.
2	☑ Diyot	☐ Eğitmen ☑ Öğrenciler	Öğrencinin bu bölümde diyotların kullanımı ve çeşitleri hakkında bilgi sahibi olması ve uygulama yapabiliyor olması hedeflenmektedir.

Temel Elektronik: Transistör

Müfredat Kodu / Ünite / Konu	İCT-400-2.6 / Temel Elektronik Komponent ve Özellikleri / Transistör
Hedeflenen Süre	45 Dakika

Transistör



Transistör, elektronik sinyalleri ve elektrik gücünü yükseltmek veya değiştirmek için kullanılan bir tür elektronik cihazdır. Her katmanı farklı bir elektrik yüküne sahip olan üç yarı iletken malzeme katmanından oluşur. Transistörler bilgisayarlar, radyolar, televizyonlar ve cep telefonları dahil olmak üzere çok çeşitli elektronik cihazlarda ve sistemlerde kullanılır. Modern elektroniğin temel bir bileşenidirler ve iletişim kurma ve bilgiye erişme şeklimizde devrim yaratmışlardır.

Konu İçeriği

- Transistörün Yapısı ve Çalışması
- NPN ve PNP Transistör Türleri
- Transistörlerin Sağlamlık Testi
- Transistörün anahtar olarak çalışması
- Transistörün Yükselteç olarak çalışması

Bilgi

Multimetre diyot kademesine alınır.

Problardan biri 3 uçtan herhangi birine (genelde orta uç) sabit tutulur ve diğer prob iki uca da tek tek değdirilir. Diğer iki ucun da multimetre değer göstermesi gerekir. Eğer sadece biri değer gösterirse ikisi de değer gösterene kadar sabit tutulan uç değiştirilir.

Sabit tutulan ucu ilk denemenizde kırmızı yapın. Bu şekilde değer bulamıyorsanız bu sefer siyah ucu sabit tutup kırmızıyı değiştirirsiniz.

Avometrenin bir probunu transistörün bir bacağında sabit tutarken. Diğer probu transistor ün diğer bacaklarına ayrı ayrı değdirdiğimizde ekranda değer gösteriyorsa o anda sabit tuttuğunuz probtaki bacak beyz bacağıdır. Değişken probta tuttuğumuz bacaklardan hangisi büyük değer gösteriyorsa o emiter bacağıdır. Diğer bacak yani küçük değer gösteren bacak ise kollektör bacağıdır. Bu değerler arasında çok büyük farklar yoktur.

Ölçülen transistörün PNP mi ? NPN mi? Olduğunu şöyle tesbit edebiliriz:

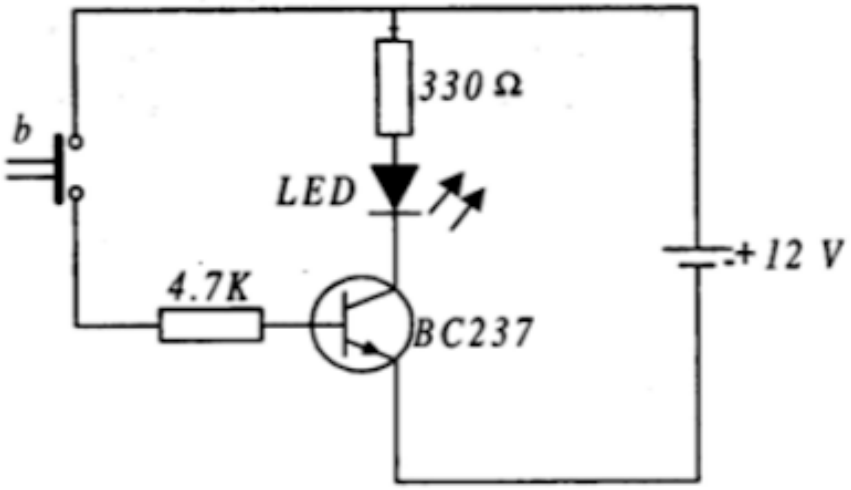
PNP : Emiter ve kollektörü buldugunuzda sabit tuttuğunuz bacaktaki (beyz) prob siyah renkli ise bu PNP tipinde olan bir transistördür.

NPN : Emiter ve kollektörü buldugunuzda sabit tuttuğunuz bacaktaki (beyz) prob kırmızı renkli ise bu transistör NPN tipi bir transistördür.

Örnek Uygulama

Transistörler devrelerde yükselteç ve anahtarlama elemanı olarak kullanılırlar. Transistörler küçük akımlarla büyük güçleri kontrol edebilen devre elemanlarıdır ve elektronik devrelerin olmazsa olmazıdır.

Bir transistörün emiter akımı (IE), kollektör akımı (IC) ile beyz akımının (IB) toplamına eşittir. IE= IB + IC ve IB= β . IB dir.



Bu devre NPN transistörünün anahtarlama elemanı olarak kullanılmasının kavranması amaçlanmaktadır. Devrede bulunan BC237 transistörü NPN tipi bir transistördür ve transistörün iletime geçebilmesi için beyz ucuna gerekli polarma verilmelidir.

Devreye enerji verildiğinde, ilk anda butona basılmamıştır ve transistörün beyz ucuna herhangi bir sinyal gelmez ve transistör iletime geçmez. Dolayısıyla transistörün beyz-emiter ucundan herhangi bir akım geçişi olmaz ve led diyot yanmaz. Butona basıldığı zaman transistörün beyzi pozitif polarma alır ve transistör iletime geçer. Böylece transistörün kollektör-emiter uçları arasından bir akım geçer ve led diyot ışık verir. Devrede bulunan transistörün iletime geçmesi için beyz-kollektör gerilimi yaklaşık 0.7 volt değerinde olmalıdır.

🛠️ *Kullanılan Malzemeler*

- 330 Ohm, 4.7 K Direnç.
- BC237 NPN transistörü.
- Led.
- Buton.
- 12 Volt Güç.

📘 *Kazanım*

- Transistörlerin nasıl çalıştığını anlamak ve elektronik devrelerde nasıl kullanılabileceklerini daha iyi anlayabilmek.
- Elektronik devrelerde çeşitli uygulamaları öğrenmek: Transistörler, elektronik devrelerde birçok farklı uygulama için kullanılır.
- İleri seviye elektronik devreleri anlamak. Transistörler, ileri seviye elektronik devrelerin önemli bir parçasıdır.
- Elektronik cihazların çalışma prensiplerini anlamak.

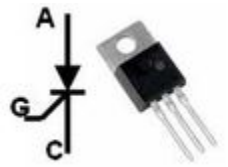
✅ *Dersin Hedefleri*

	Başlık	Hedef Kitle	Detaylar
1	☑ Transistör	☑ Eğitmen ☐ Öğrenciler	Eğitmenden, Transistörler bölümünde Konu İçeriği kısmındaki konu başlıklarını uygulamalı olarak açıklaması beklenmektedir.
2	☑ Transistör	☐ Eğitmen ☑ Öğrenciler	Öprenciden bu bölümde Transistörlerin devredeki kullanım amaçlarını bilerek devre kurabiliyor olması hedeflenmektedir.

Temel Elektronik: Tristör

Müfredat Kodu / Ünite / Konu	İCT-400-2.7 / Elektronik Komponent ve Özellikleri /
Hedeflenen Süre	45 Dakika

Tristör

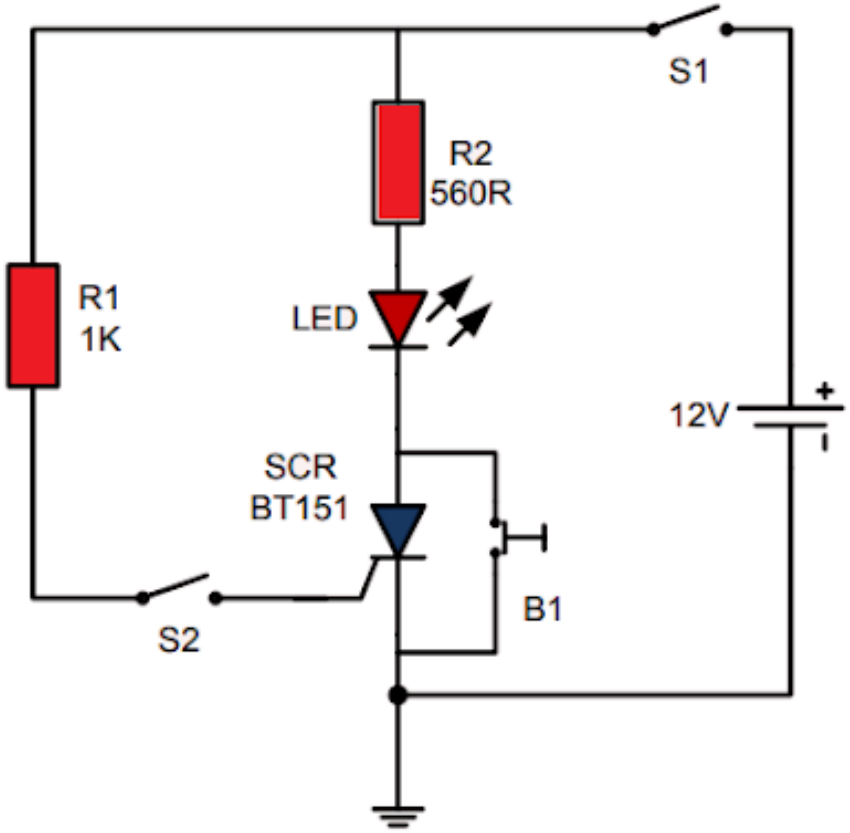


Tristörler güç elektroniği devrelerinde hızlı anahtarlama görevinde kullanılan, dört yarı iletken maddenin bileşiminden oluşan devre elemanlarıdır. Tristörlere Silikon Kontrollü Doğrultucu anlamına gelen SCR (Silicon Controlled Rectifiers) adı da verilir.

Konu İçeriği

- Tristör nedir?
- Tristörün çalışması ve özellikleri
- Tristörün tetiklenmesi
- Tristörün durdurulması

Örnek Uygulama



Yandaki devrede, tristörün çalışma mantığının öğrenilmesi amaçlanmaktadır. Devrede bulunan tristör anahtarlama görevi görmektedir. Tristörün anot, katot ve geyt olmak üzere üç adet pini bulunmaktadır. Geyt ucu tetikleme pinidir. Devreye enerji verildiğinde öncelikle S1 butonunun kapatılması gereklidir. S1 anahtarı kapatılsa bile tristör ilk anda iletimde değildir .Çünkü S2 anahtarı açıktır ve geyt ucu pozitif tetikleme alamadığından tristör iletime geçemez ve led ışık vermez. S2 anahtarı kapatılıp açıldığında tristörün geyt ucu gerekli tetiklemeyi alır ve tristör iletime geçer ve led ışık verir. S2 anahtarının açık kalması artık önemli değildir. Çünkü tristörün DC gerilimde bir kere tetikleme alması iletimde kalması için yeterlidir. Tristörü kesime sokmak için B1 anahtarına bir kez basılması yeterlidir. Bu durumda tristör kısa devre olur ve kesime gider ve lamba söner.

Kullanılan Malzemeler

- Led.
- Tristör.
- 3 Adet Buton.
- 1 Kohm Direnç.
- 560 Ohm Direnç.
- 12 Volt'luk Güç Kaynağı.

Kazanım

- Tristör kullanımının öğrenilmesi.
- Tristörlü devre kurulabiliyor olması.
- Tristörün tetiklenmesi ve durdurulmasının öğrenilmesi.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Tristör	☑ Eğitmen ☐ Öğrenciler	Eğitmenden Tristör konusunda içerik kısmındaki konu başlıklarını uygulamalı olarak anlatması hedeflenmektedir.
2	☑ Tristör	☐ Eğitmen ☑ Öğrenciler	Öğrenciden Tristörü devrede kullanabiliyor olması hedeflenmektedir.

Temel Elektronik: Optokuplör

Müfredat Kodu	İCT-400-2.8 / Elektronik Komponent ve Özellikleri / Optokuplör
Hedeflenen Süre	90 Dakika

Optokuplör

Optokuplör, bir optik ve bir elektrik bölümünden oluşan bir elektronik bileşenidir. Optokuplörler, optik ışıkları elektrik sinyallerine dönüştürme işlevini yerine getirir. Bu sayede optokuplörler, elektrik devrelerinde optik sinyalleri elektrik sinyalleriyle ayrıştırma ve izole etme gibi işlevleri gerçekleştirir. Optokuplörler, bir optik ışık kaynağı ve bir foto düğümden oluşur ve optik ışık kaynağı tarafından yayılan ışık, foto düğüm tarafından algılanarak elektrik sinyaline dönüştürülür. Optokuplörler, genellikle elektronik devrelerde kullanılır ve birçok elektronik cihazda bulunabilir.

Konu İçeriği

- Optokuplör nedir?
- Optokuplör ne işe yarar?
- Optokuplör'ün sembolü nedir?
- Optokuplör çalışma mantığı nedir?
- Optokuplör çeşitleri nelerdir?

Kazanım

- Optokuplörler: Öğrenciler, optokuplörlerin nasıl çalıştığını, optokuplörlerin optik sistemlerde nasıl kullanıldığını ve optokuplörlerin farklı tasarım seçeneklerinin avantajlarını ve dezavantajlarını öğrenebilirler.
- Optokuplörlerin uygulamaları: Öğrenciler, optokuplörlerin farklı uygulamalarını (örneğin, optikuplörlerin kullanımı telekomünikasyon sistemlerinde ve güvenlik sistemlerinde) öğrenebilirler.

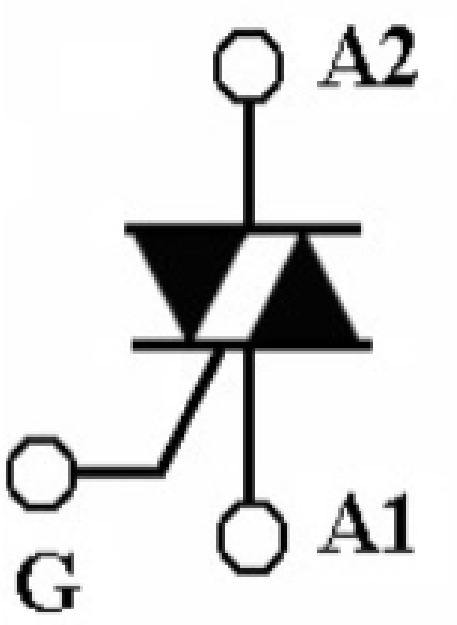
Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Optokuplör	☑ Eğitmen ☐ Öğrenciler	Optokuplör dersine giriş ve kavramların öğrencilere tanıtılması.
2	☑ Optokuplör	☐ Eğitmen ☑ Öğrenciler	Optokuplör kavramının pekiştirilmesi ve devrede kullanımı

Temel Elektronik: Triyak

Müfredat Kodu / Ünite / Konu	İCT-400-2.9 / Elektronik Komponent ve Özellikleri / Triyak
Hedeflenen Süre	90 Dakika

Triyak



Triyak, elektronik devrelerde kullanılan bir tür devre elemanıdır. Triyaklar, anahtarlama amaçlı kullanılırlar ve genellikle AC (değişen akım) devrelerinde kullanılırlar. Triyaklar, bir giriş ve iki çıkış terminalinden oluşur. Giriş terminaline bir sinyal verildiğinde, Triyakın çıkış terminallerinden birinin iletken hale gelmesiyle sinyal iletilir. Bu, Triyakların anahtarlama özelliği sayesinde olur. Triyaklar, güç anahtarı olarak da kullanılabilirler ve bu durumda, Triyakların çıkış terminallerinden birine bir motor gibi yükler bağlanır. Giriş terminaline bir sinyal verildiğinde, Triyak motoru çalıştırır ve durdurur.

Triyaklar, farklı tasarımları olmakla birlikte, genellikle üç şerit transistörden oluşur. Bu nedenle, triyakların dağıtımı üç şerit transistöre benzer şekilde yapılır.

Triyaklar, elektronik devrelerde anahtarlama işlemleri için kullanılır ve genellikle AC devrelerinde kullanılır. Örneğin, bir Triyak AC motorunu çalıştırmak ve durdurmak için kullanılabilir veya bir elektronik sistemde sinyallerin anahtarlanması için kullanılabilir.

Konu İçeriği

- Triyak nedir?
- Triyak nasıl çalışır?
- Triyak nerelerde kullanılır?
- Triyak sağlamlık testi.
- Triyak tetiklenmesi.

Bilgi

Triyak'ı avometre ile ölçerken; ölçü aleti Ohm metre konumuna X1 kademesine alınır. Sağlam bir Triyak'ın A2-A1 uçları her iki yönde de sorunsuz direnç gösterir. Triyak'ın Gate - A2 arası her iki yönde de sorunsuz direnç gösterir. Gate - A1 uçları arası her iki yönde de küçük (15-20 ohm) gösterir. Ayrıca A2 ve Gate uçları ile A2 ve A1 uçları her iki yönde de maksimum (açık devre) direnç gösterirse triyak sağlamdır.

Kazanım

- Triyak hakkında bilgi sahibi olmak.
- Triyak devreleri kurabiliyor olmak.

✅ *Dersin Hedefleri*

	Başlık	Hedef Kitle	Detaylar
1	☑ Triyak	☑ Eğitmen ☐ Öğrenciler	Triyak giriş ve kavramların öğrencilere tanıtılması.
2	☑ Triyak	☐ Eğitmen ☑ Öğrenciler	Triyak kavramlarının pekiştirilmesi ve uygulanması hedeflenmektedir.

ELEKTRONIK DEVRE UYGULAMALARI

Müfredat Kodu	İCT-400-3
Ders	Elektronik Devre Uygulamaları
Ders İçeriği	• Yarım Dalga Doğrultucu Devresi • Tam Dalga Doğrultucu Devresi • Buton İle Led Kontrol Uygulaması • NPN Transistörlü Anahtarlama Devresi • Flip Flop Devresi • Tristörlü Uygulama Devresi • Kondansatörlü Şarj Deşarj Devresi • Transistörlü Hırsız Alarm Devresi
Seviye	S2-ORTA
Ön Koşullar	Temel Elektronik kavramlar ve kompanentler konularını tamamlamış olmak
Amaç	Elektronik devre kurabiliyor ve yorumlayabiliyor olmak amaçlanmaktadır.
Hedef	Elektronik kompanentleri doğru devrede doğru bir şekilde kullanıyor olabilmek hedeflenmektedir.
Gereksinimler	Atölye Ortamı ve gerekli elektronik kompanentlerin olması
Kazanım	• Elektronik devre kurabilme • Devrelerin çalışma mantığının öğrenilmesi
Referanslar & Dokümantasyon	• http://elektronikderslerim.blogspot.com/2017/11/yarm-dalga-dogrultmac-devresi-half-wave.html • http://elektronikderslerim.blogspot.com/2017/11/tam-dalga-dogrultmac-devresi-full-wave.html • http://elektronikderslerim.blogspot.com/2017/04/delikli-plaket-uygulamas-bir-buton-bir.html • http://elektronikderslerim.blogspot.com/2016/12/npn-transistorun-anahtarlama-eleman.html • http://elektronikderslerim.blogspot.com/2016/02/ders-1flip-flop-devresi.html • http://elektronikderslerim.blogspot.com/2018/12/tristorlu-uygulama-devresi.html • http://elektronikderslerim.blogspot.com/2018/11/kondansatorlu-sarj-desarj-devresi.html • https://elektronikderslerim.blogspot.com/2018/04/transistorlu-hrsz-alarm-devresi.html
Durum	BEKLİYOR / UYGULANIYOR / TAMAMLANDI / DEĞERLENDİRİLDİ
Hedeflenen Süre	630 Dakika
Eğitim Günleri	Başlama: .../.../.... Bitiş: .../.../....

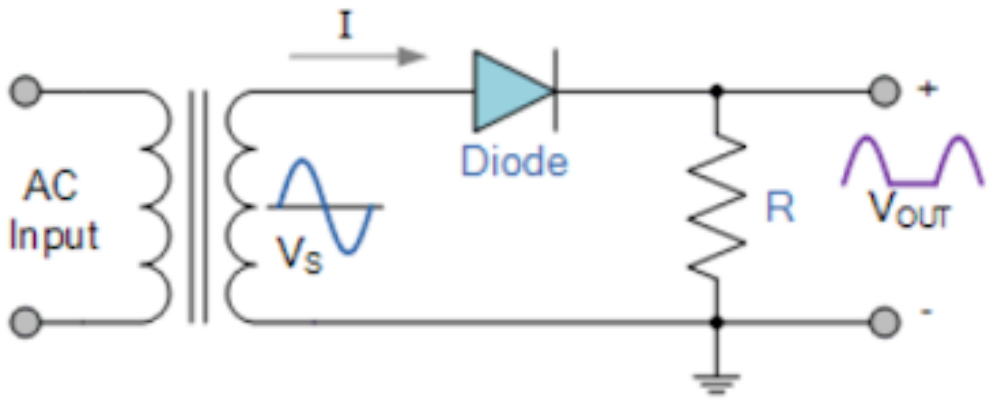
Yarım Dalga Doğrultucu Devresi

Müfredat Kodu / Ünite / Konu	İCT-400-3.1 / Elektronik Devre Uygulamaları / Yarım Dalga Doğrultucu Devresi
Hedeflenen Süre	90 Dakika

🚀 *Yarım Dalga Doğrultucu Devresi*

Yarım dalga doğrultmaç devresi , giriş sinyalinin bir alternansını kırpma işlemini gerçekleştirir. Bu devrede giriş sinyalinin pozitif ya da negatif kısmı yok edilir. Yarım dalga doğrultucu devreler güç kaynaklarında kullanılan doğrultma işlemlerinde en temel devredir.

⏳ *Uygulama*



Yarım dalga doğrultmaç devresinde, AC giriş sinyali transformatör ile düşürülür. Daha sonra bu sinyal diyot üzeriden kırpılarak yük üzerine aktarılır. Girişin pozitif alternansında diyodun anaduna pozitif sinyal gelir. Dolayısıyla diyot iletime geçer ve diyot üzerinden akım geçer ve diyot giriş sinyalini çıkışa aktarır. Böylece pozitif giriş sinyali çıkışta da görülür.

Girişin negatif alternansında diyodun anaduna eksi sinyal geldiğinden dolayı diyot açık bir anahtar gibi davranır. Dolayısıyla diyot üzerinden herhangi bir akım geçişi olmaz. Böylece negatif sinyal çıkışa aktarılmaz. Dolayısıyla girişin negatif sinyali çıkışta görülmez.

Devrede R direncine paralel bir kondansatör bağlandığında doğrultulmuş sinyal filtrelenir ve aşağıdaki sinyal çıkışta görülür.

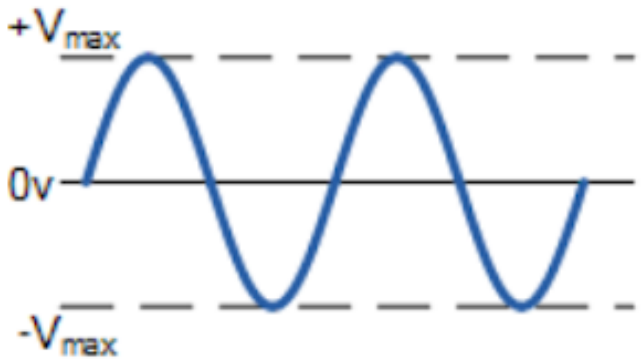
🛠️ *Kullanılan Malzemeler*

- 1N4001 tipi silisyum diyot.
- Transformatör.
- 1K lık direnç.
- Board ya da plaket.
- Jumper kablolar.
- Ölçü aleti.
- Osilaskop.

⚡ *İşlem Basamakları*

- Malzemelerin sağlamlık testini yapınız.
- Elemanları delikli plakete ya da boarda yerleştiriniz.
- Devre şemasına göre eleman bacaklarını devredeki gibi lehimleyiniz ya da borda yerleştiriniz.
- Enerji vermeden önce son bir kez devreyi kontrol ediniz.
- Devreye enerji veriniz ve devre çalışmasını gözlemleyiniz.
- Osilaskobun CH1 probunu devrenin girişine, CH2 probunu ise devrenin çıkışına bağlayınız.

- Giriş çıkış sinyallerini gözlemleyiniz.



Kazanım

- Yarım Dalga Doğrultucu Devresinin uygulanması.
- Yarım dalga doğrultmaç devresi, çalışma mantığını öğrenmek.
- Sinyal doğrultma işleminin mantığını kavramak.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Yarım Dalga Doğrultucu Devresi	☑ Eğitmen ☐ Öğrenciler	Yarım dalga doğrultmaç devresinin uygulanması hedeflenmektedir.
2	☑ Yarım Dalga Doğrultucu Devresi	☐ Eğitmen ☑ Öğrenciler	Yarım dalga doğrultmaç devresi, güç kaynaklarında doğrultma katında kullanıldığından bu devrenin çalışma mantığının öğrenilmesi ve sinyal doğrultma işleminin mantığının kavranması hedeflenmektedir.

Tam Dalga Doğrultucu Devresi

Müfredat Kodu / Ünite / Konu	İCT-400-3.2 / Elektronik Devre Uygulamaları / Tam Dalga Doğrultucu Devresi
Hedeflenen Süre	45 Dakika

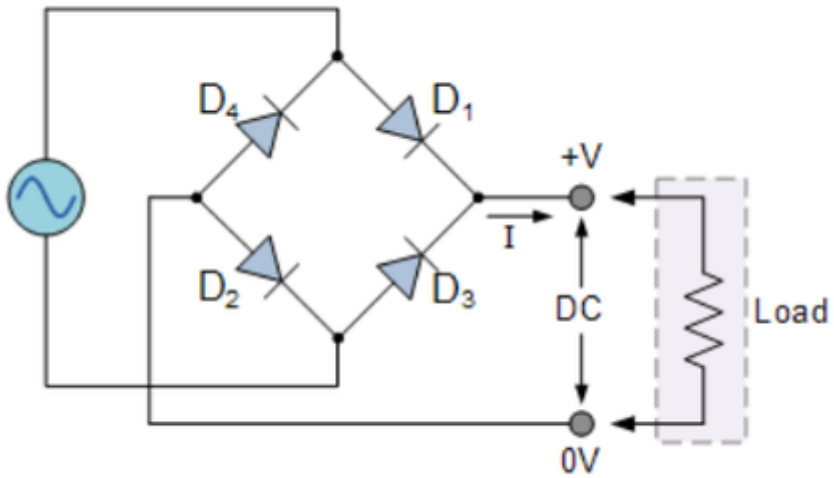
🚀 *Tam Dalga Doğrultmaç Devresi (Full Wave Rectifier)*

Tam dalga doğrultmaç devresi , giriş alternansının sadece pozitif ya da sadece negatif alternansını kırparak çıkışta bütün zaman aralıklarında sadece pozitif ya da sadece negatif alternan görünmesini sağlar.Tam dalga doğrultmaç iki şekilde yapılır:

1- Köprü tipi tam dalga doğrultmaç devresi.

2- Orta uçlu transformatörlü tam dalga doğrultmaç devresi.

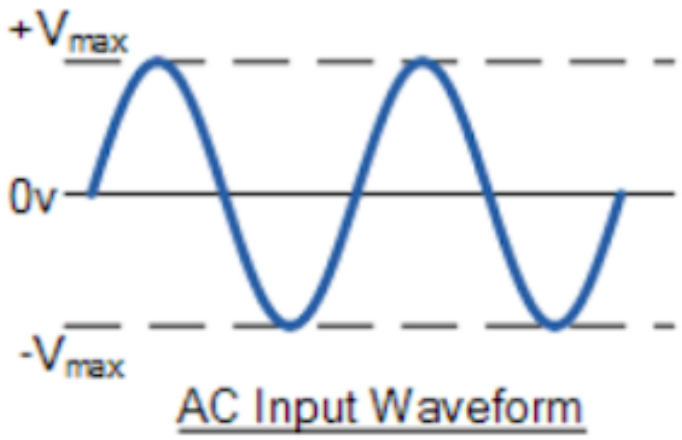
⏳ *Uygulama*

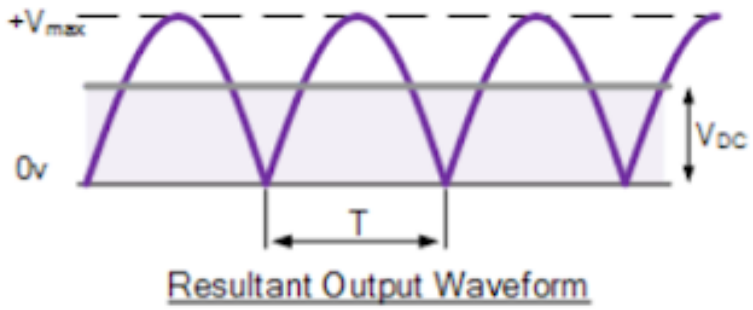


Pozitif giriş alternansta D1 ve D2 diyodunun anodu katoduna göre pozitif olduğundan bu iki diyot iletimdedir. D3 ve D4 diyotlarının ise anot uçları katot uçlarına göre negatif olduğundan bu iki diyot kesimdedir. Dolayısıyla ilk alternansta çıkış işareti pozitif olur.

Negatif giriş alternansında yani ikinci zaman aralığında D4 diyodunun katoduna negatif ve D3 diyodunun anoduna pozitif sinyal gelir. Dolayısıyla bu iki diyodun anot uçları katot uçlarına göre pozitiftir. D1 ve D2 diyotlarının ise anotları katotlarına göre nagatif olduğundan bu iki diyot kesimdedir ve bu iki diyottan akım geçmez. Dolayısıyla negatif alternansında pozitif sinyal D3 diyodu üzerinden yükün üst tarafına negatif sinyal ise D4 diyodu üzerinden yükün aşağı kısmına gelir ve böylece ikinci zaman aralığında da çıkış işareti pozitif olur.

Diğer zaman aralıklarında da çıkış işareti senkronize bir şekilde devam eder yani pozitif olur.





🛠️ *Kullanılan Malzemeler*

- 1N4001 diyot.
- Transformatör.
- 1K lık direnç.
- Board ya da plaket.
- Jumper kablolar.
- Ölçü aleti.
- Osilaskop.

⚡ *İşlem Basamakları*

- Malzemelerin sağlamlık testini yapınız.
- Elemanları delikli plakete ya da boarda yerleştiriniz.
- Devre şemasına göre eleman bacaklarını devredeki gibi lehimleyiniz ya da borda yerleştiriniz.
- Enerji vermeden önce son bir kez devreyi kontrol ediniz.
- Devreye enerji veriniz ve devre çalışmasını gözlemleyiniz.
- Osilaskobun CH1 probunu devrenin girişine, CH2 probunu ise devrenin çıkışına bağlayınız.
- Giriş çıkış sinyallerini gözlemleyiniz.

> 🟦 *Kazanım*
>
> - Bu konudan öğrencilerin kazanımı yazılmalı.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Tam Dalga Doğrultmaç Devresi (Full Wave Rectifier)	☑ Eğitmen ☐ Öğrenciler	Eğitmen bu bölümde devreyi kurudurup, osiloskop ile çıkış sinyali ölçme uygulamasını uygulamalı olarak tamamlatmalıdır.
2	☑ Tam Dalga Doğrultmaç Devresi (Full Wave Rectifier)	☐ Eğitmen ☑ Öğrenciler	Köprü tipi tam dalga doğrultmaç devresinin çalışma mantığını anlamak ve giriş çıkış sinyal şekillerini çizebilmek hedeflenmektedir.

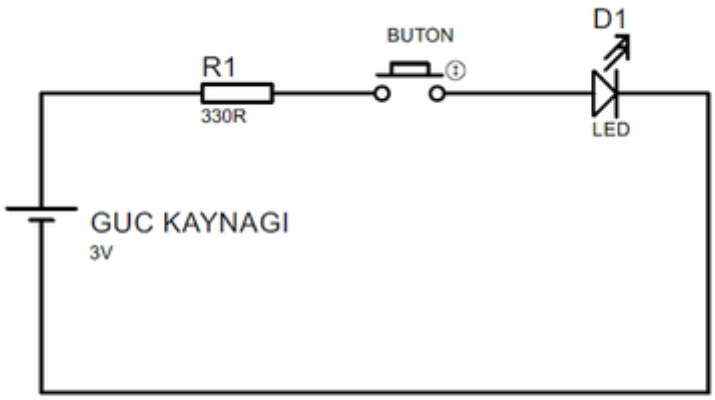
Buton İle Led Kontrol Uygulaması

eMüfredat Kodu	İCT-400-3.3 / Elektronik Devre Uygulamaları / Buton İle Led Kontrol Uygulaması
Hedeflenen Süre	45 Dakika

🚀 *Buton İle Led Kontrol Uygulaması*

Bu devre ile bir butona bağlı olan led diyodun butona bağlı olarak ışık vermesi sağlanacaktır. Bu devrenin yapılma amacı elektronik devrenin board, bakır plaket ya da delikli plakete kurulma aşamalarını gerçekleştirebilmektir.

⏳ *Uygulama*



Devreye enerji verildiği ilk anda led diyot sönük haldedir. Butona basıldığında akım R1 direnci üzerinden diyoda akar. Böylece led diyot ışık verir.

🛠️ *Kullanılan Malzemeler*

- 1N4001 Diyot.
- Transformatör.
- 1K Iık Direnç.
- Board ya da Plaket.
- Jumper Kablolar.
- Ölçü Aleti.
- Osilaskop.

⚡ *İşlem Basamakları*

- Malzemelerin sağlamlık testini yapınız.
- Elemanları delikli plakete ya da boarda yerleştiriniz.
- Devre şemasına göre eleman bacaklarını devredeki gibi lehimleyiniz.
- Enerji vermeden önce son bir kez devreyi kontrol ediniz.
- Devreye enerji veriniz ve devre çalışmasını gözlemleyiniz.
- Aşağıdaki gerekli yerleri tamamlayınız.

	Gerilim (Volt)

R1	
Led	
Güç Kaynağı	
	Akım (Amper)
Devre Akımı	

Kazanım

- Devre kurabiliyor olmak.
- Devredeki ölçümleri yapabiliyor olmak.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Buton İle Led Kontrol Uygulaması	☑ Eğitmen ☐ Öğrenciler	Buton İle Led Kontrol Uygulamasının yapımında öğrencilere yol görtermek.
2	☑ Buton İle Led Kontrol Uygulaması	☐ Eğitmen ☑ Öğrenciler	Buton İle Led Kontrol Uygulamasının tamamlanarak ölçümlerinin yapılıyor olması hedeflenmektedir.

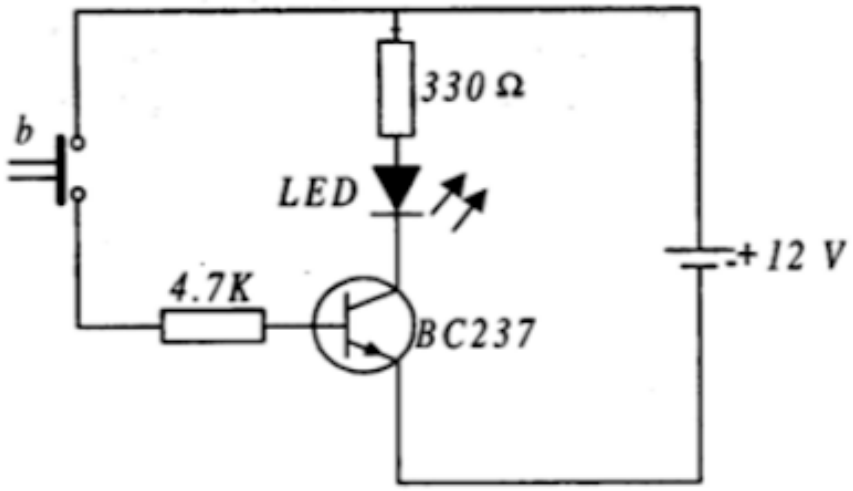
NPN Transistörlü Anahtarlama Devresi

Müfredat Kodu / Ünite / Konu	İCT-400-3.4 / Elektronik Devre Uygulamaları / NPN Transistörlü Anahtarlama Devresi
Hedeflenen Süre	45 Dakika

🚀 *Transistörlü Anahtarlama Devresi*

NPN Transistörlü Anahtarlama Devresi, transistörün anahtarlama elemanı olarak kullanılmasını anlatan önemli bir devredir. Bu devrede transistörün kesim ve doyum durumuna göre, devrede bulunan led diyot ışık verecektir.

⏳ *Uygulama*



Transistörler devrelerde yükselteç ve anahtarlama elemanı olarak kullanılırlar. Transistörler küçük akımlarla büyük güçleri kontrol edebilen devre elemanlarıdır ve elektronik devrelerin olmazsa olmazıdır.

Bir transistörün emiter akımı (IE), kollektör akımı (IC) ile beyz akımının (IB) toplamına eşittir.
IE= IB + IC ve IB= β . IB dir. Transistörlerin β değerleri transistör katologlarında verilmiştir.

Bu devre NPN transistörünün anahtarlama elemanı olarak kullanılmasının kavranması için kullanışlı bir devredir. Devrede bulunan BC237 transistörü NPN tipi bir transistördür ve transistörün iletime geçebilmesi için beyz ucuna gerekli polarma verilmelidir.

Devreye enerji verildiğinde, ilk anda butona basılmamıştır ve transistörün beyz ucuna herhangi bir sinyal gelmez ve transistör iletime geçmez. Dolayısıyla transistörün beyz-emiter ucundan herhangi bir akım geçişi olmaz ve led diyot yanmaz. Butona basıldığı zaman transistörün beyzi pozitif polarma alır ve transistör iletime geçer. Böylece transistörün kollektör-emiter uçları arasından bir akım geçer ve led diyot ışık verir. Devrede bulunan transistörün iletime geçmesi için beyz-kollektör gerilimi yaklaşık 0.7 volt değerinde olmalıdır. Bu değerler farklı transistörler için farklı değerleri alabilir.

🛠️ *Kullanılan Malzemeler*

- 330 ohm, 4.7 k Direnç.
- BC237 NPN Transistörü.
- Led.
- Buton.
- 12 Voltluk Besleme Gerilimi.

⚡ *İşlem Basamakları*

- Devreyi kurunuz ve aşağıdaki görevleri tamamlayınız.

	Volt (V)
Anahtar açıkken Transitör Beyz- Emiter arası gerilim.	
Anahtar kapalıyken Transitör Beyz- Emiter arası gerilim.	
Anahtar kapalıyken Len uçları arasındaki gerilim.	

Kazanım

- Transistörün çalışma prensibinin pekiştirilmesi.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ NPN Transistörlü Anahtarlama Devresi	☑ Eğitmen ☐ Öğrenciler	Eğitmen bu bölümde devreyi kurdurup, osiloskop ile çıkış sinyali ölçme uygulamasını uygulamalı olarak tamamlatmalıdır.
2	☑ NPN Transistörlü Anahtarlama Devresi	☐ Eğitmen ☑ Öğrenciler	Köprü tipi tam dalga doğrultmaç devresinin çalışma mantığını anlamak ve giriş çıkış sinyal şekillerini çizebilmek hedeflenmektedir.

Flip Flop Devresi

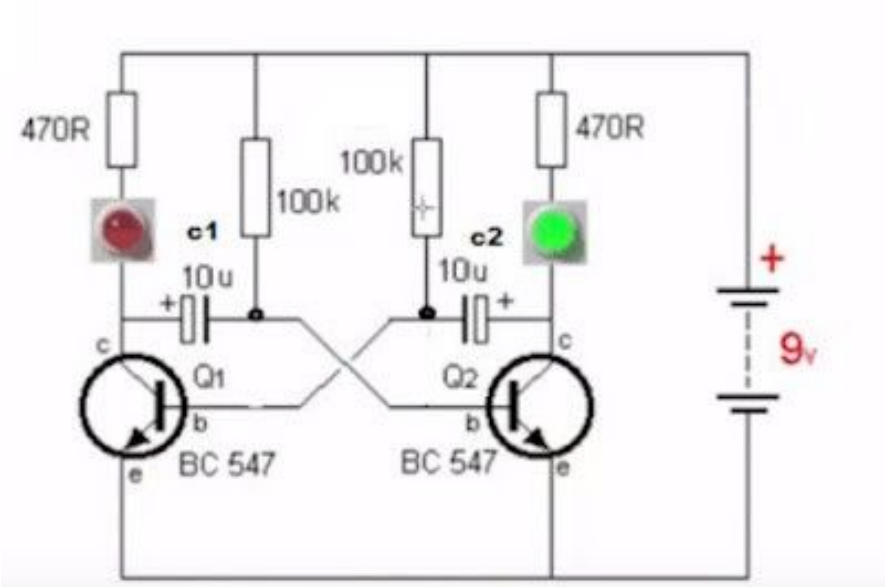
Müfredat Kodu / Ünite / Konu	İCT-400-3.5 / Elektronik Devre Uygulamaları / Flip Flop Devresi
Hedeflenen Süre	45 Dakika

Flip Flop Devresi

Flip flop devresi, karşılıklı ledlerin ya da lambaların yanıp sönmesi mantığına dayanarak oluşturulan bir elektronik devredir . Bu devrede amaç bir led yanarken diğerinin sönmesidir.

> Bir demir döküm atölyesini düşünürsek, döküm ustasının elinde iki adet çekiç olduğunu varsayalım. Usta demiri döverken elindeki iki çekici ayna anda demire vurmaz. Çekicin bir tanesini demire vururken diğerini havada tutar ve bu işi senkronize bir şekilde gerçekleştirir.

Uygulama



Devreye enerji verildiğinde kondansatörlerin şarj olma durumuna göre iki kondansatörden biri diğerine göre daha erken şarj olur. Bunun nedeni kondansatörlerin yapılarından kaynaklanmaktadır. Değerleri aynı olsa bile her kondansatör aynı anda şarj ya da deşarj olmaz. Q1 transistörüne bağlı kondansatörün diğerine göre daha erken şarj olduğunu varsayalım. Yani c2 kondansatörünün c1 kondansatörüne göre önce şarj olduğunu varsayalım. Dolayısıyla Q1 transistörünün beyzi c2 kondansatörü üzerinden şarj olur. c2 kondansatörüne bağlı led yani yeşil led yanar . Q1 transistörü iletime geçtiği anda c2 kondansatörü bu sefer deşarj olur. Tam bu sırada c1 kondansatörü şarj olur ve Q2 transistörünü iletime sokar ve kırmızı led yanar. Bu işlem senkronize bir şekilde gerçekleşir.

Kullanılan Malzemeler

- 2x470 ohm Direnç.
- 2x10 Kohm Direnç.
- 2x100 uF Kondansatör.
- 2xBC547 Transistör.
- 1 adet 9V pil ya da güç kaynağı.

> ***Kazanım***
>
> - Flip Flop devresinin çalışma mantığını anlamak.
> - Devreyi kurabiliyor olmak.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Flip Flop Devresi	☑ Eğitmen ☐ Öğrenciler	Flip Flop Devresi kurulumunda ve çalışma mantığının aktarılmasında yardımcı olunması hedeflenmektedir.
2	☑ Flip Flop Devresi	☐ Eğitmen ☑ Öğrenciler	Flip Flop Devresi kurulumunun yapılması ve çalışma mantığının öğrenilmesi hedeflenmektedir.

NOTLAR

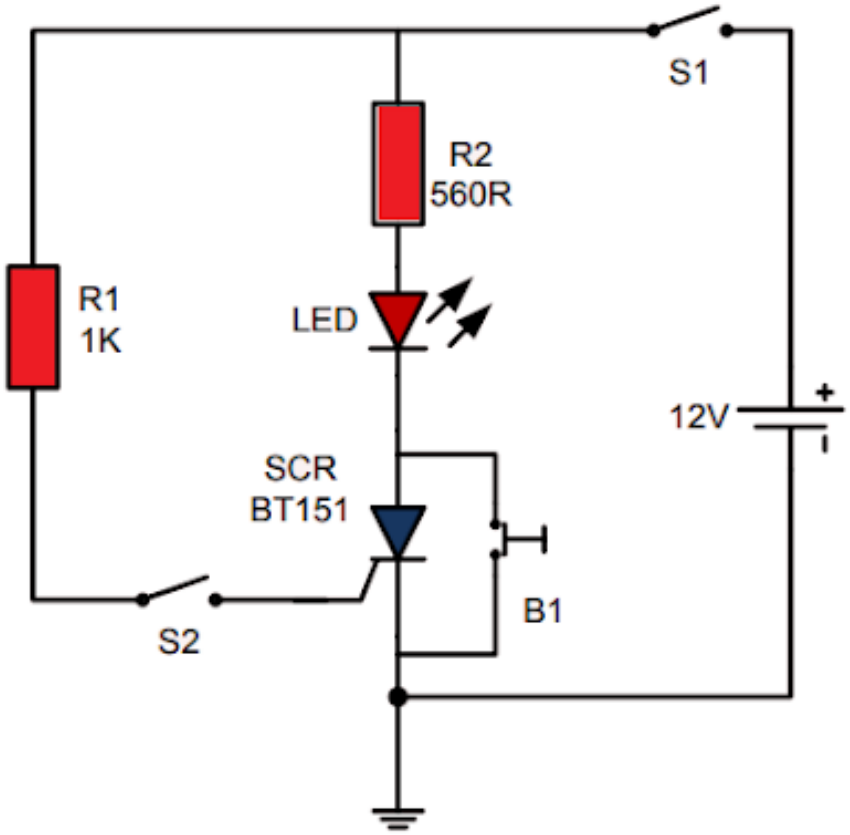
Tristörlü Uygulama Devresi

Müfredat Kodu	İCT-400-3.6 / Elektronik Devre Uygulamaları / Tristörlü Uygulama Devresi
Hedeflenen Süre	135 Dakika

🚀 *Tristörlü Uygulama Devresi*

Tristör, devrelerde anahtarlama elemanı olarak kullanılırlar. Tristörün çalışması şekli aslında transistörlere benzer. Tristörün iletime geçmesi için, geyt ucundan tetikleme alması yeterlidir.

⏳ *Uygulama*



Bu devre, tristörün çalışma mantığının öğrenilmesi için önemlidir. Devrede bulunan tristör anahtarlama görevi görmektedir. Tristörün anot, katot ve geyt olmak üzere üç adet pini bulunmaktadır. Geyt ucu tetikleme pinidir. Devreye enerji verildiğinde öncelikle S1 butonunun kapatılması gereklidir. S1 anahtarı kapatılsa bile tristör ilk anda iletimde değildir .Çünkü S2 anahtarı açıktır ve geyt ucu pozitif tetikleme alamadığından tristör iletime geçemez ve led ışık vermez. S2 anahtarı kapatılıp açıldığında tristörün geyt ucu gerekli tetiklemeyi alır ve tristör iletime geçer ve led ışık verir. S2 anahtarının açık kalması artık önemli değildir. Çünkü tristörün DC gerilimde bir kere tetikleme alması iletimde kalması için yeterlidir. Tristörü kesime sokmak için B1 anahtarına bir kez basılması yeterlidir. Bu durumda tristör kısa devre olur ve kesime gider ve lamba söner.

🛠️ *Kullanılan Malzemeler*

- Tristör.
- 3 adet Buton.
- 1 kohm Direnç.
- 560 ohm Direnç.
- 12 Volt'luk Güç Kaynağı.
- Led.

⚡ *İşlem Basamakları*

- Devrede bulunan malzemelerin sağlamlık kontrolünü yapınız.
- Devreyi bread board üzerine kurunuz.
- S2 anahtarını kapatarak tristörü iletime sokunuz ve ledin ışık verdiğini gözlemleyiniz.
- Daha sonra S2 anahtarından elinizi çekiniz.
- B1 anahtarını kapatarak tristörü kesime sokunuz ve ledin söndüğünü gözlemleyiniz.

Kazanım

- Tristör devre elemanının uygulamasını yapmak.
- Devre çalışma mantığının öğrenilmesi.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Tristörlü Uygulama Devresi	☑ Eğitmen ☐ Öğrenciler	Bu uygulamada amaç, tristörlü çalışma şeklini anlayarak bunu devre üzerinde uygulayabilmektir.
2	☑ Tristörlü Uygulama Devresi	☐ Eğitmen ☑ Öğrenciler	Bu uygulamada amaç, tristörlü çalışma şeklini anlayarak bunu devre üzerinde uygulayabilmektir.

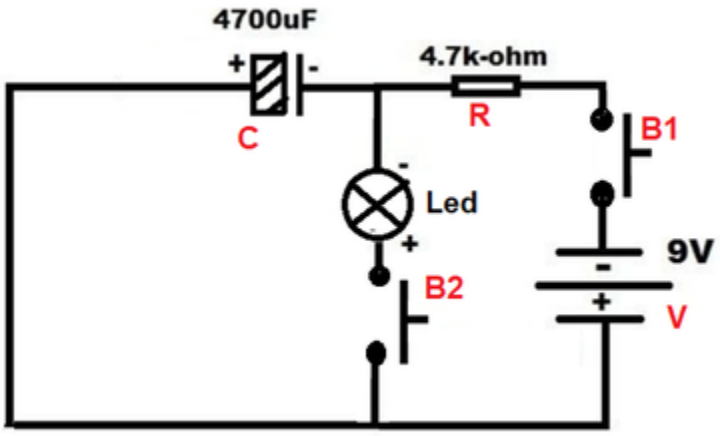
Kondansatörlü Şarj Deşarj Devresi

Müfredat Kodu / Ünite / Konu	İCT-400-3.7 / Elektronik Devre Uygulamaları / Kondansatörlü Şarj Deşarj Devresi
Hedeflenen Süre	90 Dakika

🚀 *Kondansatörlü Şarj Deşarj Devresi*

Kondansatör, elektriği şarj eden ve gerektiğinde deşarj eden devre elemanıdır. Kondansatörlerin çalışması pillerin çalışmasına benzer. Piller de üzerlerindeki elektriği belli bir süre sonra deşarj eden devre elemanlarıdır. :Bu devrede kondansatörün şarj ve deşarj olayının öğrenilmesi için gerekli devre kurulup devrenin çalışmasının öğrenilmesi amaçlanmıştır.

⏳ *Uygulama*



Devreye ilk enerji verildiği anda B1 butonuna basılı olmadığından dolayı devreden enerji akmaz , kondansatörün şarj olayı gerçekleşmez ve ledin ışık vermesi söz konusu olmaz. İlk olarak B1 butonuna basılır. Kondansatör şarj olmaya başlar. Kondansatör şarj oluktan sonra B1 butonu açık konuma getirilir ve B2 butonuna basılır. B2 butonuna basıldığında led diyot kondansatör üzerinden enerjisini alır ve ışık vermeye başlar. Kondansatör üzerindeki enerjiyi boşaltana kadar led ışık verir. Kondansatörün üzerindeki enerji boşaldığında artık ledin ışık vermesini sağlayacak herhangi bir voltaj değeri kalmamıştır. Böylece led ışık vermez. Devrenin çalışması bu döngü içerisinde devam eder.

🛠️ *Kullanılan Malzemeler*

- Bread board.
- Ölçü aleti.
- Led.
- Kutuplu Kondansatör.
- İki adet buton.
- 9 voltluk pil.

⚡ *İşlem Basamakları*

- Öncelikle devrede kullanılan malzemelerin sağlamlık kontrollerini yapınız.
- Devreyi kurmadan önce devrenin çalışma mantığını anlayınız.
- Devreyi bread board üzerine kurunuz.
- B1 butonuna basarak kondansatörü şarj ediniz. Kondansatör şarj olduktan sonra üzerindeki gerilimi ölçü aleti ile ölçünüz.
- B1 butonundan elinizi çekiniz . B2 butonuna basarak ledin yanmasını sağlayınız ve ölçü aleti ile led üzerindeki voltajı ölçünüz.

- Aşağıdaki tabloyu doldurunuz.

	Volt (V)
B1 kapalı, B2 açıkken kondansatör gerilimi	
B1 açık, B2 kapıyken kondansatör gerilimi	

Kazanım

- Kondansatör çalışma mantığını uygulamalı olarak öğrenme.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Kondansatörlü Şarj Deşarj Devresi	☑ Eğitmen ☐ Öğrenciler	Kondansatör çalışma mantığını öğrenciye aktarma hedeflenmektedir.
2	☑ Kondansatörlü Şarj Deşarj Devresi	☐ Eğitmen ☑ Öğrenciler	Kondansatör çalışma mantığını uygulamalı olarak öğrenme hedeflenmektedir.

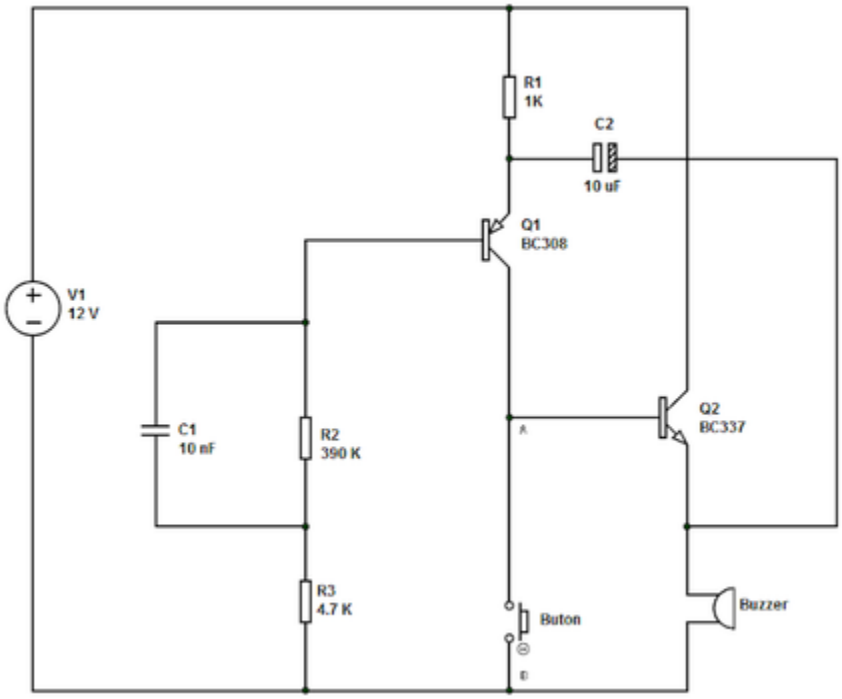
Transistörlü Hırsız Alarm Devresi

Müfredat Kodu / Ünite / Konu	İCT-400-3.8 / Elektronik Devre Uygulamaları / Transistörlü Hırsız Alarm Devresi
Hedeflenen Süre	135 Dakika

Transistörlü Hırsız Alarm Devresi

Bu devre hırsız alarm devresidir. Devrede doğru analiz yapabilmek için kullanılan devre elemanlarının çalışma yapılarının bilinmesi gerekmektedir. Bu devrenin kurulum amacı devredeki giriş elemanlarının durumuna göre çıkış elemanlarının çalışma şekillerini iyi bir şekilde anlamaktır.

Uygulama



Kullanılan Malzemeler

- 1 adet 390K direnç.
- 1 adet 4.7K direnç.
- 1 adet 1K direnç.
- 1 adet BC308 PNP Transistör.
- 1 adet BC307 NPN Transistör.
- 1 adet 10 nF kutupsuz kondansatör.
- 1 adet 10 uF kutuplu kondansatör.
- 1 adet Buzzer.
- 12 volt DC güç kaynağı.
- Bread board ya da plaket.

Kazanım

- Kondansatör çalışma mantığını uygulamalı olarak öğrenme.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Transistörlü Hırsız Alarm Devresi	☑ Eğitmen ☐ Öğrenciler	Transistörlü Hırsız Alarm Devresi çalışma mantığını öğrenciye aktarma hedeflenmektedir.
2	☑ Transistörlü Hırsız Alarm Devresi	☐ Eğitmen ☑ Öğrenciler	Transistörlü Hırsız Alarm Devresi çalışma mantığını uygulamalı olarak öğrenme hedeflenmektedir.

İcathane Eğitim Müfredatı

<YAZILIM>

HTML & CSS | Javascript | C# | C++ | ASP.NET Core | JQuery

TÜGVA İCATHANE

HACKERCAN EĞITIM MÜFREDATI

HACKERCAN EĞİTİM MÜFREDATI														
EĞİTİM İÇERİĞİ / HAFTA	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.
Hacker Can (İCT-102)														
Hacker Can : Arayüz Tanıtımı														
Hacker Can : Algoritma														
Hacker Can: Söz Dizimi														
Hacker Can: Objeler														
Hacker Can: Karakterler														
Hacker Can: Kere Döngüsü														
Hacker Can: İç İçe Kere Döngüsü														
Hacker Can: Değişkenler														
Hacker Can: Bitinceye Kadar Döngüsü														
Hacker Can: Fonksiyonlar														
Hacker Can: Parametreli Fonksiyonlar														
Hacker Can: Hazır Fonksiyonlar														
Hacker Can: if-else Kontrol Yapıları														

HACKER CAN EĞITIMI

Müfredat Kodu	İCT-102
Ders	Hacker Can
Ders İçeriği	• Hacker Can: Arayüz Tanıtımı • Hacker Can: Algoritma • Hacker Can: Söz Dizimi • Hacker Can: Objeler • Hacker Can: Karakterler • Hacker Can: Kere Döngüsü • Hacker Can: İç İçe Kere Döngüsü • Hacker Can: Değişkenler • Hacker Can: Bitinceye Kadar Döngüsü • Hacker Can: Fonksiyonlar • Hacker Can: Parametreli Fonksiyonlar • Hacker Can: Hazır Fonksiyonlar • Hacker Can: if-else Kontrol Yapıları
Seviye	İCT-S1: BAŞLANGIÇ
Ön Koşullar	Teknoloji ve yazılımla ilgili olmak.
Amaç	Hacker Can çocuklara Türkçe kodlama mantığını öğretmeyi amaçlayan bir eğitim sitesidir.
Hedef	Çocuklara erken yaşta kodlama mantığını öğretmeyi hedeflemiştir.
Gereksinimler	Bilgisayar ve Ortamda İnternet Ağı
Kazanım	• Öğrencilerin algoritma, problem çözme yeteneği ve programlama becerileri gelişir. • Çocuklara en keyifli saatlerinde eşlik ederek ve oyun tadında programlamayı öğretir. • Akıllı telefon ve tablet entegrasyonu ile internete bağlı mobil cihazlar üzerinden eğitim sürdürür. • Öğrencinin algoritma ve program çözme yeteneği gelişir.
Referanslar & Dokümantasyon	• https://www.hackercan.com/tr/ • https://www.kodpit.com/tr/hacker-can/ • http://www.cankodyaz.com/egitimler/courses/hacker_can.html
Durum	BEKLIYOR / UYGULANIYOR / TAMAMLANDI / DEĞERLENDIRILDI
Hedeflenen Süre	1260 Dakika
Eğitim Günleri	Başlama: .../.../.... Bitiş: .../.../....

HACKER CAN: ARAYÜZ TANITIMI

Müfredat Kodu / Ünite / Konu	İCT-102-1 / Hacker Can / Arayüz Tanıtımı
Hedeflenen Süre	90 Dakika

🚀 Hacker Can Nedir ?

Hacker Can öğrencilerin ana dilde kodlamayı öğrenme ve Türkçe kod yazarak yazılım geliştirme becerisi kazandırmak için hazırlanan Türkiye'nin ilk eğitim platformudur. Bu platformda okuma yazmayı bilen her öğrenci, kendi yaş grubu ve seviyesine göre programlama alanında kendini geliştirebilir. Hem Türkçe, hem İngilizce olarak kullanılabilen platformumuz öğrencilere algoritma, problem çözme yeteneği ve programlama (gerçek kod yazma) eğitimi vermeyi hedeflemektedir.

💻 Hacker Can İle Kodlama Nasıl Öğrenilir?

Hemen hemen bütün yazılım dillerinin ortak bir algoritma yapısı bulunmaktadır. Hacker Can ile bu algoritma yapısını Türkçe kodlama yapıp oyun oynayarak öğrenebilirsiniz. Tüm programlama dillerinin ortak yapısı olan değişkenler, diziler, nesneler, fonksiyonlar, döngüler ve sorgular Türkçeleştirilmiş şekilde oyun içerisinde müfredata uygun bir şekilde öğretiliyor. Kodlamanın Türkçe olması sayesinde de örneğin bir döngü mantığı çok kolay anlaşılabiliyor. Aynı zamanda İngilizce dil desteği sayesinde kodlamanızı İngilizce olarak da yapabilirsiniz. Yani hem Türkçe hem de İngilizce olarak kodlamayı öğrenebilirsiniz. Sonrasında hangi gerçek yazılım diline geçmek isterseniz isteyin, o yazılım diline adaptasyonunuz çok basit olacaktır.

🎗️ Tanıtım Arayüz

Hacker Can'ı hareket ettirirken gerçek anlamda ilk kodların yazılması. Bu kodların belli bir düzen içerisinde sırayla işlediğinin öğrenilmesi.

Ders sonucunda öğrencilerin Hacker Can'ın kod paneline ve ekranına hakim olmaları gerekmektedir.

📘 Kazanım

- Öğrencilerin algoritma, problem çözme yeteneği ve programlama (gerçek kod yazma) becerileri gelişir.
- Öğrenciye en keyifli saatlerinde eşlik ederek oyun tadında programlamayı öğretir.
- Akıllı telefon ve tablet entegrasyonu ile internete bağlı cihazlar üzerinden öğretimi sürdürür.
- Eğitim sonunda öğrenciye sertifika verilir.

✅ Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Site üzerine öğrencilerin kayıtlarının yapılması ☑ Öğrencilerin arayüz tanıtımının öğretilmesi.	☑ Eğitmen ☐ Öğrenciler	https://www.hackercan.com/tr/ sitesi üzerine öğrencilerin kayıtlarının yapılması. Öğrencilerin arayüz tanıtımının öğretilmesi.
2	☑ https://www.hackercan.com/tr/ kavramların pekiştirilmesi.	☐ Eğitmen ☑ Öğrenciler	Platform üzerindeki kavramların pekiştirilmesi.

👩‍🏫 Eğitmen Yol Haritası

1	Öğrencilere arayüzün anlatılması.
2	Örnekler kullanılarak, öğrencilerin anlamasını kolaylaştırmak adına eğitmen tarafından çözümlemeler yapılması.
3	Öğrencilere anlatılan örnekler hakkında soruların sorulup, konunun anlaşılabilirliğinin ölçülmesi.
4	Öğrenciler için tanıtılan, arayüz örneklerinden herhangi 3 örnek ile ödevlendirmenin yapılması.

HACKER CAN: ALGORITMA

Müfredat Kodu / Ünite / Konu	İCT-102-2 / Hacker Can / Algoritma
Hedeflenen Süre	1170 Dakika

Algoritma Nedir?

- Algoritma, belli bir problemi çözmek veya belirli bir amaca ulaşmak için tasarlanan yol.
- Matematikte ve bilgisayar biliminde bir işi yapmak için tanımlanan, bir başlangıç durumundan başladığında, açıkça belirlenmiş bir son durumunda sonlanan, sonlu işlemler kümesidir.
- Genellikle bilgisayar programlamada kullanılır ve tüm programlama dillerinin temeli algoritmaya dayanır.
- Aynı zamanda algoritma tek bir problemi çözecek davranışın, temel işleri yapan komutların veya deyimlerin adım adım ortaya konulmasıdır ve bu adımların sıralamasına dikkat edilmelidir.
- Bir problem çözülürken algoritmik ve sezgisel olmak üzere iki yaklaşım vardır.
- Algoritmik yaklaşımda da çözüm için olası yöntemlerden en uygun olan seçilir ve yapılması gerekenler adım adım ortaya konulur.
- Algoritmayı belirtmek için; metinsel olarak düz ifade ve akış diyagramı olmak üzere 2 yöntem kullanılır.
- Algoritmalar bir programlama dili vasıtasıyla bilgisayar tarafından işletilebilirler.

Hacker Can Algoritma

Yazılım geliştirme uzmanlığının ilk aşaması, algoritma oluşturma ve problem çözme yeteneği kazanmaktır. Bir problemin çözüm basamakları olarak da tanımlayabileceğimiz algoritma, her yazılım projesinin vazgeçilmez unsurudur. Oyun tabanlı öğrenme modeli ile öğrencilerimizin yazılım dünyasına attıkları ilk adımı, en keyifli ve verimli şekilde gerçekleştirmekteyiz.

Kazanım

- Öğrencilerin algoritma, problem çözme yeteneği ve programlama (gerçek kod yazma) becerileri gelişir.
- Öğrenciye en keyifli saatlerinde eşlik ederek oyun tadında programlamayı öğretir.
- Akıllı telefon ve tablet entegrasyonu ile internete bağlı cihazlar üzerinden öğretimi sürdürür.
- Eğitim sonunda öğrenciye sertifika verilir.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Algoritmanın tanıtımı.	☑ Eğitmen ☐ Öğrenciler	Algoritmanın öğrencilere tanıtımı.
2	☑ Uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen aracılığıyla tanıtılan, algoritmaya ait uygulamaların yapılması.

Eğitmen Yol Haritası

1	Öğrencilere algoritma mantığının anlatılması.
2	Örnekler kullanılarak, öğrencilerin anlamasını kolaylaştırmak adına eğitmen tarafından çözümlemeler yapılması.
3	Öğrencilere anlatılan örnekler hakkında soruların sorulup, konunun anlaşılabilirliğinin ölçülmesi.
4	Öğrenciler için tanıtılan, algoritma örneklerinden herhangi 3 örnek ile ödevlendirmenin yapılması.

HACKER CAN: SÖZ DIZIMI

Müfredat Kodu / Ünite / Konu	İCT-102-3 / Hacker Can / Söz Dizimi
Hedeflenen Süre	90 Dakika

🚀 Hacker Can Söz Dizimi

Bu bölümde kodlar hazır olarak gelmekte, sol üst köşede cetveli alarak karakterden mesafeye doğru ölçüm yapılır. Klavyeden ve sayılar çekmecesinden yazım yapılabilir.

📘 Kazanım

- Ders sonunda öğrencileri kod yazmaya ve ekrana daha hakim olma becerisi kazandırılır.

✅ Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Site üzerine öğrencilerin oyunu oynamaya başlaması	☑ Eğitmen ☐ Öğrenciler	https://www.hackercan.com/tr/ site üzerine öğrencilerin oyunu oynamaya başlaması. Platform üzerindeki kavramların tanıtımı.
2	☑ https://www.hackercan.com/tr/ kavramların pekiştirilmesi.	☐ Eğitmen ☑ Öğrenciler	Platform üzerindeki kavramların pekiştirilmesi.

👩‍🏫 Eğitmen Yol Haritası

1	Öğrencilere söz dizimi mantığının anlatılması.
2	Örnekler kullanılarak, öğrencilerin anlamasını kolaylaştırmak adına eğitmen tarafından çözümlemeler yapılması.
3	Öğrencilere anlatılan örnekler hakkında soruların sorulup, konunun anlaşılabilirliğinin ölçülmesi.
4	Öğrenciler için tanıtılan, söz dizimi örneklerinden herhangi 3 örnek ile ödevlendirmenin yapılması.

HACKER CAN: OBJELER

Müfredat Kodu / Ünite / Konu	İCT-102-4 / Hacker Can / Objeler
Hedeflenen Süre	90 Dakika

Hacker Can Objeler

Bu bölümde objeler çekmecesi açılmakta, içinde çadır, yıldız ve köprü bulunmaktadır.

- ‘ŞunaDön’ komutu ile başlar.
- Oyun içindeki yardımcı karakterlerin tanıtılması ve onlarında kodlar ile hareket ettirilmesi öğretilmektedir.
- Bir objenin programlanması ve ‘ŞunaDön’ komutunun öğretilmesi.
- Yardımcı karakterleri kodlar ile hareket ettirerek Hacker Can'a yardımcı olmaları sağlanmalıdır.

Ders sonucunda öğrenciler objeleri nasıl kullanacaklarını öğrenmeleri gerekmektedir.

Kazanım

- Bu bölümde objelerin doğru kullanımları amaçlanmaktadır.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Site üzerine öğrencilerin objelerle kod yazması	☑ Eğitmen ☐ Öğrenciler	https://www.hackercan.com/tr/ site üzerine öğrencilerin objelerle kod yazması. Platform üzerindeki kavramların tanıtımı.
2	☑ https://www.hackercan.com/tr/ obje kullanımının pekiştirilmesi.	☐ Eğitmen ☑ Öğrenciler	Platform üzerindeki objelerin pekiştirilmesi.

Eğitmen Yol Haritası

1	Öğrencilere objelerin anlatılması.
2	Örnekler kullanılarak, öğrencilerin anlamasını kolaylaştırmak adına eğitmen tarafından çözümlemeler yapılması.

3	Öğrencilere anlatılan örnekler hakkında soruların sorulup, konunun anlaşılabilirliğinin ölçülmesi.
4	Öğrenciler için tanıtılan, obje örneklerinden herhangi 3 örnek ile ödevlendirmenin yapılması.

NOTLAR

HACKER CAN: KARAKTERLER

Müfredat Kodu / Ünite / Konu	İCT-102-5 / Hacker Can / Karakterler
Hedeflenen Süre	90 Dakika

🚀 Hacker Can Karakterler

Bilişimde karakter kodlaması kavramı bir çeşit kodlama sistemi kullanılarak kodlanmış karakter gruplarını temsil etmektedir. Soyutlama düzeyi ve kullanıldığı bağlama bağlı olarak karakterlere karşılık gelen kod noktaları ve bunların oluşturdukları kod alanı, doğal sayılar, elektrik sinyalleri vb. şeklinde algılanabilir. Metinsel verilerin işlenmesi, depolanması ve iletimi esnasında karakter kodlamaları kullanılır.

- Bu bölümde yeni pencere olarak "Karakter" penceresi açılıyor.

📘 Kazanım

- Bu bölümde karakterlerin doğru kullanımları amaçlanmaktadır.

✅ Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Site üzerine öğrencilerin karakterleri kullanarak kod yazmaları hedeflenmektedir.	☑ Eğitmen ☐ Öğrenciler	https://www.hackercan.com/tr/ Site üzerine öğrencilerin karakterleri kullanarak kod yazmaları hedeflenmektedir.
2	☑ https://www.hackercan.com/tr/ karakterlerin pekiştirilmesi.	☐ Eğitmen ☑ Öğrenciler	Platform üzerindeki karakterlerin pekiştirilmesi.

👩‍🏫 Eğitmen Yol Haritası

1	Öğrencilere karakterlerin mantığının anlatılması.
2	Örnekler kullanılarak, öğrencilerin anlamasını kolaylaştırmak adına eğitmen tarafından çözümlemeler yapılması.
3	Öğrencilere anlatılan örnekler hakkında soruların sorulup, konunun anlaşılabilirliğinin ölçülmesi.
4	Öğrenciler için tanıtılan, karakter örneklerinden herhangi 3 örnek ile ödevlendirmenin yapılması.

NOTLAR

HACKER CAN: KERE DÖNGÜSÜ

Müfredat Kodu / Ünite / Konu	İCT-102-6 / Hacker Can / Kere Döngüsü
Hedeflenen Süre	90 Dakika

🚀 Kere Döngüsü Nedir?

Programlamada en çok kullanılan özelliklerden biri de döngülerdir. Döngüler aracılığı ile belli şartlar gerçekleşinceye kadar veya gerçekleştikçe bir komut parçasını çalıştırmak mümkündür. Bir programda bir komut parçasının belli koşullar gerçekleştikçe veya gerçekleşinceye kadar defalarca icra etmesi gerekebilir. Bu defalarca çalışmayı sağlayan komutlara döngü (çevrim) yapıları denir. Döngü (loop) deyimleri, bir kümenin belli bir koşul altında tekrar edilmesi için kullanılır.

📚 Hacker Can Kere Döngüsü

- Aynı işlemin birden fazla yapılması gerektiği durumlarda, işlemin daha kısa yapılmasını sağlar.
- Hacker Can'ı bir kare üzerinde yürütmek istenirse aynı kodları 5 kere yazmak yerine bu komut kullanılır.

Ders sonunda öğrenciler kere mantığını kavramalı. Artık yazacakları kodlarda kere mantığı üzerinden düşünmelilerdir.

📘 Kazanım

- Bu bölümde kere döngüsünün doğru kullanımları amaçlanmaktadır.

✅ Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Site üzerine öğrencilerin kere döngüsünün doğru kullanılması hedeflenmiştir.	☑ Eğitmen ☐ Öğrenciler	https://www.hackercan.com/tr/ site üzerine öğrencilerin kere döngüsünün doğru kullanılması hedeflenmiştir.
2	☑ https://www.hackercan.com/tr/ kere döngüsünün pekiştirilmesi.	☐ Eğitmen ☑ Öğrenciler	Platform üzerindeki kere mantığının pekiştirilmesi.

👩‍🏫 Eğitmen Yol Haritası

1	Öğrencilere kere döngüsünün anlatılması.
2	Örnekler kullanılarak, öğrencilerin anlamasını kolaylaştırmak adına eğitmen tarafından çözümlemeler yapılması.
3	Öğrencilere anlatılan örnekler hakkında soruların sorulup, konunun anlaşılabilirliğinin ölçülmesi.
4	Öğrenciler için tanıtılan, kere döngüsü örneklerinden herhangi 3 örnek ile ödevlendirmenin yapılması.

HACKER CAN: İÇ İÇE KERE DÖNGÜSÜ

Müfredat Kodu / Ünite / Konu	İCT-102-7 / Hacker Can / İç İçe Kere Döngüsü
Hedeflenen Süre	90 Dakika

🚀 Hacker Can İç İçe Kere Döngüsü

Bir döngü yapısının içine başka bir döngü yapısının yerleştirilmesiyle elde edilen yapıya iç içe döngü adı verilir. İç içe döngülerde hatırlamamız gereken anahtar bilgi, önce içteki döngülerin tamamlanması gerektiğidir.

- Kere içinde kere kullanımının öğrenilmesi.
- Tekrar eden aynı işlemler içinde farklı tekrar eden aynı işlemlerin kullanılması.

Bu ders sonucunda öğrenciler normal kere kullanımını pekiştirmeli, aynı zamanda da kere içinde başka bir kere ile daha kısa ve hızlı kodlar yazabileceğini öğrenmelidir.

📘 Kazanım

- Bu bölümde iç içe kere döngüsünün doğru kullanımları amaçlanmaktadır.

✅ Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Site üzerinde öğrencilerin iç içe kere döngüsünü öğrenmeleri	☑ Eğitmen ☐ Öğrenciler	https://www.hackercan.com/tr/ site üzerinde öğrencilerin iç içe kere döngüsünü öğrenmeleri.
2	☑ https://www.hackercan.com/tr/ iç içe kere döngüsünü pekiştirilmesi.	☐ Eğitmen ☑ Öğrenciler	Platform üzerindeki iç içe kere döngüsünü pekiştirilmesi.

👩‍🏫 Eğitmen Yol Haritası

1	Öğrencilere iç içe kere döngüsünün anlatılması.

2	Örnekler kullanılarak, öğrencilerin anlamasını kolaylaştırmak adına eğitmen tarafından çözümlemeler yapılması.
3	Öğrencilere anlatılan örnekler hakkında soruların sorulup, konunun anlaşılabilirliğinin ölçülmesi.
4	Öğrenciler için tanıtılan, iç içe kere döngüsü örneklerinden herhangi 3 örnek ile ödevlendirmenin yapılması.

NOTLAR

HACKER CAN: DEĞİŞKENLER

Müfredat Kodu / Ünite / Konu	İCT-102-8 / Hacker Can / Değişkenler
Hedeflenen Süre	90 Dakika

🚀 Hacker Can Değişkenler

- Değişken tanımlama, kullanma ve değişkenlerle işlemler yapma.
- Herhangi bir sayı veya yazıyı bir değişkene atayıp ihtiyaç duyulan herhangi bir yerde kullanmak.

Bu ders sonucunda öğrenciler uzun gördükleri birçok şeyi kısa yazılan değişkenlere atayıp kodları kısaltmayı öğrenmelidir. Ayrıca birden fazla yerde değişken kullanıp daha sonradan değişiklik yapmak istenildiğinde tek bir yerden değiştirmeyi öğrenmelidir.

Kazanım

- Bu bölümde değişkenlerin doğru kullanımları amaçlanmaktadır.

✅ Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Site üzerinde öğrencilerin değişkenleri öğrenmeleri.	☑ Eğitmen ☐ Öğrenciler	https://www.hackercan.com/tr/ site üzerinde öğrencilerin değişkenleri öğrenmeleri.
2	☑ https://www.hackercan.com/tr/ değişkenlerin pekiştirilmesi.	☐ Eğitmen ☑ Öğrenciler	Platform üzerindeki değişkenlerin pekiştirilmesi.

👩‍🏫 Eğitmen Yol Haritası

1	Öğrencilere değişkenlerin anlatılması.
2	Örnekler kullanılarak, öğrencilerin anlamasını kolaylaştırmak adına eğitmen tarafından çözümlemeler yapılması.
3	Öğrencilere anlatılan örnekler hakkında soruların sorulup, konunun anlaşılabilirliğinin ölçülmesi.

4	Öğrenciler için tanıtılan, değişken örneklerinden herhangi 3 örnek ile ödevlendirmenin yapılması.

NOTLAR

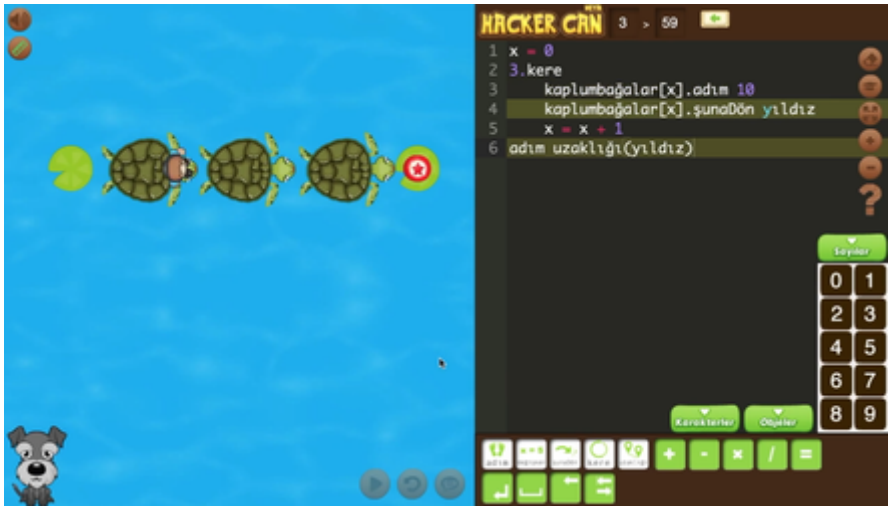
HACKER CAN: BITINCEYE KADAR DÖNGÜSÜ

Müfredat Kodu	İCT-102-9 / Hacker Can / Bitinceye Kadar Döngüsü
Hedeflenen Süre	90 Dakika

🚀 Hacker Can Bitinceye Kadar Döngüsü

Algoritmik olarak bir dizideki elemanların sayısınca nasıl bir döngü oluşturacağını öğrenmek. Kere ile bitinceye kadar arasındaki farkın öğretilmesi.

Bir dizinin tüm elemanlarını sıra ile döndürerek hepsi bitinceye kadar döngünün içerisindeki kodları yürütmek. Bu derste basit düzey olarak ele alınacaktır. Bitinceye kadar döngüsünün kere döngüsünden farkı bizim istediğimiz kadar değil dizide bulunan eleman sayısı kadar dönmesidir.



Bu ders sonucunda öğrenciler bitinceye kadar döngüsüne giriş yapmış olup ana mantığını kavramalıdır. Aynı zamanda da kere döngüsünden farkını iyi anlamalarıdır.

Kazanım

- Bu bölümde bitinceye kadar döngüsünü doğru kullanmaları amaçlanmaktadır.

✅ Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Site üzerinde öğrencilerin bitinceye kadar döngüsünü öğrenmeleri.	☑ Eğitmen ☐ Öğrenciler	https://www.hackercan.com/tr/ site üzerinde öğrencilerin bitinceye kadar döngüsünü öğrenmeleri.
2	☑ https://www.hackercan.com/tr/ bitinceye kadar döngüsünün pekiştirilmesi.	☐ Eğitmen ☑ Öğrenciler	Platform üzerindeki bitinceye kadar döngüsünün pekiştirilmesi.

👩‍🏫 Eğitmen Yol Haritası

1	Öğrencilere bitinceye kadar döngüsünün anlatılması.
2	Örnekler kullanılarak, öğrencilerin anlamasını kolaylaştırmak adına eğitmen tarafından çözümlemeler yapılması.

3	Öğrencilere anlatılan örnekler hakkında soruların sorulup, konunun anlaşılabilirliğinin ölçülmesi.
4	Öğrenciler için tanıtılan, iç içe kere döngüsü örneklerinden herhangi 3 örnek ile ödevlendirmenin yapılması.

NOTLAR

HACKER CAN: FONKSIYONLAR

Müfredat Kodu / Ünite / Konu	İCT-102-10 / Hacker Can / Fonksiyonlar
Hedeflenen Süre	90 Dakika

🚀 Hacker Can Fonksiyonlar

Fonksiyon yazma, fonksiyonları kullanmayı öğrenme, fonksiyonların kod bloklarını barındırması ve bu fonksiyonların gerekli satırlarda kullanılması.

Bu ders sonucunda öğrenciler fonksiyonları neden kullanmak zorunda olduklarını iyi anlamalıdırlar. Gerçek hayattan da bazı örnekler verilerek fonksiyonların önemi pekiştirilmelidir. Örnek: Öğretmenler ders anlatırken öğrenciler not almıştır. Sınav zamanı gelince tüm sistemi tekrar tarayıp notları yeniden yazmak yerine aldıkları notları kullanırlar.

📘 Kazanım

- Bu bölümde fonksiyonları doğru kullanmaları amaçlanmaktadır.

✅ Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Site üzerinde öğrencilerin fonksiyonları öğrenmeleri.	☑ Eğitmen ☐ Öğrenciler	https://www.hackercan.com/tr/ site üzerinde öğrencilerin fonksiyonları öğrenmeleri.
2	☐ https://www.hackercan.com/tr/ fonksiyonların pekiştirilmesi.	☐ Eğitmen ☑ Öğrenciler	Platform üzerindeki fonksiyonları pekiştirilmesi.

👩‍🏫 Eğitmen Yol Haritası

1	Öğrencilere fonksiyonların anlatılması.
2	Örnekler kullanılarak, öğrencilerin anlamasını kolaylaştırmak adına eğitmen tarafından çözümlemeler yapılması.
3	Öğrencilere anlatılan örnekler hakkında soruların sorulup, konunun anlaşılabilirliğinin ölçülmesi.
4	Öğrenciler için tanıtılan, fonksiyon örneklerinden herhangi 3 örnek ile ödevlendirmenin yapılması.

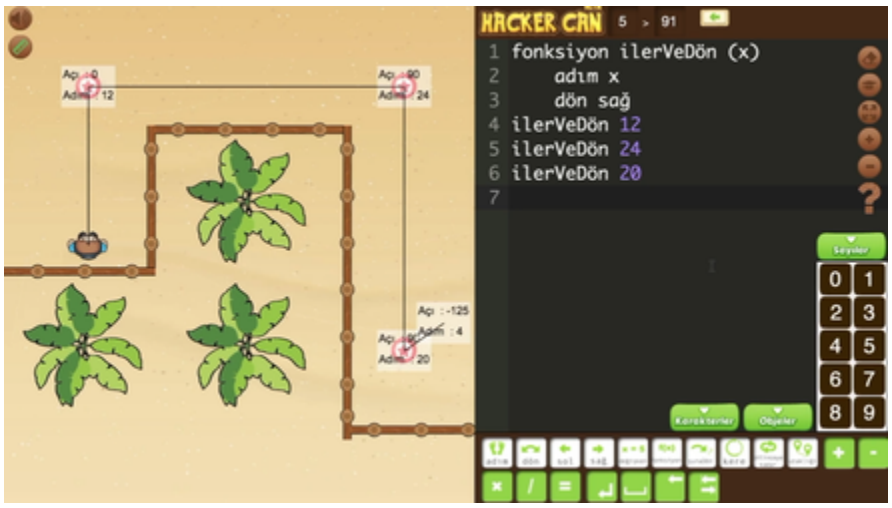
NOTLAR

HACKER CAN: PARAMETRELI FONKSIYONLAR

Müfredat Kodu / Ünite / Konu	İCT-102-11 / Hacker Can / Parametreli Fonksiyonlar
Hedeflenen Süre	90 Dakika

🚀 Hacker Can Parametreli Fonksiyonlar

- Parametreli fonksiyonları öğrenme ve daha önce öğrendikleri fonksiyonlarla birlikte kullanabilme.
- Fonksiyonlar yardımı ile sahneleri oynama.



- Öğrenciler fonksiyonları çok iyi kavramış olmalıdır.
- Zor bölümleri kendilerinin yaptıklarından emin olunmalıdır.
- Fonksiyonlar temel yapı taşlarından olduğu için iyi öğrendiklerinden emin olunmalıdır.
- Öğrencilere gerçek kodlama hayatında fonksiyon kütüphaneleri oluşturup daha hızlı ve rahat kodlama yapabilecekleri öğretilmelidir.

📘 Kazanım

- Bu bölümde öğrencinin parametreleri öğrenme ve daha önce öğrendiği fonksiyonlarla birlikte kullanmaları amaçlanmıştır.

✅ Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Site üzerinde öğrencilerin parametreli fonksiyonları öğrenmeleri.	☑ Eğitmen ☐ Öğrenciler	https://www.hackercan.com/tr/ site üzerinde öğrencilerin parametreli öğrenmeleri.
2	☑ https://www.hackercan.com/tr/ parametreli fonksiyonların pekiştirilmesi.	☐ Eğitmen ☑ Öğrenciler	Platform üzerindeki parametreli fonksiyonların pekiştirilmesi.

👩‍🏫 Eğitmen Yol Haritası

1	Öğrencilere parametreli fonksiyonların anlatılması.
2	Örnekler kullanılarak, öğrencilerin anlamasını kolaylaştırmak adına eğitmen tarafından çözümlemeler yapılması.

3	Öğrencilere anlatılan örnekler hakkında soruların sorulup, konunun anlaşılabilirliğinin ölçülmesi.
4	Öğrenciler için tanıtılan, parametreli fonksiyon örneklerinden herhangi 3 örnek ile ödevlendirmenin yapılması.

HACKER CAN: HAZIR FONKSIYONLAR

Müfredat Kodu / Ünite / Konu	İCT-102-12 / Hacker Can / Hazır Fonksiyonlar
Hedeflenen Süre	135 Dakika

🚀 Hacker Can Hazır Fonksiyonlar

- Sistemdeki hazır fonksiyonların kullanımını öğrenme.
- Sistem içindeki hazır fonksiyonları kullanıp bölümleri geçme.

Öğrenciler oyun içinde bölümleri tamamlamak için ihtiyaçları olan hazır fonksiyonları etkin bir şekilde kullanmalıdır. Bu fonksiyonların daha önce öğrendikleri (bir önceki ünite) fonksiyonlar gibi çalıştığını ama sistemin içinde hazır olarak bulunduğunu bilmelidir.

Kazanım

- Bu bölümde hazır fonksiyonları doğru kullanmaları amaçlanmaktadır.

✅ Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Site üzerinde öğrencilerin hazır fonksiyonları öğrenmeleri.	☑ Eğitmen ☐ Öğrenciler	https://www.hackercan.com/tr/ site üzerinde öğrencilerin hazır öğrenmeleri.
2	☐ https://www.hackercan.com/tr/ hazır fonksiyonların pekiştirilmesi.	☐ Eğitmen ☑ Öğrenciler	Platform üzerindeki hazır fonksiyonların pekiştirilmesi.

👩‍🏫 Eğitmen Yol Haritası

1	Öğrencilere hazır fonksiyonların anlatılması.
2	Örnekler kullanılarak, öğrencilerin anlamasını kolaylaştırmak adına eğitmen tarafından çözümlemeler yapılması.
3	Öğrencilere anlatılan örnekler hakkında soruların sorulup, konunun anlaşılabilirliğinin ölçülmesi.
4	Öğrenciler için tanıtılan, hazır fonksiyon örneklerinden herhangi 3 örnek ile ödevlendirmenin yapılması.

HACKER CAN: IF-ELSE KONTROL YAPILARI

Müfredat Kodu / Ünite / Konu	İCT-102-13 / Hacker Can / if-else Kontrol Yapıları
Hedeflenen Süre	135 Dakika

🚀 Hacker Can if-else Kontrol Yapıları

Bu bölümde if-else yapısını öğrenme ve algoritma mantığını daha da üst sevilere çıkarma.

Klasik tek bir yönde gitme ve sorunu çözme yerine farklı ihtimaller olduğu durumlarda algoritma üretip sorunu çözme.

if-else ile daha önce öğrendiklerini birleştirme.

Zorlu bölümleri if-else ile geçebilme ve gerekli olan tüm kodları kullanma ve bölümleri tamamlama.

Öğrenciler tekdüzelikten kurtarılmalıdır. Önlerine çıkan belirsizlikleri düzenli bir kod yazım şekli ve düzgün tasarlanan algoritma ile çözmeyi öğrenmelidir. İf else mantığı yapay zekanın temel taşlarından olup iyi kavranmalıdır

Öğrenciler algoritmik mantıklarını büyük ölçüde tamamlamış olup, artık yardıma ihtiyaç duymadan önlerindeki bölümleri geçmeyi başarabilmelidir.

Kazanım

- Bu bölümde if-else kontrol yapılarını doğru kullanmaları amaçlanmaktadır.

✅ Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Site üzerinde öğrencilerin if-else kontrol yapılarını öğrenmeleri.	☑ Eğitmen ☐ Öğrenciler	https://www.hackercan.com/tr/ site üzerinde öğrencilerin if-else kontrol yapılarını öğrenmeleri.
2	☑ https://www.hackercan.com/tr/ if-else kontrol yapılarının pekiştirilmesi.	☐ Eğitmen ☑ Öğrenciler	Platform üzerindeki if-else kontrol yapılarının pekiştirilmesi.

👩‍🏫 Eğitmen Yol Haritası

1	Öğrencilere if-else kontrol yapılarının anlatılması.

2	Örnekler kullanılarak, öğrencilerin anlamasını kolaylaştırmak adına eğitmen tarafından çözümlemeler yapılması.
3	Öğrencilere anlatılan örnekler hakkında soruların sorulup, konunun anlaşılabilirliğinin ölçülmesi.
4	Öğrenciler için tanıtılan, if-else kontrol yapı örneklerinden herhangi 3 örnek ile ödevlendirmenin yapılması.

MIT APP INVENTOR EĞITIM MÜFREDATI

MIT APP INVENTOR EĞİTİM MÜFREDATI														
EĞİTİM İÇERİĞİ / HAFTA	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.
MIT App Inventor : Arayüz Tanıtımı														
MIT App Inventor : User Interface Menüsü														
MIT App Inventor : Layout Menüsü														
MIT App Inventor : Media Menüsü														
MIT App Inventor : Drawing and Animations Menüsü														
MIT App Inventor : Sensors Menüsü														
MIT App Inventor : Social Menü														
MIT App Inventor : Storage Menüsü														
MIT App Inventor : Connectivity Menüsü														
MIT App Inventor : Değişkenler														
Renklendirme Uygulaması														
Boyama Uygulaması														
Pusula Uygulaması														
Alarm Uygulaması														
Web Tarayıcısı Uygulaması														
Qr Kod Okuyucu Uygulaması														
Telefonu Sallayarak Ekran Temizleme Uygulaması														
Fotoğraf Çekme Uygulaması														
Ses Tanıma Uygulaması														
Layout Bileşenleri Uygulaması														

MIT APP INVENTOR EĞITIMI

Müfredat Kodu	İCT-104
Ders	MIT App Inventor
Ders İçeriği	• MIT App Inventor: Arayüz Tanıtımı • MIT App Inventor: User Interface Menüsü • MIT App Inventor: Layout Menüsü • MIT App Inventor: Media Menüsü • MIT App Inventor: Drawing and Animations Menüsü • MIT App Inventor: Sensors Menüsü • MIT App Inventor: Social Menü • MIT App Inventor: Storage Menüsü • MIT App Inventor: Connectivity Menüsü • MIT App Inventor: Değişkenler • Renklendirme Uygulaması • Boyama Uygulaması • Pusula Uygulaması • Alarm Uygulaması • Web Tarayıcısı Uygulaması • Qr Kod Okuyucu Uygulaması • Telefonu Sallayarak Ekran Temizleme Uygulaması • Fotoğraf Çekme Uygulaması • Ses Tanıma Uygulaması • Layout Bileşenleri Uygulaması
Seviye	İCT-S2: ORTA
Ön Koşullar	Maker alanına ilgili olmak
Amaç	Uygulama geliştirmeye yeni başlayanların Android işletim sistemi için blok kodlama yöntemiyle uygulama geliştirmesine yardımcı olmayı amaçlar.
Hedef	MIT App Inventor projesi, teknoloji tüketiminden teknoloji yaratmaya geçmek için tüm insanları, özellikle gençleri güçlendirerek yazılım geliştirmeyi demokratikleştirmeyi hedefliyor.
Gereksinimler	Bilgisayar, Ortamda İnternet Ağı, Mail adresi
Kazanım	• Algoritmik düşünce. • Problem çözme. • Tasarım yetkinliklerini kazanma.
Referanslar & Dokümantasyon	• http://www.cankodyaz.com/egitimler/courses/appinventor.html
Durum	BEKLIYOR / UYGULANIYOR / TAMAMLANDI / DEĞERLENDIRILDI
Hedeflenen Süre	1260 Dakika
Eğitim Günleri	Başlama: .../.../... Bitiş: .../.../...

MIT APP INVENTOR: ARAYÜZ TANITIMI

Müfredat Kodu / Ünite / Konu	İCT-104-1 / MIT App Inventor / Arayüz Tanıtımı
Hedeflenen Süre	90 Dakika

🚀 MIT App Inventor Arayüz Tanıtımı

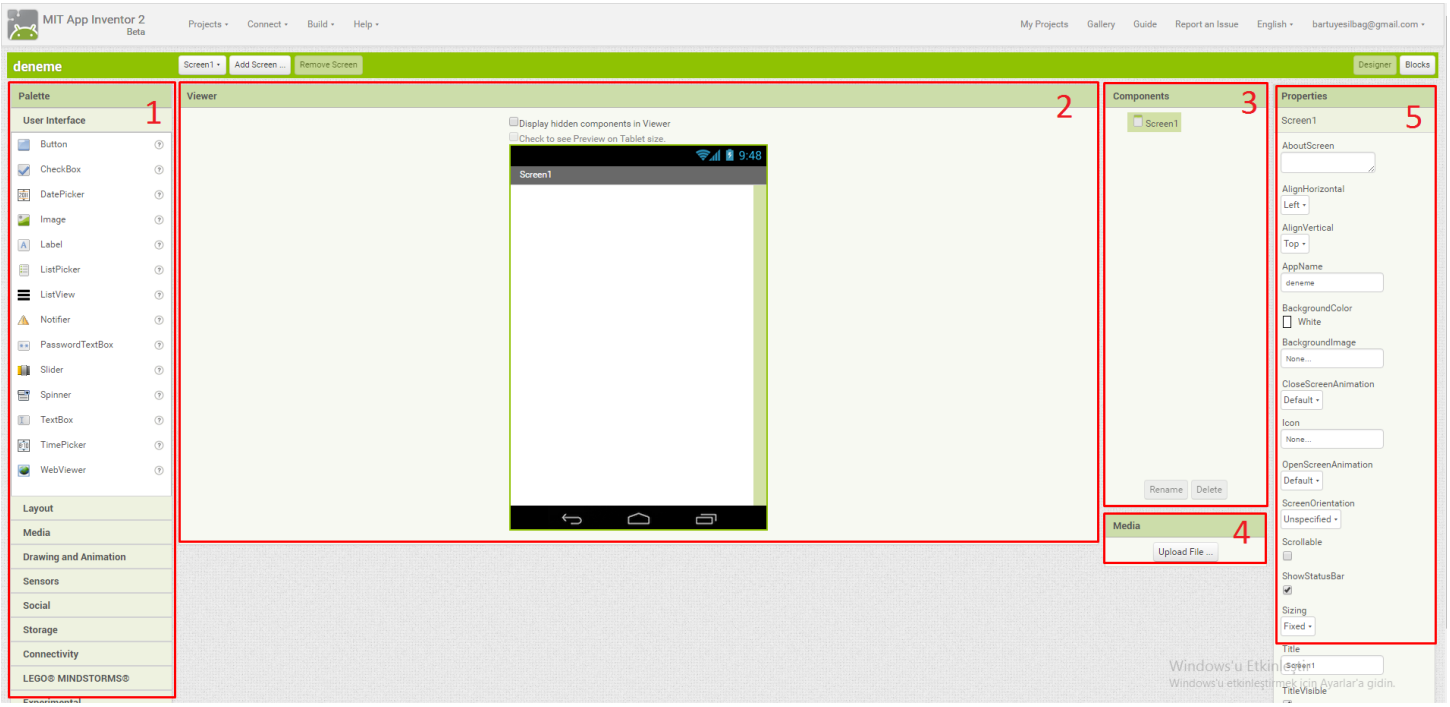
Sisteme giriş yaptıktan sonra karşımıza çıkan anketler ve karşılama mesajlarından sonra MIT App Inventor arayüzü bizi karşılamaktadır.

Üst kısımda proje oluşturma, projenizin detaylarına ve daha önce yaptığınız projelerin bilgisayarınızdan online ortama aktarılmasını sağlayan menüler olduğu gibi platform üzerinde oluşturduğunuz uygulamaları bilgisayarınıza aktarmanızı sağlayan menülerde mevcuttur.

Bu alanda ;

- Connect menüsünü kullanarak MIT'nin kendi oluşturduğu emülatörü sayesinde geliştirdiğiniz uygulamaları telefonunuz üzerinde deneyebilirsiniz.
- Build menüsünü kullanarak oluşturduğunuz uygulamayı kendi telefonunuza ve bilgisayarınıza indirebilirsiniz.
- Help menüsü altında platform üzerinde kullanabileceğiniz yardım menüleri mevcuttur.
- My Project menüsü ile geliştirdiğiniz uygulamaları görüntüleyebilirsiniz.
- Galery bölümü ile MIT App Inventor üzerinde geliştirilen uygulamaları görüntüleyebilirsiniz.

📸 Arayüz



✅ Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar

1	☑ Arayüzün tanıtımı	☑ Eğitmen ☐ Öğrenciler	Eğitim ve uygulamaların yapılacağı cihazlar üzerinde kurulumların tamamlanması amaçlanmaktadır. Öğrencilere ait cihazlar üzerine gerekli kurulumların yapılması.Öğrencilere portal arayüzünün tanıtılması.Öğrencilerin portal hakkındaki sorularına cevap verilmesi, eğer öğrencilerin soruları yoksa eğitmenin öğrencilere portal üzerindeki paneller hakkında soru sorulması beklenmektedir.
2	☑ Arayüzün keşfedilmesi	☐ Eğitmen ☑ Öğrenciler	Öğrencilerin uygulamaya erişim sağlaması ve uygulamayı keşfetmesi amaçlanmaktadır.

MIT APP INVENTOR: USER INTERFACE MENÜSÜ

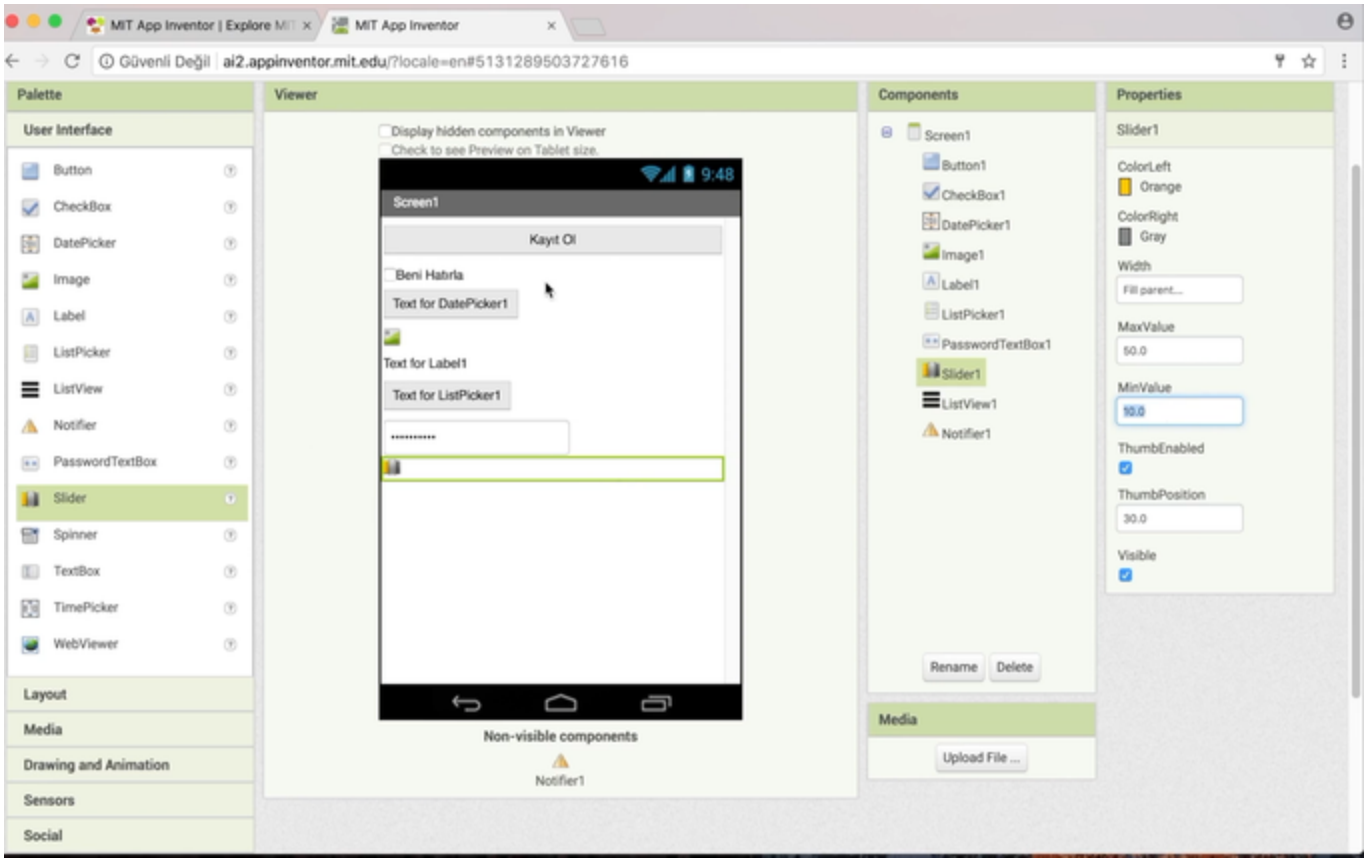
Müfredat Kodu / Ünite / Konu	İCT-104-2 / MIT App Inventor / User Interface Menüsü
Hedeflenen Süre	20 Dakika

🚀 User Interface Menüsü

Bu ünitede User Interface seçeneği altındaki araçları inceleyeceğiz.

- Buton aracı ile ekrana buton ekleyebilirsiniz.
- CheckBox aracı ile beni hatırla, çoklu seçim gibi alanların yapımını tamamlayabilirsiniz.
- DatePicker aracı ile telefondaki tarihi uygulama üzerine çekebilirsiniz.
- İmage aracı ile uygulamaya dahil edilmek istenen görüntüleri ekleyebilirsiniz.
- Label, daha çok bilgilendirme amacı olarak uygulama ekranlarında yer alan kısa metinlerin yer alabileceği araçtır.
- ListPicker, daha önceden oluşturulan listelerin uygulamaya çekilmesini sağlayan araçtır.
- Listview aracı bir çok verinin listelendiği yerlerde listenin içeriğini sunabilmek için kullanılır.
- Notifier özelliğinde bu alan ekranımıza görüntülenebilir olarak gelmez, görünürlük alanında özellikleri kısıtlı olmasına rağmen kod alanında daha fazla özelliğe sahiptir.
  Kod kısmında ekrana mesaj vermek, hata mesajlarını ekrana yansıtmak gibi özellikleri vardır.
- PasswordTextBox aracı, daha çok formlarda ve üye giriş ekranları gibi şifrenizi gireceğiniz alanlarda yazdığınız karakterleri gizleyerek ekranda gösteren bir araçtır.
- Slider alanı ise ekranda iki sayı aralığında bulunan tüm sayısal değerleri parmağınızla sürükleyerek kolay bir şekilde elde edebileceğiniz araçtır.
- Spinner (açılır menü özelliği) , bir grup listenin daha az yer kaplayacak şekilde oluşturulmuş şekilidir.
- Textbox alanı, buton veya password textbox alanı gibi birçok alanda kullandığımız metin girmemizi sağlayan bir bileşendir.
- Timepicker alanı, telefonumuzun tarihini uygulamaya taşıyan bir araçtır.
- WebViewer özelliği, android uygulamaları içerisinde web sayfalarını çalıştırabilen bir tarayıcı oluşturmaktadır.

📸 User Interface Arayüzü



Kazanım

- Öğrencilerin arka plan rengi, yazı tipleri, yazı fontları, yükseklik, genişlik gibi kavramlar başta olmak üzere kodlama alanında belli başlı kullanılan kavramlarla tanışması.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ User interface menüsünün anlatılması	☑ Eğitmen ☐ Öğrenciler	Öğrencilere user interface alanının ve araçlarının tanıtılması amaçlanmaktadır.Öğrencilere User Interface menüsü üzerindeki araçların tanıtılması.Tanıtılan araçlar hakkında soruların sorulup, konunun anlaşılabilirliğinin ölçülmesi.Konunun daha iyi kavranması için, öğrencilerinizle birlikte uygulamalar bölümü altındaki https://hackercan.atlassian.net/wiki/spaces/ICT/pages/65113191#%F0%9F%9A%80-Alarm-Uygulamas%C4%B1 'nı yapabilirsiniz.
2	☑ User interface menüsünün keşfedilmesi.	☐ Eğitmen ☑ Öğrenciler	Öğrencilerin, öğretilen User Interface alanı üzerindeki araçları kullanarak örnek uygulama yapması amaçlanmaktadır.

MIT APP INVENTOR: LAYOUT MENÜSÜ

Müfredat Kodu / Ünite / Konu	İCT-104-3 / MIT App Inventor / Layout Menüsü
Hedeflenen Süre	20 Dakika

🚀 Layout Menüsü

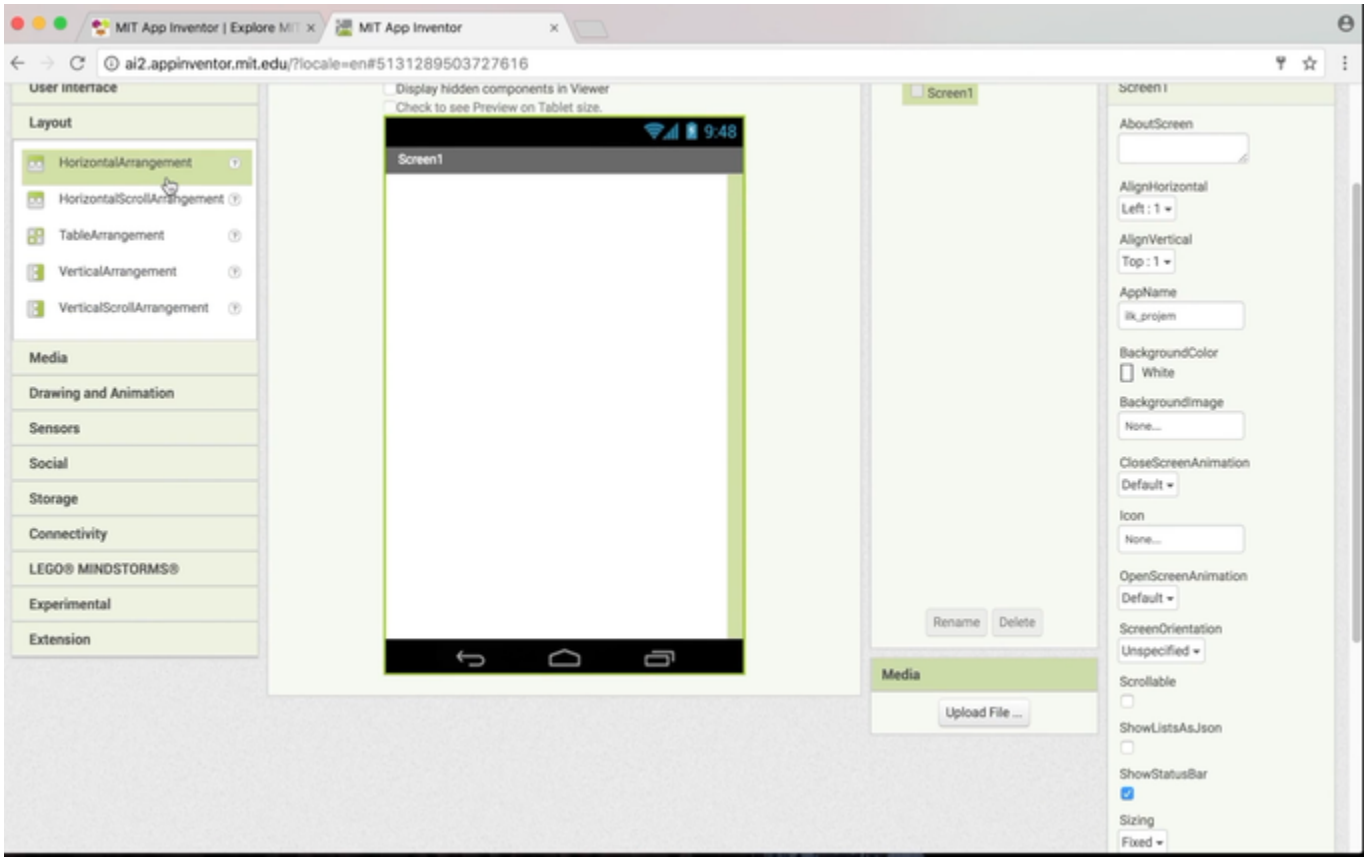
Bu ünitede Layout Menüsü altındaki hizalama araçlarını öğreneceğiz.

Bu araçların kullanım amacı; farklı inçlere sahip olan cihazların (tablet, telefon) üzerindeki görüntülerin ekranlarda farklı konumlanmasını sağlar.

5 adet hizalama aracı vardır.

- HorizontalArrangement; içine atılan nesneleri yan yana hizalar.
- HorizontalScrollArangement; içine atılan nesneler yatay olarak hizalanır, scroll özelliği kazandırır.
- TableArrangement; içerisinde oluşturulan hücreleri tablo haline getirir.
- VerticalArrangement; içine atılan nesneler dikey olarak hizalanır.
- VerticalScrollArrangement; içine atılan nesneler dikey olarak hizalanır, scroll özelliği kazandırır.

📸 Layout Menüsü Arayüzü



Kazanım

- Bu bölümde öğrenciler; hizalama ve responsive görünüm hakkında bilgi sahibi olur.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Layout menüsünün tanıtılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere Layout menüsünün tanıtılması ve hizalama üzerinde örnekler yapılması beklenmektedir. Öğrencilere Layout menüsü üzerindeki araçların tanıtılması. Eğitmen tarafından, tanıtılan araçların kullanılarak öğrencilerin konuyu daha rahat anlaması adına uygulamaların yapılması.Tanıtılan araçlar hakkında soruların sorulup, konunun anlaşılabilirliğinin ölçülmesi.Konunun daha iyi kavranması için, öğrencilerinizle birlikte uygulamalar başlığı altındaki https://hackercan.atlassian.net/wiki/spaces/ICT/pages/65113757#%F0%9F%96%A5%EF%B8%8F-Layout-Bile%C5%9Fenleri-ile-%C3%96rnek-Uygulamalar bölümünü yapabilirsiniz.
2	☑ Layout menüsüne ait uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Yeni öğrenilen hizalama konusu hakkında uygulamaların yapılması ve konunun pekiştirilmesi hedeflenmektedir.

MIT APP INVENTOR: MEDIA MENÜSÜ

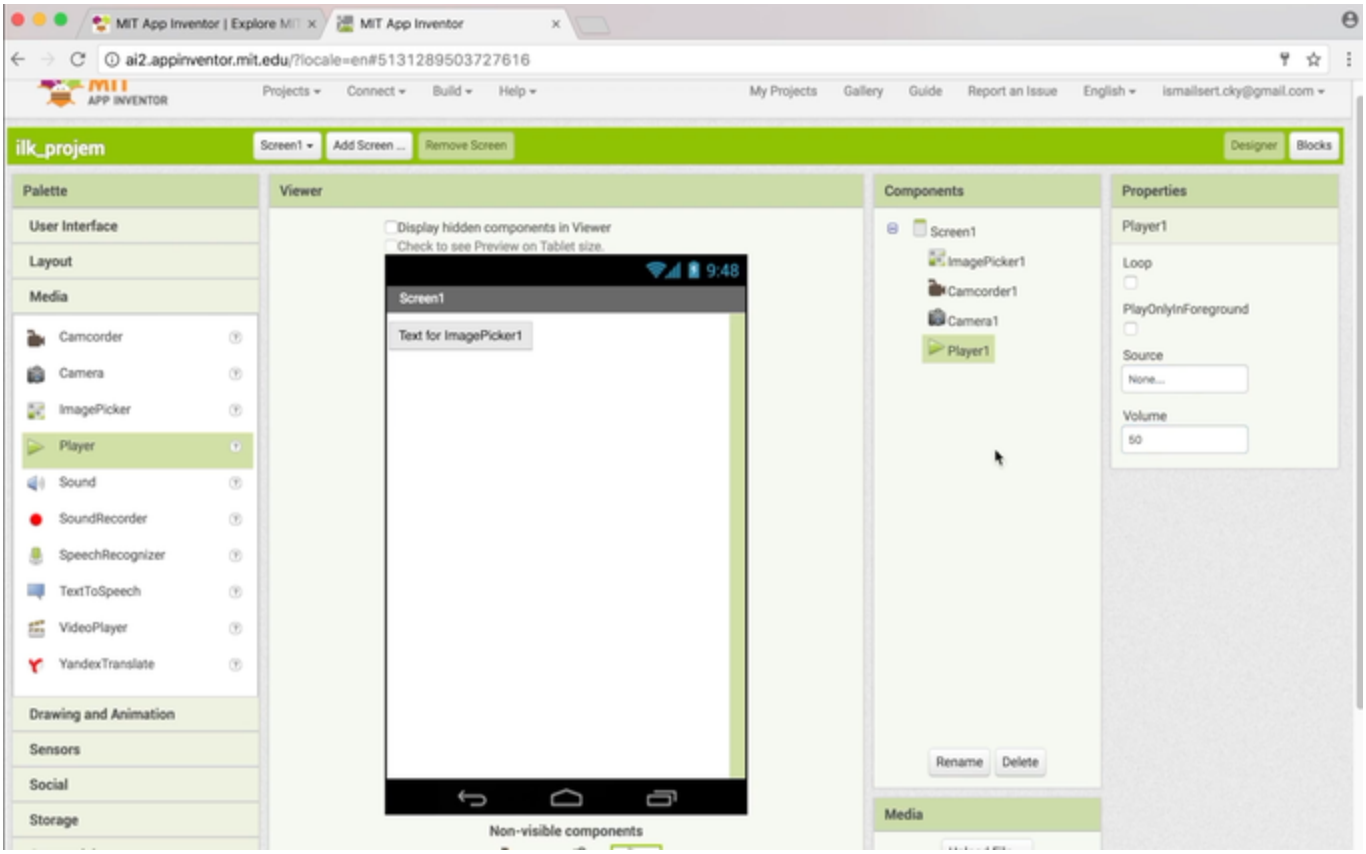
Müfredat Kodu / Ünite / Konu	İCT-104-4 / MIT App Inventor / Media Menüsü
Hedeflenen Süre	25 Dakika

🚀 Media Menüsü

Bu ünitede Media Menüsü altındaki Media araçlarını inceleyeceğiz.

- Camcorder; Android cihazınızda bulunan kamera kaydedici yazılımını kullanarak uygulama üzerinde video kaydetmeyi sağlar.
- Camera; Android cihazınızda bulunan kameranın yazılımını aktif hale getirerek blocks kısmındaki taşlarla etkileşime geçirmenize yardımcı olur.
- İmagePicker; Telefonunuz üzerinde kayıtlı olan görsellere ulaşabilmenizi sağlayan araçtır.
- Player aracı; görünüm alanında görüntülenmez fakat block kısmındaki yönlendirmelerle etkileşime geçirilir. İçerisine atılan sesin tekrar etmesi, arka planda çalınması gibi işlevlere sahiptir.
- Sound özelliği; çalınması istenilen ses bu araç ile uygulamaya dahil edilebilir.
- SoundRecorder; görünüm alanında görüntülenemez fakat ses kaydı yapabilmek için block taşları arasındaki komutları uygun şekilde dizmeniz gerekir.
- SpeechRecognizer; bu araç ile konuşmanızı yazıya çevirebilirsiniz.
- TextToSpeech; ekrandaki yazıyı sesli bir şekilde ifade etmektedir.
- VideoPlayer; uygulamaya dahil etmek istediğiniz videoları bu alan sayesinde ekleyebilirsiniz.
- YandexTranslate; internet bağlantısı kullanarak ekrandaki kelimeleri farklı dillere çevirebilme imkanını sağlayan araçtır.

📸 Media Menüsü Arayüzü



Kazanım

- Bu bölümde öğrenciler, Media menüsüne ait araçların kullanımı öğrenir ve uygulamalarla öğrendiklerini pekiştirir.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Media menüsünün tanıtılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere Media Menüsünün tanıtılması ve uygulamalarla pekiştirilmesi beklenmektedir. Öğrencilere Media menüsü üzerindeki araçların tanıtılması.Tanıtılan araçların kullanılarak öğrencilerin anlamasını kolaylaştırmak adına eğitmen tarafından uygulamaların yapılması. Tanıtılan araçlar hakkında soruların sorulup, konunun anlaşılabilirliğinin ölçülmesi. Öğrenciler için, tanıtılan Media menüsüne ait araçlardan herhangi 3 aracı kullanarak uygulama ödevlendirmesinin yapılması.Konunun daha iyi kavranması için, öğrencilerinizle birlikte uygulamalar bölümü altındaki https://hackercan.atlassian.net/wiki/spaces/ICT/pages/65113458#%F0%9F%9A%80-Foto%C4%9Fraf-%C3%87ekme-Uygulamas%C4%B1 'nı yapabilirsiniz.
2	☑ Media menüsüne ait uygulamaların tamamlanması.	☐ Eğitmen ☑ Öğrenciler	Öğrencilerin Media menüsüne erişimi ile kamera, ses kayıt cihazı, galeri, translate gibi birçok aracı öğrenmesi hedeflenmektedir.

MIT APP INVENTOR: DRAWING AND ANIMATIONS MENÜSÜ

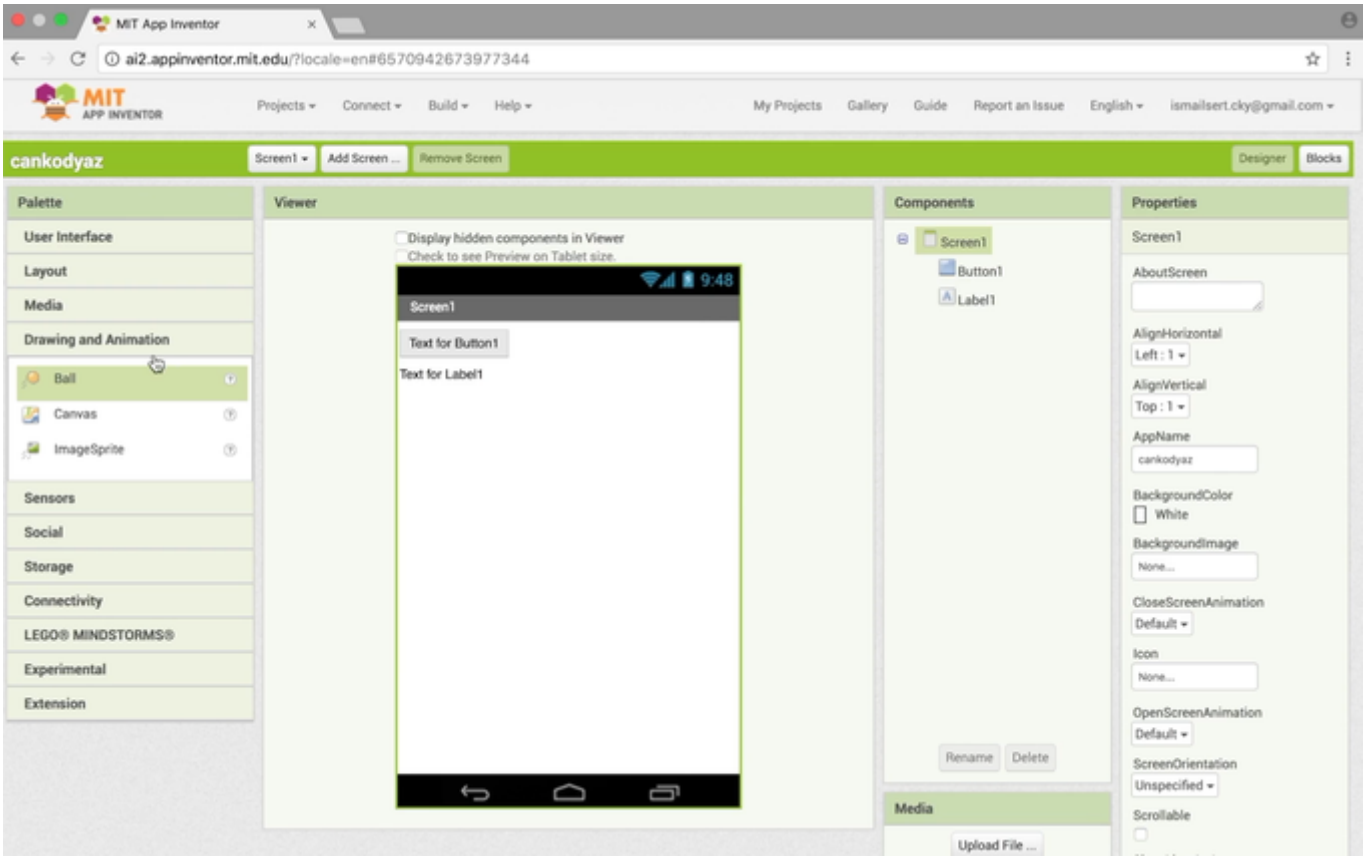
Müfredat Kodu / Ünite / Konu	İCT-104-5 / MIT App Inventor / Drawing and Animations Menüsü
Hedeflenen Süre	25 Dakika

Drawing and Animations Menüsü

Drawing and Animations Menüsü genel anlamda android ekranı üzerinde çizim yapılabilecek ve nesne hareketi verilebilecek komponentlere sahiptir.

- Canvas aracı: Ekrana çizim yapabilme imkanı sağlayan bir araçtır.
- ImageSprite: Canvas'ın içinde yer alabilen ve Canvas'ın içerisinde hareket edebilen araçtır.
- Ball: Ekranda sensörleri kullanarak hareket eden bir topu oluşturabilirsiniz.

Drawing and Animations Menüsü Arayüzü



Kazanım

- Bu bölümde öğrenciler; çizim ve hareketli nesne oluşturabilme öğrenmektedir.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar

1	☑ Drawing and Animations menüsünün tanıtılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere çizim ve nesne hareketlerinin anlatılması amaçlanmaktadır.Öğrencilere Drawn and Animations menüsünün tanıtılması.Tanıtılan araçların kullanılarak öğrencilerin anlamasını kolaylaştırmak adına eğitmen tarafından uygulamaların yapılması.Tanıtılan araçlar hakkında soruların sorulup, konunun anlaşılabilirliğinin ölçülmesi.Öğrencilere, yukarıda tanıtılan 3 aracı da kullanılarak çizim ödevlendirmesi yapılması.Konunun daha iyi kavranılması için, öğrencilerinizle birlikte uygulamalar bölümü altındaki https://hackercan.atlassian.net/wiki/spaces/ICT/pages/65113356#%F0%9F%9A%80-Telefonu-Sallayarak-Ekran-Temizleme-Uygulamas%C4%B1 'nı yapabilirsiniz.
2	☑ Bölüme ait uygulamaların tamamlanması.	☐ Eğitmen ☑ Öğrenciler	Eğitmenle birlikte çizim ve nesneye ait uygulamaların yapılması, pekiştirilmesi hedeflenmektedir.

MIT APP INVENTOR: SENSORS MENÜSÜ

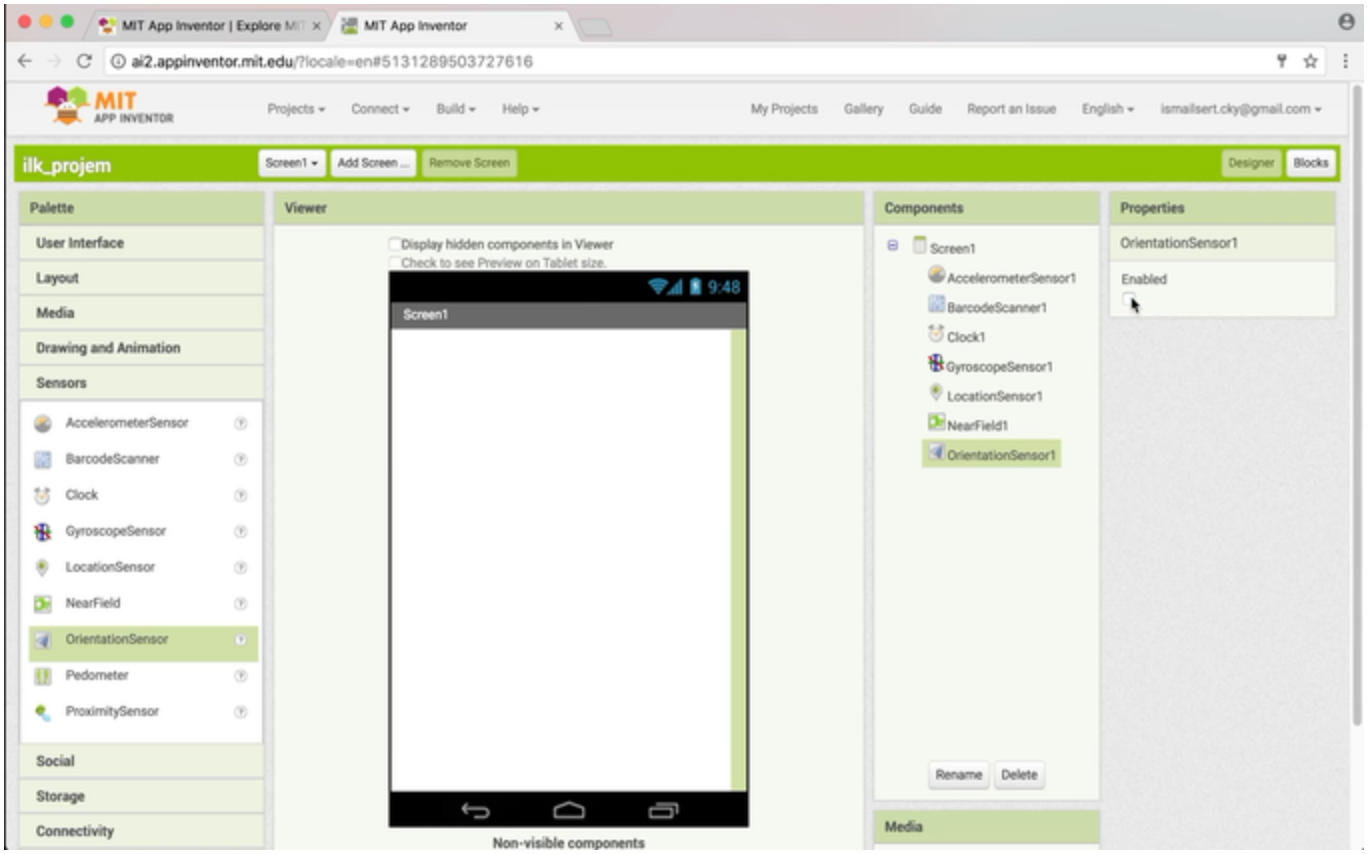
Müfredat Kodu / Ünite / Konu	İCT-104-6 / MIT App Inventor / Sensors Menüsü
Hedeflenen Süre	18 Dakika

🚀 Sensors Menüsü

Bu ünitede Sensors Menüsü altındaki sensörleri inceleyeceğiz.

- AccelerometerSensor: Telefonun ivmesini ölçen (yukarı aşağı), bu değerleri oluşturan ve saklayan sensördür.
- BarcodeScanner: MIT App Inventorun resmi uygulamasında da bulabileceğiniz, barkodu okuyan ve barkod yönetimi sağlayan araçtır.
- Clock: Sistem saatini kullanıp, çeşitli değişiklikler yaparak uygulama içerisinde etkileşim kurmayı amaçlamaktadır.
- GyroscopeSensor: Telefonunuzun dengede olup olmama durumlarını ölçmenize sağlayan araçtır.
- LocationSensor: Telefon üzerindeki konumuzunu bildiren sensördür.
- NearField: NFC olarak bilinen bu sensör, yakın temasta bulunan ve okunabilen kartları okuyarak size değerler döndüren sensördür.
- OrientationSensor: Yönlendirme sensörüdür.
- Pedometer: Telefonunuza yüklediğiniz adımsayar gibi uygulamaların işlemini yapan sensördür.
- ProximitySensor: Telefon ekranınız üst kısmında yer alan ve konuşma sırasında ekranınızın kapanmasını sağlayan yakınlık sensörüdür

📸 Sensors Menüsü Arayüzü



Kazanım

- Bu bölümde öğrenciler; telefon ivmesini ölçmeyi, barkod okutabilmeyi, konum ve sensörleri kullanmayı öğrenmektedir.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Sensors menüsünün tanıtılması.	☑ Eğitmen ☐ Öğrenciler	Eğitmen tarafından sensör menülerinin tanıtılması ve özellikleri hakkında bilgi vermesi beklenmektedir.Öğrencilere Sensors menüsünün tanıtılması.Tanıtılan araçların kullanılarak öğrencilerin anlamasını kolaylaştırmak adına eğitmen tarafından uygulamaların yapılması. Tanıtılan araçlar hakkında soruların sorulup, konunun anlaşılabilirliğinin ölçülmesi.Konunun daha iyi kavranılabilmesi için, öğrencilerinizle birlikte uygulamalar bölümü altındaki https://hackercan.atlassian.net/wiki/spaces/ICT/pages/65113141#%F0%9F%9A%80-Pusula-Uygulamas%C4%B1 'nı yapabilirsiniz.
2	☑ Sensors menüsüne ait uygulamaların tamamlanması.	☐ Eğitmen ☑ Öğrenciler	Bu alanda öğrencinin telefon ivmesini ölçmeyi, barkod okumayı, konum ve sensörleri kullanmayı öğrenmesi hedeflenmektedir.

MIT APP INVENTOR: SOCIAL MENÜ

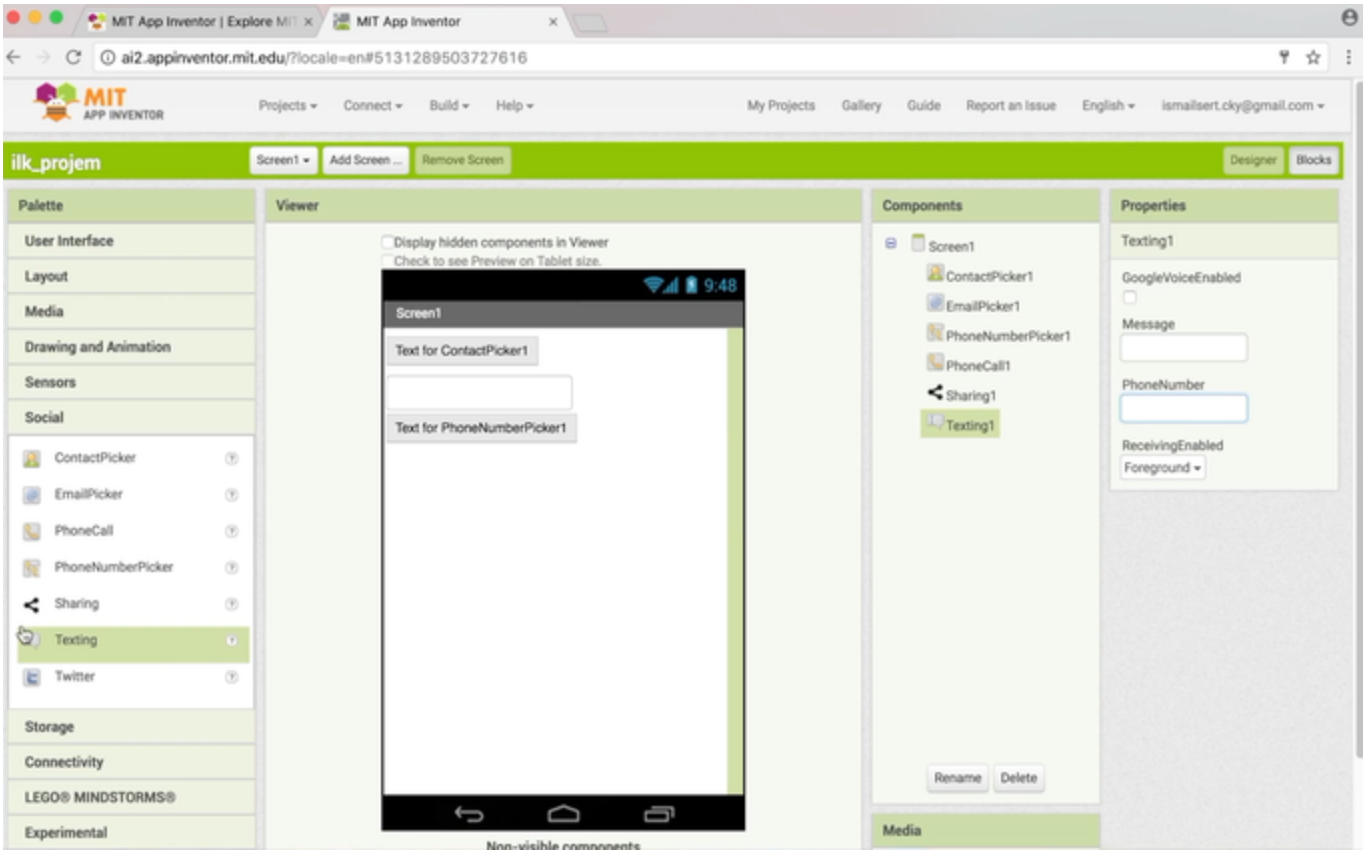
Müfredat Kodu / Ünite / Konu	İCT-104-7 / MIT App Inventor / Social Menü
Hedeflenen Süre	18 Dakika

🚀 Social Menü

Bu ünitede Social Menü altındaki araçları inceleyeceğiz.

- ContactPicker: İsim listenize erişim sağlayabilen araçtır.
- EmailPicker: Bu araç sayesinde e-posta gönderimini gerçekleştirebilecek listelere erişim sağlayabilirsiniz.
- PhoneCall: Bu özellik ile parametre olarak girdiğiniz telefon numarasını arayabilirsiniz.
- PhoneNumberPicker: Telefon numaralarınızın listelendiği alana erişim sağlayarak mevcut telefon numaralarını çekebileceğiniz bir araçtır.
- Sharing: Resim, link gibi verileri paylaşmanıza yardım eden araçtır.
- Texting: Sms göndermeyi sağlayan araçtır.
- Twitter: Twitter hizmetinin bir uygulamasını oluşturan araçtır.

📸 Social Menü Arayüzü



Kazanım

- Bu bölümde öğrenciler; uygulama üzerinden isim listesi, mail listesi, telefon rehberi gibi araçlara erişim sağlamayı ve link paylaşımı yapıp sms göndermeyi öğrenmekte ve uygulamalarla öğrendiklerini pekiştirmektedir.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Social menünün tanıtılması.	☑ Eğitmen ☐ Öğrenciler	Uygulama içerisinde kullanılabilecek sosyal araçların tanıtılması beklenmektedir . Öğrencilere Social menüsünün tanıtılması. Tanıtılan araçların kullanılarak öğrencilerin anlamasını kolaylaştırmak adına eğitmen tarafından uygulamaların yapılması. Öğrencilere tanıtılan araçlar hakkında soruların sorulup, konunun anlaşılabilirliğinin ölçülmesi. Öğrencilere, yukarıda tanıtılan araçlardan herhangi 3 tanesi üzerinde uygulama yapmasını ödevlendirmek.
2	☑ Bölüm içerisindeki uygulamaların tamamlanması.	☐ Eğitmen ☑ Öğrenciler	Bu alanda öğrenciler uygulama üzerinden isim listesi, mail listesi, telefon rehberi gibi araçlara erişim sağlamayı ve link paylaşımı yapıp sms göndermeyi öğrenmeleri hedeflenmektedir.

MIT APP INVENTOR: STORAGE MENÜSÜ

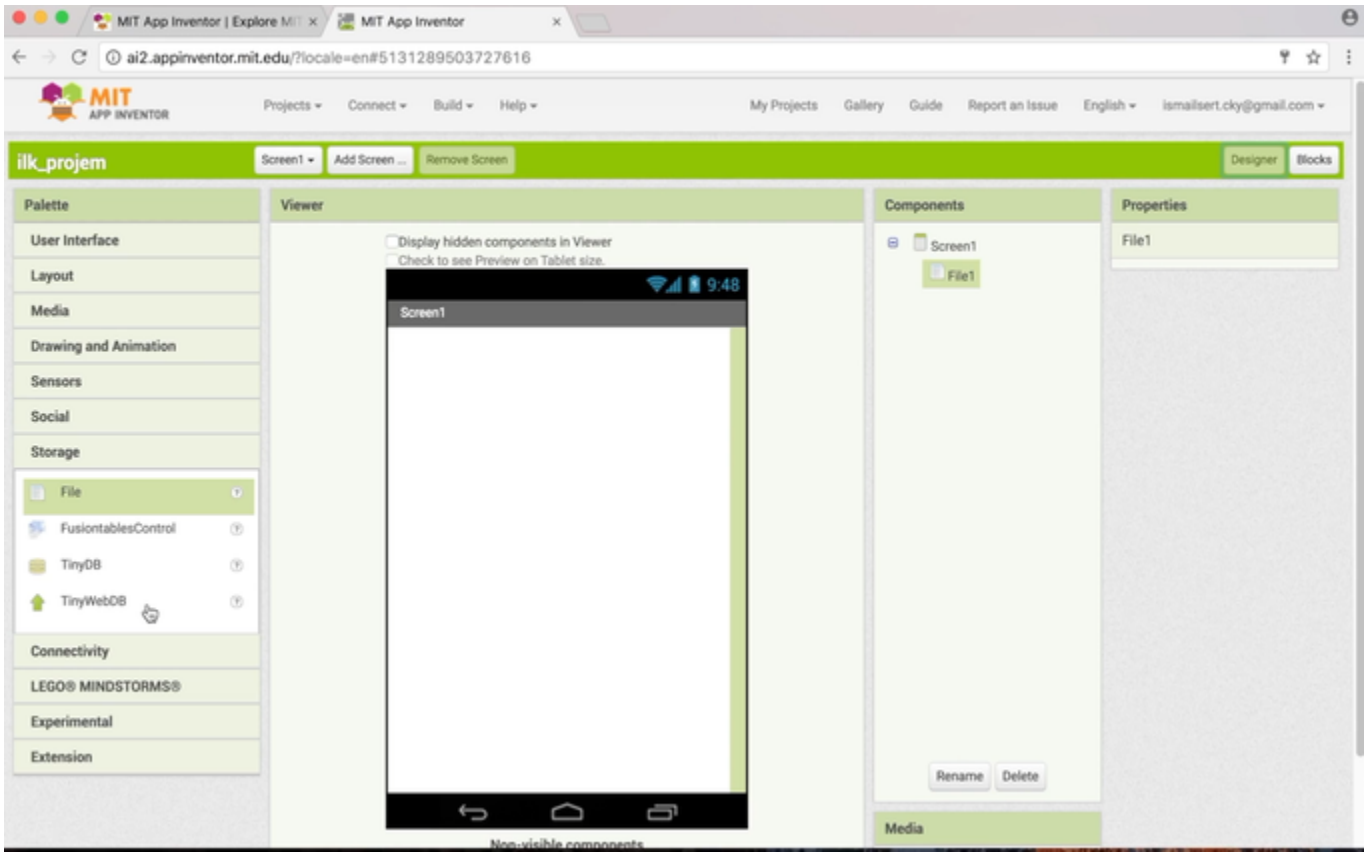
Müfredat Kodu / Ünite / Konu	İCT-104-8 / MIT App Inventor / Storage Menüsü
Hedeflenen Süre	18 Dakika

🚀 Storage Menüsü

Storage menüsü ile telefonunuzun hafızasına yada uzaktaki bir sunucuya dosya yükleme, kayıt ekleme gibi işlemler gerçekleştirebilirsiniz.

- File: Dosya kaydı, dosya oluşturma, dosya okuma gibi işlemlerin gerçekleştirildiği araçtır.
- TinyWebDB: Uzaktaki sunucuya ulaşabilen, web tabanlı bir veri yönetim sistemidir.

📸 Storage Menüsü Arayüzü



Kazanım

- Bu bölümde öğrenciler; veri yönetimi, uzaktaki bir sunucuya dosya yükleme ve kayıt ekleme işlemlerini öğrenir.

✅ Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar

1	☑ Storage menüsünün tanıtılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere konuya ait kavramların tanıtılması ve uygulanması beklenmektedir. Öğrencilere Storage menüsünün tanıtılması. Tanıtılan araçların kullanılarak öğrencilerin anlamasını kolaylaştırmak adına eğitmen tarafından uygulamaların yapılması. Tanıtılan araçlar hakkında soruların sorulup, konunun anlaşılabilirliğinin ölçülmesi. Öğrencilere, yukarıda tanıtılan araçlar üzerinde uygulama yapmasını ödevlendirmek.
2	☑ Storage menüsüne ait uygulamaların tamamlanması.	☐ Eğitmen ☑ Öğrenciler	Bu alanda öğrencilere uzaktan erişim, veri yönetimi gibi konuların öğretilmesi amaçlanmıştır.

MIT APP INVENTOR: CONNECTIVITY MENÜSÜ

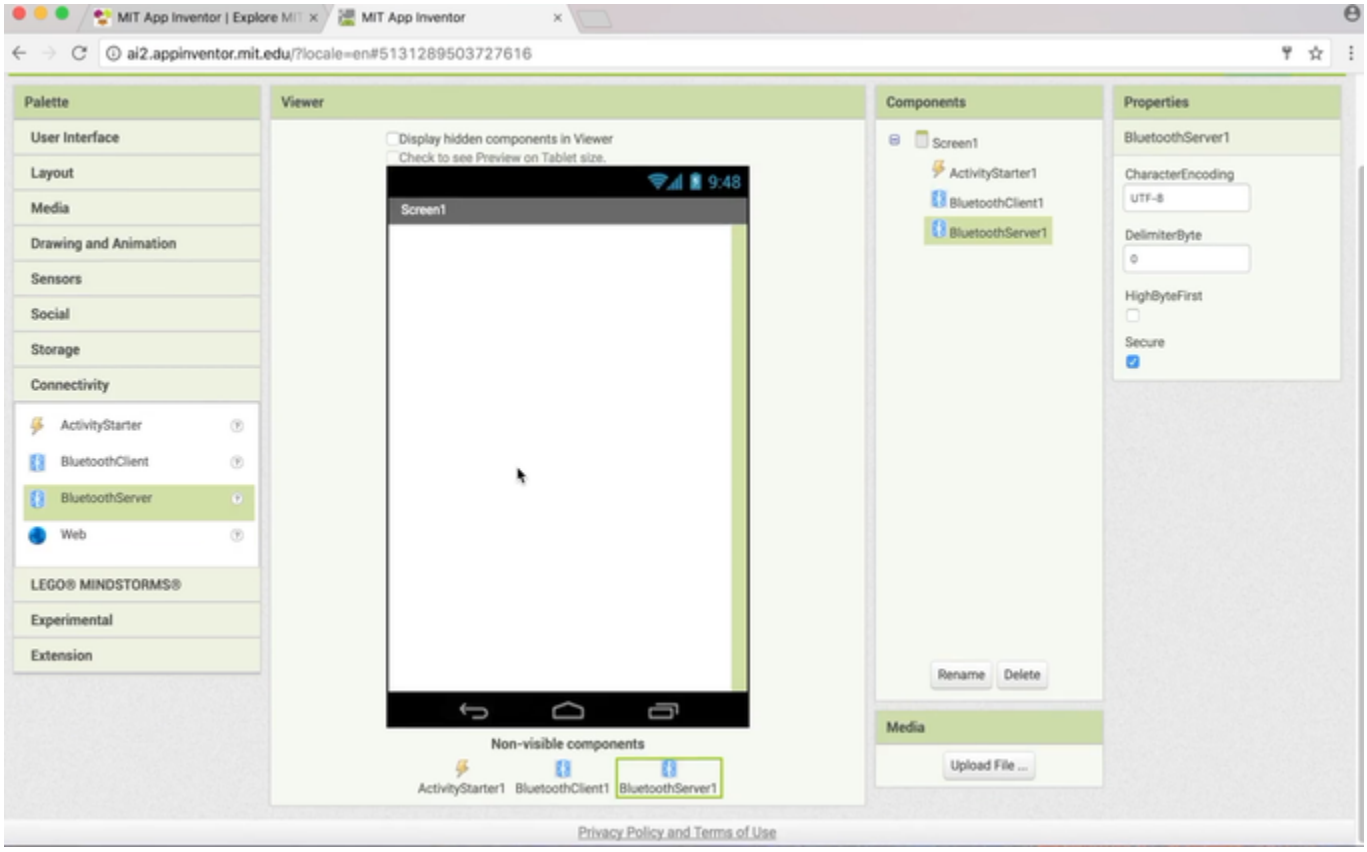
Müfredat Kodu / Ünite / Konu	İCT-104-9 / MIT App Inventor / Connectivity Menüsü
Hedeflenen Süre	18 Dakika

Connectivity Menüsü

Bu ünitede Connectivity Menü altındaki araçları inceleyeceğiz.

- ActivityStarter: Bu bileşen sayesinde telefonunuzun içerisindeki uygulamaları çağırıp çalıştırabilirsiniz.
- BluetoothClient: Bu bileşeni eklediğiniz cihaz yalnızca arama yapabilir ve bağlantı sağlayabilir.
- BluetoothServer: Bu bileşeni eklediğiniz cihaz yayın , veri aktarımı, veri gönderimi işlemlerini gerçekleştirebilir.
- Web: Bu araç User interface alanında bulunan WebViewer'ın daha detaylı halidir.

Connectivity Menüsü Arayüzü



Kazanım

- Bu bölümde öğrenciler; telefon içerisindeki uygulamaları çağırıp, çalıştırmayı öğrenmektedir.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Connectivity menüsünün tanıtılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere Connectivity menüsü altındaki kavramların tanıtılması. Öğrencilere Connectivity menüsünün tanıtılması. Tanıtılan araçların kullanılarak öğrencilerin anlamasını kolaylaştırmak adına eğitmen tarafından uygulamaların yapılması. Tanıtılan araçlar hakkında soruların sorulup, konunun anlaşılabilirliğinin ölçülmesi. Öğrencilere, yukarıda tanıtılan araçlar üzerinde uygulama yapmasını ödevlendirmek.
2	☑ Connectivity menüsüne ait uygulamaların tamamlanması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen aracılığıyla öğrencilerin Connectivity Menüsü altındaki araçlara ait uygulama yapması hedeflenmektedir.

MIT APP INVENTOR: DEĞİŞKENLER

Müfredat Kodu / Ünite / Konu	İCT-104-10 / MIT App Inventor / Değişkenler
Hedeflenen Süre	18 Dakika

🚀 Değişkenler

Değişkenler, bilgisayarın ram belleğinde tutulan geçici hafıza odacıklarıdır.

1.Değişken : "initialize global name to"

Bu blok global değişkenler oluşturmak için kullanılır. Herhangi bir değer türünü argüman olarak alır. İsim üzerine tıklandığında bu global değişkenin ismi değiştirilir. Global değişkenler tüm prosedürlerde veya olaylarda kullanılır, bu nedenle bu blok tek başına duracaktır.

Global değişkenler bir uygulama çalışırken değiştirilebilir.Bu bloğu istediğiniz zaman yeniden adlandırabilirsiniz ve eski ada atıfta bulunan tüm ilişkili bloklar otomatik olarak güncellenir.

2.Değişken : "get"

Bu blok, oluşturmuş olabileceğiniz değişkenleri almak için bir yol sağlar.

3.Değişken : "set"

Bu blok get ile aynı kuralları izler. Açılır menüde yalnızca kapsam dahilindeki değişkenler kullanılabilir. Bir v değişkeni seçildikten sonra, v'ye yeni bir değer vermek için bir blok ekleyebilirsiniz.

4.Değişken : "initialize local name to - in (do)"

Bu blok, yalnızca bloğun DO kısmında çalıştırdığınız prosedürde kullanılan yeni değişkenler oluşturmanıza olanak tanıyan bir mutatördür. Bu şekilde, bu prosedürdeki tüm değişkenler prosedür her çalıştırıldığında aynı değerle başlayacaktır.

NOT: Bu blok, bir DO bloğu olduğu için aşağıda açıklanan bloktan farklıdır. Buna deyimler ekleyebilirsiniz. İfadeler bir şeyler yapar. Bu nedenle bu bloğun içinde deyim bloklarının eklenmesi için boşluk vardır.

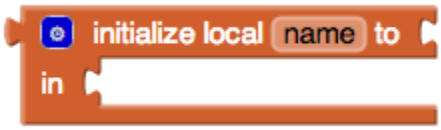
Bu bloktaki değişkenleri istediğiniz zaman yeniden adlandırabilirsiniz ve programınızın başka yerlerinde eski ada atıfta bulunan tüm ilgili bloklar otomatik olarak güncellenecektir.

5.Değişken : initialize local name to - in (return)

Bu blok, yalnızca bloğun RETURN kısmında çalıştırdığınız prosedürde kullanılan yeni değişkenler oluşturmanıza olanak tanıyan bir mutatördür. Bu şekilde, bu prosedürdeki tüm değişkenler prosedür her çalıştırıldığında aynı değerle başlayacaktır.

NOT: Bu blok, bir RETURN bloğu olduğu için yukarıda açıklanan bloktan farklıdır. Buna ifadeler ekleyebilirsiniz. İfadeler bir değer döndürür. Bu nedenle bu blokta ifadeleri takmak için bir soket vardır.

Bu bloktaki değişkenleri istediğiniz zaman yeniden adlandırabilirsiniz ve programınızın başka yerlerinde eski isme atıfta bulunan tüm ilgili bloklar otomatik olarak güncellenecektir.



Kazanım

- Bu bölümde öğrenciler, değişkenler konusunu öğrenir ve uygulamalarla öğrendiklerini pekiştirir.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Değişkenler konusunun anlatılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere değişkenlerin tanıtılması, Öğrencilere değişken atama yönteminin gösterilmesi, Atanan değişkenlerin kullanılarak öğrencilerin anlamasını kolaylaştırmak adına eğitmen tarafından uygulamaların yapılması, Öğrencilere değişkenler konusu hakkında soruların sorulup, konunun anlaşılabilirliğinin ölçülmesi, Öğrencilere, değişken atama ve değişkenleri kullanma hakkında ödevlendirme yapmak, Konunun daha iyi kavranması için, öğrencilerinizle birlikte uygulamalar bölümü altındaki https://hackercan.atlassian.net/wiki/spaces/ICT/pages/65113191#%F0%9F%9A%80-Alarm-Uygulamas%C4%B1'nı yapabilirsiniz beklenmektedir.
2	☑ Değişkenler konusuna ait uygulamaların tamamlanması.	☐ Eğitmen ☑ Öğrenciler	Öğrencilerin değişkenler ve değişken çeşitlerini uygulayarak öğrenmesi hedeflenmektedir.

RENKLENDIRME UYGULAMASI

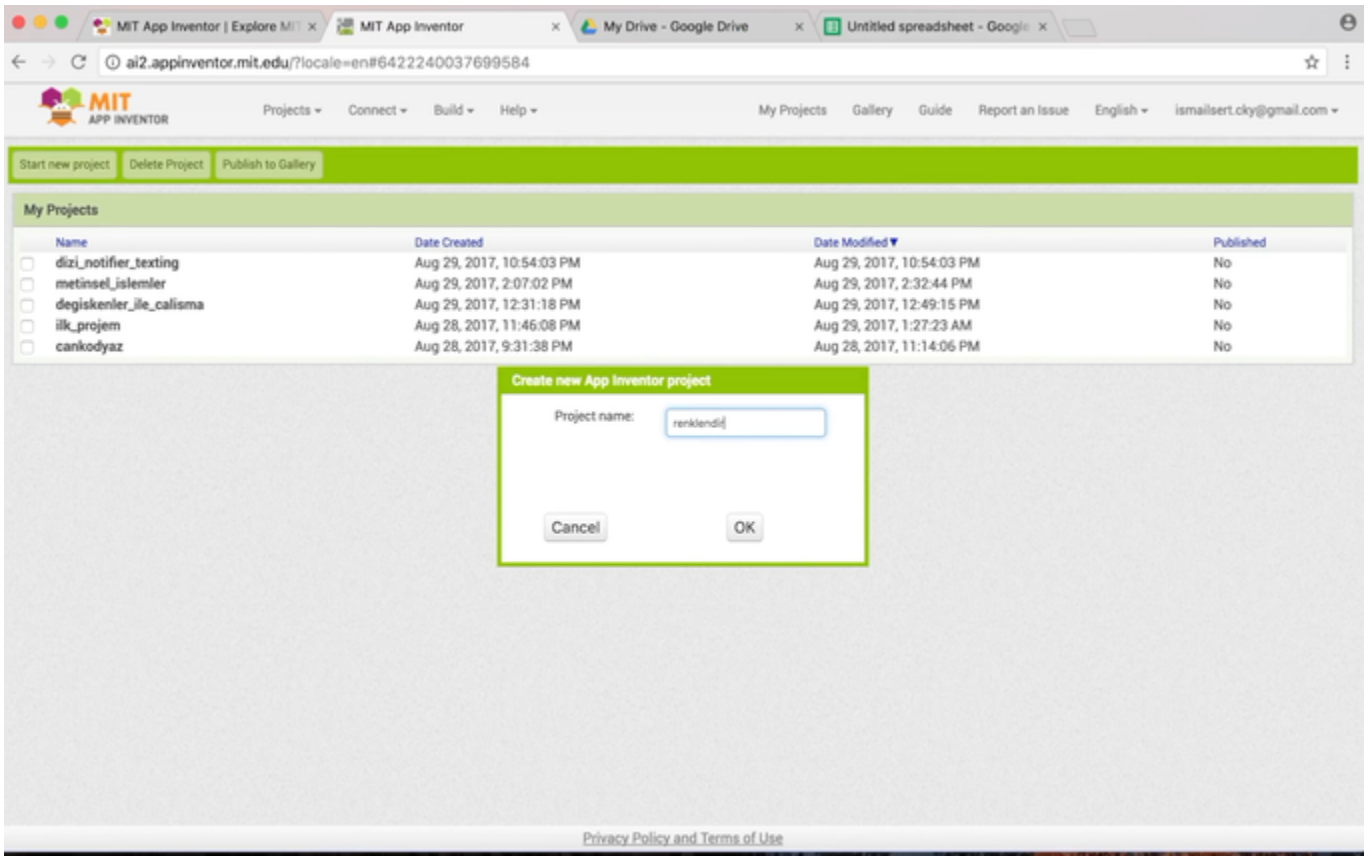
Müfredat Kodu / Ünite / Konu	İCT-104-11.1 / MIT App Inventor Uygulamaları / Renklendirme Uygulaması
Hedeflenen Süre	45 Dakika

Renklendirme Uygulaması

Bu bölümde Renklendirme Uygulaması ile ekranı salladığımızda arka plan renginin değişmesini hedefliyoruz.

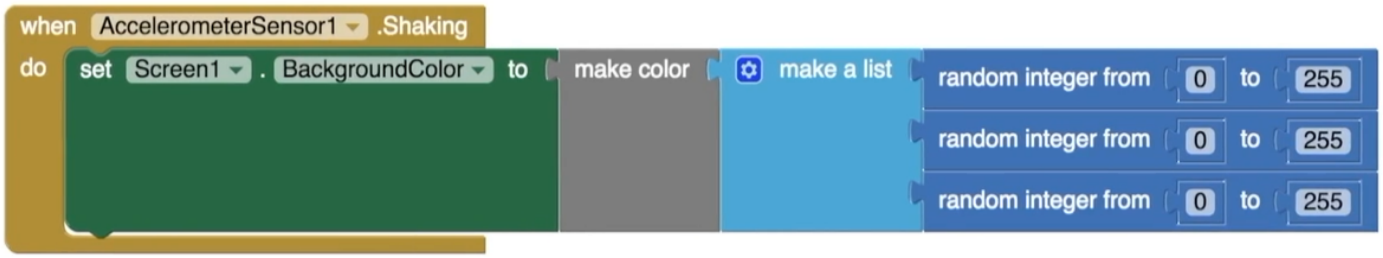
Uygulama

1. Adım : İlk olarak yeni bir proje ekliyoruz.



2. Adım : Sensors paneli üzerinde bulunan AccelerometerSensor aracını screen üzerine ekliyoruz ve kod blokları alanına geçiyoruz.

Kod Bloğu



Kazanım

- Bu alan öğrencilerin AccelerometerSensor sensoru ile ivme, BackgroundColor aracı ile arkaplan rengini öğrenmesini hedeflemektedir.

BOYAMA UYGULAMASI

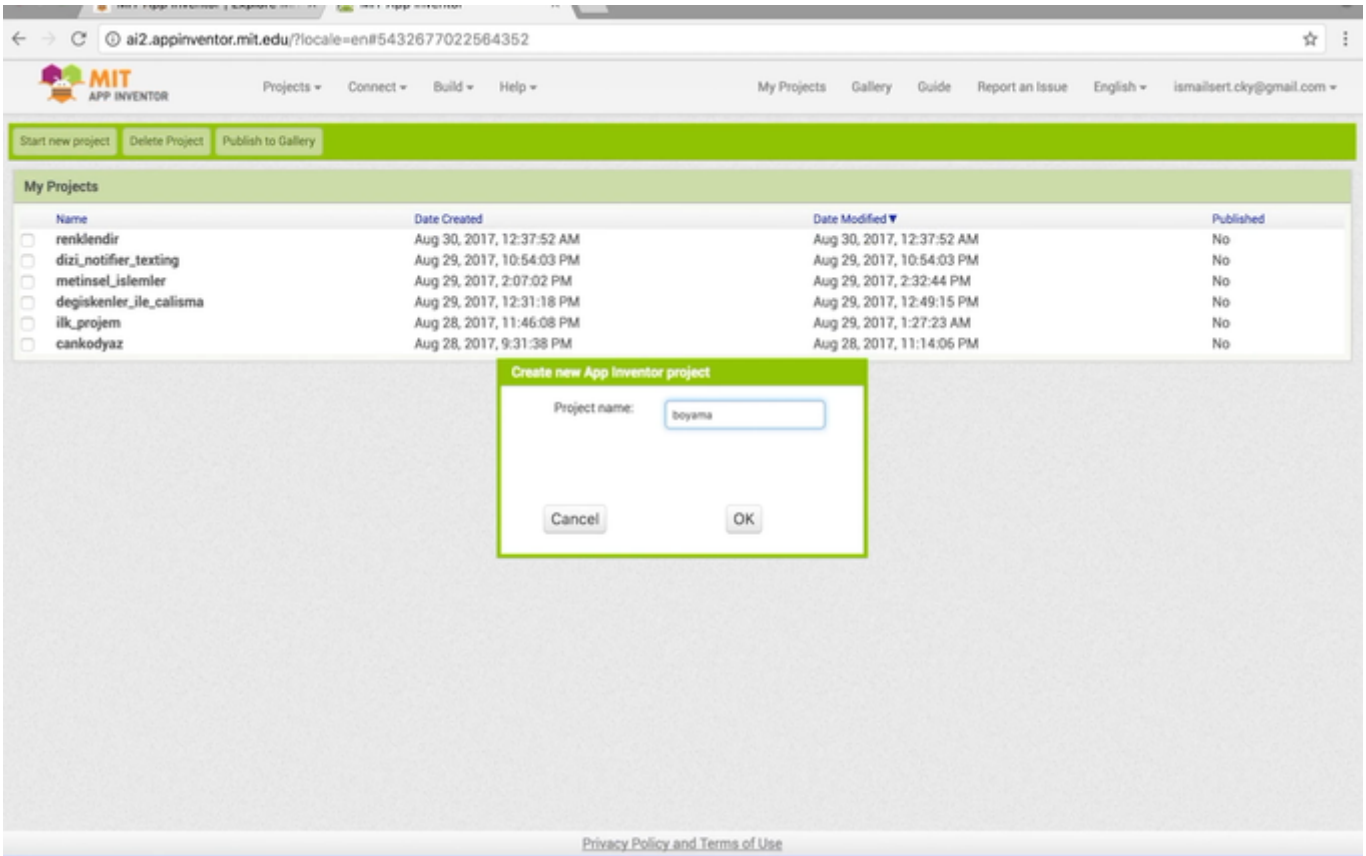
Müfredat Kodu / Ünite / Konu	İCT-104-11.2 / MIT App Inventor Uygulamaları / Boyama Uygulaması
Hedeflenen Süre	45 Dakika

🚀 Boyama Uygulaması

Bu bölümde Boyama Uygulaması yapmayı hedefliyoruz.

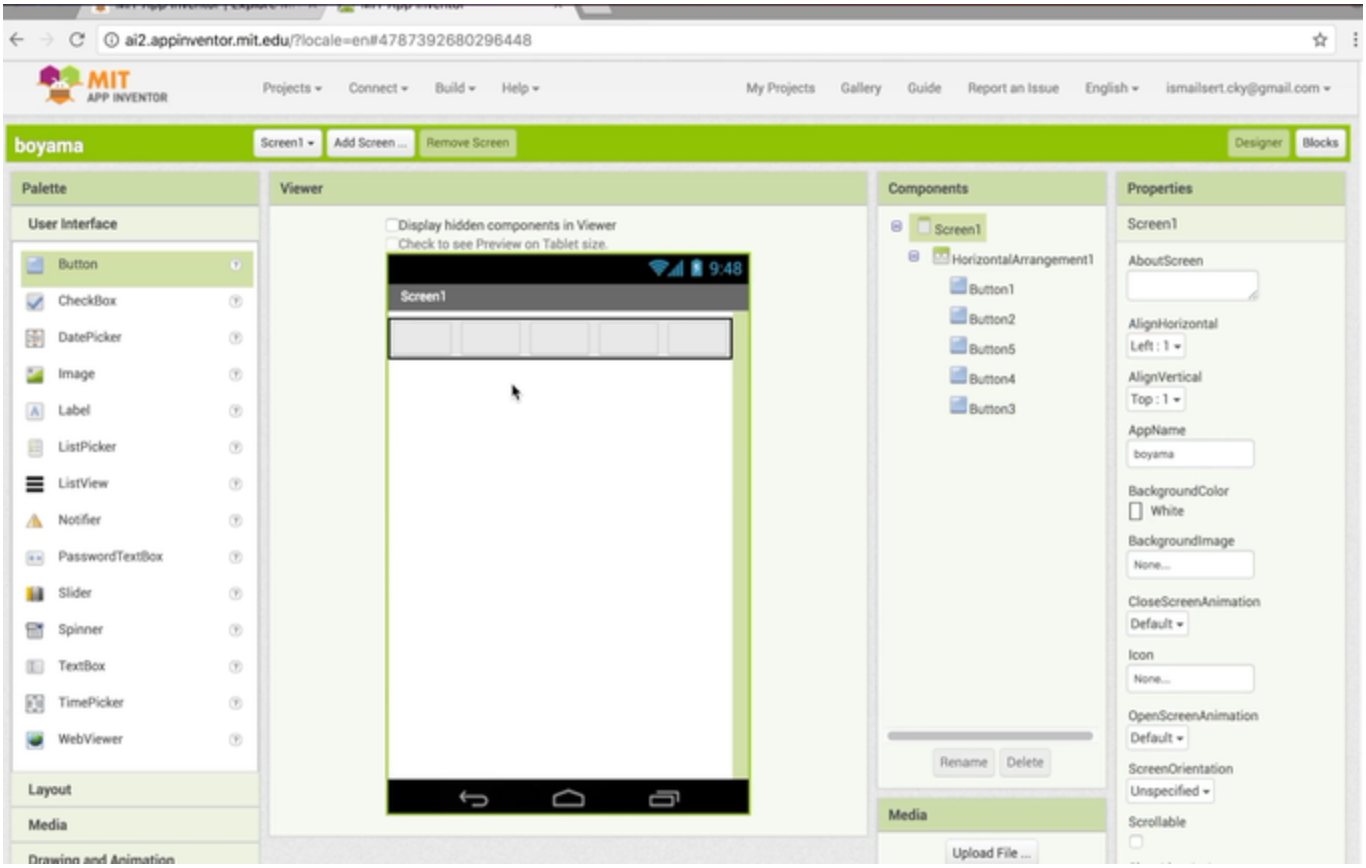
💻 Uygulama

1. Adım : İlk olarak yeni bir proje oluşturuyoruz.

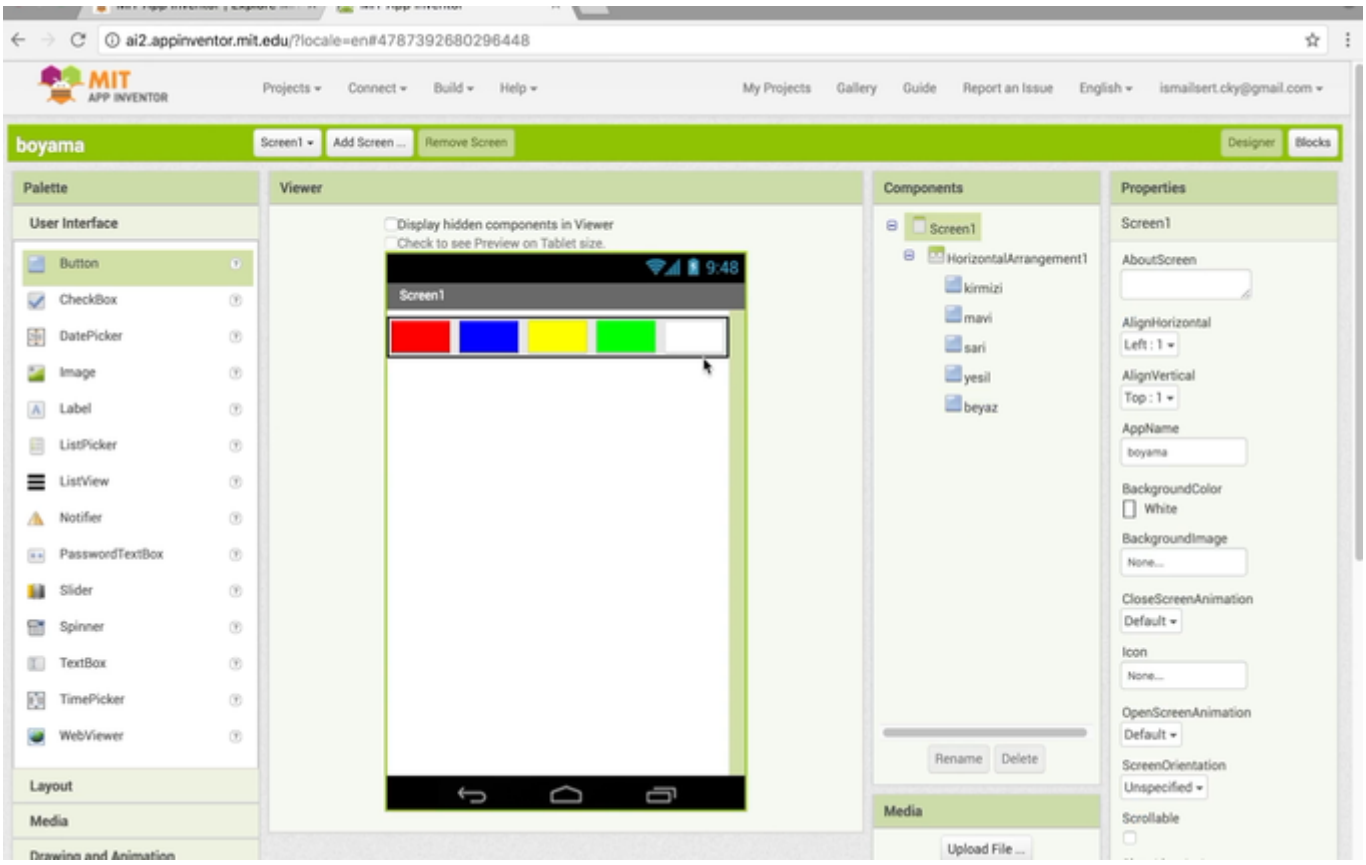


2. Adım : Görsel kısmını oluşturmaya başlıyoruz.

- Sayfanın hizalanması için Layout paneli altındaki HorizontalArrangement aracını screen üzerine ekliyoruz.
- Uygulamamızın üst tarafında renklerin seçilebilmesi için yan yana 5 tane buton ekliyoruz.
- Butonlara ait değerleri Height: 30px; ve Width: Fill parent olarak ayarlıyoruz.
- Butonların içerisindeki yazıları (text değerini) temizliyoruz.

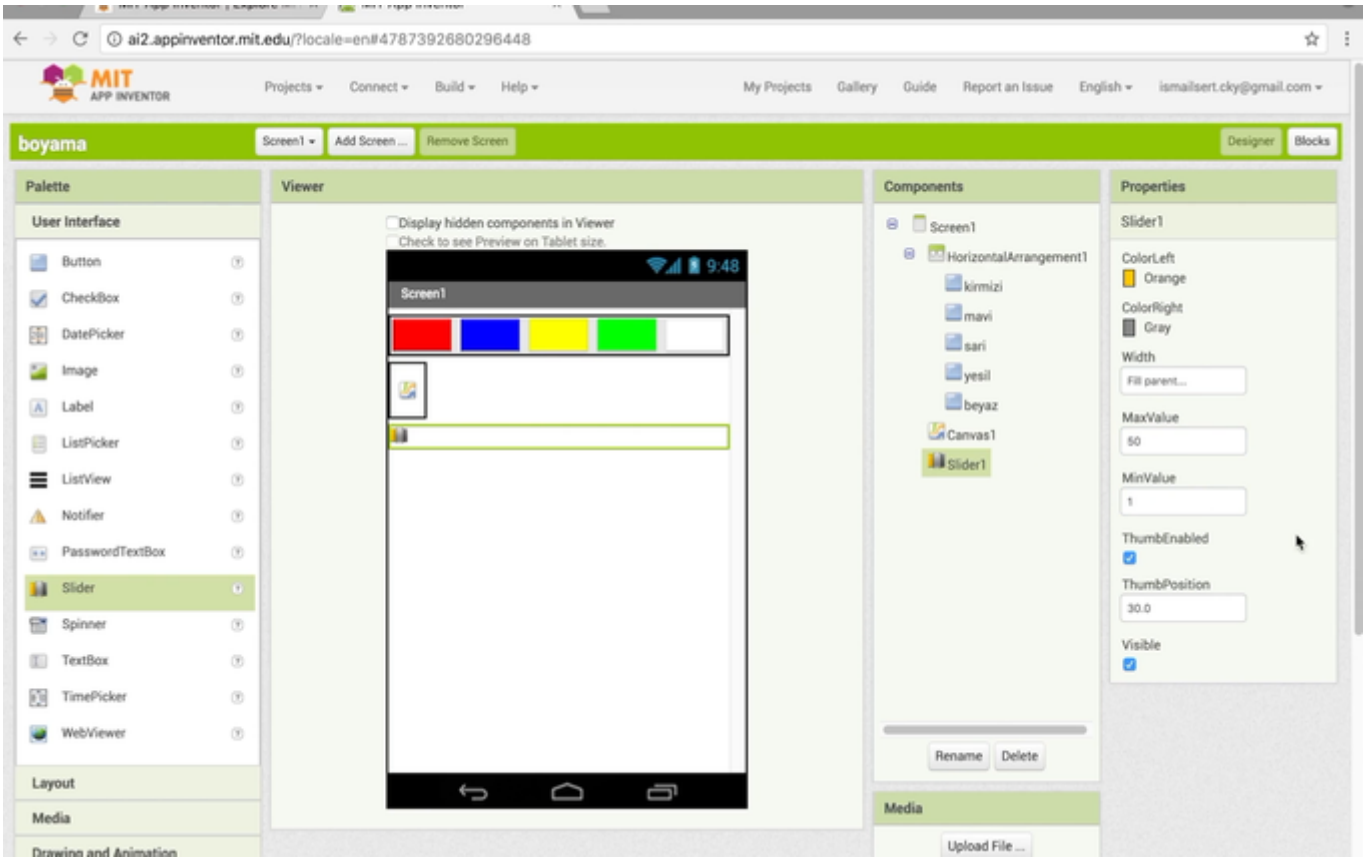


3. Adım : Renk eklemek istediğimiz buton seçiliyken sağ taraftaki properties panelinde bulunan BackgroundColor ayarını istediğimiz rengi seçerek güncelleyebiliriz.

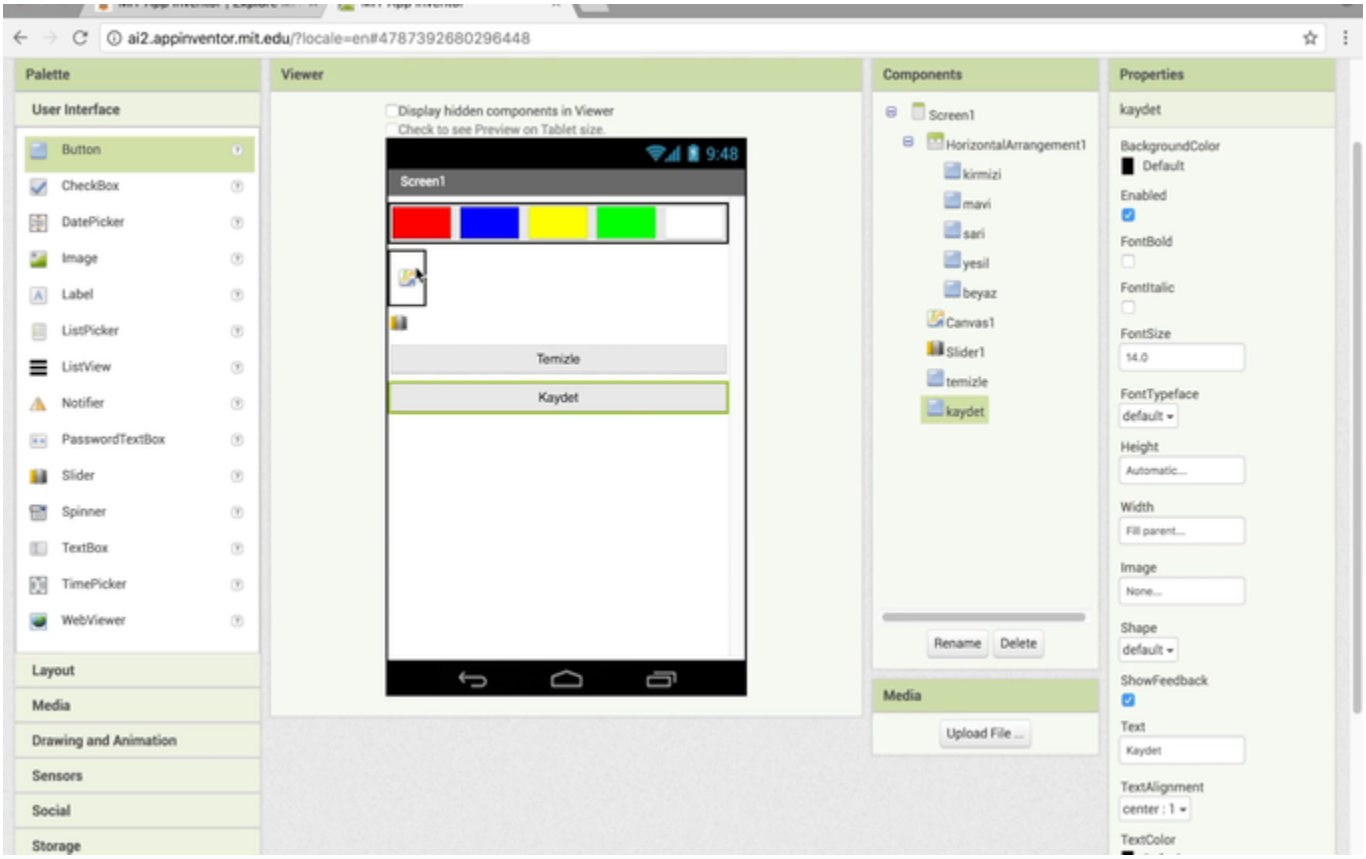


4. Adım : Drawing and Animation başlığı altındaki Canvas aracını screen üzerine ekliyoruz.

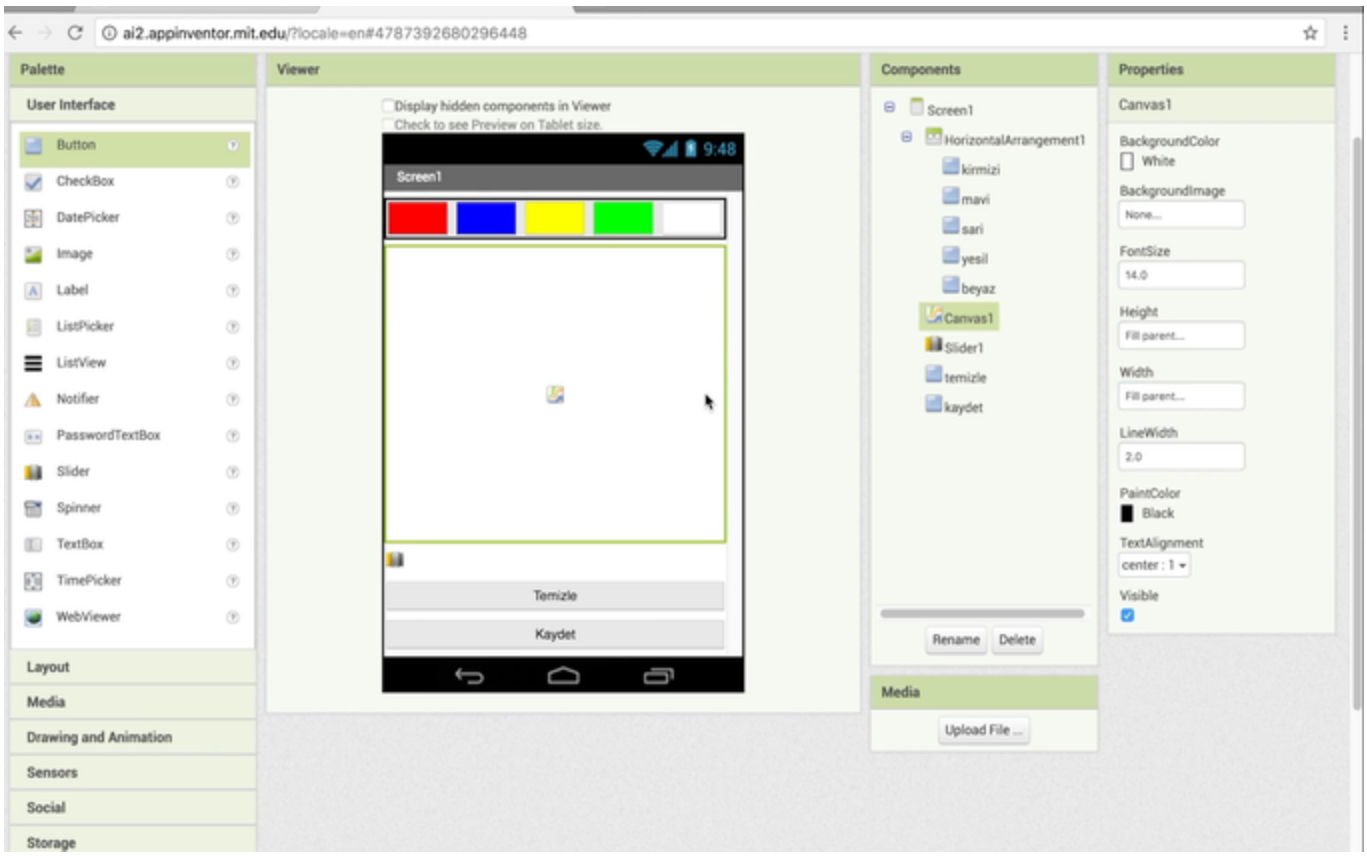
5. Adım : Kalemimizin kalınlığını belirtmek için User Interface paneli altındaki Slider aracını ekrana ekliyoruz.



6. Adım : User Interface paneli altından 2 adet buton ekliyoruz.Eklediğimiz butonların genişliklerini "Fill Parent" olarak güncelledikten sonra isimlerini düzenliyoruz.



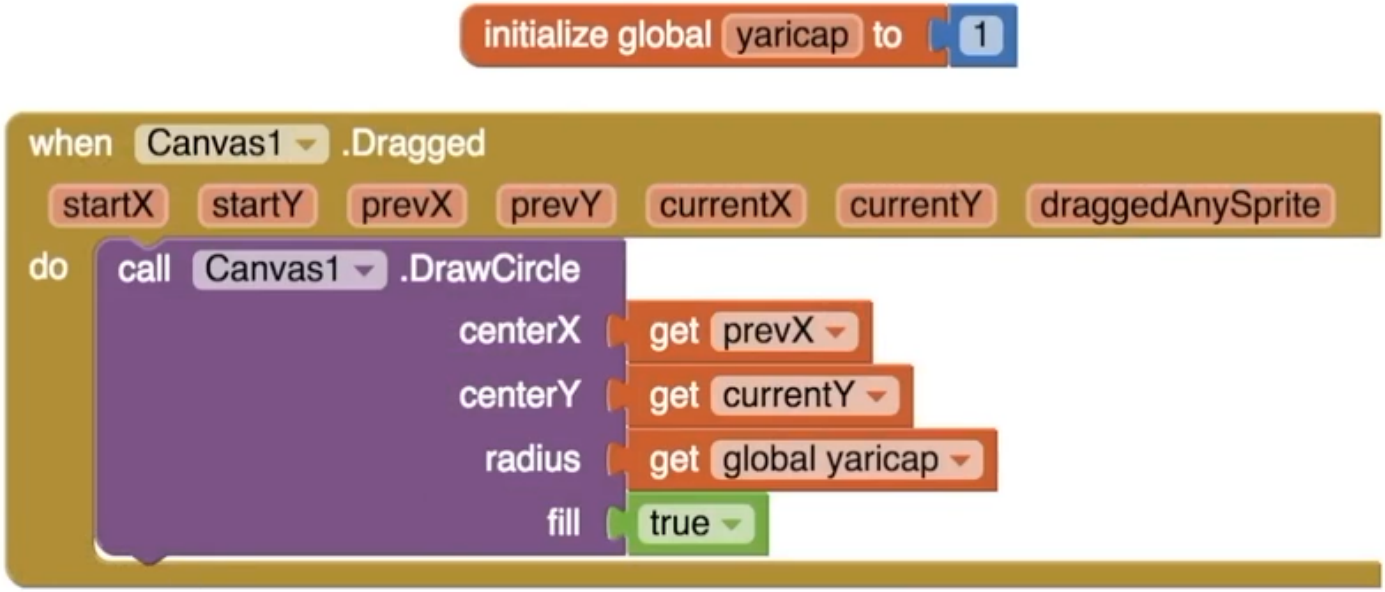
7. Adım : Canvas aracına ait weight ve height değerlerini "Fill parent" olarak güncelliyoruz.

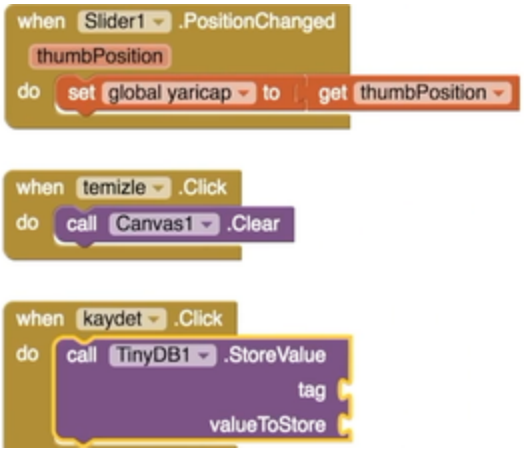


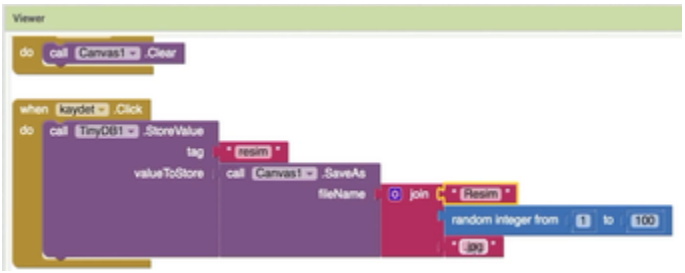
8. Adım : Son adımda Kaydet butonuna tıkladığımızda ihtiyacımız olan TinyDb aracını ekran üzerine ekliyoruz ve kod bloğuna alanına geçiyoruz.

⌨ Kod Bloğu

when kirmizi .Click
do set Canvas1 . PaintColor to
when mavi .Click
do set Canvas1 . PaintColor to
when sari .Click
do set Canvas1 . PaintColor to
when yesil .Click
do set Canvas1 . PaintColor to
when beyaz .Click
do set Canvas1 . PaintColor to


initialize global yaricap to 1
when Canvas1 .Dragged
startX startY prevX prevY currentX currentY draggedAnySprite
do call Canvas1 .DrawCircle
centerX get prevX
centerY get currentY
radius get global yaricap
fill true





Kazanım

- Bu alanda öğrencilerin buton, hizalama, nesne hareketi, görüntü ekleme yollarını öğrenmesi amaçlanmıştır.

PUSULA UYGULAMASI

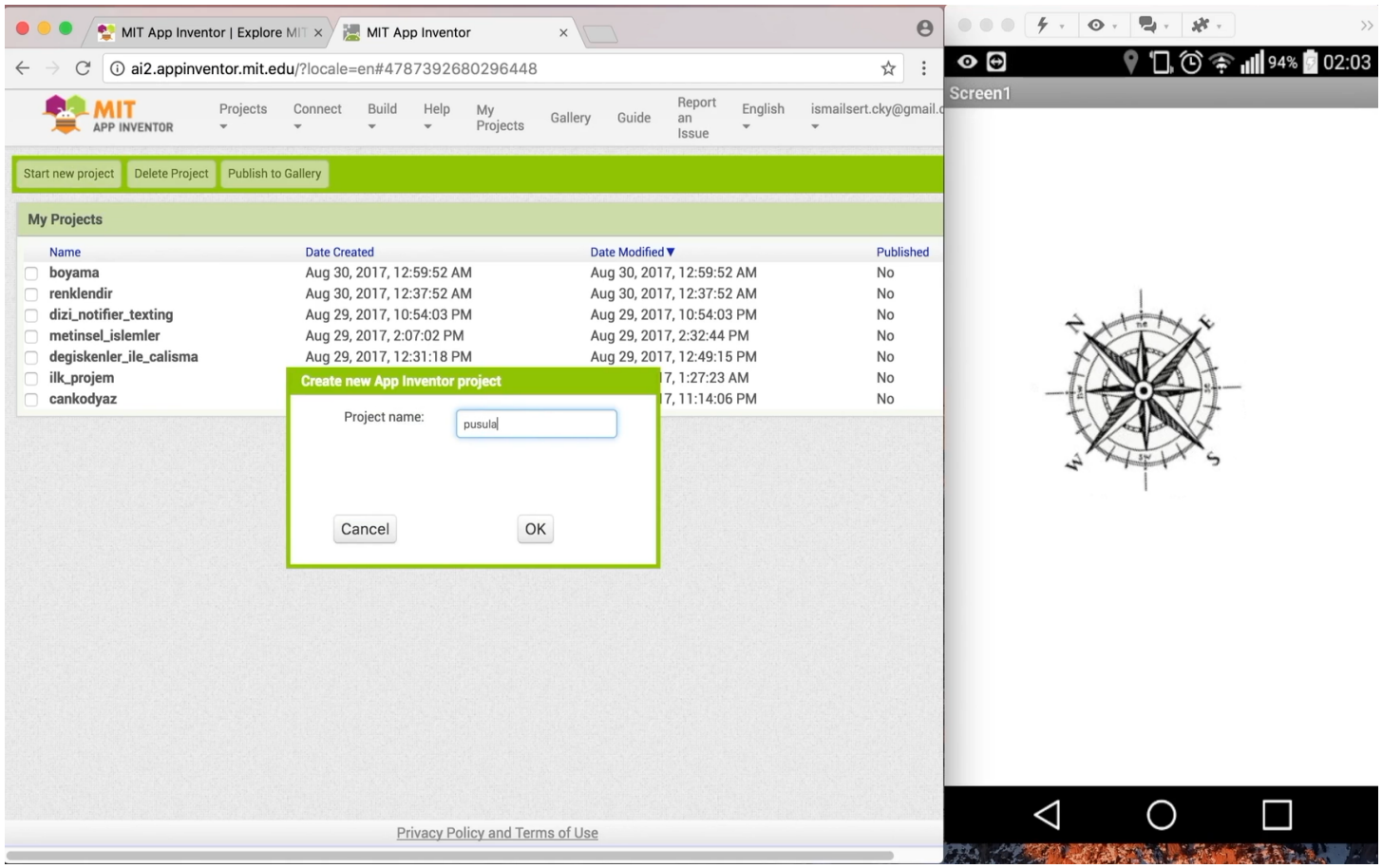
Müfredat Kodu / Ünite / Konu	İCT-104-11.3 /MIT App Inventor Uygulamaları / Pusula Uygulaması
Hedeflenen Süre	90 Dakika

🚀 Pusula Uygulaması

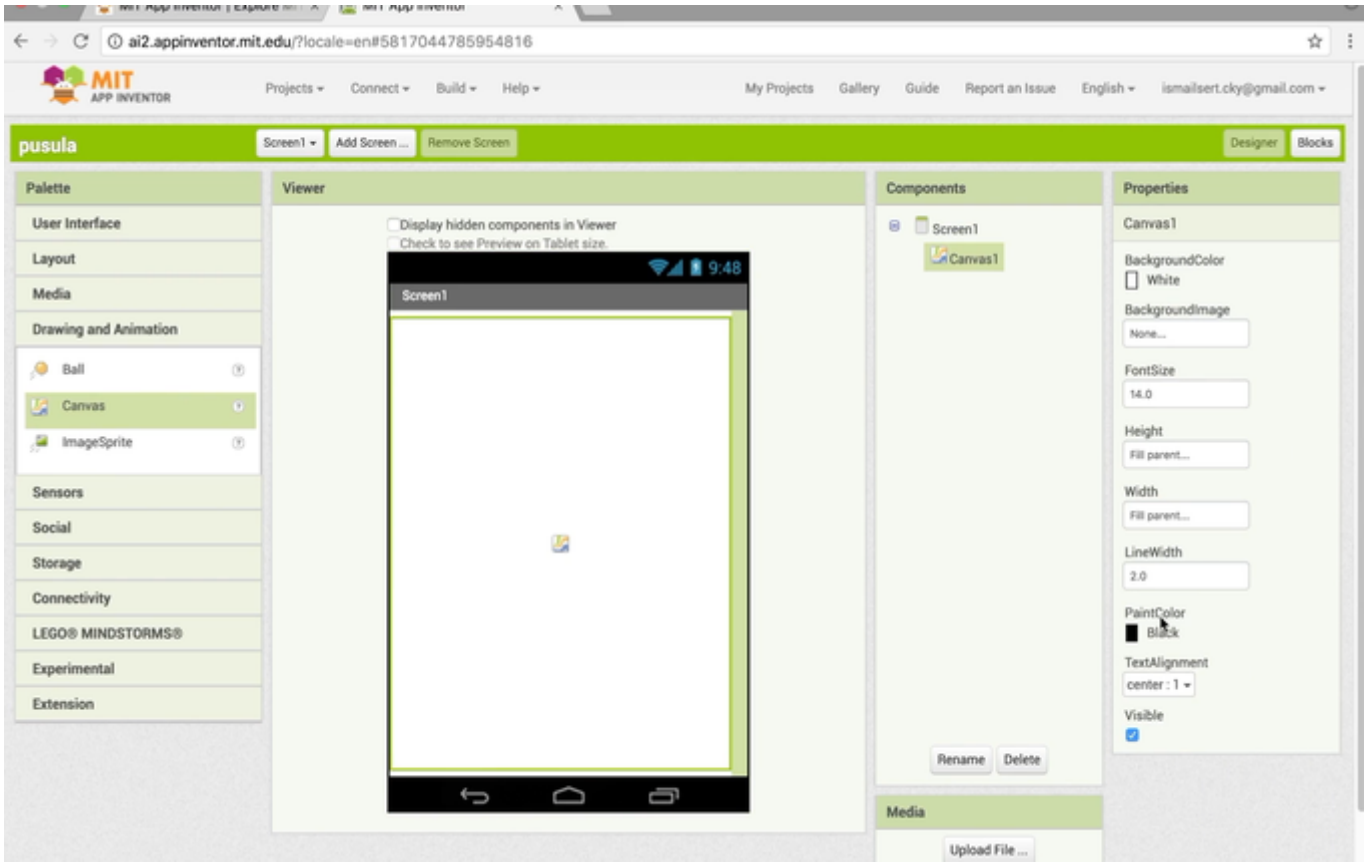
Bu uygulamada telefonumuzun oryantasyon sensörünü kullanarak ve bir görsel ile eşleştirerek ekranda pusula görüntüsünü oluşturmayı hedefliyoruz.

Uygulama

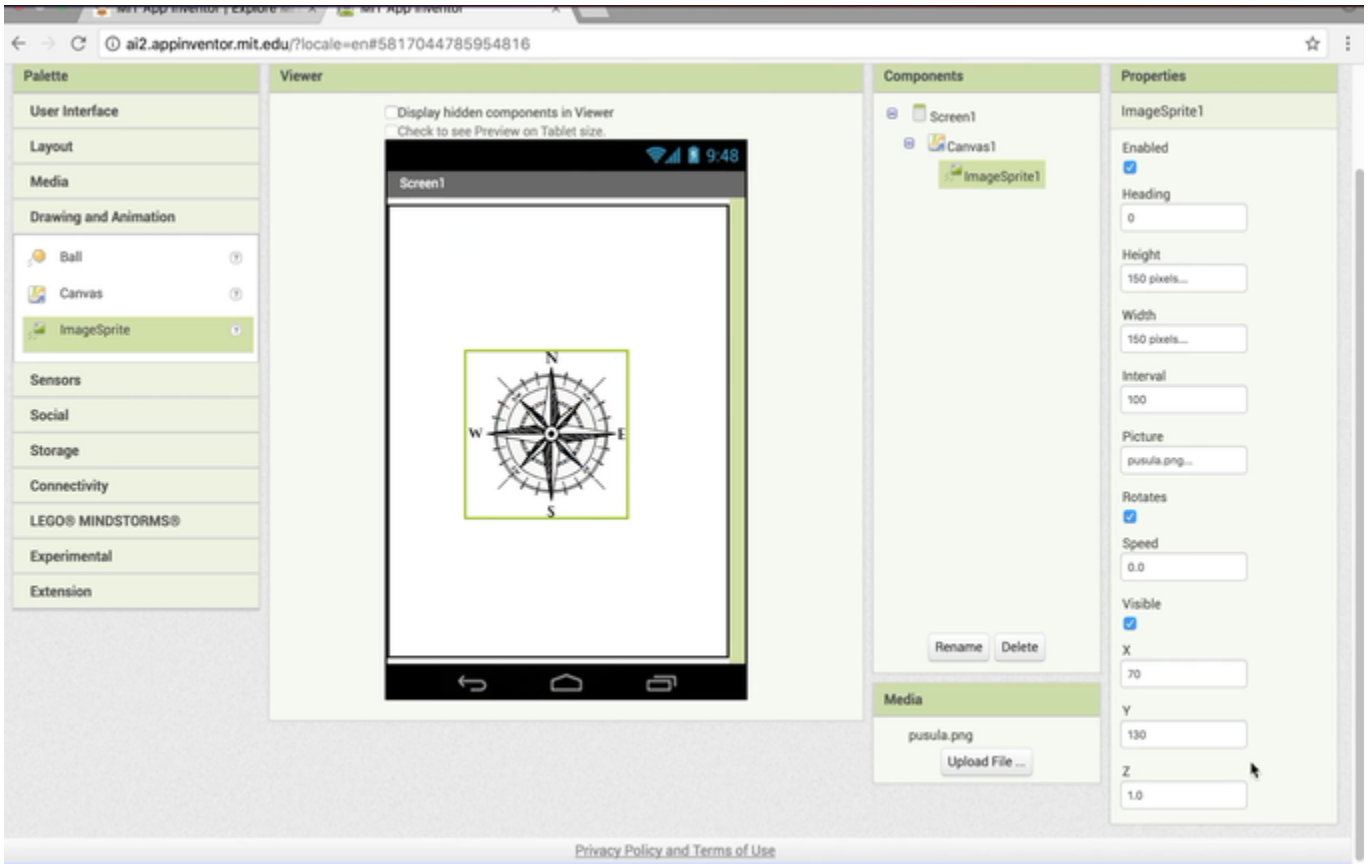
1. Adım : İlk olarak yeni bir proje ekliyoruz.



2. Adım : Drawing and animation başlığı altındaki Canvas aracını screen üzerine ekliyoruz ve width, hight degerlerini 'Fill parent' olarak değiştiriyoruz.

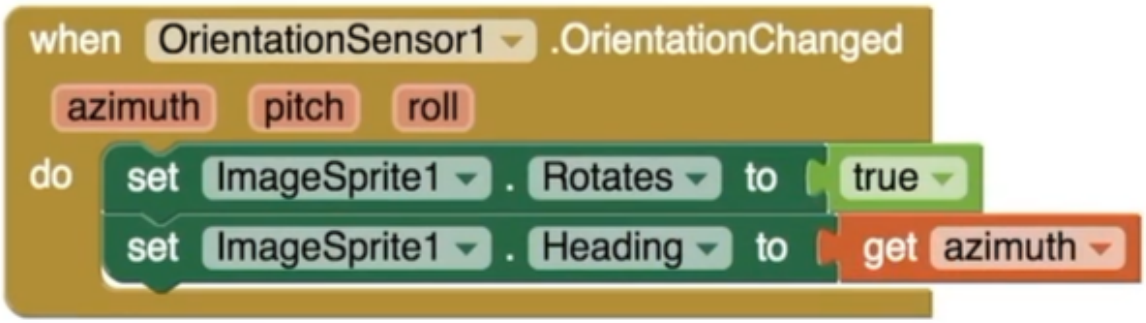


3. Adım : Canvas içerisine bir ImageSprite yerleştiriyoruz. ImageSprite aracına ait properties alanında bulunan picture bölümü ile eklemek istediğimiz pusula görüntüsünü seçiyoruz.



4. Adım : OrientationSensor aracını ekleyerek kod blokları kısmına geçiyoruz.

Kod Bloğu



Kazanım

- Bu bölüm öğrencilerin Canvas ve ImageSprinte araçlarını keşfetmesini sağlar.

ALARM UYGULAMASI

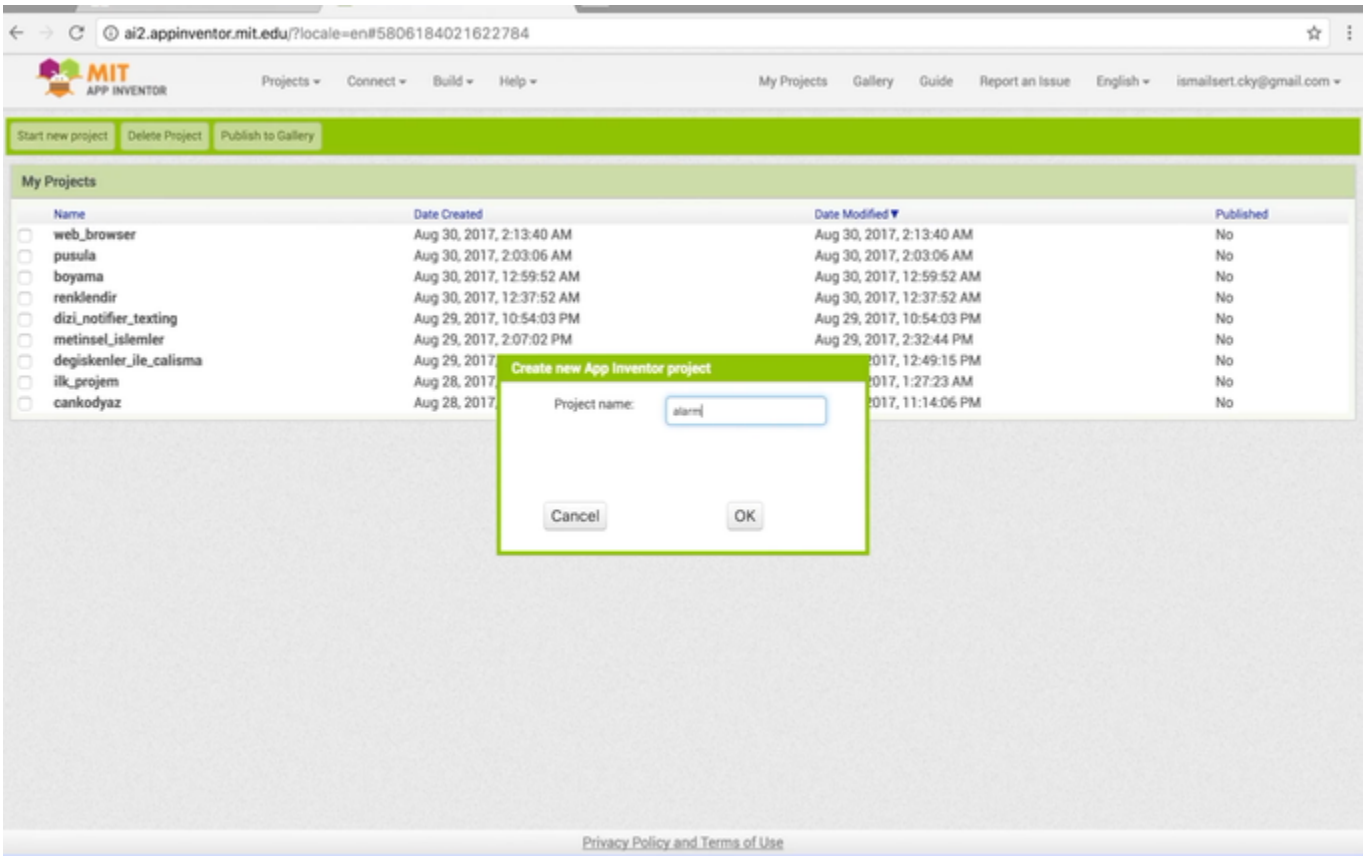
Müfredat Kodu / Ünite / Konu	İCT-104-11.4 / MIT App Inventor Uygulamaları / Alarm Uygulaması
Hedeflenen Süre	90 Dakika

Alarm Uygulaması

Bu bölümde bir alarm uygulaması yapmayı hedefliyoruz.

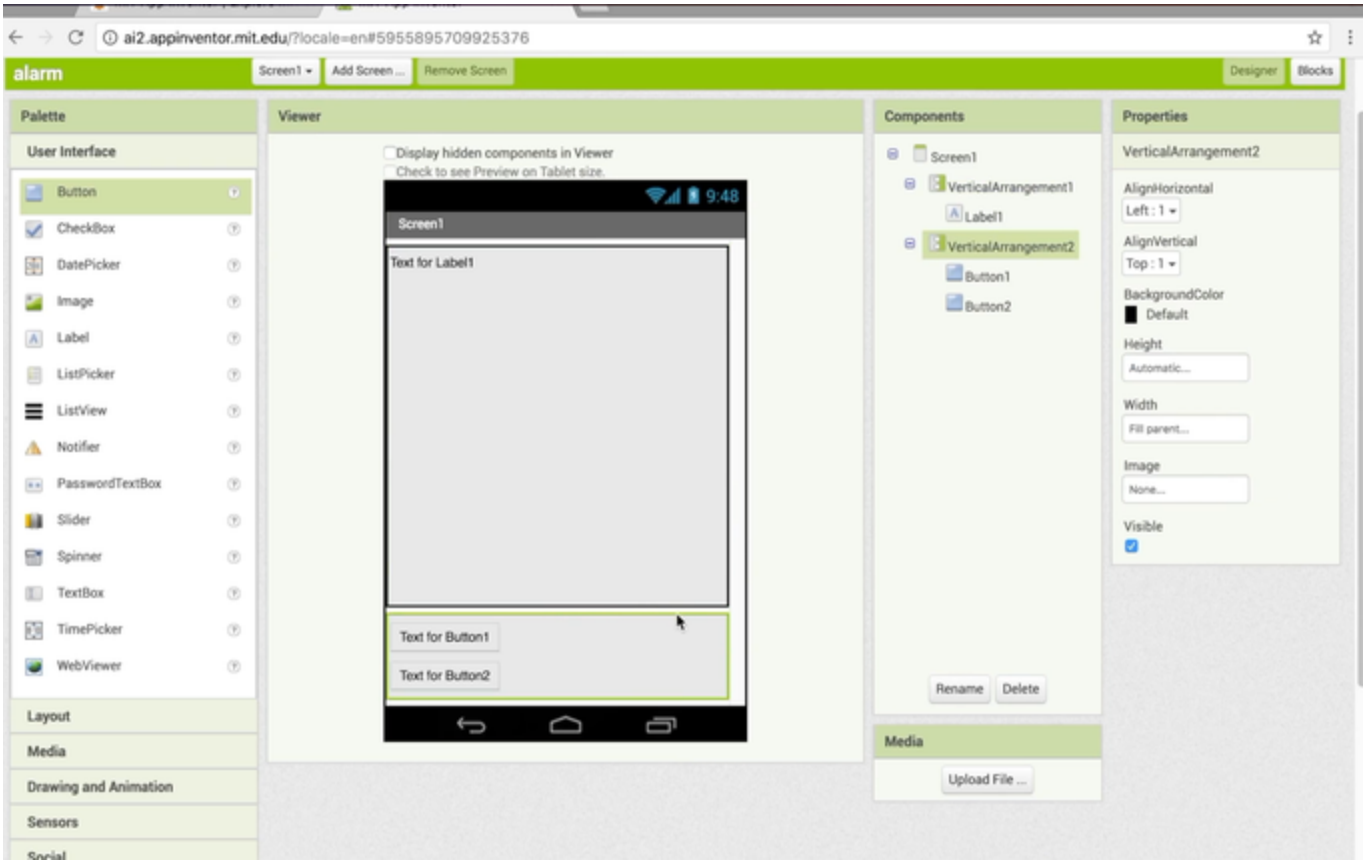
Uygulama

1. Adım : İlk olarak yeni bir proje oluşturuyoruz.

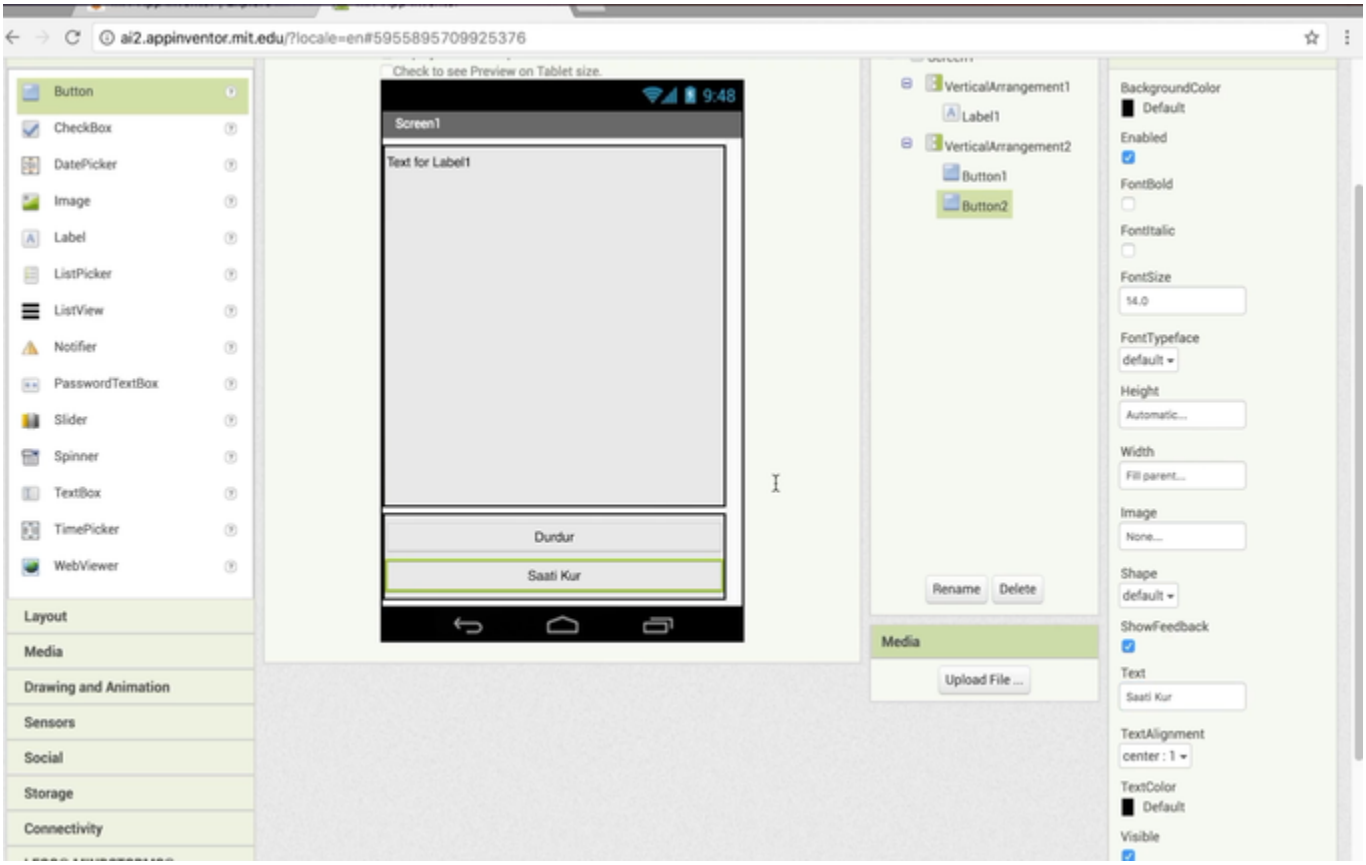


2. Adım : Görsel kısmını oluşturmak için layout başlığı altındaki VerticalArrangement aracını kullanıyoruz.

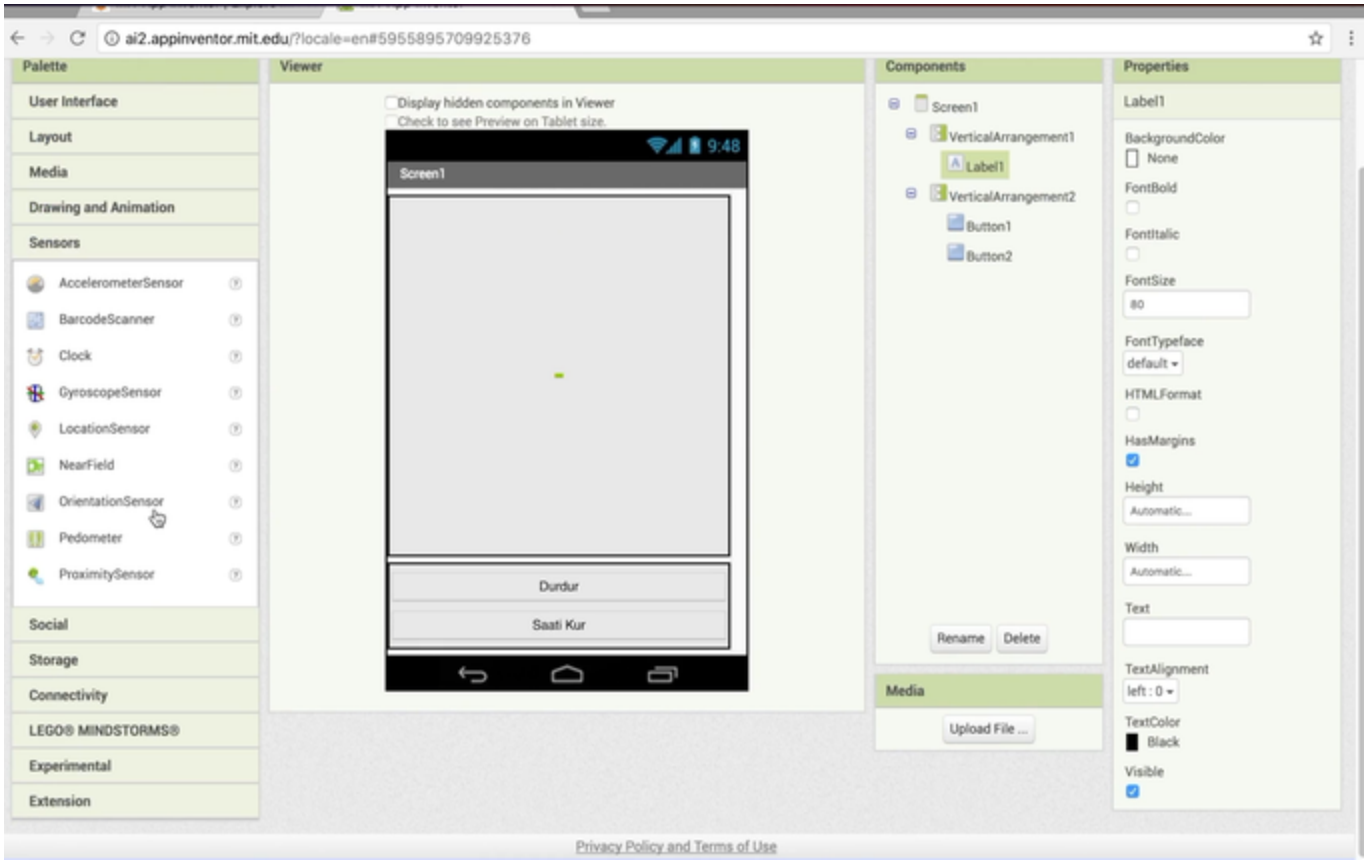
- Birinci oluşturduğumuz VerticalArrangement aracının içerisine bir tane label aracı ekliyoruz.
- İkinci oluşturduğumuz VerticalArrangement aracının içerisine 2 adet buton ekliyoruz.



3. Adım : Butonların width değerlerini 'Fill parent' değeri yaparak butonları ekrana sığdırıyoruz ve buton isimlerini değiştiriyoruz.

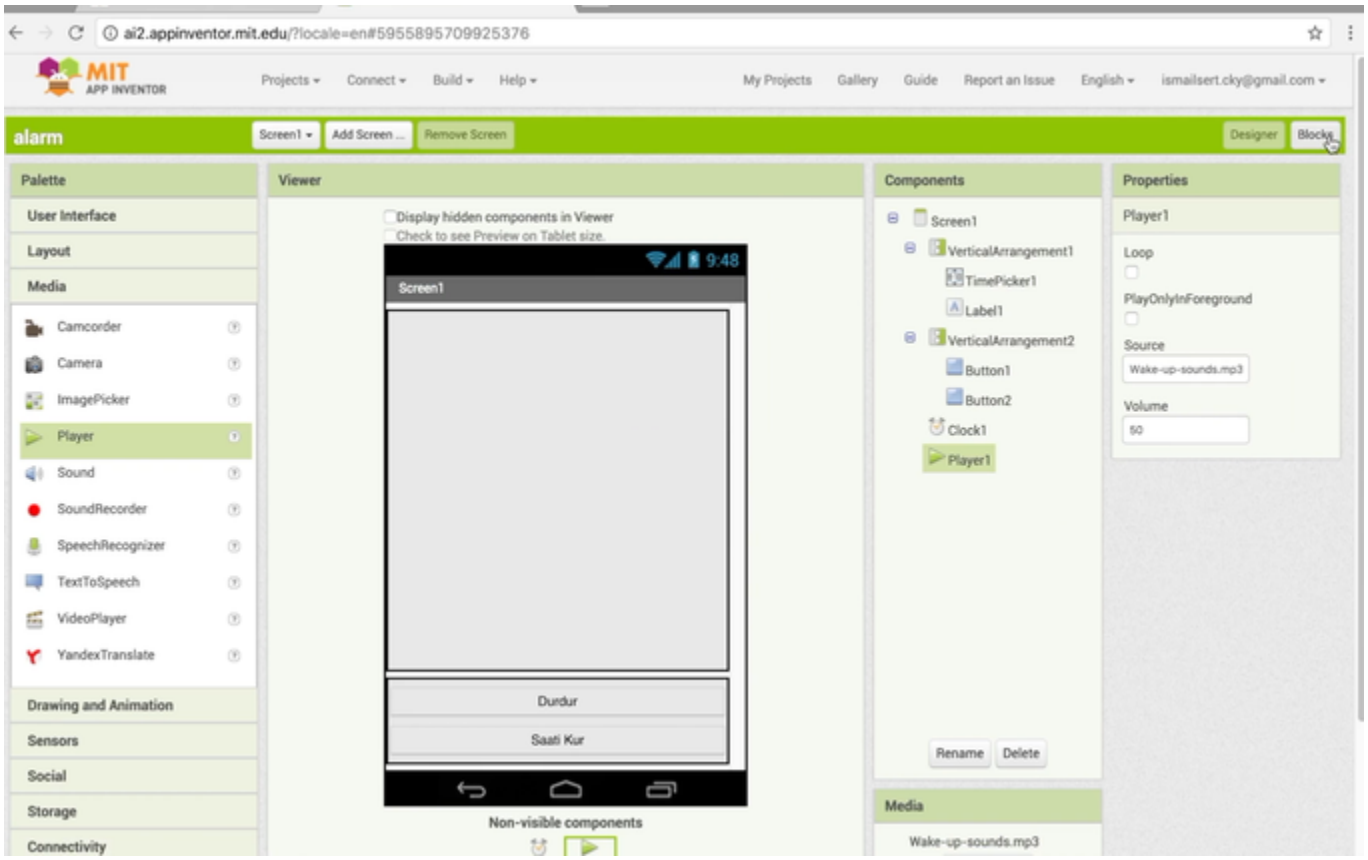


4. Adım : Label aracının font size değerini 80 yapıp, properties alanı üzerinden VerticalArrangement değerine ait hizalama değerlerini AlignHorizontal: Center : 3 , AlignVertical : Center : 2 , olarak hizalıyoruz ve labelin içerisindeki yazıyı siliyoruz.

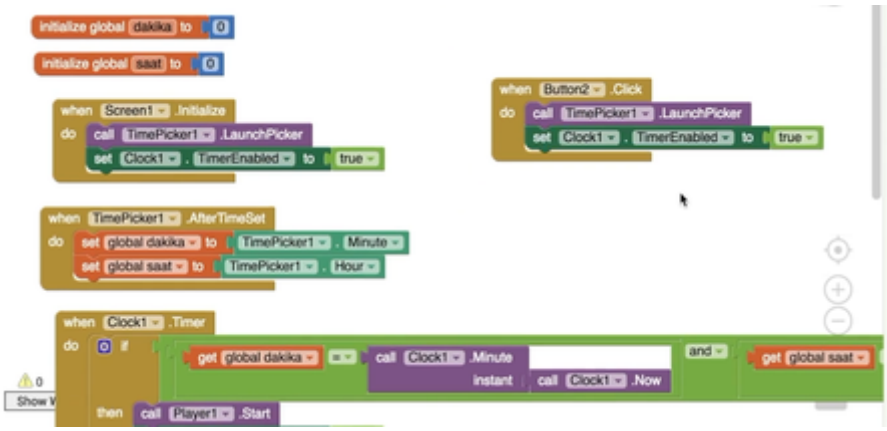


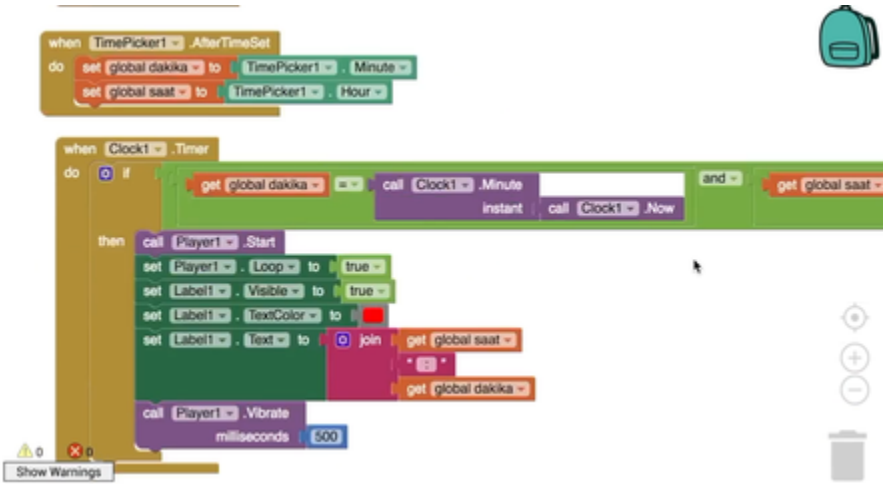
5. Adım : Sensorler bölümündeki Clock sensörünü ve User Interface başlığı altından timepicker aracını screen üzerine ekliyoruz.

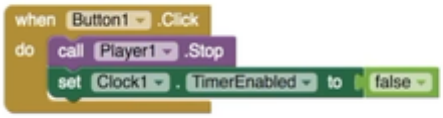
6. Adım : Alarm sesi oluşturmak için media bölümü altında bulunan player aracını screen üzerine ekliyoruz. Sağ tarafta properties alanındaki source bölümünde bulunan Upload File alanı üzerinden bir alarm sesi yüklüyoruz ve kod blokları kısmına geçiyoruz.



Kod Bloğu







Kazanım

- Bu alan öğrencilerin hizalama araçlarını pekiştirmesinin yanı sıra sensörler ve zaman araçlarını tanımasını sağlar.

WEB TARAYICISI UYGULAMASI

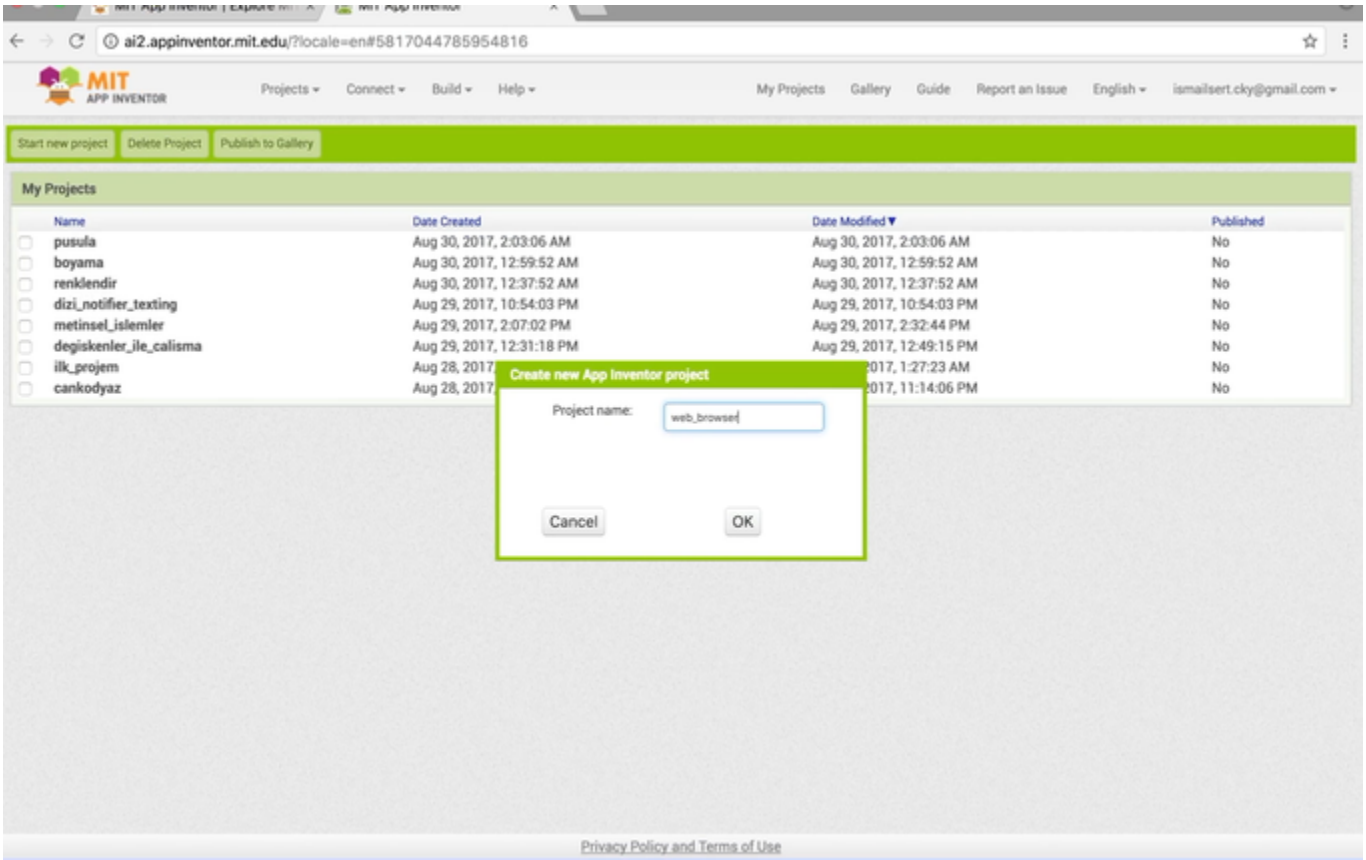
Müfredat Kodu / Ünite / Konu	İCT-104-11.5 / MIT App Inventor Uygulamaları / Web Tarayıcısı Uygulaması
Hedeflenen Süre	90 Dakika

🚀 Web Tarayıcısı Uygulaması

Bu bölümde basit bir web tarayıcısı oluşturmayı hedefliyoruz.

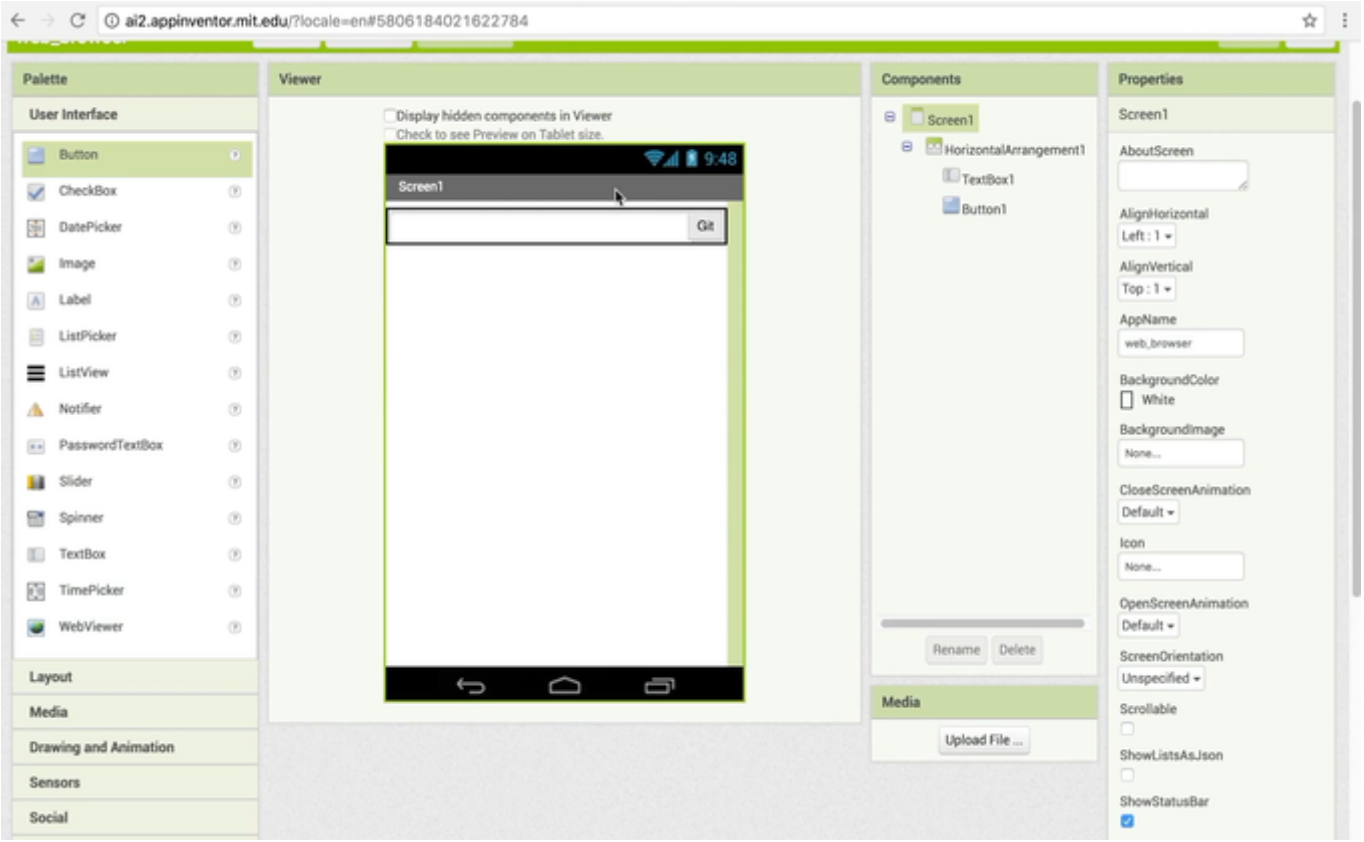
💻 Uygulama

1. Adım : Yeni bir proje oluşturuyoruz.



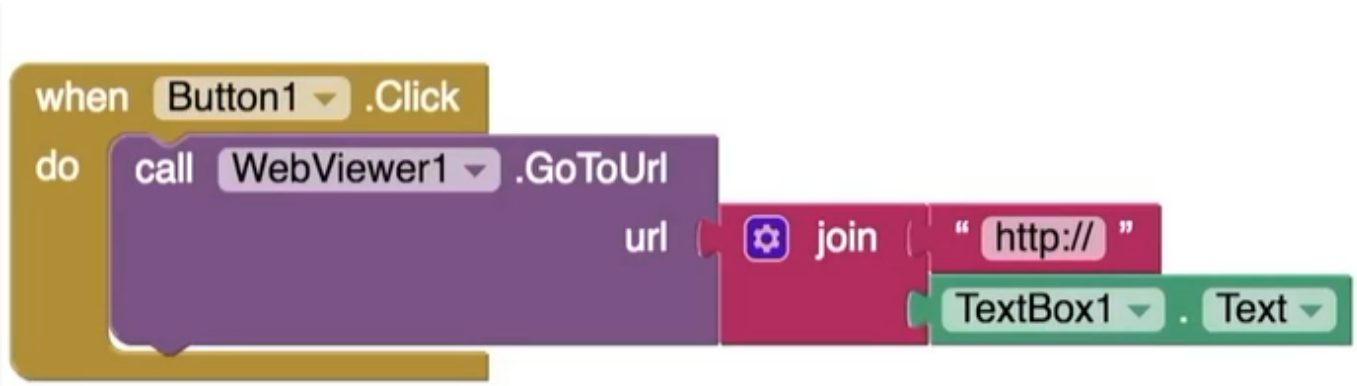
2. Adım : Layout başlığı altındaki HorizontalArrangement aracını sürükleyip screen üzerine bırakıyoruz.

- HorizontalArrangement aracının içerisine bir textbox ve buton ekliyoruz.



3. Adım : Son olarak WebViewer aracını screen üzerine yerleştiriyoruz.

Kod Bloğu



Kazanım

- Bu alan öğrencilerin kod blokları ve webViewer aracı ile birlikte web tarayıcısı sistemini tanımasını sağlar.

QR KOD OKUYUCU UYGULAMASI

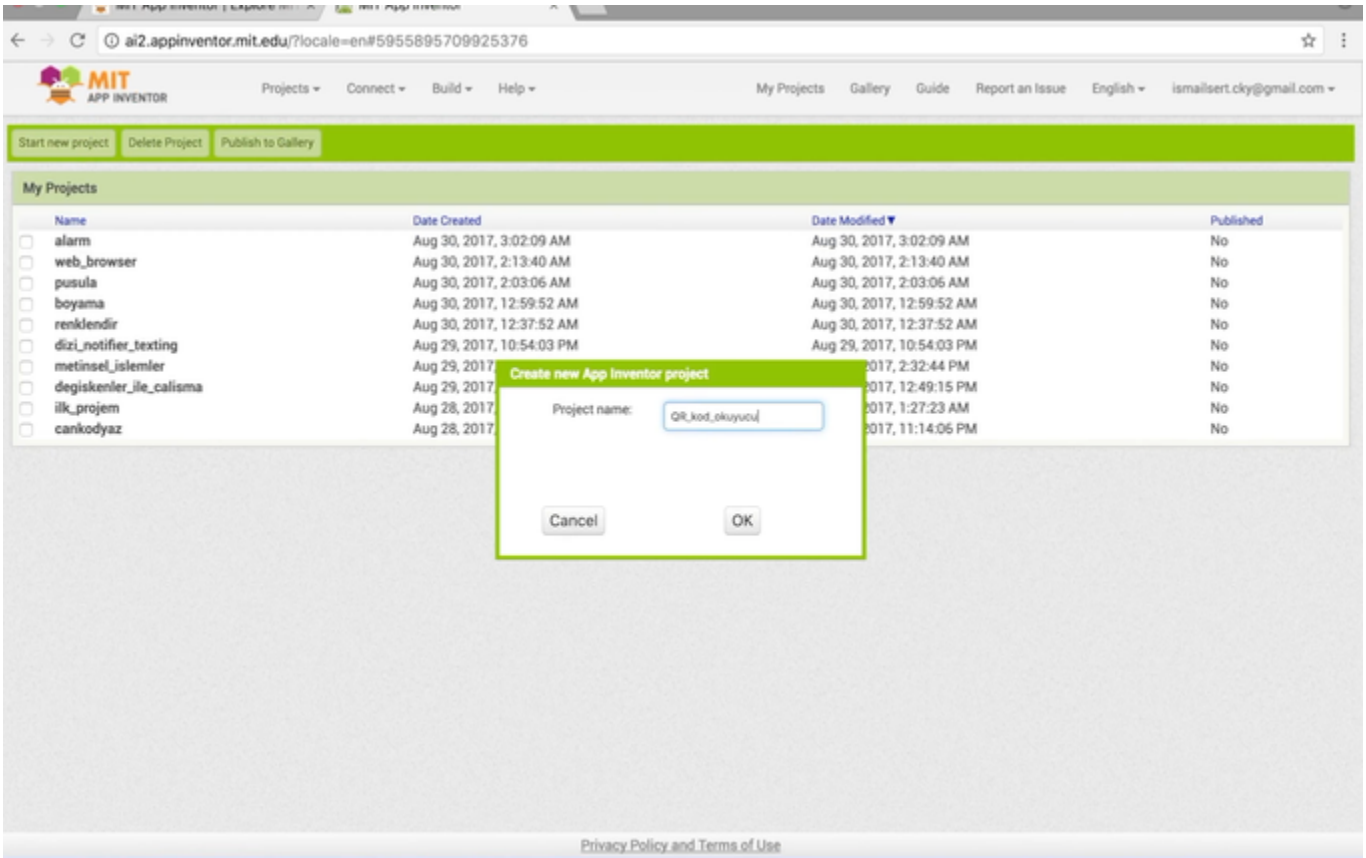
Müfredat Kodu / Ünite / Konu	İCT-104-11.6 / MIT App Inventor Uygulamaları / Qr Kod Okuyucu Uygulaması
Hedeflenen Süre	90 Dakika

🚀 Qr Kod Okuyucu Uygulaması

Bu bölümde bir Qr Kod Okuyucu Uygulaması yapmayı hedefliyoruz.

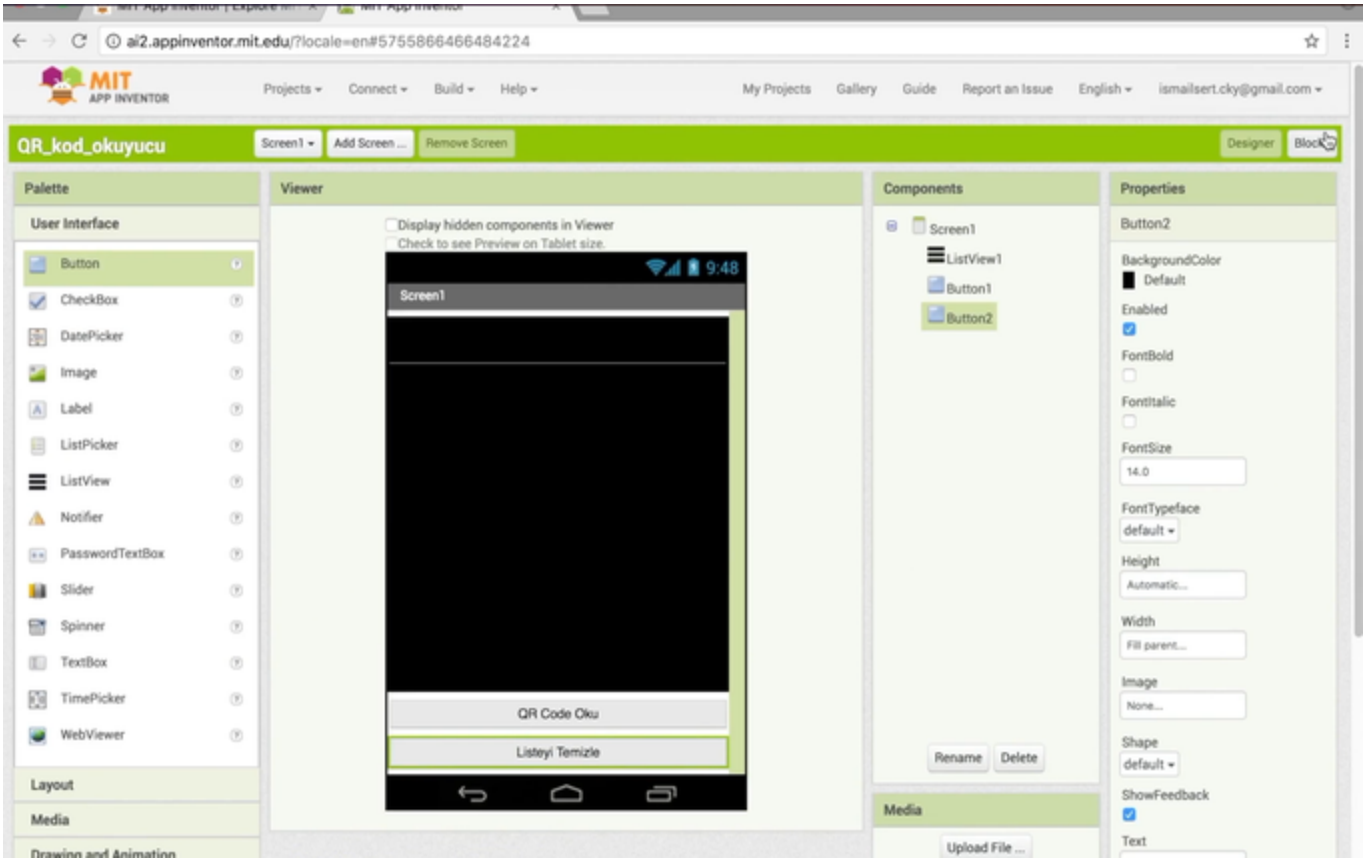
💻 Uygulama

1. Adım : İlk olarak yeni bir proje oluşturuyoruz.



2. Adım : Görsel kısmını oluşturmaya başlıyoruz.

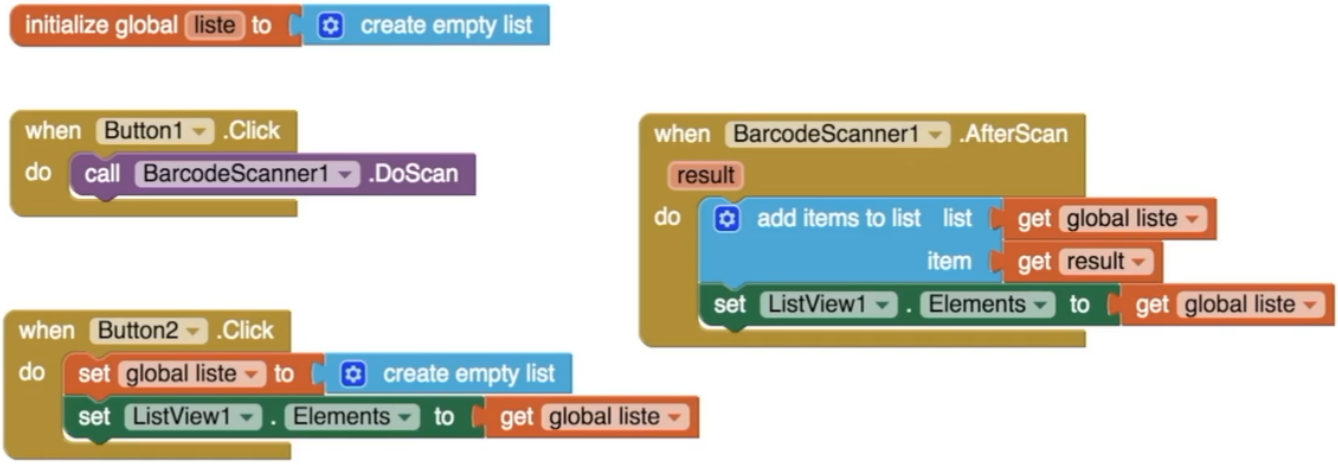
- Screen üzerine ilk önce listView ekleyip butonlarımızı oluşturuyoruz.



3. Adım : Sensors başlığı altındaki BarcodeScanner ayarını screen üzerine ekliyoruz.

> ⚠ BarcodeScanneri ekledikten sonra properties altındaki UseExternalScanner seçeneğini, seçili durumdan kaldırmalıyız. Bu seçenek, telefonunuzda başka QR kod okuyucusu varsa onu aktif hale getirmektedir.

Kod Bloğu



Kazanım

- Bu alan öğrencilerin kod blokları ve QR kod ile birlikte barkod sistemini tanıması sağlar.

TELEFONU SALLAYARAK EKRAN TEMIZLEME UYGULAMASI

Müfredat Kodu / Ünite / Konu	İCT-104-11.7 / MIT App Inventor Uygulamaları / Telefonu Sallayarak Ekran Temizleme Uygulaması
Hedeflenen Süre	90 Dakika

🚀 Telefonu Sallayarak Ekran Temizleme Uygulaması

Bu bölümde Drawing and Animation panelinden yararlanarak bir uygulama yapmayı hedefliyoruz.

✍️ Gerekli Malzemeler

> *Android telefon,*
>
> *MIT AI2 telefon*
>
> *Google hesabı,*
>
> *Usb şarj kablosu*

💻 Uygulama

1. Adım : http://ai2.appinventor.mit.edu/ adresine girerek App Inventor'u açalım ve buradan New Project diyerek yeni proje başlatalım.

2. Adım : Uygulamayı başlattığınızda telefonunuzun ekranında çıkacak olan görüntünün oluşturulması adımına geçelim.

Bu tasarımı oluşturmak için gerekli olan App Inventor araçları şunlardır ;
1 x Canvas,
1 x Accelerometer,

3. Adım : Drawing and Animation bölümünden Canvas ekleyelim ve özelliklerini Width: Fill Parent, Height: Fill Parent olacak şekilde değiştirelim.

4.Adım : Bu işlemleri tamamladıktan sonra uygulamanın telefonda nasıl göründüğüne bakmak için; connect menüsünden "AI Companion" seçelim.

5.Adım : Bilgisayarınızın ekranına bir QR kod ve ayrıca giriş kodu görseli gelecektir.

→ Bu "QR Code" veya "Code" değerlerini Android telefonunuzdaki AppInventor2 uygulamasını çalıştırdığınızda ekrana gelecek olan alanları kullanarak giriş yapabilirsiniz.

6. Adım : Telefonunuzun ekranına gelen görüntüde "connect with code" ve "scan QR code" şeklinde 2 buton olacaktır. Buradaki seçeneklerden birini seçerek ilerleyelim.

→ Seçim yapıldıktan sonra hazırlamış olduğumuz tasarım Android telefonun ekranında görüntülenmiş olacak.

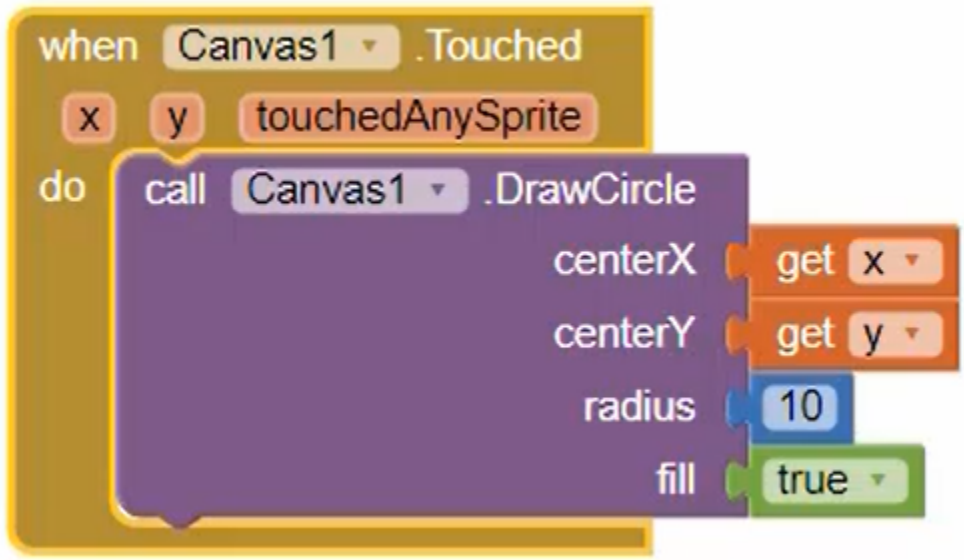
⌨️ Kod Bloğu

1. Adım: Adım : Blocks bölümünden Canvas1 aracını seçelim ve açılan ekrandan "when Canvas1.Touched.. do" bloğunu alalım.

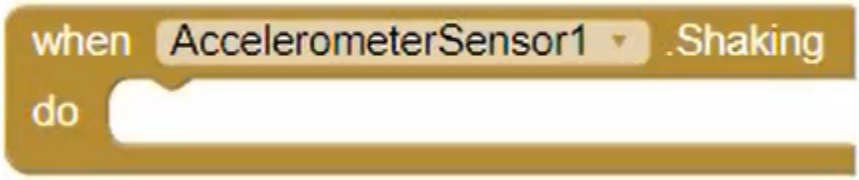
2.Adım : Blocks bölümünden Canvas1 aracını seçelim ve açılan ekrandan "call Canvas1 .DrawCircle" bloğunu alalım.

→ .DrawCircle bloğunun centerX ve centerY değerleri için .Touched bloğunda yer alan x ve y işaretlerinin üzerine mouse ile gidildiğinde açılan pencereden uygun blokları yerlerine yerleştirelim.

Radius değeri için ise Math menüsünden “0” bloğunu alalım ve buraya bir sayı yazalım. (Örneğin; 10)



3.Adım: Blocks bölümünden Accelerometer aracına tıklayalım ve “AccelerometerSensor1. Shaking do” bloğunu alalım.



4.Adım: Bu adımda telefonu salladığımızda ekranın temizlenmesi için gerekli olan kod bloğunu Canvas1 aracına tıklayarak alalım.

5.Adım: Bu adımda blokları birleştirelim ve uygulamayı telefonda çalıştıralım.

Kazanım

- Bu bölüm öğrencilere; problem çözme, öğrendiklerini paylaşma, eleştirel ve analitik düşünme becerisini kazandırır, android telefon sistemi ve QR kod sistemi ile tanıştırır.

FOTOĞRAF ÇEKME UYGULAMASI

Müfredat Kodu / Ünite / Konu	İCT-104-11.8 / MIT App Inventor Uygulamaları / Fotoğraf Çekme Uygulaması
Hedeflenen Süre	90 Dakika

🚀 Fotoğraf Çekme Uygulaması

Bu bölümde Fotoğraf Çekme Uygulamasını yapmayı hedefliyoruz.

💻 Uygulama

1. Adım : İlk olarak sisteme giriş yapıyoruz.

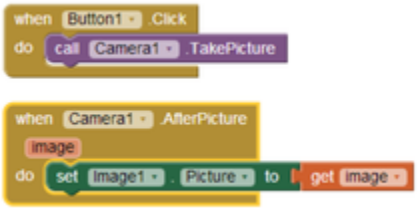
2. Adım : Projemize isim veriyoruz.

→ Karşımıza çıkan kodlama ekranı üzerinden telefonun formatını değiştirerek ekranda nasıl göründüğünü görebilirsiniz.

3.Adım : Ekranımıza “buton”, “Image” ve media bölümünden “Camera” bileşenleri ekliyoruz. “Image1” in “Width” özelliğini “FillParent” yapıyoruz. Butonumuzun “Text” özelliğini “Fotograf Çek” olarak değiştiriyoruz.

4.Adım : Blok bölümüne geçiyoruz ve butonumuzun “click” olayına “Camera1” bileşeninin fotograf çekmeye başlamasını tetikliyoruz. “When.. Camera1..AfterPicture” olayı ile de “Camera1” bileşeninin “getimage” değerini “image1” bileşeninin “picture” değişkenine kilitliyoruz.

Burada dikkat edilmesi gereken en önemli nokta “When..Camera1..AfterPicture” olay bloğunda, köşede “image” özelliği vardı; bu özelliğin üzerinde fareyi bekletirseniz “getimage” (cihazın kamera uygulamasının çektiği fotoğraf yolunu verir) özelliğini gösteren baloncuk görüntülenir, bu baloncuğun içinden “getimage” bloğunu şeçip “image1” bileşeninin “picture” değişkenine kilitliyoruz.



5. Adım : Projemiz tamamlandı. Tamamlanan projeyi denemek için programın bize sunduğu QR kodu ile denememizi sağlayan basit bir yöntem var. Öncelikle üstteki sekmelerden **Build** sekmesine geldiğimizde site uygulamamızı derleyip bize bir QR kod veriyor.

6.Adım : QR kodu okutturmak için öncelikle, https://play.google.com/store/apps/details?id=edu.mit.appinventor.aicompanion3 adresinden MIT AI2 Companion isimli uygulamayı indirmemiz gerekiyor. Bu kodu telefona indirdiğimiz uygulamamıza okuttuğumuzda yaptığımız uygulamayı indirip telefona kuruyor ve işlem tamamlanıyor.

7. Adım : İndirme işleminde farklı kaynaktan dosya indirimine izin vermeniz gerekiyor. Uygulama telefonunuza indiğinde App Inventor simgesiyle görünür. Tıkladığınızda açılır.

Burada dikkat edilecek noktaya gelirsek cihazının GPS özelliğini açmanız gerekir. Eğer güncellemeyi daha sık aralıklarla almak isterseniz. “LocationSensor1” bileşeninin “set..LocationSensor1..Timeinterval” bloğunundaki özelliğine milisaniye (Örn:1 saniye için 1000 değeri girilecek) olarak değer atayabilirsiniz. Örneğin bu bloğu “Button1” in “Click” olayı içine yerleştirin.

Kazanım

- Bu bölümde öğrenciler media menüsüne ait araçları kullanarak öğrendiklerini pekiştirir.

SES TANIMA UYGULAMASI

Müfredat Kodu / Ünite / Konu	İCT-104-11.9 / MIT App Inventor Uygulamaları / Ses Tanıma Uygulaması
Hedeflenen Süre	180 Dakika

🚀 Ses Tanıma Uygulaması

Bu bölümde Ses Tanıma Uygulamasını yapmayı hedefliyoruz.

💻 Uygulama

1. Adım : İlk olarak sisteme giriş yapıyoruz.

2. Adım : Projemize isim veriyoruz.

→ Karşımıza çıkan kodlama ekranı üzerinden telefonun formatını değiştirerek ekranda nasıl göründüğünü görebilirsiniz.

3.Adım : Bu uygulama için ekranımıza resimde görüldüğü üzere Bir adet “SpeechRegognizer”, bir adet “Button”, tanınan seslerin anlık yazılacağı bir “TextBox” bileşeni, “TextBox” bileşeninin yanına bir etiket ve tanınan konuşmaların sırasıyla eklendiği bir “ListView” bileşeni ekliyoruz. Bunlara aşağıdaki gibi yapacağı görevlere göre isimler ekliyoruz.

4.Adım : Properties kısmından ise, Background Color ile buton rengini değiştirebilir, FontSize ile buton üstünde yazan yazı boyutu değiştirilebilir, veya Height/Width değerleri ile oynayarak buton boyutunu değiştirebiliriz. Burada değiştirmek istediklerimizi değiştirdikten sonra sağ üstteki sekmeden Block kısmına geçebiliriz.

5. Adım : İlk olarak projemize bir değişken oluşturuyoruz. Bunu Variables bölümünden initialize global kısmından seçiyoruz. İsmini elaman_listesi yapıyoruz. Daha sonra onun devamına boş liste oluştur bloğunu eklememiz gerekiyor. Bunun için list bölümünden create empty list bloğunu ekliyoruz.

Block kısmında “Button1” ile ses tanıma işlemi başlatılıyor. “When..SpeechRegognizer1..AfterGettingText” olayı ile ses tanıma işlemi bittikten sonra bu olay bloğunun köşesinde bulunan kırmızı renkli “result” (Tanınmış sesin metin hali) yazısının üzerine fare ile durduğumuz zaman çıkan “get..result” bloğunu “TextBox1” bileşeninin “Text” değişkenine kilitliyoruz.

“ListView1” bileşeni ile yaptığımız ise, “liste_eleman” isminde bir değişken oluşturup bu değişkene “SpeechRegognizer1” bileşenin “result” bloğunu add..item..list” bloğu yardımı ile “ListView1” bileşeninin sonuna bir eleman daha ekliyoruz. Böylece sesin tanınmasını ve text box'ta listelenmesini sağlayacağız. Ayrıca tanınan bu sesleri ise listview kısmında listeyeceğiz.

6.Adım : Projemiz tamamlandı. Tamamlanan projeyi denemek için programın bize sunduğu QR kodu ile denememizi sağlayan basit bir yöntem var. Öncelikle üstteki sekmelerden Build sekmesine geldiğimizde site uygulamamızı derleyip bize bir QR kod veriyor.

---

7. Adım : QR kodu okutmak için öncelikle, https://play.google.com/store/apps/details?id=edu.mit.appinventor.aicompanion3 adresinden MIT AI2 Companion isimli uygulamayı indirmemiz gerekiyor. Bu kodu telefona indirdiğimiz uygulamamıza okuttuğumuzda yaptığımız uygulamayı indirip telefona kuruyor ve işlem tamamlanıyor.

---

8. Adım : İndirme işleminde farklı kaynaktan dosya indirimine izin vermeniz gerekiyor. Uygulama telefonunuza indiğinde App Inventor simgesiyle görünür.

Kazanım

- Bu bölümde öğrenciler media menüsüne ait araçları kullanarak öğrendiklerini pekiştirir.

LAYOUT BİLEŞENLERİ UYGULAMASI

Müfredat Kodu / Ünite / Konu	İCT-104-11.10 / MIT App Inventor Uygulamaları / Layout Bileşenleri Uygulaması
Hedeflenen Süre	180 Dakika

🚀 Layout Bileşenleri Uygulaması

Bu bölümde Layout Bileşenleri Uygulamasını yapmayı hedefliyoruz.

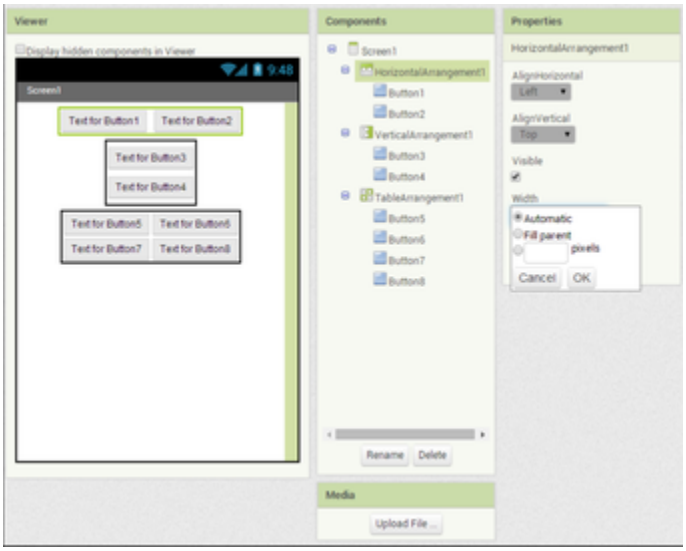
💻 Layout Bileşenleri ile Örnek Uygulamalar

Yeni bir proje oluşturuyoruz. “Components” Alanı içinde “Screen1” üzerine tıklıyoruz, ve “Properties” alanı içindeki “AlignHorizontal” (Yatay Hizala) özelliğini “Center” (Ortala) olarak değiştiriyoruz. Böylece ekrana yerleştireceğimiz bileşenler ortalanacak görünümü daha iyi algılanacak.

Sırasıyla “Layout” bölümünden “HorizontalArrangement”(Yatay Yerleşim Düzenleme), “VerticalArrangement” (Dikey Yerleşim Düzenleme) ve “TableArrangement”(Tabloda Yerleşim Düzenleme) bileşenlerini ekrana ekliyoruz. Bu bileşenlerin genel görünümü kare şeklinde kutucuklardır. Yapı olarak ise sadece “TableArrangement” içine bileşen yerleştirirken fark edebileceğimiz; seçili olarak 2 satır ve 2 sütun şeklinde hücreler vardır. Bu bileşenin hücre sayısını “Properties” alanından ayarlayabilirsiniz.

Bileşenlerimizin içine “User Interface” bölümünden “Button” bileşenini ekliyoruz;

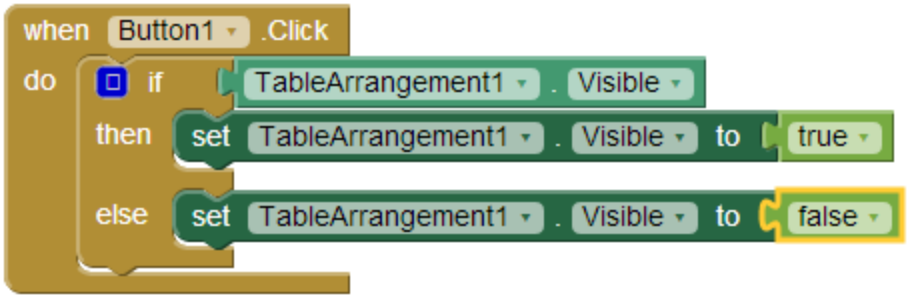
- “HorizontalArrangement” (Yatay Yerleşim Düzenleme) içine yerleştirilen butonlar yan yana yerleşir.
- “VerticalArrangement” (Dikey Yerleşim Düzenleme) içine yerleştirilen butonlar ise üst üste yerleşir. Daha doğru bir anlatımla dikey sıralanır.
- “TableArrangement” (Tabloda Yerleşim Düzenleme) içine yerleştirilen butonların yerleştirme anında görünen hücrelere yerleştiğini fark edeceksiniz.



Tüm bu yerleşim bileşenlerinin “Properties” alanında “Width” (Genişlik) ve “Height” (Yükseklik) özelliklerinin olduğunu göreceksiniz. Bu özellikler:

1. Automatic: Bileşenin içine yerleştirilen bileşenlerin boyutu oranında genişlik veya yükseklik otomatik ayarlanır.
2. FillParent: Bileşen bir üst düzenleme bileşeninin içini doldurur. İç içe birçok düzenleme bileşeni olabileceği gibi bu örnekteki gibi sadece “Screen1” içinde de olabilir. Bu durumda üst düzenleme bileşeni “Screen1” olarak kabul edilir ve o bileşenin içi doldurulur.
3. Pixels: Boyutu piksel olarak ayarlamak için buraya istediğimiz değeri girebiliriz.

Bu bileşenlerin “Visible” (Görünürlük) özelliğini kullanarak; programın çalışması esnasında istediğiniz bileşenleri olduğu gibi düzenleme, bileşenin içine yerleştirdiğiniz tüm bileşenlerin görünür olması veya gizlenmesini sağlayabilirsiniz. Aşağıdaki resimde ilgili bloklar ile “TableArrangement1” bileşenini gösterip gizleyen bir blok dizisi vardır. Bu bloklar “Button1” bileşeninin “Click” olayına kilitlenmiştir.



Kazanım

- Bu bölümde öğrenciler layout menüsüne ait araçları kullanarak öğrendiklerini pekiştirir.

PYTHON EĞITIM MÜFREDATI

PYTHON EĞİTİM MÜFREDATI														
EĞİTİM İÇERİĞİ / HAFTA	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.
Python: Tanıtım ve Programlamaya Giriş														
Python: print() Fonksiyonu														
Python: Yorum Satırları														
Python: Kaçış Dizileri														
Python: Veri Türleri														
Python: Değişkenler														
Python: Karakter Dizileri														
Python: Kullanıcıdan Bilgi Almak														
Python: Operatörler														
Python: Kontrol Yapıları														
Python: Hata Yakalama														
Python: Fonksiyonlar														
Python: Nesne Tabanlı Programlama														

PYTHON EĞITIMI

üfredat Kodu	İCT-301-1
Ders	PYTHON
Ders İçeriği	• PYTHON: Tanıtım • PYTHON: Kurulum • PYTHON: print() Fonksiyonu • PYTHON: Yorum Satırları • PYTHON: Kaçış Dizileri • PYTHON: String (Karakter) • PYTHON: Integer (Sayılar) • PYTHON: Sayı Metotları • PYTHON: Listeler • PYTHON: Liste Metotları • PYTHON: Kümeler (Set) • PYTHON: Değişkenler • PYTHON: Karakter Dizileri • PYTHON: Kullanıcıdan Bilgi Almak • PYTHON: Operatörler • PYTHON: if-else • PYTHON: While Döngüsü • PYTHON: For Döngüsü • PYTHON: Hata Yakalama • PYTHON: Fonksiyonlar • PYTHON: Gömülü Fonksiyonlar • PYTHON: Nesne Tabanlı Programlama (Bölüm 1) • PYTHON: Nesne Tabanlı Programlama (Bölüm 2)
Seviye	İCT-S2: ORTA
Ön Koşullar	Teknoloji ve yazılımla ilgili olmak.
Amaç	Öğrencilere web sitesi yazılımı oluşturmak, görevleri otomatikleştirme ve veri analizi yapmayı öğretmek amaçlanmaktadır.
Hedef	Bu bölümde yazılım geliştirme süreci konusunda bilgi edinmek, bu süreçte gerekli olan araçları tanıtmak hedeflemektedir.
Gereksinimler	Bilgisayar ve Ortamda İnternet Ağı
Kazanım	Öğrenciler; • Python programlama dilini kullanarak algoritma ve uygulama geliştirmeyi öğrenir, • Örnek Python programları ile hangi problemlere çözüm üretebileceğini kavrar.
Referanslar & Dokümantasyon	• Python Programlama Dili — Yazbel Python Belgeleri • https://www.kodkampusu.com/python-kacis-karakterleri/
Durum	BEKLIYOR / UYGULANIYOR / TAMAMLANDI / DEĞERLENDIRILDI
Hedeflenen Süre	1260 dakika
Eğitim Günleri	Başlama: .../.../.... Bitiş: .../.../....

PYTHON: TANITIM

Müfredat Kodu / Ünite / Konu	İCT-301-1.1.1 / Tanıtım ve Programlamaya Giriş / Tanıtım
Hedeflenen Süre	45 dakika

🚀 Python Nedir?

Tahmin edebileceğiniz gibi Python (C, C++, Perl, Ruby ve benzerleri gibi) bir programlama dilidir ve tıpkı öteki programlama dilleri gibi, önünüzde duran kara kutuya, yani bilgisayara hükmetmenizi sağlar.

Bu programlama dili Guido Van Rossum adlı Hollandalı bir programcı tarafından 90'lı yılların başında geliştirilmeye başlanmıştır. Çoğu insan, isminin Python olmasına aldanarak, bu programlama dilinin, adını piton yılanından aldığını düşünür. Ancak zannedildiğinin aksine bu programlama dilinin adı piton yılanından gelmez. Guido Van Rossum bu programlama dilini, *The Monty Python* adlı bir İngiliz komedi grubunun, *Monty Python's Flying Circus* adlı gösterisinden esinlenerek adlandırmıştır. Ancak her ne kadar gerçek böyle olsa da, Python programlama dilinin pek çok yerde bir yılan figürü ile temsil edilmesi neredeyse bir gelenek halini almıştır.

Dediğimiz gibi, Python bir programlama dilidir. Üstelik pek çok dile kıyasla öğrenmesi kolay bir programlama dilidir. Bu yüzden, eğer daha önce hiç programlama deneyiminiz olmamışsa, programlama maceranıza Python'la başlamayı tercih edebilirsiniz.

💻 Platform Desteği

Python programlama dili pek çok farklı işletim sistemi ve platform üzerinde çalışabilir. GNU/Linux, Windows, Mac OS X, AS/400, BeOS, MorphOS, MS-DOS, OS/2, OS/390, z/OS, RiscOS, S60, Solaris, VMS, Windows CE, HP-UX, iOS ve Android gibi, belki adını dahi duymadığınız pek çok ortamda Python uygulamaları geliştirebilirsiniz. Ayrıca herhangi bir ortamda yazdığınız bir Python programı, üzerinde hiçbir değişiklik yapılmadan veya ufak değişikliklerle başka ortamlarda da çalıştırılabilir.

Biz bu belgelerde Python programlama dilini GNU/Linux ve Microsoft Windows işletim sistemi üzerinden anlatacağız. Ancak sıkı sıkıya bel bağlayacağımız özel bir GNU/Linux dağıtımı veya Windows sürümü yok. Bu yüzden, hangi GNU/Linux dağıtımını veya hangi Windows sürümünü kullanıyor olursanız olun, buradaki bilgiler yardımıyla Python programlama dilini öğrenebilir, öğrendiklerinizi kendi işletim sisteminize uyarlayabilirsiniz.

✅ Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Python programlama dilinin tanıtılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere Python programlama dilinin tanıtılması ve kullanım amaçlarının öğretilmesi amaçlanmaktadır.
2	☑ Python programlama dili keşfetmek.	☐ Eğitmen ☑ Öğrenciler	Öğrencilerin bu derste Python programlama dilini keşfetmesi amaçlanmaktadır.

PYTHON: KURULUM

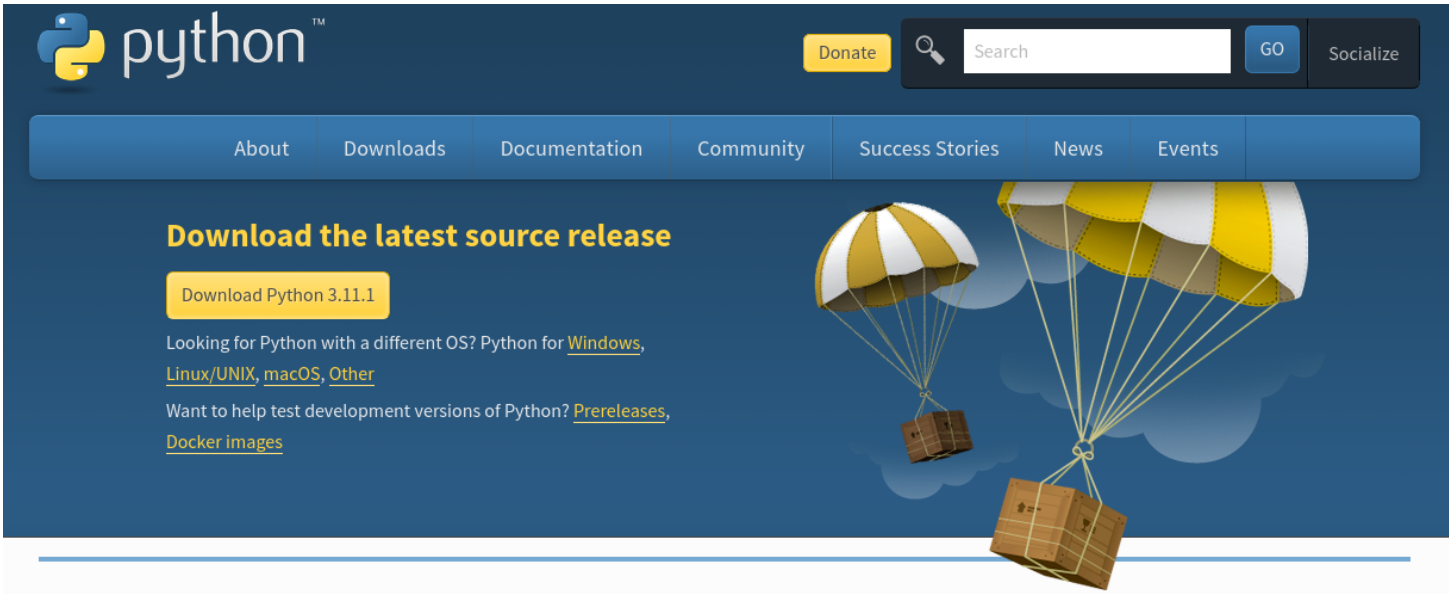
Müfredat Kodu / Ünite / Konu	İCT-301-1.1.2 / Tanıtım ve Programlamaya Giriş / Kurulum
Hedeflenen Süre	45 dakika

🚀 Python Nasıl Kurulur?

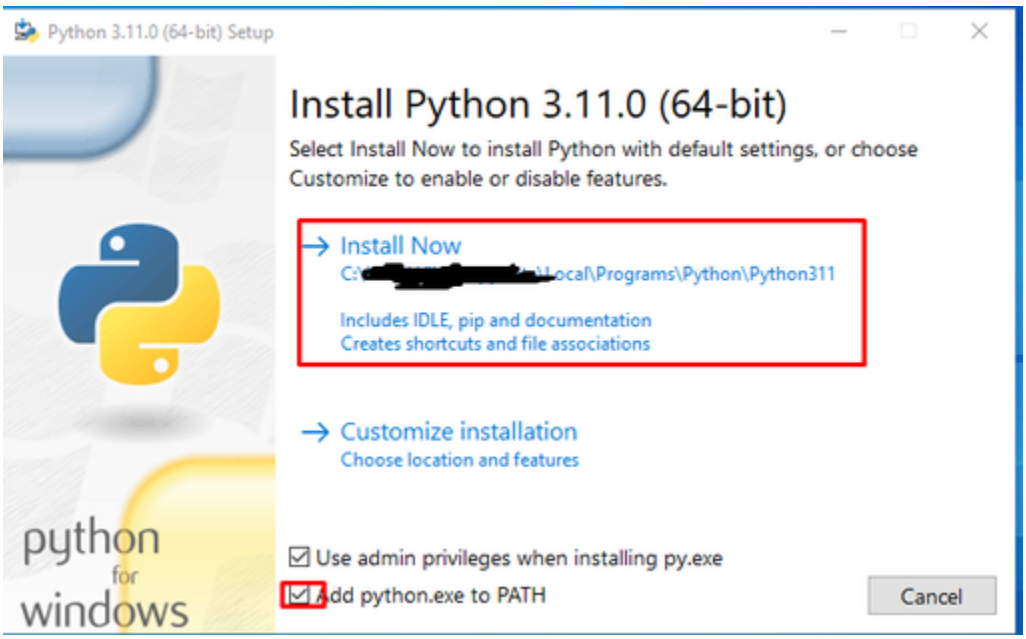
Python ile program yazabilmemiz için bu programlama dilinin bilgisayarımızda kurulu olması gerekiyor. Bu programlama dilini kurmanızın gerekip gerekmediği, kullandığınız işletim sistemine bağlıdır. Biz burada Windows kullanıcılarının durumunu inceleyeceğiz.

Windows sürümlerinin hiçbirinde Python kurulu olarak gelmez. O yüzden Windows kullanıcıları, Python'ı sitesinden indirip kuracak.

Bunun için öncelikle https://www.python.org/downloads/ adresine gidiyoruz. Bu adrese gittiğinizde, üzerinde 'Download Python 3.11.0' yazan bir düğme göreceksiniz.



Kurulum dosyasına çift tıkladığınızda karşınıza ilk gelen ekranda, pencerenin alt tarafında şu kutucukları göreceksiniz:



✅ Install launcher for all users (recommended)

✅ Add Python 3.11 to PATH

Burada ilk kutucuk zaten seçilidir. Bunu bu şekilde bırakabilirsiniz. İkinci kutucuk ise Python'ı yola eklememizi, böylece yalnızca python komutu vererek Python'ı başlatabilmemizi sağlayacak. O yüzden oradaki ikinci kutucuğu da işaretliyoruz.

Aynı pencerenin üst tarafında ise şu seçenekleri göreceksiniz:

→ Install Now

→ Customize Installation

Burada 'Install Now' yazan kısma tıklayarak kurulumu başlatıyoruz.

Eğer Python'ın bilgisayarda nereye kurulacağını ve başka birtakım kurulum özelliklerini değiştirmek istiyorsanız 'Customize Installation' yazılı kısma tıklayabilirsiniz. Ben bu dokümanda sizin 'Install Now' yazan kısma tıklayarak kurulum yaptığınızı varsayacağım.

💻 Python Kurulum ve Çalışma Dizini

Python programlama dilini, kullandığımız işletim sistemine nasıl kurabileceğimizi bilmek kadar önemli bir konu da Python'ı hangi dizine kurduğumuzu bilmektir. Zira programcılık maceramız boyunca karşılaşacağımız bazı sorunlar, Python'ın kurulu olduğu dizine gitmemizi gerektirecek, üstelik kendi yazdığımız bazı programlarda da Python'ın kurulu olduğu dizinde çeşitli işlemler yapmak ihtiyacı duyacağız. Ayrıca bazı durumlarda, o anda çalışan Python sürümünün hangi konumdan çalıştığını tespit etmemiz de gerekebilir.

İşte bu sebeplerden, Python'ın hangi dizine kurulduğunu mutlaka biliyor olmamız lazım.

Windows'ta Python programlama dilini aynen bu kitapta gösterdiğimiz şekilde kurduysanız, Python %LOCALAPPDATA%\Programs\Python dizini içine kurulacaktır. Ancak eğer kurulum penceresinde 'Customize Installation' düğmesine basarak kurulumu özelleştirdiyseniz ve 'Install for all users' seçeneğini işaretlediyseniz Python %PROGRAMFILES% veya %PROGRAMFILES(x86) adlı çevre değişkenlerinin işaret ettiği dizin içine kurulacaktır.

✅ Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Kurulumun yapılması. ☑ Öğrencilere editörün tanıtılması.	☑ Eğitmen ☐ Öğrenciler	Eğitim ve uygulamaların yapılacağı bilgisayarlara Python editörünün kurulması. Öğrencilere editör arayüzün tanıtılması.
2	☑ Uygulamaya erişim sağlamak. ☑ Arayüzün incelenmesi.	☐ Eğitmen ☑ Öğrenciler	Eğitmenler tarafından kurulumu sağlanan editöre erişim sağlamak. Öğrenciler tarafından editör arayüzünün incelenmesi.

PYTHON: PRINT() FONKSIYONU

Müfredat Kodu / Ünite / Konu	İCT-301-1.2 / Python / print() Fonksiyonu
Hedeflenen Süre	30 dakika

🚀 print() Fonksiyonu

print() fonksiyonunun görevi ekrana çıktı vermemizi sağlamaktır. Hemen bununla ilgili bir örnek verelim:

```
>>> print("Python programlama dili")

Python programlama dili
```

Bildiğiniz gibi burada gördüğümüz “Python programlama dili” bir karakter dizisidir. İşte print() fonksiyonunun görevi bu karakter dizisini ekrana çıktı olarak vermektir.

Daha önce de dediğimiz gibi, etkileşimli kabuk bir test ortamı olması açısından rahat bir ortamdır. Bu sebeple bu ortamda ekrana çıktı verebilmek için print() fonksiyonunu kullanmak zorunda değilsiniz. Yani başında print() olsa da olmasa da etkileşimli kabuk ekrana yazdırmak istediğiniz şeyi yazdırır. Ama iyi bir alışkanlık olması açısından, ekrana herhangi bir şey yazdıracağınızda ben size print() fonksiyonunu kullanmanızı tavsiye ederim.

print() son derece güçlü bir fonksiyondur. Gelin isterseniz bu güçlü ve faydalı fonksiyonu derin derin incelemeye koyulalım.

🖥️ Nasıl Kullanılır?

Yukarıda verdiğimiz örnekte ilk gözümüze çarpan şey, karakter dizisini print() fonksiyonunun parantezleri içine yazmış olmamızdır. Biz bir fonksiyonun parantezleri içinde belirtilen öğelere ‘parametre’ dendiğini geçen bölümde öğrenmiştik. Tıpkı öğrendiğimiz öteki fonksiyonlar gibi, pr int() fonksiyonu da birtakım parametreler alır.

Elbette, eğer istersek burada doğrudan “Python programlama dili” adlı karakter dizisini kullanmak yerine, önce bu karakter dizisini bir değişkene atayıp, sonra da print() fonksiyonunun parantezleri içinde bu değişkeni kullanabiliriz. Yani:

```
>>> dil = "Python programlama dili"
>>> print(dil)

Python programlama dili
```

👷 Bir Fonksiyon Olarak print()

print() ifadesinin bir fonksiyon olduğunu söylemiştik hatırlarsanız. Dediğimiz gibi, fonksiyonlarla ilgili ayrıntılı açıklamaları ilerleyen derslerde vereceğiz. Ancak şimdi dilerseniz bundan sonra anlatacaklarımızı daha iyi kavrayabilmemiz için, fonksiyonlar hakkında bilmemiz gereken bazı temel şeyleri öğrenmeye çalışalım.

Gördüğünüz gibi, print() fonksiyonunu şöyle kullanıyoruz:

```
>>> print("Aramak istediğiniz kelimeyi yazın: ")
```

Burada print() bir fonksiyon, “Aramak istediğiniz kelimeyi yazın:” adlı karakter dizisi ise bu fonksiyonun parametresidir. Daha önce len() adlı başka bir fonksiyon daha öğrenmiştik hatırlarsanız. Onu da şöyle kullanıyorduk:

```
>>> len("elma")
```

Burada da len() bir fonksiyon, “elma” adlı karakter dizisi ise bu fonksiyonun parametresidir. Aslında biçim olarak print() ve len() fonksiyonlarının birbirinden hiçbir farkı olmadığını görüyorsunuz.

■ print() Fonksiyonunun Parametreleri

Şimdiye kadar verdiğimiz örneklerde belki çok da belli olmuyordur, ama aslında print() fonksiyonu son derece güçlü bir araçtır. İşte şimdi biz bu fonksiyonun gücünü gözler önüne seren özelliklerini incelemeye başlayacağız. Bu bölümü dikkatle takip etmeniz, ilerde yapacağımız çalışmaları daha rahat anlayabilmeniz açısından büyük önem taşır.

▲ *sep Parametresi*

print() fonksiyonu ile ilgili olarak yukarıda verdiğimiz örnekleri incelediğimizde, bu fonksiyonun kendine özgü bir davranış şekli olduğunu görüyoruz. Mesela bir önceki bölümde verdiğimiz şu örneğe bakalım:

```
>>> print('Arda', 'Kahveci')
Arda Kahveci
```

Burada print() fonksiyonunu iki farklı parametre ile birlikte kullandık. Bu fonksiyon, kendisine verdiğimiz bu parametreleri belli bir düzene göre birbiriyle birleştirdi. Bu düzen gereğince print(), kendisine verilen parametreleri birleştirirken, parametreler arasına bir boşluk yerleştiriyor. Bunu daha net görmek için şöyle bir örnek daha verelim:

```
>>> print("Python", "PHP", "C++", "C", "Erlang")
Python PHP C++ C Erlang
```

▲ *end Parametresi*

print() fonksiyonunun end adlı özel bir parametresi daha bulunur. Tıpkı sep parametresi gibi, end parametresi de print() fonksiyonunda görünmese bile her zaman oradadır.

Bildiğiniz gibi, sep parametresi print() fonksiyonuna verilen parametreler birleştirilirken araya hangi karakterin gireceğini belirliyordu. end parametresi ise bu parametrelerin sonuna neyin geleceğini belirler.

print() fonksiyonu ön tanımlı olarak, parametrelerin sonuna ‘satır başı karakteri’ ekler. Peki bu satır başı karakteri (veya ‘yeni satır karakteri’) denen şey de ne oluyor?

Dilerseniz bunu bir örnek üzerinde görelim.

Şöyle bir kodumuz olsun:

```
>>> print("Pardus ve Ubuntu birer GNU/Linux dağıtımıdır.")
```

Bu kodu yazıp Enter tuşuna bastığımız anda print() fonksiyonu iki farklı işlem gerçekleştirir:

1. Öncelikle karakter dizisini ekrana yazdırır.
2. Ardından bir alt satıra geçip bize >>> işaretini gösterir.

İşte bu ikinci işlem, karakter dizisinin sonunda bir adet satır başı karakteri olmasından, daha doğrusu print() fonksiyonunun, satır başı karakterini karakter dizisinin sonuna eklemesinden kaynaklanır.

Kazanım

- Öğrenci bu alanda print() fonksiyonunu kavrar.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ print() fonksiyonunun anlatılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere print() fonksiyonunun anlatılması ve uygulamalarla pekiştirilmesi, öğrencilerle birlikte editör üzerinde uygulamaların yapılması. amaçlanmaktadır.
2	☑ Uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen eşliğinde editör üzerinde uygulamaların yapılması amaçlanmaktadır.

PYTHON: YORUM SATIRLARI

Müfredat Kodu / Ünite / Konu	İCT-301-1.3 / Python / Yorum Satırları
Hedeflenen Süre	30 dakika

🚀 Yorum İşareti

Python'da yazdığımız kodları başkalarının da anlayabilmesini sağlamak için, programımızın yorumlarla desteklenmesi tavsiye edilir. Elbette programınızı yorumlarla desteklemeseniz de programınız sorunsuz bir şekilde çalışacaktır. Ama programı yorumlarla desteklemek en azından nezaket gereğidir.

Ayrıca işin başka bir boyutu daha var. Sizin yazdığınız kodları nasıl başkaları okurken zorlanıyorsa, kendi yazdığınız kodları okurken siz bile zorlanabilirsiniz. Özellikle uzun süredir ilgilenmediğiniz eski programlarınızı gözden geçirirken böyle bir sorunla karşılaşabilirsiniz. Programın içindeki bir kod parçası, programın ilk yazılışının üzerinden 5-6 ay geçtikten sonra size artık hiçbir şey ifade etmiyor olabilir. Kodlara bakıp, 'Acaba burada ne yapmaya çalışmışım?' diye düşündüğünüz zamanlar da olacaktır. İşte bu tür sıkıntıları ortadan kaldırmak veya en aza indirmek için kodlarımızın arasına açıklayıcı notlar ekleyeceğiz.

Python'da yorumlar # işareti ile gösterilir. Mesela alttaki kod örneğini inceleyelim:

```
isim    = "Baki"
soyisim = "Güzel"
işsis   = "Windows" #işletim sistemi
şehir   = "İstanbul"

#isim, soyisim, işsis ve şehir adlı değişkenleri
#alt alta, düzgün bir şekilde ekrana basıyoruz.
#Uygun yerlerde alt satıra geçebilmek için "\n"
#adlı kaçış dizisini kullanıyoruz.
print("isim          : ", isim,    "\n",
    "soyisim       : ", soyisim, "\n",
    "işletim sistemi: ", işsis,   "\n",
    "şehir         : ", şehir,   "\n",
    sep="") #parametreler arasında boşluk bırakmıyoruz.
```

Burada dikkat edeceğimiz nokta her yorum satırının başına # işaretini koymayı unutmamaktır.

Yazdığımız yorumlar Python'a hiç bir şey ifade etmez. Python bu yorumları tamamen görmezden gelecektir. Bu yorumlar bilgisayardan ziyade kodları okuyan kişi için bir anlam taşır.

Elbette yazdığınız yorumların ne kadar faydalı olacağı, yazdığınız yorumların kalitesine bağlıdır. Dediğimiz gibi, yerli yerinde kullanılmış yorumlar bir programın okunurluğunu artırır, ama her tarafı yorumlarla kaplı bir programı okumak da bazen hiç yorum girilmemiş bir programı okumaktan daha zor olabilir! Dolayısıyla Python'da kodlarımıza yorum eklerken önemli olan şey, kaş yapmaya çalışırken göz çıkarmamaktır. Yani yorumlarımızı, bir kodun okunurluğunu artırmaya çalışırken daha da bozmayacak şekilde yerleştirmeye dikkat etmeliyiz.

Kazanım

- Bu bölümde öğrencilerin yorum satırını tanıması sağlanmıştır.

✅ Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Yorum Satırı konusunun anlatılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere Yorum Satırı konusunun anlatılması ve uygulamalarla pekiştirilmesi amaçlanmaktadır.
2	☑ Uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen eşliğinde editör üzerinde uygulamaların yapılması hedeflenmektedir.

PYTHON: KAÇIŞ DIZILERI

Müfredat Kodu / Ünite / Konu	İCT-301-1.4 / Python / Kaçış Dizileri
Hedeflenen Süre	30 dakika

🚀 Kaçış Dizileri

⚡ Python kaçış karakterleri karakter (string) dizisi içerisinde yer alan ve özel anlamı olan ifadelerdir. Dizelere özel karakterler eklemenize izin verir. Bunu yapmak için, kaçmak istediğiniz karakterin önüne bir ters eğik çizgi (\ ) eklemeniz yeterlidir.

Eğer bir karakter dizisi içinde çift tırnak işareti geçiyorsa, o karakter dizisini tek tırnakla; eğer tek tırnak geçiyorsa da o karakter dizisini çift tırnakla tanımlayarak bu sorunun üstesinden gelebiliyoruz. Ama daha önce de söylediğimiz gibi, 'kaçış dizileri' adı verilen birtakım araçları kullanarak, mesela içinde çift tırnak geçen karakter dizilerini yine çift tırnakla tanımlayabiliriz.

▲ *Satır Başı (\n)*

Python'daki en temel kaçış dizisi \ işaretidir. Bu kaçış dizisi başka karakterlerle birleşerek, farklı işlevlere sahip yeni kaçış dizileri de oluşturabilir.

Satır başı karakterine 'yeni satır karakteri' dendiği de olur.

```
>>> print("birinci satır\nikinci satır\nüçüncü satır")

birinci satır
ikinci satır
üçüncü satır
```

Gördüğünüz gibi, \n adlı kaçış dizisi, bir alt satıra geçilmesini sağlıyor. İşte aslında \n kaçış dizisi de, \ ile 'n' harfinin birleşmesinden oluşmuş bir kaçış dizisidir. Burada \ işaretinin görevi, 'n' harfinin özel bir anlam kazanmasını sağlamaktır. \ işareti ile 'n' harfi birleştiğinde 'satır başı karakteri' denen özel bir karakter dizisi ortaya çıkarıyor.

▲ *Sekme (\t)*

Python'da \ işareti sadece 'n' harfiyle değil, başka harflerle de birleşebilir. Örneğin \ işaretini 't' harfiyle birleştirerek yine özel bir anlam ifade eden bir kaçış dizisi elde edebiliriz:

```
>>> print("abc\tdef")

abc def
```

Burada \t adlı kaçış dizisi, "abc" ifadesinden sonra sanki Tab (sekme) tuşuna basılmış gibi bir etki oluşturarak "def" ifadesini sağa doğru itiyor.

▲ *Aynı Satır Başı (\r)*

Bu kaçış dizisi, bir karakter dizisinde aynı satırın en başına dönülmesini sağlar. Bu kaçış dizisinin işlevini tanımına bakarak anlamak biraz zor olabilir. O yüzden dilerseniz bu kaçış dizisinin ne işe yaradığını bir örnek üzerinde göstermeye çalışalım:

```
>>> print("Merhaba\rZalim Dünya!")

Zalim Dünya!
```

Burada olan şey şu: Normal şartlar altında, print() fonksiyonu içine yazdığımız bir karakter dizisindeki bütün karakterler soldan sağa doğru tek tek ekrana yazdırılır:

```
>>> print("Merhaba Zalim Dünya!")

Merhaba Zalim Dünya!
```

Ancak eğer karakter dizisinin herhangi bir yerine \r adlı kaçış dizisini yerleştirirsek, bu kaçış dizisinin bulunduğu konumdan itibaren aynı satırın başına dönülecek ve \r kaçış dizisinden sonra gelen bütün karakterler satır başındaki karakterlerin üzerine yazacaktır.

Kazanım

- Öğrenciler bu bölümde kaçış dizileri ve işlevlerini kavrar.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Kaçış Dizileri konusunun anlatılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere Kaçış Dizileri konusunun anlatılması ve uygulamalarla pekiştirilmesi amaçlanmaktadır.
2	☐ Uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen eşliğinde editör üzerinde uygulamaların yapılması amaçlanmaktadır.

PYTHON: STRING (KARAKTER)

Müfredat Kodu / Ünite / Konu	İCT-301-1.5.1 / Python Veri Türleri / String (Karakter)
Hedeflenen Süre	45 dakika

🚀 String (Karakter)

Teknik olarak söylemek gerekirse, yazdığımız;

```
“Merhaba Dünya!”
```

ifadesi bir karakter dizisidir. İngilizcede buna *string* adı verilir ve programlama açısından son derece önemli bir kavramdır bu. Kavramın adından da rahatlıkla anlayabileceğiniz gibi, bir veya daha fazla karakterden oluşan öğelere karakter dizisi (*string*) diyoruz. Karakter dizileri bütün programcılık maceramız boyunca karşımıza çıkacak. O yüzden bu kavramı ne kadar erken öğrenirsek o kadar iyi. Peki bir verinin karakter dizisi olup olmamasının bize ne faydası var? Yani yukarıdaki cümle karakter dizisi olmuş olmamış bize ne?

Python’da, o anda elinizde bulunan bir verinin hangi tipte olduğunu bilmek son derece önemlidir. Çünkü bir verinin ait olduğu tip, o veriyle neler yapıp neler yapamayacağınızı belirler. Python’da her veri tipinin belli başlı özellikleri vardır. Dolayısıyla, elimizdeki bir verinin tipini bilmezsek o veriyi programlarımızda etkin bir şekilde kullanamayız. İşte yukarıda örneğini verdiğimiz “Merhaba Dünya!” adlı karakter dizisi de bir veri tipidir. Python’da karakter dizileri dışında başka veri tipleri de bulunur. Biraz sonra başka veri tiplerini de inceleyeceğiz.

Dikkat ederseniz “Merhaba Dünya!” adlı karakter dizisini tırnak içinde gösterdik. Bu da çok önemli bir bilgidir. Eğer bu cümleyi tırnak içine almazsak programımız hata verecektir:

```
Merhaba Dünya!

  File "<stdin>", line 1
    Merhaba Dünya!
              ^
SyntaxError: invalid syntax
```

Zaten tırnak işaretleri, karakter dizilerinin ayırt edici özelliğidir. Öyle ki, Python’da tırnak içinde gösterdiğiniz her şey bir karakter dizisidir. Örneğin şu bir karakter dizisidir:

```
"a"
```

Dilerseniz biz karakter dizilerine elimizin alışması için birkaç örnek daha verelim:

```
"Elma"

"Guido Van Rossum"

"Python programlama dili"
```

Yukarıdaki örneklerin hepsi birer karakter dizisidir. Dikkat ettiyseniz yukarıdaki karakter dizilerinin hepsinin ortak özelliği tırnak içinde gösteriliyor olmasıdır. Dediğimiz gibi, tırnak işaretleri karakter dizilerinin ayırt edici özelliğidir.

Peki bir verinin karakter dizisi olup olmadığından nasıl emin olabilirsiniz?

Eğer herhangi bir verinin karakter dizisi olup olmadığı konusunda tereddütünüz varsa, type() adlı bir fonksiyondan yararlanarak o verinin tipini sorgulayabilirsiniz. Bu fonksiyonu şöyle kullanıyoruz:

```
type("Elma")

<class 'str'>
```

Bu 'fonksiyon' kelimesinin kafanızı karıştırmasına izin vermeyin. İleride fonksiyonları oldukça ayrıntılı bir şekilde inceleyeceğimiz için, type() *ifadesinin bir fonksiyon olduğunu bilmeniz şimdilik yeterli olacaktır. Üstelik fonksiyon konusunu ayrıntılı bir şekilde anlatma vakti geldiğinde siz fonksiyonlara dair pek çok şeyi zaten öğrenmiş olacaksınız.*

Burada amacımız "Elma" adlı öğenin tipini denetlemek. Denetlenecek öğeyi type() fonksiyonunun parantezleri arasında belirttiğimize dikkat edin. (Fonksiyonların parantezleri içinde belirtilen değerlere teknik dilde parametre adı verilir.)

Yukarıdaki çıktıda bizi ilgilendiren kısım, sondaki 'str' ifadesi. Tahmin edebileceğiniz gibi, bu ifade *string* kelimesinin kısaltmasıdır. Bu kelimenin Türkçede karakter dizisi anlamına geldiğini söylemiştik. O halde yukarıdaki çıktıya bakarak, "Elma" öğesinin bir karakter dizisi olduğunu söyleyebiliyoruz.

type() fonksiyonu yardımıyla kendi kendinize bazı denemeler yaparak konuyu iyice sindirmenizi tavsiye ederim. Mesela "½{656$#gfd" ifadesinin hangi sınıfa girdiğini kontrol etmekle başlayabilirsiniz.

Kazanım

- Öğrenciler karakter dizileri ve type() fonksiyonunu öğrenir.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ String(Karakter) konusunun anlatılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere String (Karakter) konusunun anlatılması ve uygulamalarla pekiştirilmesi amaçlanmaktadır.
2	☑ Uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen eşliğinde editör üzerinde uygulamaların yapılması amaçlanmaktadır.

PYTHON: INTEGER (SAYILAR)

Müfredat Kodu / Ünite / Konu	İCT-301-1.5.2 / Python Veri Türleri / Integer (Sayılar)
Hedeflenen Süre	45 dakika

🚀 Integer (Sayılar)

Dedik ki, Python'da birtakım veri tipleri bulunur ve karakter dizileri de bu veri tiplerinden yalnızca biridir. Veri tipi olarak karakter dizilerinin dışında, biraz önce aritmetik işleçler vesilesiyle sözünü ettiğimiz, bir de 'sayı' (*integer*) adlı bir veri tipi vardır.

Herhalde sayıların ne anlama geldiğini tarif etmeye gerek yok. Bunlar bildiğimiz sayılardır. Mesela:

```
23

4567

2.3

(10+2j)
```

Python'da sayıların farklı alt türleri bulunur. Mesela tam sayılar, kayan noktalı sayılar, karmaşık sayılar...

Yukarıdaki örnekler arasında geçen 23 ve 4567 birer tamsayıdır. İngilizcede bu tür sayılara *integer* adı verilir.

2.3 ise bir kayan noktalı sayıdır (*floating point number* veya kısaca *float*). Bu arada kayan noktalı sayılarda basamak ayracı olarak virgül değil, nokta işareti kullandığımıza dikkat edin.

Sayıları temel olarak öğrendiğimize göre etkileşimli kabuğu basit bir hesap makinesi niyetine kullanabiliriz:

```
5 + 2

7

25 * 25

625

5 / 2

2.5

10 - 3

7
```

Öncelikle aritmetik işleçleri ve görevlerini şöylece sıralayalım:

İşleç	Görevi
+	toplama
-	çıkarma
*	çarpma
/	bölme

Yukarıdaki örneklerde bir şey dikkatinizi çekmiş olmalı: Karakter dizilerini tanımlarken tırnak işaretlerini kullandık. Ancak sayılarda tırnak işareti yok. Daha önce de dediğimiz gibi, tırnak işaretleri karakter dizilerinin ayırt edici özelliğidir. Python'da tırnak içinde gösterdiğiniz her şey bir karakter dizisidir. Mesela şu örneklere bakalım:

```
34657

34657
```

Bu bir sayıdır. Peki ya şu?

```
"34657"

'34657'
```

Bu ise bir karakter dizisidir. Dilerseniz biraz önce öğrendiğimiz type() fonksiyonu yardımıyla bu verilerin tipini sorgulayalım:

```
type(34657)

<class 'int'>
```

Buradaki 'int' ifadesi İngilizce "*integer*", yani tamsayı kelimesinin kısaltmasıdır. Demek ki 34657 sayısı bir tam sayıymış. Bir de şuna bakalım:

```
type("34657")

<class 'str'>
```

Gördüğünüz gibi, 34657 sayısını tırnak içine aldığımızda bu sayı artık sayı olma özelliğini yitiriyor ve bir karakter dizisi oluyor. Şu anda bu çok önemsiz bir ayrıntıymış gibi gelebilir size, ama aslında son derece önemli bir konudur bu. Bu durumun etkilerini şu örneklerde görebilirsiniz:

```
23 + 65

88
```

Burada normal bir şekilde iki sayıyı birbiriyle topladık.

Bir de şuna bakın:

```
"23" + "65"

'2365'
```

Burada ise Python iki karakter dizisini yan yana yazmakla yetindi; yani bunları birleştirdi. Python açısından “23” ve 23 birbirinden farklıdır. “23” bir karakter dizisi iken, 23 bir sayıdır. Aynı şey “65” ve 65 için de geçerlidir. Yani Python açısından “65” ile “Merhaba Dünya!” arasında hiç bir fark yoktur. Bunların ikisi de karakter dizisi sınıfına girer. Ancak 65 ile “65” birbirinden farklıdır. 65 bir sayı iken, “65” bir karakter dizisidir.

Bu bilgi, özellikle aritmetik işlemlerde büyük önem taşır. Bunu dilerseniz şu örnekler üzerinde gösterelim:

```
45 + "45"

Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

Gördüğünüz gibi, yukarıdaki kodlar hata veriyor. Bunun sebebi bir sayı (45) ile bir karakter dizisini (“45”) birbiriyle toplamaya çalışmamızdır. Asla unutmayın, aritmetik işlemler ancak sayılar arasında yapılır. Karakter dizileri ile herhangi bir aritmetik işlem yapılamaz.

Bir de şuna bakalım:

```
45 + 45

90
```

Bu kodlar ise düzgün çalışır. Çünkü burada iki sayıyı aritmetik işleme soktuk ve başarılı olduk.

Son olarak şu örneği verelim:

```
"45" + "45"

'4545'
```

Burada + işlecinin toplama anlamına gelmediğine dikkat edin. Bu işleç burada iki karakter dizisini birleştirme görevi üstleniyor. Yani yukarıdaki örneğin şu örnekten hiçbir farkı yoktur:

```
>>> "google." + "com"

'google.com'
```

Bu iki örnekte de yaptığımız şey karakter dizilerini birbiriyle birleştirmektir.

Gördüğünüz gibi, + işlecinin sağındaki ve solundaki değerler birer karakter dizisi ise bu işleç bu iki değeri birbiriyle birleştiriyor. Ama eğer bu değerler birer sayı ise + işleci bu değerleri birbiriyle aritmetik olarak topluyor.

- işleci de + işlecine benzer bir iş yapar. Yani eğer * işleci bir sayı ve bir karakter dizisi ile karşılaşırsa, o karakter dizisini, verilen sayı kadar tekrarlar. Örneğin:

```
>>> "w" * 3
'www'
```

Burada * işleci bir karakter dizisi ("w") ve bir sayı (3) arasında işlem yaptığı için, karakter dizisini, ilgili sayı kadar tekrarlıyor. Yani "w" karakter dizisini 3 kez tekrarlıyor.

Bir de şuna bakalım:

```
>>> 25 * 3
75
```

Burada ise * işleci iki adet sayı arasında işlem yaptığı için bu değerleri birbiriyle aritmetik olarak çarpıyor ve 75 değerini elde etmemizi sağlıyor.

Böylece karakter dizileri ile sayılar arasındaki farkı öğrenmiş olduk. Bu bilgiler size önemsizmiş gibi gelebilir, ama aslında karakter dizileri ile sayılar arasındaki farkı anlamak, Python programlama dilinin önemli bir bölümünü öğrenmiş olmak demektir. İleride yazacağınız en karmaşık programlarda bile, bazen programınızın çalışmamasının (veya daha kötüsü yanlış çalışmasının) nedeninin karakter dizileri ile sayıları birbirine karıştırmanız olduğunu göreceksiniz. O yüzden burada öğrendiğiniz hiçbir bilgi kırıntısını baştan savmamanızı (ve sabırsızlık ya da acelecilik etmemenizi) tavsiye ederim.

Kazanım

- Öğrenciler bu bölümde; integer, float gibi sayı tipleriyle tanışır, aritmek işleçler ve görevlerini öğrenir.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Sayılar konusunun anlatılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere Sayılar konusunun anlatılması ve uygulamalarla pekiştirilmesi amaçlanmaktadırç.
2	☑ Uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen eşliğinde editör üzerinde uygulamaların yapılması amaçlanmaktadır.

PYTHON: SAYI METOTLARI

Müfredat Kodu / Ünite / Konu	İCT-301-1.5.3 / Python Veri Türleri / Sayı Metotları
Hedeflenen Süre	45 dakika

🚀 Sayı Metotları

Tıpkı öteki veri tiplerinde olduğu gibi, sayıların da bazı metotları bulunur. Bu metotları kullanarak sayılar üzerinde çeşitli işlemler gerçekleştirebiliriz.

✨ Tam Sayıların Metotları

Dediğimiz gibi, Python'da birkaç farklı sayı tipi bulunur. Biz ilk olarak tam sayı (*integer*) denen sayı tipinin metot ve niteliklerini inceleyeceğiz. Şimdilik bizi ilgilendiren tek bir metot var.

Bu metodun adı bit_length().

🔺 *bit_length() Metodu*

Bilgisayarlar hakkında bilmemiz gereken en önemli bilgilerden biri şudur: Bilgisayarlar ancak ve ancak sayılarla işlem yapabilir. Bilgisayarların işlem yapabildiği sayılar da onlu sistemdeki sayılar değil, ikili sistemdeki sayılardır. Yani 0'lar ve 1'ler.

Bilgisayar terminolojisinde bu 0'lar ve 1'lerden oluşan her bir basamağa 'bit' adı verilir. Yani ikili sayma sisteminde '0' ve '1' sayılarından herbiri 1 bit'tir. Mesela onlu sistemde 2 sayısının ikili sistemdeki karşılığı olan 10 sayısı iki bit'lik bir sayıdır. Onlu sistemdeki 100 sayısının ikili sistemdeki karşılığı olan 1100100 sayısı ise yedi bitlik bir sayıdır.

Bu durumu daha net bir şekilde görebilmek için şu kodları yazalım:

```
>>> for i in range(11):
...     print(i, bin(i)[2:], len(bin(i)[2:]), sep="\t")
...
0       0       1
1       1       1
2       10      2
3       11      2
4       100     3
5       101     3
6       110     3
7       111     3
8       1000    4
9       1001    4
10      1010    4
```

Burada ikinci sütundaki sayılar ilk sütundaki sayıların ikili sistemdeki karşılıklarıdır. Üçüncü sütundaki sayılar ise her bir sayının kaç bit olduğunu, yani bir bakıma ikili sayma sisteminde kaç basamağa sahip olduğunu gösteriyor.

İşte herhangi bir tam sayının kaç bit'lik bir yer kapladığını öğrenmek için, tam sayıların metotlarından biri olan bit_length() metodundan yararlanacağız:

```
>>> sayı = 10
>>> sayı.bit_length()

4
```

Kayan Noktalı Sayıların Metotları

Python'da tam sayılar dışında kayan noktalı sayıların da olduğunu biliyoruz. Biz bu metotlar arasından, as_integer_ratio() ve is_integer() adlı metotlarla ilgileneceğiz.

as_integer_ratio() Metodu

Bu metot, birbirine bölündüğünde ilgili kayan noktalı sayıyı veren iki adet tam sayı verir bize. Örnek üzerinden açıklayalım:

```
>>> sayı = 4.5
>>> sayı.as_integer_ratio()

(9, 2)
```

9 sayısını 2 sayısına böldüğümüzde 4.5 sayısını elde ederiz. İşte as_integer_ratio() metodu, bu 9 ve 2 sayılarını bize ayrı ayrı verir.

is_integer() Metodu

Bir kayan noktalı sayının ondalık kısmında 0 harici bir sayının olup olmadığını kontrol etmek için bu metodu kullanıyoruz. Örneğin:

```
>>> (12.0).is_integer()

True

>>> (12.5).is_integer()

False
```

Kazanım

- Öğrenci tam sayı ve kayan noktalı sayı metotlarını öğrenir, uygulamalarla birlikte konuyu pekiştirir.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Sayıların Metotları konusunun anlatılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere Sayıların Metotları konusunun anlatılması ve uygulamalarla pekiştirilmesi amaçlanmaktadır.
2	☑ Uygulamaların yapılması.	☐ Eğitmen ☑	Eğitmen eşliğinde editör üzerinde uygulamaların yapılması amaçlanmaktadır.

		Öğrenciler	

PYTHON: LISTELER

Müfredat Kodu / Ünite / Konu	İCT-301-1.5.4 / Python Veri Türleri / Listeler
Hedeflenen Süre	45 dakika

🚀 Listeler

Bu bölümde birer veri tipi olarak listelerden söz etmenin yanı sıra liste metotlarından da bahsedeceğiz. Listeleri öğrendikten sonra Python'daki hareket imkanınızın bir hayli genişlediğine tanık olacaksınız.

🧐 Liste Tanımlamak

Listeleri tanımaya, bu veri tipini nasıl tanımlayacağımızı öğrenerek başlayalım.

Hatırlarsanız bir karakter dizisi tanımlayabilmek için şöyle bir yol izliyorduk:

```
>>> kardiz = "karakter dizisi"
```

Yani herhangi bir öğeyi karakter dizisi olarak tanımlayabilmek için yapmamız gereken tek şey o öğeyi tırnak içine almaktı. Herhangi bir öğeyi (tek, çift veya üç) tırnak içine aldığımızda karakter dizimizi tanımlamış oluyoruz. Liste tanımlamak için de buna benzer bir şey yapıyoruz. Dikkatlice bakın:

```
>>> liste = ["öğe1", "öğe2", "öğe3"]
```

Gördüğünüz gibi, liste tanımlamak da son derece kolay. Bir liste elde etmek için, öğeleri birbirinden virgülle ayırıp, bunların hepsini köşeli parantezler içine alıyoruz.

Karakter dizilerini anlatırken, herhangi bir nesnenin karakter dizisi olup olmadığından emin olmak için type() fonksiyonundan yararlanabileceğimizi söylemiştik. Eğer bir nesne type() fonksiyonuna <class 'str'> cevabı veriyorsa o nesne bir karakter dizisidir. Listeler için de buna benzer bir sorgulama yapabiliriz:

```
>>> liste = ["öğe1", "öğe2", "öğe3"]
>>> type(liste)

<class 'list'>
```

Bu çıktıdan anlıyoruz ki, liste veri tipi type() fonksiyonuna <class 'list'> cevabı veriyor. Dolayısıyla, eğer bir nesne type() fonksiyonuna <class 'list' > cevabı veriyorsa o nesnenin bir liste olduğunu rahatlıkla söyleyebiliriz.

Yukarıda tanımladığımız liste adlı listeye baktığımızda dikkatimizi bir şey çekiyor olmalı. Bu listeye şöyle bir baktığımızda, aslında bu listenin, içinde üç adet karakter dizisi barındırdığını görüyoruz. Gerçekten de listeler, bir veya daha fazla veri tipini içinde barındıran kapsayıcı bir veri tipidir. Mesela şu listeye bir bakalım:

```
>>> liste = ["Ahmet", "Mehmet", 23, 65, 3.2]
```

Gördüğünüz gibi, liste içinde hem karakter dizileri (“Ahmet”, “Mehmet”), hem de sayılar (23, 65, 3.2) var

Listelerin Öğelerine Erişmek

Listelerin öğelerine ulaşmak için şu yöntemi kullanıyoruz:

```
>>> meyveler = ["elma", "armut", "çilek", "kiraz"]
>>> meyveler[0]

'elma'
```

Bir karakter dizisinin 0. öğesini aldığımızda o karakter dizisinin ilk karakterini almış oluyoruz. Bir listenin 0. öğesini aldığımızda ise o listenin ilk öğesini almış oluyoruz.

Sayma yöntemi olarak ise karakter dizileri ve listelerde aynı mantık geçerli. Hem listelerde hem de karakter dizilerinde Python saymaya 0’dan başlıyor. Yani karakter dizilerinde olduğu gibi, listelerde de ilk öğenin sırası 0.

Eğer bu listenin öğelerinin hepsine tek tek ulaşmak isterseniz for döngüsünden yararlanabilirsiniz:

```
meyveler = ["elma", "armut", "çilek", "kiraz"]

for meyve in meyveler:
    print(meyve)
```

Listelerin Öğelerini Değiştirmek

Listeler değiştirilebilen (*mutable*) bir veri tipidir. Dolayısıyla listeler üzerinde doğrudan değişiklik yapabiliriz. Bir liste üzerinde değişiklik yapabilmek için o listeyi yeniden tanımlamamıza gerek yok. Şu örneği dikkatlice inceleyin:

```
>>> renkler = ["kırmızı", "sarı", "mavi", "yeşil", "beyaz"]
>>> print(renkler)

['kırmızı', 'sarı', 'mavi', 'yeşil', 'beyaz']

>>> renkler[0] = "siyah"
>>> print(renkler)

['siyah', 'sarı', 'mavi', 'yeşil', 'beyaz']
```

Liste öğelerini nasıl değiştirdiğimize çok dikkat edin. Yukarıdaki örnekte renkler adlı listenin 0. öğesini değiştirmek istiyoruz. Bunun için şöyle bir formül kullandık:

```
renkler[öğe_sırası] = yeni_öğe
```

Listeye Öğe Eklemek

Listeler büyüyüp küçülebilen bir veri tipidir. Yani Python'da bir listeye istediğiniz kadar öğe ekleyebilirsiniz. Diyelim ki elimizde şöyle bir liste var:

```
>>> liste = [2, 4, 5, 7]
```

Bu listeye yeni bir öğe ekleyebilmek için şöyle bir kod yazabiliriz:

```
>>> liste + [8]

[2, 4, 5, 7, 8]
```

Listeden Öğe Çıkarmak

Bir listeden öğe silmek için del adlı ifadeden yararlanabilirsiniz. Örneğin:

```
>>> liste = [1, 5, 3, 2, 9]
>>> del liste[-1]
>>> liste

[1, 5, 3, 2]
```

Listeleri Silmek

Python'da listeleri tamamen silmek de mümkündür. Örneğin:

```
>>> liste = [1, 5, 3, 2, 9]
>>> del liste
>>> liste

Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'liste' is not defined
```

Listeleri Kopyalamak

Diyelim ki, yazdığınız bir programda, var olan bir listeyi kopyalamak, yani aynı listeden bir tane daha üretmek istiyorsunuz. Mesela elimizde şöyle bir liste olsun:

```
>>> li1 = ["elma", "armut", "erik"]
```

Amacımız bu listeden bir tane daha oluşturmak. İlk olarak aklınıza şöyle bir yöntem gelmiş olabilir:

```
>>> li2 = li1
```

Gerçekten de bu yöntem bize aynı öğelere sahip iki liste verdi:

```
>>> print(li1)

["elma", "armut", "erik"]

>>> print(li2)

["elma", "armut", "erik"]
```

Gelin şimdi ilk listemiz olan li1 üzerinde bir değişiklik yapalım. Mesela bu listenin “elma” olan ilk öğesini “karpuz” olarak değiştirelim:

```
>>> li1[0] = "karpuz"
>>> print(li1)

["karpuz", "armut", "erik"]
```

Gördüğünüz gibi, li1 adlı listenin ilk öğesini başarıyla değiştirdik. Şimdi şu noktada, li2 adlı öbür listemizin durumunu kontrol edelim:

```
>>> print(li2)

["karpuz", "armut", "erik"]
```

O da ne! Biz biraz önce li1 üzerinde değişiklik yapmıştık, ama görünüşe göre bu değişiklikten li2 de etkilenmiş. Muhtemelen beklediğiniz şey bu değildi. Yani siz li2 listesinin içeriğinin aynı kalıp, değişiklikten yalnızca li1 listesinin etkilenmesini istiyordunuz. Biraz sonra bu isteğinizi nasıl yerine getirebileceğinizi göstereceğiz. Ama önce dilerseniz, bir liste üzerindeki değişiklikten öteki listenin de neden etkilendiğini anlamaya çalışalım.

Hatırlarsanız, listelerin değiştirilebilir (*mutable*) bir veri tipi olduğunu söylemiştik. Listeler bu özellikleriyle karakter dizilerinden ayrılıyor. Zira biraz önce li1 ve li2 üzerinde yaptığımız işlemin bir benzerini karakter dizileri ile yaparsak farklı bir sonuç alırız. Dikkatlice bakın:

```
>>> a = "elma"
```

Burada, değeri “elma” olan a adlı bir karakter dizisi tanımladık. Şimdi bu karakter dizisini kopyalayalım:

```
>>> b = a
```

```
>>> a
'elma'
>>> b
'elma'
```

Böylece aynı değere sahip iki farklı karakter dizimiz olmuş oldu.

Şimdi a adlı karakter dizisi üzerinde değişiklik yapalım. Ama biz biliyoruz ki, bir karakter dizisini değiştirmenin tek yolu, o karakter dizisini yeniden tanımlamaktır:

```
>>> a = "E" + a[1:]
>>> a
'Elma'
```

Burada yaptığımız şeyin bir 'değişiklik' olmadığına dikkatinizi çekmek isterim. Çünkü aslında biz burada var olan a adlı değişken üzerinde bir değişiklik yapmak yerine, yine a adı taşıyan başka bir değişken oluşturuyoruz.

Peki bu 'değişiklikten' öbür karakter dizisi etkilendi mi?

```
>>> b
'elma'
```

Gördüğünüz gibi, bu değişiklik öteki karakter dizisini etkilememiş. Bunun sebebinin, karakter dizilerinin değiştirilemeyen (*immutable*) bir veri tipi olması olduğunu söylemiştik.

Kazanım

- Öğrencilerin listeleri ve listeye ait öğeleri kavramasını, liste öğeleri üzerinde işlem yapabilmesini sağlar.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Listeler konusunun anlatılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere Listeler konusunun anlatılması ve uygulamalarla pekiştirilmesi amaçlanmaktadır.
2	☑ Uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen eşliğinde editör üzerinde uygulamaların yapılması amaçlanmaktadır.

PYTHON: LISTE METOTLARI

Müfredat Kodu / Ünite / Konu	İCT-301-1.5.5 / Python Veri Türleri / Liste Metotları
Hedeflenen Süre	45 dakika

🚀 Liste Metotları

Burada, geçen bölümde kaldığımız yerden devam edeceğiz listeleri anlatmaya. Ağırlıklı olarak bu bölümde listelerin metotlarından söz edeceğiz. 'Metot' kavramını karakter dizilerinden hatırlıyorsunuz. Karakter dizilerini anlatırken bol miktarda metot görmüştük.

Python'da bütün veri tipleri bize birtakım metotlar sunar. Bu metotlar yardımıyla, ilgili veri tipi üzerinde önemli değişiklikler veya sorgulamalar yapabiliyoruz.

▲ *append() Metodu*

append kelimesi İngilizcede 'eklemek, ilave etmek, iliştirmek' gibi anlamlara gelir. append() metodunun görevi de kelime anlamıyla uyumludur. Bu metodu, bir listeye öğe eklemek için kullanıyoruz. Mesela:

```
>>> liste = ["elma", "armut", "çilek"]
>>> liste.append("erik")
```

Bu metot, yeni öğeyi listenin en sonuna ekler. Mesela yukarıdaki örnekte "erik" adlı karakter dizisi listede "çilek" adlı karakter dizisinin sağına eklendi.

▲ *extend() Metodu*

extend kelimesi İngilizcede 'genişletmek, yaymak' gibi anlamlara gelir. İşte extend() adlı metot da kelime anlamına uygun olarak listeleri 'genişletir'.

Şöyle bir düşündüğünüzde extend() metodunun append() metoduyla aynı işi yaptığını zannedebilirsiniz. Ama aslında bu iki metot işleyiş olarak birbirinden çok farklıdır.

append() metodunu kullanarak yazdığımız şu koda dikkatlice bakın:

```
li1 = [1, 3, 4]
li2 = [10, 11, 12]
li1. append(li2)

print(li1)
```

append() metodunu bir listeye her defasında sadece tek bir öğe eklenmesine izin verir. Yukarıda olduğu gibi, eğer bu metodu kullanarak bir listeye yine bir liste eklemeye çalışırsanız, eklediğiniz liste tek bir öğe olarak eklenecektir. Yani yukarıdaki kodlar size şöyle bir çıktı verecektir:

```
[1, 3, 4, [10, 11, 12]]
```

Gördüğünüz gibi, [10, 11, 12] listesi öteki listeye tek bir liste halinde eklendi. İşte extend() metodu bu tür durumlarda işinize yarayabilir. Mesela yukarıdaki örneği bir de extend() metodunu kullanarak yazalım:

```
li1 = [1, 3, 4]
li2 = [10, 11, 12]
li1. extend(li2)

print(li1)
```

Bu defa şöyle bir çıktı alıyoruz:

```
[1, 3, 4, 10, 11, 12]
```

Gördüğünüz gibi, extend() metodu tam da kelime anlamına uygun olarak listeyi yeni öğelerle genişletti.

insert() Metodu

Bildiğiniz gibi, + işleci, append() ve extend() metotları öğeleri listenin sonuna ekliyor. Peki biz bir öğeyi listenin sonuna değil de, liste içinde başka bir konuma eklemek istersek ne yapacağız? İşte bunun için insert() adlı başka bir metottan yararlanacağız.

insert kelimesi 'yerleştirmek, sokmak' gibi anlamlara gelir. insert() metodu da bu anlama uygun olarak, öğeleri listenin istediğimiz bir konumuna yerleştirir. Dikkatlice inceleyin:

```
>>> liste = ["elma", "armut", "çilek"]
>>> liste.insert(0, "erik")
>>> print(liste)

['erik', 'elma', 'armut', 'çilek']
```

Gördüğünüz gibi insert() metodu iki parametre alıyor. İlk parametre, öğenin hangi konuma yerleştirileceğini, ikinci parametre ise yerleştirilecek öğenin ne olduğunu gösteriyor. Yukarıdaki örnekte "erik" öğesini listenin 0. konumuna, yani listenin en başına yerleştiriyoruz.

remove() Metodu

Bu metot listeden öğe silmemizi sağlar. Örneğin:

```
>>> liste = ["elma", "armut", "çilek"]
>>> liste.remove("elma")
>>> liste

['armut', 'çilek']
```

reverse() Metodu

Python'da listelerin öğelerini ters çevirmek için listelerin reverse() metodunu kullanabilirsiniz:

```
>>> liste = ["elma", "armut", "çilek"]
>>> liste.reverse()
>>> liste

['çilek', 'armut', 'elma']
```

pop() Metodu

Tıpkı remove() metodu gibi, bu metot da bir listeden öğe silmemizi sağlar:

```
>>> liste = ["elma", "armut", "çilek"]
>>> liste.pop()
```

Ancak bu metot, remove() metodundan biraz farklı davranır. pop() metodunu kullanarak bir liste öğesini sildiğimizde, silinen öğe ekrana basılacaktır. Bu metot parametresiz olarak kullanıldığında listenin son öğesini listeden atar. Alternatif olarak, bu metodu bir parametre ile birlikte de kullanabilirsiniz. Örneğin:

```
>>> liste.pop(0)
```

Bu komut listenin 0. öğesini listeden atar ve atılan öğeyi ekrana basar.

sort() Metodu

Yine listelerin önemli bir metodu ile karşı karşıyayız. sort() adlı bu önemli metot bir listenin öğelerini belli bir ölçüte göre sıraya dizmemizi sağlar. Basit bir örnek verelim. Diyelim ki elimizde şöyle bir liste var:

```
üyeler = ['Ahmet', 'Mehmet', 'Ceylan', 'Seyhan', 'Mahmut', 'Zeynep',
         'Abdullah', 'Kadir', 'Kemal', 'Kamil', 'Selin', 'Senem',
         'Sinem', 'Tayfun', 'Tuna', 'Tolga']
```

Bu listedeki isimleri mesela alfabe sırasına dizmek için sort() metodunu kullanabiliriz:

```
>>> üyeler.sort()
>>> print(üyeler)

['Abdullah', 'Ahmet', 'Ceylan', 'Kadir', 'Kamil', 'Kemal', 'Mahmut',
 'Mehmet', 'Selin', 'Senem', 'Seyhan', 'Sinem', 'Tayfun', 'Tolga',
 'Tuna', 'Zeynep']
```

Bu metot elbette yalnızca harfleri alfabe sırasına dizmek için değil sayıları sıralamak için de kullanılabilir:

```
>>> sayılar = [1, 0, -1, 4, 10, 3, 6]
>>> sayılar.sort()
>>> print(sayılar)

[-1, 0, 1, 3, 4, 6, 10]
```

Gördüğünüz gibi, sort() metodu öğeleri artan sıralamaya tabi tutuyor. Yani öğeler 'a, b, c' veya 1, 2, 3 şeklinde sıralanıyor.

count() Metodu

Karakter dizileri ile listelerin ortak metotlarından biri de count() metodudur. Tıpkı karakter dizilerinde olduğu gibi, listelerin count() metodu da bir öğenin o veri tipi içinde kaç kez geçtiğini söyler:

```
>>> liste = ["elma", "armut", "elma", "çilek"]
>>> liste.count("elma")

2
```

Karakter dizilerinin count() metoduyla ilgili söylediğimiz her şey listelerin count() metodu için de geçerlidir.

copy() Metodu

Hatırlarsanız, geçen bölümde, listeleri, birbirlerini etkilemeyecek şekilde kopyalamak için şu iki yöntemi kullanmıştık:

```
>>> liste1 = ["ahmet", "mehmet", "özlem"]
>>> liste2 = liste1[:]
```

ve:

```
>>> liste2 = list(liste1)
```

İşte aynı iş için yukarıdakilere ek olarak copy() adlı bir metottan da yararlanabiliriz. Dikkatlice bakın:

```
>>> liste2 = liste1.copy()
```

Hangi yöntemi seçeceğiniz size kalmış...

clear() Metodu

Listelerle ilgili olarak ele alacağımız son metodun adı clear(). Bu metodun görevi bir listenin içeriğini silmektir.

Diyelim ki elimizde şöyle bir liste var:

```
>>> liste = [1, 2, 3, 5, 10, 20, 30, 45]
```

Bu listenin içini boşaltmak için clear() metodunu kullanabiliriz:

```
>>> liste.clear()
>>> liste

[]
```

Bu metodun del sözcüğünden farklı olduğunu dikkat edin. clear() metodu listenin içeriğini boşaltırken, del sözcüğü listeyi olduğu gibi ortadan kaldırır.

Kazanım

- Öğrencilerin liste ve demetlerin metotlarını kavraması.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Liste Metotları konusunun anlatılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere Liste Metotları konusunun anlatılması ve uygulamalarla pekiştirilmesi amaçlanmaktadır.
2	☐ Uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen eşliğinde editör üzerinde uygulamaların yapılması amaçlanmaktadır.

PYTHON: KÜMELER (SET)

Müfredat Kodu / Ünite / Konu	İCT-301-1.5.6 / Python Veri Türleri / Kümeler
Hedeflenen Süre	45 dakika

🚀 Kümeler

Tıpkı listeler, demetler, karakter dizileri, sayılar ve dosyalar gibi kümeler de Python'daki veri tiplerinden biridir. Adından da az çok tahmin edebileceğiniz gibi kümeler, matematikten bildiğimiz "küme" kavramıyla sıkı sıkıya bağlantılıdır. Bu veri tipi, matematikteki kümelerin sahip olduğu bütün özellikleri taşır. Yani matematikteki kümelerden bildiğimiz kesişim, birleşim ve fark gibi özellikler Python'daki kümeler için de geçerlidir.

🔺 *Küme Oluşturmak*

Kümelerin bize sunduklarından faydalanabilmek için elbette öncelikle bir küme oluşturmamız gerekiyor. Küme oluşturmak çok kolay bir işlemdir. Örneğin boş bir kümeyi şöyle oluşturuyoruz:

```
>>> boş_küme = set()
```

Kümelerin ayırt edici bir işareti yoktur. Küme oluşturmak için set() adlı özel bir fonksiyondan yararlanıyoruz.

⚠️ Burada dikkat etmemiz gereken en önemli konu sayılardan küme oluşturamayız:

```
>>> n = 10
>>> küme = set(n)

TypeError: 'int' object is not iterable
```

🔺 *Kümelerin Yapısı*

Bir önceki başlık altında kümelerin nasıl tanımlanacağını inceledik. Gelin şimdi de biraz kümelerin yapısından bahsedelim.

Örneğin şöyle bir küme tanımlayalım:

```
>>> kardiz = "Python Programlama Dili"
>>> küme = set(kardiz)
>>> print(küme)

{'g', 'D', 'a', ' ', 'o', 'n', 'm', 'l', 'i', 'h', 't', 'r', 'P', 'y'}
```

Burada bir şey dikkatinizi çekmiş olmalı. Bir öğeyi küme olarak tanımlayıp ekrana yazdırdığımızda elde ettiğimiz çıktı, o öğe içindeki her bir alt öğeyi tek bir kez içeriyor. Yani mesela "Python Programlama Dili" içinde iki adet "P" karakteri var, ama çıktıda bu iki "P" karakterinin yalnızca biri

görünüyor. Buradan anlıyoruz ki, kümeler aynı öğeyi birden fazla tekrar etmez. Bu çok önemli bir özelliktir ve pek çok yerde işimize yarar. Aynı durum karakter dizisi dışında kalan öteki veri tipleri için de geçerlidir. Yani mesela eğer bir listeyi küme haline getiriyorsak, o listedeki öğeler küme içinde yalnızca bir kez geçecektir. Listede aynı öğeden iki-üç tane bulunsa bile, kümemiz bu öğeleri teke indirecektir.

Kümelerin Metotları

clear() Metodu

Kümelerle ilgili olarak inceleyeceğimiz ilk metot clear(). Bu metodu daha önce sözlükleri çalışırken de görmüştük. Sözlüklerde bu metodun görevi sözlüğün içini boşaltmak idi. Burada da aynı vazifeyi görür:

```
>>> km = set("adana")
>>> for i in km:
...     print(i)
...
a
d
n

>>> km.clear()
>>> km
set()
```

Burada önce “km” adlı bir küme oluşturduk. Daha sonra da clear() metodunu kullanarak bu kümenin bütün öğelerini sildik. Artık elimizde boş bir küme var.

copy() Metodu

Listeler ve sözlükleri incelerken copy() adlı bir metot öğrenmiştik. Bu metot aynı zamanda kümelerle birlikte de kullanılabilir. Üstelik işlevi de aynıdır:

```
>>> km = set("kahramanmaraş")
>>> yedek = km.copy()
>>> yedek

{'a', 'r', 'h', 'k', 'm', 'ş', 'n'}

>>> km

{'a', 'h', 'k', 'm', 'n', 'r', 'ş'}
```

Burada bir şey dikkatinizi çekmiş olmalı. “km” adlı kümeyi “yedek” adıyla kopyaladık, ama bu iki kümenin çıktılarına baktığımız zaman öğe sıralamasının birbirinden farklı olduğunu görüyoruz. Biliyorsunuz, tıpkı sözlüklerde olduğu gibi, kümeler de sırasız veri tipleridir. Bu yüzden, elde ettiğimiz çıktıda öğeler rastgele diziliyor. Dolayısıyla öğelere sıralarına göre erişemiyoruz. Aynen sözlüklerde olduğu gibi...

add() Metodu

Kümelerden bahsederken, bunların değiştirilebilir bir veri tipi olduğunu söylemiştik. Dolayısıyla kümeler, üzerlerinde değişiklik yapmamıza müsaade eden metotlar da içerir. Örneğin add() bu tür metotlardan biridir. *Add* kelimesi Türkçe'de "eklemek" anlamına gelir. Adından da anlaşılacağı gibi, bu metot yardımıyla kümelerimize yeni öğeler ilave edebileceğiz. Hemen bunun nasıl kullanıldığına bakalım:

```
>>> küme = set(["elma", "armut", "kebap"])
>>> küme.add("çilek")
>>> print(küme)

{'elma', 'armut', 'kebap', 'çilek'}
```

Gördüğünüz gibi, add() metodunu kullanarak, kümemize çilek adlı yeni bir öğe ekledik. Eğer kümede zaten varolan bir öğe eklemeye çalışırsak kümede herhangi bir değişiklik olmayacaktır. Çünkü, daha önce de söylediğimiz gibi, kümeler her bir öğeyi tek bir sayıda barındırır.

Eğer bir kümeye birden fazla öğeyi aynı anda eklemek isterseniz for döngüsünden yararlanabilirsiniz:

```
>>> yeni = [1,2,3]
>>> for i in yeni:
...     küme.add(i)
...

>>> küme

{1, 2, 3, 'elma', 'kebap', 'çilek', 'armut'}
```

Burada yeni adlı listeyi kümeye for döngüsü ile ekledik. Ama bu işlemi yapmanın başka bir yolu daha vardır. Bu işlem için Python'da ayrı bir metot bulunur. Bu metodun adı update() metodudur. Sırası gelince bu metodu da göreceğiz.

Bu arada, yeri gelmişken kümelerin önemli bir özelliğinden daha söz edelim. Bir kümeye herhangi bir öğe ekleyebilmemiz için, o öğenin değiştirilemeyen (*immutable*) bir veri tipi olması gerekiyor.

▲ *difference() Metodu*

Bu metot iki kümenin farkını almamızı sağlar. Örneğin:

```
>>> k1 = set([1, 2, 3, 5])
>>> k2 = set([3, 4, 2, 10])

>>> k1.difference(k2)

{1, 5}
```

▲ *difference_update() Metodu*

Bu metot, difference() metodundan elde edilen sonuca göre bir kümenin güncellenmesini sağlar. Yani? Hemen bir örnek verelim:

```
>>> k1 = set([1, 2, 3])
>>> k2 = set([1, 3, 5])
>>> k1.difference_update(k2)
>>> print(k1)

{2}

>>> print(k2)

{1, 3, 5}
```

Gördüğünüz gibi, bu metot k1'in k2'den farkını aldı ve bu farkı kullanarak k1'i yeniden oluşturdu. k1 ile k2 arasındaki tek fark 2 adlı öğe idi. Dolayısıyla difference_update() metodunu uyguladığımızda k1'in öğelerinin silinip yerlerine 2 adlı öğenin geldiğini görüyoruz.

remove() Metodu

Bu metot da bir önceki bölümde gördüğümüz discard() metoduyla aynı işlevi yerine getirir. Eğer bir kümeden öğe silmek istersek remove() metodunu da kullanabiliriz:

```
>>> hayvanlar.remove("köpek")
```

Peki discard() varken remove() metoduna ne gerek var? Ya da tersi.

Bu iki metot aynı işlevi yerine getirse de aralarında önemli bir fark vardır. Hatırlarsanız discard() metoduyla, kümede olmayan bir öğeyi silmeye çalışırsak herhangi bir hata mesajı almayacağımızı söylemiştik. Eğer remove() metodunu kullanarak, kümede olmayan bir öğeyi silmeye çalışırsak, discard() metodunun aksine, hata mesajı alırız:

```
>>> hayvanlar.remove("fare")

Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
KeyError: 'fare'
```

Bu iki metot arasındaki bu fark önemli bir farktır. Bazen yazdığınız programlarda, duruma göre her iki özelliğe de ihtiyacınız olabilir.

update() Metodu

Bu metot, bir kümeyi güncellememizi sağlar.

```
>>> küme = set(["elma", "armut", "kebap"])
>>> yeni = [1, 2, 3]
>>> küme.update(yeni)
>>> print(küme)

{1, 2, 3, 'elma', 'armut', 'kebap'}
```

🔺 *pop() Metodu*

İnceleyeceğimiz son metot pop() metodu olacak. Gerçi bu metot bize hiç yabancı değil. Bu metodu listeler konusundan hatırlıyoruz. Orada öğrendiğimize göre, bu metot listenin bir öğesini silip ekrana basıyordu. Aslında buradaki fonksiyonu da farklı değil. Burada da kümelerin öğelerini silip ekrana basıyor:

```
>>> a = set(["elma", "armut", "kebap"])
>>> a.pop()

'elma'
```

Peki bu metot hangi ölçüte göre kümeden öğe siliyor? Herhangi bir ölçüt yok. Bu metot, küme öğelerini tamamen rastgele siliyor.

Böylelikle Python'da Listeler, Demetler, Sözlükler ve Kümeler konusunu bitirmiş olduk. Bu konuları sık sık tekrar etmek, hiç olmazsa arada sırada göz gezdirmek bazı şeylerin zihnimizde yer etmesi açısından oldukça önemlidir.

❄️ Dondurulmuş Kümeler (Frozenset)

Daha önce de söylediğimiz gibi, kümeler üzerinde değişiklik yapabiliyoruz. Zaten kümelerin add() ve remove() gibi metotlarının olması bu durumu teyit ediyor. Ancak kimi durumlarda, öğeleri üzerinde değişiklik yapılamayan kümelere de ihtiyaç duyabilirsiniz. Hatırlarsanız listeler ve demetler arasında da buna benzer bir ilişki var. Demetler çoğu zaman, üzerinde değişiklik yapılamayan bir liste gibi davranır. İşte Python aynı imkanı bize kümelerde de sağlar. Eğer öğeleri üzerinde değişiklik yapılamayan bir küme oluşturmak isterseniz set() yerine frozenset() fonksiyonunu kullanabilirsiniz. Dilerseniz hemen bununla ilgili bir örnek verelim:

```
>>> dondurulmuş = frozenset(["elma", "armut", "ayva"])
```

Dondurulmuş kümeleri bu şekilde oluşturuyoruz. Şimdi bu dondurulmuş kümenin metotlarına bakalım:

```
>>> dir(dondurulmuş)

['__and__', '__class__', '__contains__', '__delattr__', '__doc__', '__eq__',
 '__format__', '__ge__', '__getattribute__', '__gt__', '__hash__',
 '__init__', '__iter__', '__le__', '__len__', '__lt__', '__ne__', '__new__',
 '__or__', '__rand__', '__reduce__', '__reduce_ex__', '__repr__', '__ror__',
 '__rsub__', '__rxor__', '__setattr__', '__sizeof__', '__str__', '__sub__',
 '__subclasshook__', '__xor__', 'copy', 'difference', 'intersection',
 'isdisjoint', 'issubset', 'issuperset', 'symmetric_difference', 'union']
```

Gördüğünüz gibi, add(), remove(), update() gibi, değişiklik yapmaya yönelik metotlar listede yok.

Dondurulmuş kümeler ile normal kümeler arasında işlev olarak hiçbir fark yoktur.

Kazanım

- Öğrenciler bu bölümde kümeleri, küme yapılarını, küme üreteçlerini, küme metotlarını ve dondurulmuş kümeleri kavrar.

✅ Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Kümeler konusunun anlatılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere Kümeler konusunun anlatılması ve uygulamalarla pekiştirilmesi.
2	☑ Uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen eşliğinde editör üzerinde uygulamaların yapılması.

PYTHON: DEĞİŞKENLER

Müfredat Kodu / Ünite / Konu	İCT-301-1.6 / Python / Değişkenler
Hedeflenen Süre	90 dakika

🚀 Değişkenler

Python'da bir program içinde değerlere verilen isimlere değişken denir. Hemen bir örnek verelim:

```
>>> n = 5
```

Burada 5 sayısını bir değişkene atadık. Değişkenimiz ise n. Ayrıca 5 sayısını bir değişkene atamak için = işaretinden yararlandığımıza da çok dikkat edin. Buradan, = işaretinin Python programlama dilinde değer atama işlemleri için kullanıldığı sonucunu çıkarıyoruz.

n = 5 gibi bir komut yardımıyla 5 değerini n adlı değişkene atamamız sayesinde artık ne zaman 5 sayısına ihtiyaç duysak bu n değişkenini çağırmamız yeterli olacaktır:

```
>>> n
5
>>> n * 10
50
>>> n / 2
2.5
```

Gördüğünüz gibi, 5 değerini bir değişkene atadıktan sonra, bu 5 değerini kullanmamız gereken yerlerde sadece değişkenin adını kullandığımızda değişkenin değerini Python otomatik olarak yerine koyabiliyor. Yani n = 5 komutuyla n adlı bir değişken tanımladıktan sonra, artık ne zaman 5 sayısına ihtiyaç duysak n değişkenini çağırmamız yeterli olacaktır. Python o 5 değerini otomatik olarak yerine koyar.

✍️ Değişken Adı Belirleme Kuralları

Python programlama dilinde, değişken adı olarak belirleyebileceğimiz kelime sayısı neredeyse sınırsızdır. Yani hemen hemen her kelimeyi değişken adı olarak kullanabiliriz. Ama yine de değişken adı belirlerken dikkat etmemiz gereken bazı kurallar var. Bu kuralların bazıları zorunluluk, bazıları ise yalnızca tavsiye niteliğindedir.

Şimdi bu kuralları göz atalım:

- Değişken adları bir sayı ile başlayamaz.
- Değişken adları aritmetik işleçlerle başlayamaz.
- Değişken adları ya bir alfabe harfiyle ya da _ işaretiyle başlamalıdır:

- Değişken adları içinde Türkçe karakterler kullanabilirsiniz. Ancak ileride beklenmedik uyum sorunları çıkması ihtimaline karşı değişken adlarında Türkçe karakter kullanmaktan kaçınmak isteyebilirsiniz.
  - Aşağıdaki kelimeleri değişken adı olarak kullanamazsınız:

```
['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', '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']
```

Bunlar Python'da özel anlam ifade eden kelimelerdir. Etkileşimli kabuk zaten bu kelimeleri değişken adı olarak kullanmanıza izin vermeyecektir.

Kazanım

- Öğrenciler değişken ve değişken adı belirleme kurallarını öğrenir.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Değişkenler konusunun anlatılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere Değişkenler konusunun anlatılması ve uygulamalarla pekiştirilmesi amaçlanmaktadır.
2	☑ Uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen eşliğinde editör üzerinde uygulamaların yapılması amaçlanmaktadır.

PYTHON: KARAKTER DIZILERI

Müfredat Kodu / Ünite / Konu	İCT-301-1.7 / Python / Karakter Dizileri
Hedeflenen Süre	90 dakika

🚀 Karakter Dizileri

Karakter dizileri, adından da anlaşılacağı gibi, karakterlerin bir araya gelmesiyle oluşan bir dizidir. Karakter dizileri; tek, çift veya üç tırnak içinde gösterilen, öteki veri tiplerinden de bu tırnaklar aracılığıyla ayırt edilen özel bir veri tipidir. Teknik olarak ifade etmek gerekirse, bir nesneyi type() fonksiyonu yardımıyla sorguladığımızda, eğer <class 'str'> çıktısı alıyorsak bu nesne bir karakter dizisidir.

🖥️ Karakter Dizilerinin Öğelerine Erişmek

Python ile programlama yaparken karakter dizileri ile iki şekilde karşılaşabilirsiniz: Birincisi, bir karakter dizisini doğrudan kendiniz tanımlamış olabilirsiniz. İkincisi, karakter dizisi size başka bir kaynak aracılığıyla gelmiş olabilir (mesela input() fonksiyonu yardımıyla kullanıcıdan aldığınız bir veri).

Python'da kendi tanımladığınız ya da herhangi başka bir kaynaktan gelen karakter dizilerine şu şekilde erişebiliriz:

```
>>> nesne = "karakter dizisi"
```

Burada değeri "karakter dizisi" olan nesne adlı bir değişken tanımladık. Yazdığımız programlarda bu değişkene erişmek için, değişkenin adını kullanmamız yeterlidir.

✍️ Karakter Dizilerini Dilimlemek

Bu bölümde karakter dizilerini 'dilimlemekten' söz edeceğiz. Peki 'dilimlemek' derken neyi kastediyoruz? Aslında burada gerçek anlamda 'karpuz gibi dilimlemekten' söz ediyoruz... Şu örnek, ne demek istediğimizi daha net ortaya koyacaktır:

```
>>> site = "www.google.com"
>>> site[4:11]

'google'

>>> site[12:15]

'com'

>>> site[0:3]

'www'
```

Gördüğünüz gibi, karakter dizisine köşeli parantez içinde bazı değerler vererek bu karakter dizisini dilim dilim ayırdık.

♻️ Karakter Dizilerini Ters Çevirmek

Eğer amacınız bir karakter dizisini ters çevirmek, yani karakter dizisi içindeki her bir öğeyi tersten yazdırmaksa biraz önce öğrendiğimiz dilimleme yöntemini kullanabilirsiniz. Dikkatlice bakın:

```
>>> kardiz[::-1]
'midemtedaV iseçhaB lüG anaS'
```

Gördüğünüz gibi, "Sana Gül Bahçesi Vadetmedim" adlı karakter dizisi içindeki bütün karakterler sondan başa doğru ekrana dizildi.

⚠️ Üç Önemli Fonksiyon

Karakter dizilerinin temel özellikleri hakkında söyleyeceklerimizin sonuna geldik sayılır. Biraz sonra karakter dizilerinin çok önemli bir parçası olan metotlardan söz edeceğiz. Ama isterseniz metotlara geçmeden önce, çok önemli üç fonksiyondan söz edelim. Bu fonksiyonlar sadece karakter dizileri ile değil, başka veri tipleri ile çalışırken de işlerimizi bir hayli kolaylaştıracak.

▲ *dir() Fonksiyonu*

İngilizcede 'karakter dizisi'nin karşılığının *string*, bu kelimenin kısaltmasının da 'str' olduğunu hatırlıyor olmalısınız. İşte dir() fonksiyonuna parametre olarak bu 'str' kelimesini verdiğimizde, Python bize karakter dizilerinin bütün metotlarını listeliyor.

Karakter dizileri dışında, şimdiye kadar öğrendiğimiz başka bir veri tipi de sayılar. Biz Python'da sayıların tam sayılar (*integer*), kayan noktalı sayılar (*float*) ve karmaşık sayılar (*complex*) olarak üçe ayrıldığını da biliyoruz. Örnek olması açısından dir() fonksiyonunu bir de sırasıyla, tam sayılar, kayan noktalı sayılar ve karmaşık sayılar üzerinde de uygulayalım:

```
>>> dir(int)
>>> dir(float)
>>> dir(complex)
```

Gördüğünüz gibi, dir() fonksiyonunu kullanmak için, metotlarını listelemek istediğimiz nesneyi alıp dir() fonksiyonuna parametre olarak veriyoruz. Örneğin yukarıda karakter dizileri için str; tam sayılar için int; kayan noktalı sayılar için float; karmaşık sayılar için ise complex parametrelerini kullandık.

▲ *enumerate() Fonksiyonu*

Eğer yazdığınız bir programda numaralandırmaya ilişkin işlemler yapmanız gerekiyorsa Python'ın size sunduğu çok özel bir fonksiyondan yararlanabilirsiniz. Bu fonksiyonun adı enumerate().

Gelelim bu fonksiyonun nasıl kullanılacağına... Önce şöyle bir deneme yapalım:

```
>>> enumerate("google")
<enumerate object at 0x00E3BC88>
```

Yukarıdaki çıktıyı daha iyi anlayabilmek için bir for döngüsü kullanmak daha açıklayıcı olabilir:

```
>>> for i in enumerate("google"):
...     print(i)
...
(0, 'g')
(1, 'o')
(2, 'o')
(3, 'g')
(4, 'l')
(5, 'e')
```

Gördüğünüz gibi, gerçekten de bu fonksiyon bize bir öğe (mesela ‘g’ harfi) ve bu öğeye ait bir sıra numarası (mesela 0) veriyor.

help() Fonksiyonu

Python’la ilgili herhangi bir konuda yardıma ihtiyacınız olduğunda, internetten araştırma yaparak pek çok ayrıntılı belgeye ulaşabilirsiniz. Ama eğer herhangi bir nesne hakkında hızlı bir şekilde ve İngilizce olarak yardım almak isterseniz help() adlı özel bir fonksiyondan yararlanabilirsiniz.

Bu fonksiyonu iki farklı şekilde kullanıyoruz. Birinci yöntemde, etkileşimli kabuğa help() yazıp Enter düğmesine basıyoruz.

Python bu komutu verdiğimizde özel bir yardım ekranı açıyor bize. Bu ekranda >>> yerine help> ifadesinin olduğuna dikkat edin. Mesela dir() fonksiyonu hakkında bilgi almak için help> ifadesinden hemen sonra, hiç boşluk bırakmadan, şu komutu verebiliriz:

```
help> dir
```

Bu komut bize şu çıktıyı veriyor:

```
Help on built-in function dir in module builtins:

dir(...)
    dir([object]) -> list of strings

    If called without an argument, return the names in the current scope.
    Else, return an alphabetized list of names comprising (some of) the attributes
    of the given object, and of attributes reachable from it.
    If the object supplies a method named __dir__, it will be used; otherwise
    the default dir() logic is used and returns:
      for a module object: the module's attributes.
      for a class object:  its attributes, and recursively the attributes of its bases.
      for any other object: its attributes, its class's attributes, and
      recursively the attributes of its class's base classes.
```

Gördüğünüz gibi, dir() fonksiyonunun ne işe yaradığı ve nasıl kullanıldığı konusunda ayrıntılı bir bilgi ediniyoruz. Bu arada, hakkında bilgi almak istediğimiz fonksiyonu parantezsiz yazdığımıza dikkat edin.

Kazanım

- Öğrenci karakter dizilerinin öğelerine erişim sağlamayı ve diziler üzerinde değişiklik yapmayı öğrenir.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Karakter Dizileri konusunun anlatılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere Karakter Dizileri konusunun anlatılması ve uygulamalarla pekiştirilmesi amaçlanmaktadır.
2	☑ Uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen eşliğinde editör üzerinde uygulamaların yapılması amaçlanmaktadır.

PYTHON: KULLANICIDAN BILGI ALMAK

Müfredat Kodu / Ünite / Konu	İCT-301-1.8 / Python / Kullanıcıdan Bilgi Almak
Hedeflenen Süre	90 dakika

🚀 Kullanıcıdan Bilgi Almak

Şimdiye kadar Python programlama dili ile ilgili epey bilgi edindik. Ama muhtemelen buraya kadar öğrendiklerimiz sizi heyecanlandırmaktan bir hayli uzaktı. Zira şu ana kadar hep tek yönlü bir programlama faaliyeti yürüttük.

Mesela şimdiye kadar öğrendiklerimizi kullanarak ancak şöyle bir program yazabildik:

```
isim = "Arda"

print("Merhaba", isim, end="!\n")
```

Bu programı çalıştırdığımızda şöyle bir çıktı alacağımızı biliyorsunuz:

```
Merhaba Arda!
```

Bu programın ne kadar sıkıcı olduğunu herhalde söylemeye gerek yok. Bu programda isim değişkenini doğrudan kendimiz yazdığımız için programımız hiçbir koşulda Merhaba Arda dışında bir çıktı veremez. Çünkü bu program, tek yönlü bir programlama faaliyetinin ürünüdür.

İlk önce input() fonksiyonunu anlatarak yola koyulalım.

🔺 *input() Fonksiyonu*

input() da daha önce öğrendiğimiz type(), len() ve print() gibi bir fonksiyondur. Esasında biz bu fonksiyonu ilk kez burada görmüyoruz. Windows kullanıcıları, yazdıkları bir programı çift tıklayarak çalıştırabilmek için bu fonksiyonu kullandıklarını hatırlıyor olmalılar. Mesela şu programı ele alalım:

```
#!/usr/bin/env python3

kartvizit = """
Süper Hacker
Arda Kahveci
Tel: 0212 123 23 23
Faks: 0212 123 23 24
e.posta: hacker_arda@google.com
"""

print(kartvizit)
```

Bu programı yazıp kaydettikten sonra bu programın simgesi üzerine çift tıkladığımızda siyah bir komut ekranının çok hızlı bir şekilde açılıp kapandığını görürüz. Aslında programımız çalışıyor, ama programımız yapması gereken işi yaptıktan hemen sonra kapandığı için biz program penceresini görmüyoruz.

Programımızın çalıştıktan sonra hemen kapanmamasını sağlamak için son satıra bir input() fonksiyonu yerleştirmemiz gerektiğini biliyoruz:

```
#!/usr/bin/env python3

kartvizit = """
Süper Hacker
Arda Kahveci
Tel: 0212 123 23 23
Faks: 0212 123 23 24
e.posta: hacker_arda@google.com
"""

print(kartvizit)

input()
```

Bu sayede programımız kullanıcıdan bir giriş bekleyecek ve o girişi alana kadar da kapanmayacaktır. Programı kapatmak için Enter düğmesine basabiliriz.

eval() Fonksiyonu

Dilerseniz basit bir eval() örneği verelim:

```
print("""
Basit bir hesap makinesi uygulaması.

İşleçler:

   +   toplama
   -   çıkarma
   *   çarpma
   /   bölme

Yapmak istediğiniz işlemi yazıp ENTER
tuşuna basın. (Örneğin 23 ve 46 sayılarını
çarpmak için 23 * 46 yazdıktan sonra
ENTER tuşuna basın.)
""")

veri = input("İşleminiz: ")
hesap = eval(veri)

print(hesap)
```

Bu programı çalıştırarak, “İşleminiz: “ ifadesinden sonra, örneğin, 45 * 76 yazıp Enter tuşuna basarsak programımız bize 3420 çıktısı verecektir. Yani programımız hesap makinesi işlevini yerine getirip 45 sayısı ile 76 sayısını çarpacaktır. Dolayısıyla, yukarıdaki programı kullanarak her türlü aritmetik işlemi yapabilirsiniz. Hatta bu program, son derece karmaşık aritmetik işlemlerin yapılmasına dahi müsaade eder.

Kazanım

- Öğrenci bu bölümde kullanıcıdan bilgi alma konularını öğrenir ve uygulamalarla öğrendiği konuyu pekiştirir.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Kullanıcıdan Bilgi Alma konusunun anlatılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere Kullanıcıdan Bilgi Alma konusunun anlatılması ve uygulamalarla pekiştirilmesi amaçlanmaktadır.
2	☑ Uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen eşliğinde editör üzerinde uygulamaların yapılması amaçlanmaktadır.

PYTHON: OPERATÖRLER

Müfredat Kodu / Ünite / Konu	İCT-301-1.9 / Python / Operatörler
Hedeflenen Süre	90 dakika

🚀 Operatörler

Bu bölümde, aslında pek de yabancısı olmadığımız ve hatta önceki derslerimizde üstünkörü de olsa değindiğimiz bir konuyu çok daha ayrıntılı bir şekilde ele alacağız. Burada anlatacağımız konu size yer yer sıkıcı gelebilir. Ancak bu konuyu hakkıyla öğrenmenizin, programcılık maceranız açısından hayati önemde olduğunu rahatlıkla söyleyebilirim.

Gelelim konumuza...

Bu bölümün konusu operatörler. Peki nedir bu 'operator' denen şey?

Sağında ve solunda bulunan değerler arasında bir ilişki kuran işaretlerdir. Bir operatörün sağında ve solunda bulunan değerlere ise işlenen (*oper and*) adı veriyoruz.

Biz bu bölümde operatörleri altı başlık altında inceleyeceğiz:

- Aritmetik Operatörler
- Karşılaştırma Operatörler
- Bool Operatörler
- Değer Atama Operatörler
- Aitlik Operatörler
- Kimlik Operatörler

Gördüğünüz gibi, işlememiz gereken konu çok, gitmemiz gereken yol uzun. O halde hiç vakit kaybetmeden, aritmetik operatörlerle yolculuğumuza başlayalım.

🔺 *Aritmetik İşleçler*

Dedik ki, sağında ve solunda bulunan değerler arasında bir ilişki kuran işaretlere operatör(operator) adı verilir. Aritmetik operatörler; matematikte kullanılan ve sayılarla aritmetik işlemler yapmamızı sağlayan yardımcı araçlardır. Dilerseniz bu tanımı bir örnekle somutlaştıralım:

```
>>> 45 + 33
78
```

Burada da 23 ve 46 değerleri birer işlenendir. Bu iki değer arasında yer alan * işareti ise, işlenenler arasında bir çarpma ilişkisi kuran bir operatördür. Ancak bir noktaya özellikle dikkatinizi çekmek istiyorum. Daha önceki derslerimizde de değindiğimiz gibi, + ve * operatörleri Python'da birden fazla anlama gelir. Örneğin yukarıdaki örnekte + operatörü, işlenenler arasında bir toplama ilişkisi kuruyor. Ama aşağıdaki durum biraz farklıdır:

```
>>> "google" + ".com"
'google.com'
```

Burada + operatörü işlenenler (“google” ve “.com”) arasında bir birleştirme ilişkisi kuruyor.

■ *% modülüs//taban bölme*

İlk önce modülüsün ne olduğunu ve ne işe yaradığını anlamaya çalışalım. Şu bölme işlemine bir bakın: Burada 02 sayısı bölme işleminin kalanıdır. İşte modülüs denen operatör de bölme işleminden kalan bu değeri gösterir. Yani:

```
>>> 30 % 4

2
```

Gördüğünüz gibi modülüs operatörü (%) gerçekten de bölme işleminden kalan sayıyı gösteriyor... Peki bu bilgi ne işimize yarar? Mesela bu bilgiyi kullanarak bir sayının tek mi yoksa çift mi olduğunu tespit edebiliriz:

```
sayı = int(input("Bir sayı girin: "))

if sayı % 2 == 0:
print("Girdiğiniz sayı bir çift sayıdır.")
else:
print("Girdiğiniz sayı bir tek sayıdır.")
```

■ *f bölünen % bölen == 0:*

Programımızın temelini bu kod oluşturuyor. Çünkü bir sayının bir sayıya tam bölünüp bölünmediğini bu kodla belirliyoruz. Eğer bir sayı başka bir sayıya bölündüğünde kalan değer, yani modülüs 0 ise, o sayı öbür sayıya tam bölünüyor demektir.Ayrıca bir sayının son basamağını elde etmek için de modülüsten yararlanabilirsiniz. Herhangi bir tamsayı 10’a bölündüğünde kalan (yani modülüs), bölünen sayının son basamağı olacaktır:

```
>>> 65 % 10

5

>>> 543 % 10

3
```

▲ *Karşılaştırma Operatörleri*

Adından da anlaşılacağı gibi, karşılaştırma operatörleri, işlenenler (*operands*) arasında bir karşılaştırma ilişkisi kuran operatörlerdir. Bu operatörleri şöyle sıralayabiliriz:

==	eşittir
!=	eşit değildir
>	büyüktür

<	küçüktür
>=	büyük eşittir
<=	küçük eşittir

Bu operatörlerin hiçbiri size yabancı değil, zira bunların hepsini aslında daha önceki derslerde verdiğimiz örneklerde kullanmıştık. Burada da bunlarla ilgili basit bir örnek vererek yolumuza devam edelim:

```
parola = "xyz05"

soru = input("parolanız: ")

if soru == parola:
    print("doğru parola!")

elif soru != parola:
    print("yanlış parola!")
```

Burada soru değişkeniyle kullanıcıdan alınan verinin, programın başında tanımladığımız parola değişkeninin değerine eşit olup olmadığını sorguluyoruz. Buna göre, eğer kullanıcıdan gelen veri parolayla eşleşiyorsa (if soru == parola), kullanıcıyı parolanın doğru olduğu konusunda bilgilendiriyoruz (print("doğru parola!")). Ama eğer kullanıcıdan gelen veri parolayla eşleşmiyorsa (elif soru != parola), o zaman da kullanıcıya parolanın yanlış olduğunu bildiriyoruz (print("yanlış parola!")).

Bool Operatörleri

Bu bölümde bool operatörlerinden söz edeceğiz, ancak bool operatörlerine geçmeden önce biraz bool kavramından bahsetmemiz yerinde olacaktır.

Nedir bu bool denen şey?

Bilgisayar bilimi iki adet değer üzerine kuruludur: 1 ve 0. Yani sırasıyla True ve False. Bilgisayar biliminde herhangi bir şeyin değeri ya True, ya da False'tur. İşte bu True ve False olarak ifade edilen değerlere bool değerleri adı verilir (George Boole adlı İngiliz matematikçi ve filozofun adından). Türkçe olarak söylemek gerekirse, True değerinin karşılığı Doğru, False değerinin karşılığı ise Yanlış'tır.

Örneğin:

```
>>> a = 1
```

Burada a adlı bir değişken tanımladık. Bu değişkenin değeri 1. Şimdi bu değişkenin değerini sorgulayalım:

```
>>> a == 1 #a değeri 1'e eşit mi?

True
```

Gördüğünüz gibi, a == 1 sorgusu True (Doğru) çıktısı veriyor. Çünkü a değişkeninin değeri gerçekten de 1. Bir de şunu deneyelim:

```
>>> a == 2

False
```

Burada da a değişkeninin değerinin 2 sayısına eşdeğer olup olmadığını sorguladık. a değişkeninin değeri 2 olmadığı için de Python bize False (Yanlış) çıktısı verdi.

Gördüğünüz gibi, bool operatörleri herhangi bir ifadenin doğruluğunu veya yanlışlığını sorgulamak için kullanılabiliyor. Buna göre, eğer bir sorgulamanın sonucu doğru ise True, eğer yanlış ise False çıktısı alıyoruz.

Değer Atama Operatörleri

Bu noktaya kadar yaptığımız çalışmalarda sadece tek bir değer atama operatörü gördük. Bu operatör = operatörüdür. Adından da anlaşılacağı gibi, bu operatörün görevi bir değişkene değer atamaktır. Mesela:

```
>>> a = 23
```

Burada = operatörü a değişkenine 23 değerini atama işlevi görüyor.

Python'daki tek değer atama operatörü elbette = değildir. Bunun dışında başka değer atama operatörleri de bulunur. Tek tek inceleyelim:

+= operatörü

Bu operatörün ne işe yaradığını anlamak için şöyle bir örnek düşünün:

```
>>> a = 23
```

a değerine mesela 5 ekleyip bu değeri 28'e eşitlemek için ne yapmamız lazım? Tabii ki şunu:

```
>>> a = a + 5
>>> print(a)

28
```

-= operatörü

Bir önceki += operatörü toplama işlemi yapıp, ortaya çıkan değeri tekrar aynı değişkene atıyordu. -= operatörü de buna benzer bir işlem gerçekleştirir:

```
>>> a = 23
>>> a -= 5
>>> print(a)

18
```

Yukarıdaki kullanım şununla tamamen aynıdır:

```
>>> a = 23
>>> a = a - 5
>>> print(a)

18
```

Ancak tıpkı += operatöründe olduğu gibi, -= operatörü de alternatifine göre daha hızlı çalışan bir araçtır.

/= operatörü

Bu operatörün çalışma mantığı da yukarıdaki operatörlerle aynıdır:

```
>>> a = 30
>>> a /= 3
>>> print(a)

10
```

Yukarıdaki işlem de şununla tamamen aynıdır:

```
>>> a = 30
>>> a = a / 3
>>> print(a)

10
```

***= operatörü**

Bu da ötekiler gibi, çarpma işlemi yapıp, bu işlemin sonucunu aynı değişkene atar:

```
>>> a = 20
>>> a *= 2
>>> print(a)

40
```

%= operatörü

Bu operatörümüz ise bölme işleminden kalan sayıyı aynı değişkene atar:

```
>>> a = 40
>>> a %= 3
```

```
>>> print(a)

1
```

**= operatörü

Bu operatörün ne yaptığını tahmin etmek zor değil. Bu operatörümüz, bir sayının kuvvetini hesapladıktan sonra çıkan değeri aynı değişkene atıyor:

```
>>> a = 12
>>> a **= 2
>>> print(a)

144
```

//= operatörü

Değer atama operatörlerinin sonuncusu olan //= operatörünün görevi ise taban bölme işleminin sonucunu aynı değişkene atamaktır:

```
>>> a = 5
>>> a //= 2
>>> print(a)

2
```

Aitlik Operatörleri

Aitlik operatörleri, bir karakter dizisi ya da sayının, herhangi bir veri tipi içinde bulunup bulunmadığını sorgulamamızı sağlayan operatörlerdir.

Python'da bir tane aitlik operatörü bulunur. Bu operatör de in operatörüdür. Bu operatörü şöyle kullanıyoruz:

```
>>> a = "abcd"
>>> "a" in a

True

>>> "f" in a

False
```

Gördüğünüz gibi, in adlı bu operatör, bir öğenin, veri tipi içinde bulunup bulunmadığını sorguluyor. Eğer bahsedilen öğe, veri tipi içinde geçiyorsa True çıktısı, eğer geçmiyorsa False çıktısı alıyoruz.

Henüz bu in operatörünü verimli bir şekilde kullanmamızı sağlayacak araçlardan yoksunuz. Ancak birkaç sayfa sonra öğreneceğimiz yeni araçlarla birlikte bu operatörü çok daha düzgün ve verimli bir şekilde kullanabilecek duruma geleceğiz.

Kimlik Operatörleri

Python'da her şeyin (ya da başka bir deyişle her nesnenin) bir kimlik numarası (*identity*) vardır. Kabaca söylemek gerekirse, bu kimlik numarası denen şey esasında o nesnenin bellekteki adresini gösterir.

Peki bir nesnenin kimlik numarasına nasıl ulaşırız?

Python'da bu işi yapmamızı sağlayacak id() adlı bir fonksiyon bulunur (İngilizcedeki *identity* (kimlik) kelimesinin kısaltması). Şimdi bir örnek üzerinde bu id() fonksiyonunu nasıl kullanacağımıza bakalım:

```
>>> a = 100
>>> id(a)

137990748
```

Çıktıda gördüğümüz 137990748 sayısı a değişkeninin tuttuğu 100 sayısının kimlik numarasını gösteriyor.

Kazanım

- Öğrenci bu bölümde Operatörler konusunu öğrenir ve uygulamalarla öğrendiği konuyu pekiştirir.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Operatörler konusunun anlatılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere Operatörler konusunun anlatılması ve uygulamalarla pekiştirilmesi amaçlanmaktadır.
2	☑ Uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen eşliğinde editör üzerinde uygulamaların yapılması amaçlanmaktadır.

PYTHON: IF-ELSE

Müfredat Kodu / Ünite / Konu	İCT-301-1.10.1 / Python Kontrol Yapıları / if-else
Hedeflenen Süre	45 dakika

🚀 Koşul Deyimleri

Python programlama dilinde koşullu durumları belirtmek için üç adet deyimden yararlanıyoruz:

- if
- elif
- else

İsterseniz önce if deyimi ile başlayalım...

🔺 *if Koşulu*

İngilizce bir kelime olan '*if*', Türkçede 'eğer' anlamına gelir. Anlamından da çıkarabileceğimiz gibi, bu kelime bir koşul bildiriyor. Yani '*eğer bir şey falanca ise...*' ya da '*eğer bir şey filanca ise...*' gibi... İşte biz Python'da bir koşula bağlamak istediğimiz durumları if deyimi aracılığıyla göstereceğiz.

Gelin isterseniz bu deyimi nasıl kullanacağımıza dair ufacık bir örnek vererek işe başlayalım:

Öncelikle elimizde şöyle bir değişken olsun:

```
n = 255
```

Yukarıda verdiğimiz değişkenin değerinin bir karakter dizisi değil, aksine bir sayı olduğunu görüyoruz. Şimdi bu değişkenin değerini sorgulayalım:

```
if n > 10:
```

Burada sayının 10'dan büyük olup olmadığına bakıyoruz.

Burada gördüğümüz > işaretinin ne demek olduğunu açıklamaya gerek yok sanırım. Hepimizin bildiği 'büyüktür' işareti Python'da da aynen bildiğimiz şekilde kullanılıyor. Mesela 'küçüktür' demek isteseydik, < işaretini kullanacaktık.

Burada değeri 26 olan a adlı bir değişken belirledik. Yani a değişkenine değer olarak 26 sayısını atadık. Ayrıca burada, değer atama işleminin ardından Enter tuşuna bastıktan sonra Python hiçbir şey yapmadan bir alt satıra geçti. Bir de şuna bakalım:

```
>>> a == 26

True
```

Burada ise yaptığımız şey a değişkeninin değerinin 26 olup olmadığını sorgulamak a == 26 komutunu verdikten sonra Python bize True diye bir çıktı verdi. Bu çıktının anlamını biraz sonra öğreneceğiz. Ama şimdi isterseniz konuyu daha fazla dağıtmayalım. Biz şimdilik sadece = ve == işaretlerinin birbirinden tamamen farklı anlamlara geldiğini bilelim yeter.

Ne diyorduk?

```
if n > 10:
```

Bu ifadeyle Python'a şöyle bir şey demiş oluyoruz:

"Eğer n sayısının değeri 10'dan büyükse..."

Burada kullandığımız işaretlere dikkat edin. En sonda bir adet : işaretinin olduğunu gözden kaçırmıyoruz. Bu tür işaretler Python için çok önemlidir. Bunları yazmayı unutursak Python gözümüzün yaşına bakmayacaktır.

Dedik ki, if n > 10: ifadesi, 'eğer n değişkeninin değeri 10'dan büyükse...' anlamına gelir. Bu ifadenin eksik olduğu apaçık ortada. Yani belli ki bu cümlenin bir de devamı olması gerekiyor. O halde biz de devamını getirelim:

```
if n > 10:
    print("sayı 10'dan büyüktür!")
```

Burada çok önemli bir durumla karşı karşıyayız. Dikkat ederseniz, ikinci satırı ilk satıra göre girintili yazdık. Elbette bunu şirinlik olsun diye yapmadık. Python programlama dilinde girintiler çok büyük önem taşır. Hatta ne kadarlık bir girinti verdiğiniz bile önemlidir. Eğer Python kodlarına duyarlı bir metin düzenleyici kullanıyorsanız, kullandığınız metin düzenleyici çoğu durumda sizin yerinize uygun bir şekilde girintilemeyi yapacaktır.

Daha önce de söylediğimiz gibi, if deyimi dışında Python'da koşullu durumları ifade etmek için kullandığımız, elif ve else adlı iki deyim daha vardır. Bunlar if ile birlikte kullanılırlar. Gelin isterseniz bu iki deyimden, adı elif olana bakalım.

elif Koşulu

Python'da, if deyimleriyle birlikte kullanılan ve yine koşul belirten bir başka deyim de elif deyimidir. Buna şöyle bir örnek verebiliriz:

```
yaş = int(input("Yaşınız: "))

if yaş == 18:
    print("18 iyidir!")

elif yaş < 0:
    print("Yok canım, daha neler!...")

elif yaş < 18:
    print("Genç bir kardeşimizsin!")

elif yaş > 18:
    print("Eh, artık yaş yavaş yavaş kemale eriyor!")
```

elif'i de incelediğimize göre, koşul bildiren deyimlerin sonuncusuna göz atabiliriz: else

else Koşulu

if ile elif arasındaki farkı biliyoruz. 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.

Ancak her durum için bir if bloğu yazmak bir süre sonra yorucu ve sıkıcı olacaktır. İşte bu noktada devreye else deyimi girecek. else'in anlamı kabaca şudur:

> "Eğer yukarıdaki koşulların hiçbiri gerçekleşmezse..."

Gelin isterseniz bununla ilgili şöyle bir örnek verelim:

```
soru = input("Bir meyve adı söyleyin bana:")

if soru == "elma":
    print("evet, elma bir meyvedir...")

elif soru == "karpuz":
    print("evet, karpuz bir meyvedir...")

elif soru == "armut":
    print("evet, armut bir meyvedir...")

else:
    print(soru, "gerçekten bir meyve midir?")
```

Eğer kullanıcı soruya 'elma', 'karpuz' veya 'armut' cevabı verirse, evet, ... bir meyvedir çıktısı verilecektir. Ama eğer kullanıcı bu üçü dışında bir cevap verirse, ... gerçekten bir meyve midir? çıktısını görürüz. Burada else deyimi, programımıza şu anlamı katıyor:

Eğer kullanıcı yukarıda belirlenen meyve adlarından hiç birini girmez, bunların yerine bambaşka bir şey yazarsa, o zaman else bloğu içinde belirtilen işlemi gerçekleştir.

Kazanım

- Öğrenciler bu bölümde if-else ve elif ile koşul durumlarını kavrar.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ if-else konusunun anlatılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere if-else konusunun anlatılması ve uygulamalarla pekiştirilmesi amaçlanmaktadır.
2	☑ Uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen eşliğinde editör üzerinde uygulamaların yapılması amaçlanmaktadır.

PYTHON: WHILE DÖNGÜSÜ

Müfredat Kodu / Ünite / Konu	İCT-301-1.10.2 / Python Kontrol Yapıları / While Döngüsü
Hedeflenen Süre	45 dakika

🚀 While Döngüsü

İngilizce bir kelime olan *while*, Türkçede '... iken, ... olduğu sürece' gibi anlamlara gelir. Python'da while bir döngüdür. Bir önceki bölümde söylediğimiz gibi, döngüler sayesinde programlarımızın sürekli olarak çalışmasını sağlayabiliriz.

Bu bölümde Python'da while döngüsünün ne olduğunu ve ne işe yaradığını anlamaya çalışacağız. Öncelikle while döngüsünün temellerini kavrayarak işe başlayalım.

Basit bir while döngüsü kabaca şuna benzer:

```
a = 1

while a == 1:
```

Burada a adlı bir değişken oluşturduk. Bu değişkenin değeri 1, bir sonraki satırda ise while a == 1: gibi bir ifade yazdık. En başta da söylediğimiz gibi *while* kelimesi, '... iken, olduğu sürece' gibi anlamlar taşıyor. Python programlama dilindeki anlamı da buna oldukça yakındır. Burada while a == 1 ifadesi programımıza şöyle bir anlam katıyor:

"a değişkeninin değeri 1 olduğu sürece..."

Gördüğünüz gibi cümlemiz henüz eksik. Yani belli ki bunun bir de devamı olacak. Ayrıca while ifadesinin sonundaki : işaretinden anladığımız gibi, bundan sonra gelecek satır girintili yazılacak. Devam edelim:

```
a = 1

while a == 1:
    print("bilgisayar çıldırdı!")
```

Burada Python'a şu emri vermiş olduk:

- a değişkeninin değeri 1 olduğu sürece, ekrana 'bilgisayar çıldırdı!' yazısını dök!

Bu programı çalıştırdığımızda Python verdiğimiz emre sadakatle uyacak ve a değişkeninin değeri 1 olduğu müddetçe de bilgisayarımızın ekranına 'bilgisayar çıldırdı!' yazısını dökecektir. Programımızın içinde a değişkeninin değeri 1 olduğu ve bu değişkenin değerini değiştirecek herhangi bir şey bulunmadığı için Python hiç sıkılmadan ekrana 'bilgisayar çıldırdı!' yazısını basmaya devam edecektir. Eğer siz durdurmazsanız bu durum sonsuza kadar devam edebilir. Bu çılgınlığa bir son vermek için klavyenizde Ctrl+C veya Ctrl+Z tuşlarına basarak programı durmaya zorlayabilirsiniz.

Burada programımızı sonsuz bir döngüye sokmuş olduk (*infinite loop*). Esasında sonsuz döngüler genellikle bir program hatasına işaret eder. Yani çoğu durumda programcının arzu ettiği şey bu değildir. O yüzden doğru yaklaşım, döngüye soktuğumuz programlarımızı durduracak bir ölçüt belirlemektir. Yani öyle bir kod yazmalıyız ki, a değişkeninin 1 olan değeri bir noktadan sonra artık 1 olmasın ve böylece o noktaya

ulaşıldığında programımız dursun. Kullanıcının Ctrl+C tuşlarına basarak programı durdurmak zorunda kalması pek hoş olmuyor. Gelin isterseniz bu soyut ifadeleri biraz somutlaştıralım.

Öncelikle şu satırı yazarak işe başlıyoruz:

```
a = 1
```

Burada normal bir şekilde a değişkenine 1 değerini atadık. Şimdi devam ediyoruz:

```
a = 1

while a < 10:
```

while ile verdiğimiz ilk örnekte while a == 1 gibi bir ifade kullanmıştık. Bu ifade;

"a'nın değeri 1 olduğu müddetçe..."

gibi bir anlama geliyordu.

while a < 10 ifadesi ise;

"a'nın değeri 10'dan küçük olduğu müddetçe..."

anlamına gelir. İşte burada programımızın sonsuz döngüye girmesini engelleyecek bir ölçüt koymuş olduk. Buna göre, a değişkeninin şimdiki değeri 1'dir. Biz, a'nın değeri 10'dan küçük olduğu müddetçe bir işlem yapacağız. Devam edelim:

```
a = 1

while a < 10:
    print("bilgisayar yine çıldırdı!")
```

Ne oldu? İstediğimizi elde edemedik, değil mi? Programımız yine sonsuz döngüye girdi. Bu sonsuz döngüyü kırmak için Ctrl+C (veya Ctrl+Z)'ye basmamız gerekecek yine...

Sizce buradaki hata nereden kaynaklandı? Yani neyi eksik yaptık da programımız sonsuz döngüye girmekten kurtulamadı? Aslında bunun cevabı çok basit. Biz yukarıdaki kodları yazarak Python'a şu emri vermiş olduk:

- a'nın değeri 10'dan küçük olduğu müddetçe ekrana 'bilgisayar yine çıldırdı!' yazısını bas!

a değişkeninin değeri 1. Yani 10'dan küçük. Dolayısıyla Python'ın ekrana o çıktıyı basmasını engelleyecek herhangi bir şey yok...

Şimdi bu problemi nasıl aşacağımızı görelim:

```
a = 1
```

```
while a < 10:
    a += 1
    print("bilgisayar yine çıldırdı!")
```

Burada a += 1 satırını ekledik kodlarımızın arasına. += işlecini anlatırken söylediğimiz gibi, bu satır, a değişkeninin değerine her defasında 1 ekliyor ve elde edilen sonucu tekrar a değişkenine atıyor. En sonunda a'nın değeri 10'a ulaşınca da, Python ekrana 'bilgisayar yine çıldırdı!' cümlesini yazmayı bırakıyor. Çünkü while döngüsü içinde belirttiğimiz ölçüte göre, programımızın devam edebilmesi için a değişkeninin değerinin 10'dan küçük olması gerekiyor. a'nın değeri 10'a ulaştığı anda bu ölçüt bozulacaktır.

Kazanım

- Öğrenci bu bölümde Döngüler konu başlığı altındaki While Döngüsünü, döngünün mantığını ve amaçlarını kavrar.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ While Döngüsü konusunun anlatılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere While Döngüsü konusunun anlatılması ve uygulamalarla pekiştirilmesi amaçlanmaktadır.
2	☐ Uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen eşliğinde editör üzerinde uygulamaların yapılması amaçlanmaktadır.

PYTHON: FOR DÖNGÜSÜ

Müfredat Kodu / Ünite / Konu	İCT-301-1.10.3 / Python Kontrol Yapıları / For Döngüsü
Hedeflenen Süre	45 dakika

🚀 For Döngüsü

Etrafta yazılmış Python programlarının kaynak kodlarını incelediğinizde, içinde for döngüsü geçmeyen bir program kolay kolay bulamazsınız. Belki while döngüsünün kullanılmadığı programlar vardır. Ancak for döngüsü Python'da o kadar yaygındır ve o kadar geniş bir kullanım alanına sahiptir ki, hemen hemen bütün Python programları bu for döngüsünden en az bir kez yararlanır.

Peki nedir bu for döngüsü denen şey?

for da tıpkı while gibi bir döngüdür. Yani tıpkı while döngüsünde olduğu gibi, programlarımızın birden fazla sayıda çalışmasını sağlar. Ancak for döngüsü while döngüsüne göre biraz daha yeteneklidir. while döngüsü ile yapamayacağınız veya yaparken çok zorlanacağınız şeyleri for döngüsü yardımıyla çok kolay bir şekilde halledebilirsiniz.

Yalnız, söylediğimiz bu cümleden, for döngüsünün while döngüsüne bir alternatif olduğu sonucunu çıkarmayın. Evet, while ile yapabildiğiniz bir işlemi for ile de yapabilirsiniz çoğu zaman, ama bu döngülerin, belli vakalar için tek seçenek olduğu durumlar da vardır. Zira bu iki döngünün çalışma mantığı birbirinden farklıdır.

Şimdi gelelim for döngüsünün nasıl kullanılacağına...

Dikkatlice bakın:

```
tr_harfler = "şçöğüİı"

for harf in tr_harfler:
    print(harf)
```

Burada öncelikle tr_harfler adlı bir değişken tanımladık. Bu değişken Türkçeye özgü harfleri tutuyor. Daha sonra bir for döngüsü kurarak, tr_harfler adlı değişkenin her bir öğesini tek tek ekrana yazdırdık.

Peki bu for döngüsünü nasıl kurduk?

for döngülerinin söz dizimi şöyledir:

```
for değişken_adı in değişken:
    yapılacak_işlem
```

Bu soyut yapıları kendi örneğimize uygulayarak durumu daha net anlamaya çalışalım:

```
tr_harfler adlı değişken içindeki her bir öğeyi harf olarak adlandır:
    ve harf olarak adlandırılan bu öğeleri ekrana yazdır.
```

Yukarıdaki örnekte bir for döngüsü yardımıyla tr_harfler adlı değişken içindeki her bir öğeyi ekrana yazdırdık. Esasında for döngüsünün yeteneklerini düşündüğümüzde bu örnek pek heyecan verici değil. Zira aynı işi aslında print() fonksiyonu ile de yapabilirdik:

```
tr_harfler = "şçöğüİı"
print(*tr_harfler, sep="\n")
```

Aslında bu işlemi while ile de yapmak mümkün (Bu kodlardaki, henüz öğrenmediğimiz kısmı şimdilik görmezden gelin):

```
tr_harfler = "şçöğüİı"
a = 0

while a < len(tr_harfler):
    print(tr_harfler[a], sep="\n")
    a += 1
```

while döngüsü kullanıldığında işi uzattığımızı görüyorsunuz. Dediğimiz gibi, for döngüsü while döngüsüne göre biraz daha yeteneklidir ve while ile yapması daha zor (veya uzun) olan işlemleri for döngüsü ile çok daha kolay bir şekilde yapabiliriz. Ayrıca for döngüsü ile while döngüsünün çalışma mantıkları birbirinden farklıdır. for döngüsü, üzerinde döngü kurulabilecek veri tiplerinin her bir öğesinin üzerinden tek tek geçer ve bu öğelerin her biri üzerinde bir işlem yapar. while döngüsü ise herhangi bir ifadenin bool değerini kontrol eder ve bu değerin bool değeri False olana kadar, belirlenen işlemi yapmayı sürdürür.

Kazanım

- Öğrenci bu bölümde Döngüler konu başlığı altındaki For Döngüsünü, döngünün mantığını ve amaçlarını kavrar.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ For döngüsü konusunun anlatılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere For döngüsü konusunun anlatılması ve uygulamalarla pekiştirilmesi amaçlanmaktadır.
2	☐ Uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen eşliğinde editör üzerinde uygulamaların yapılması amaçlanmaktadır.

PYTHON: HATA YAKALAMA

Müfredat Kodu / Ünite / Konu	İCT-301-1.11 / Python / Hata Yakalama
Hedeflenen Süre	45 dakika

🚀 Hata Türleri

Biz bu bölümde hataları üç farklı başlık altında ele alacağız:

1. Programcı Hataları (*Error*)
2. Program Kusurları (*Bug*)
3. İstisnalar (*Exception*)

Öncelikle programcı hatalarından bahsedelim.

Programcıdan kaynaklanan hatalar doğrudan doğruya programı yazan kişinin dikkatsizliğinden ötürü ortaya çıkan bariz hatalardır. Örneğin şu kod bir programcı hatası içerir:

```
>>> print "Merhaba Python!"
```

Bu kodu çalıştırdığınızda şöyle bir hata mesajı görürsünüz:

```
>>> print "Merhaba Python!"

File "<stdin>", line 1
  print "Merhaba Python!"
                 ^
SyntaxError: invalid syntax
```

Bu hata mesajında bizi ilgilendiren kısım son cümlede yer alıyor: SyntaxError, yani Söz dizimi hatası.

🔺 *try... except...*

Python'da hata yakalama işlemleri için try... except... bloklarından yararlanılır. Hemen bir örnek verelim:

```
ilk_sayı    = input("ilk sayı: ")
ikinci_sayı = input("ikinci sayı: ")

try:
   sayı1 = int(ilk_sayı)
   sayı2 = int(ikinci_sayı)
   print(sayı1, "/", sayı2, "=", sayı1 / sayı2)
except ValueError:
   print("Lütfen sadece sayı girin!")
```

Biliyoruz ki, bir veriyi sayıya dönüştürmek istediğimizde eğer kullanıcı sayı değerli bir veri yerine harf değerli bir veri girerse programımız çöker. Dolayısıyla int(ilk_sayı) ve int(ikinci_sayı) kodları, kullanıcının gireceği veri türüne göre hata üretme potansiyeline sahiptir. O yüzden, burada hata vereceğini bildiğimiz o kodları try bloğu içine aldık.

Yine bildiğimiz gibi, veri dönüştürme işlemi sırasında kullanıcının uygun olmayan bir veri girmesi halinde üretilecek hata bir ValueError'dır. Dolayısıyla except bloğu içine yazacağımız hata türünün adı da ValueError olacaktır. O yüzden ValueError adlı hatayı yakalayabilmek için şu satırları yazdık:

```
except ValueError:
    print("Lütfen sadece sayı girin!")
```

try... except... as...

Bildiğiniz gibi, Python bir programın çalışması esnasında hata üretirken çıktıda hata türünün adıyla birlikte kısa bir hata açıklaması veriyor. Yani mesela şöyle bir çıktı üretiyor:

```
ValueError: invalid literal for int() with base 10: 'f'
```

Burada 'ValueError' hata türünün adı, 'invalid literal for int() with base 10: 'f'' ise hatanın açıklamasıdır. Eğer istersek, yazdığımız programda bu hata açıklamasına erişebiliriz. Dikkatlice bakın:

```
ilk_sayı    = input("ilk sayı: ")
ikinci_sayı = input("ikinci sayı: ")

try:
    sayı1 = int(ilk_sayı)
    sayı2 = int(ikinci_sayı)
    print(sayı1, "/", sayı2, "=", sayı1 / sayı2)
except ValueError as hata:
    print(hata)
```

Bu programı çalıştırıp sayı değerli olmayan bir veri girersek hata çıktısı şöyle olacaktır:

```
invalid literal for int() with base 10: 'f'
```

try... except... else...

Daha önce de dediğimiz gibi, Python'da hata yakalama işlemleri için çoğunlukla try... except... bloklarını bilmek yeterli olacaktır. İşlerimizin büyük kısmını sadece bu blokları kullanarak halledebiliriz. Ancak Python bize bu konuda, zaman zaman işimize yarayabilecek başka araçlar da sunmaktadır. İşte try... except... else... blokları da bu araçlardan biridir. Bu bölümde kısaca bu blokların ne işe yaradığından söz edeceğiz.

Öncelikle try... except... else... bloğunun ne işe yaradığına bakalım. Esasında biz bu else deyimini daha önce de 'koşullu ifadeler' konusunu işlerken görmüştük. Buradaki kullanımı da zaten hemen hemen aynıdır. Diyelim ki elimizde şöyle bir şey var:

```
try:
    bölünen = int(input("bölünecek sayı: "))
    bölen = int(input("bölen sayı: "))
    print(bölünen/bölen)
except ValueError:
    print("hata!")
```

Burada eğer kullanıcı sayı yerine harf girerse ValueError hatası alırız. Bu hatayı except ValueError: ifadesiyle yakalıyoruz ve hata verildiğinde kullanıcıya bir mesaj göstererek programımızın çökmesini engelliyoruz. Ama biliyoruz ki, bu kodları çalıştırdığımızda Python'ın verebileceği tek hata ValueError değildir. Eğer kullanıcı bir sayıyı 0'a bölmeye çalışırsa Python ZeroDivisionError adlı hatayı verecektir. Dolayısıyla bu hatayı da yakalamak için şöyle bir şey yazabiliriz:

```
try:
    bölünen = int(input("bölünecek sayı: "))
    bölen = int(input("bölen sayı: "))
    print(bölünen/bölen)
except ValueError:
    print("Lütfen sadece sayı girin!")
except ZeroDivisionError:
    print("Bir sayıyı 0'a bölemezsiniz!")
```

Bu şekilde hem ValueError hatasını hem de ZeroDivisionError hatasını yakalamış oluruz. Bu kodların özelliği, except... bloklarının tek bir try... bloğunu temel almasıdır. Yani biz burada bütün kodlarımızı tek bir try... bloğu içine tıkıştırıyoruz. Bu blok içinde gerçekleşen hataları da daha sonra tek tek except... blokları yardımıyla yakalıyoruz.

try... except... finally...

try... except... else... yapılarının dışında, Python'ın bize sunduğu bir başka yapı da try... except... finally... yapılarıdır. Genel olarak Python'da dosyalarla çalışabilmek için öncelikle bilgisayarda bulunan bir dosyayı okuma veya yazma kipinde açarız. Dosyayı açtıktan sonra bu dosyayla ihtiyacımız olan birtakım işlemler gerçekleştiririz. Dosyayla işimiz bittikten sonra ise dosyamızı mutlaka kapatmamız gerekir. Ancak eğer dosya üzerinde işlem yapılırken bir hata ile karşılaşılırsa dosyamızı kapatma işlemini gerçekleştirdiğimiz bölüme hiç ulaşılamayabilir. İşte finally... bloğu böyle bir durumda işimize yarayacaktır:

```
try:
    dosya = open("dosyaadı", "r")
    ...burada dosyayla bazı işlemler yapıyoruz...
    ...ve ansızın bir hata oluşuyor...
except IOError:
    print("bir hata oluştu!")
finally:
    dosya.close()
```

Burada finally... bloğu içine yazdığımız dosya.close() ifadesi dosyamızı güvenli bir şekilde kapatmaya yarıyor. Bu blok, yazdığımız program hata verse de vermese de işletilecektir.

Kazanım

- Öğrenci bu bölümde hata yakalama yöntemlerini öğrenir ve uygulamalarla pekiştirir.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Hata Yakalama konusunun anlatılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere Hata Yakalama konusunun anlatılması ve uygulamalarla pekiştirilmesi amaçlanmaktadır.
2	☑ Uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen eşliğinde editör üzerinde uygulamaların yapılması amaçlanmaktadır.

PYTHON: FONKSIYONLAR

Müfredat Kodu / Ünite / Konu	İCT-301-1.12 / Python / Fonksiyonlar
Hedeflenen Süre	180 dakika

🚀 Fonksiyonlar

Bu bölümde amacımız fonksiyonları enine boyuna inceleyerek, okurun bilgi dağarcığında fonksiyonlara ilişkin sağlam bir altyapı oluşturmaktır. Okur, bu bölümü bitirdikten sonra fonksiyonlara ilişkin olarak bilmesi gereken her şeyi öğrenmiş olacak.

✍️ Fonksiyon Tanımlamak ve Çağırmak

kayıt_oluştur() adlı fonksiyon için bu ada sahip fonksiyonu aşağıdaki şekilde tanımlamalıyız:

```
def kayıt_oluştur(isim, soyisim, işsis, şehir):
    print("-"*30)

    print("isim          : ", isim)
    print("soyisim       : ", soyisim)
    print("işletim sistemi: ", işsis)
    print("şehir         : ", şehir)

    print("-"*30)
```

İlk bakışta bu kodlar size hiçbir şey ifade etmemiş olabilir. Ama hiç endişe etmeyin. Biz birazdan bu satırların ne anlama geldiğini bütün ayrıntılarıyla anlatacağız. Siz şimdilik anlamadığınız kısımları görmezden gelip okumaya devam edin.

Yukarıdaki kodlar yardımıyla fonksiyonumuzu tanımlamış olduk. Artık elimizde, tıpkı print() veya input() gibi, kayıt_oluştur() adlı 'ev yapımı' bir fonksiyon var. Dolayısıyla bu yeni fonksiyonumuzu, daha önce öğrendiğimiz fonksiyonları nasıl kullanıyorsak aynı şekilde kullanabiliriz. Yani aşağıdaki gibi komutlar yazabiliriz:

Yalnız fonksiyonumuzu tanımlayıp bitirdikten sonra, bu fonksiyonu kullanırken, kodlarımızın hizalamasına dikkat ediyoruz. Fonksiyonu kullanmak için yazdığımız kodları def ifadesinin hizasına getiriyoruz. Yani:

```
def kayıt_oluştur(isim, soyisim, işsis, şehir):
    print("-"*30)

    print("isim          : ", isim)
    print("soyisim       : ", soyisim)
    print("işletim sistemi: ", işsis)
    print("şehir         : ", şehir)

    print("-"*30)

kayıt_oluştur("Fırat", "Özgül", "Ubuntu", "İstanbul")
kayıt_oluştur("Mehmet", "Öztaban", "Debian", "Ankara")
```

Parametreler ve Argümanlar

Şimdiye kadar yaptığımız örnekler sayesinde aslında parametrelerin neye benzediğini ve ne işe yaradığını öğrenmiştik. Bu bölümde ise sizi 'argüman' adlı bir kavramla tanıştırıp, argüman ile parametre arasındaki benzerlik ve farklılıkları inceleyeceğiz. Bunun yanı sıra, parametre kavramını da bu bölümde daha derinlikli bir şekilde ele alacağız.

O halde hemen yola koyulalım.

Parametrenin ne olduğunu biliyorsunuz. Bunlar fonksiyon tanımlarken parantez içinde belirttiğimiz, fonksiyon gövdesinde yapılan işin değişken öğelerini gösteren parçalardır. Mesela:

```
def kopyala(kaynak_dosya, hedef_dizin):
    çıktı = "{} adlı dosya {} adlı dizin içine kopyalandı!"
    print(çıktı.format(kaynak_dosya, hedef_dizin))
```

Burada kopyala() adlı bir fonksiyon tanımladık. Bu fonksiyon toplam iki adet parametre alıyor: kaynak_dosya ve hedef_dizin. Gördüğünüz gibi, bu iki parametre gerçekten de fonksiyon gövdesinde yapılan işin değişken öğelerini gösteriyor. Bu fonksiyonun üreteceği çıktı, fonksiyonu çağıran kişinin bu iki parametreye vereceği değerlere bağlı olarak şekillenecek.

Sıralı (veya İsimsiz) Parametreler

Python'da şöyle bir fonksiyon tanımlayabileceğimizi biliyoruz:

```
def kayıt_oluştur(isim, soyisim, işsis, şehir):
    print("-"*30)

    print("isim          : ", isim)
    print("soyisim       : ", soyisim)
    print("işletim sistemi: ", işsis)
    print("şehir         : ", şehir)

    print("-"*30)
```

Yukarıda tanımladığımız bu fonksiyonu şu şekilde çağırabiliriz:

```
kayıt_oluştur("Ahmet", "Öz", "Debian", "Ankara")
```

Bu fonksiyonda, yazdığımız parametrelerin sırası büyük önem taşır. Mesela yukarıdaki fonksiyonu şöyle çağırdığımızı düşünün:

```
kayıt_oluştur("Debian", "Ankara", "Öz", "Ahmet")
```

Eğer fonksiyon parametrelerini bu sırayla kullanırsak aldığımız çıktı hatalı olacaktır:

```
------------------------------
isim          :  Debian
soyisim       :  Ankara
işletim sistemi:  Öz
şehir         :  Ahmet
------------------------------
```

Gördüğünüz gibi, isim, soyisim ve öteki bilgiler birbirine karışmış. İşte Python'da, veriliş sırası önem taşıyan bu tür parametrelere 'sıralı parametreler' (veya isimsiz parametreler) adı verilir.

İsimli Parametreler

Bir önceki bölümde verdiğimiz şu örneği yeniden ele alalım:

```
def kayıt_oluştur(isim, soyisim, işsis, şehir):
    print("-"*30)

    print("isim          : ", isim)
    print("soyisim       : ", soyisim)
    print("işletim sistemi: ", işsis)
    print("şehir         : ", şehir)

    print("-"*30)
```

Bu fonksiyonu çağırırken parametrelerin sırasını doğru vermenin, alacağımız çıktının düzgün olması bakımından büyük önem taşıdığını biliyoruz. Ancak özellikle parametre sayısının çok olduğu fonksiyonlarda parametre sırasını akılda tutmak zor olabilir. Böyle durumlarda parametreleri isimleri ile birlikte kullanmayı tercih edebiliriz:

```
kayıt_oluştur(soyisim="Öz", isim="Ahmet", işsis="Debian", şehir= "Ankara")
```

Böylece fonksiyon parametrelerini istediğimiz sıra ile kullanabiliriz. Ancak burada dikkat etmemiz gereken bazı noktalar var. Python'da isimli bir parametrenin ardından sıralı bir parametre gelemez. Yani şu kullanım yanlıştır:

```
kayıt_oluştur(soyisim="Öz", isim="Ahmet", "Debian", "Ankara")
```

Bu kodlar bize şu hatayı verir:

```
  File "<stdin>", line 1
SyntaxError: non-keyword arg after keyword arg
```

Bu yüzden, eğer isimli parametreler kullanacaksak, isimli parametrelerden sonra sıralı parametre kullanmamaya dikkat ediyoruz.

✍️ return Deyimi

Bu bölümde return adlı bir deyimden söz edeceğiz. Özellikle Python programlama dilini öğrenmeye yeni başlayanlar bu deyimin ne işe yaradığını anlamakta zorlanabiliyor. Biz burada bu deyimi anlaşılır hale getirebilmek için elimizden geleni yapacağız. Öncelikle çok basit bir örnek verelim:

```
def ismin_ne():
    isim = input("ismin ne? ")
    print(isim)
```

Bu çok basit bir fonksiyon. Bu fonksiyonu nasıl çağıracağımızı biliyoruz:

```
ismin_ne()
```

Fonksiyonu bu şekilde çağırdıktan sonra, fonksiyon tanımında yer alan input() fonksiyonu sayesinde kullanıcıya ismi sorulacak ve verdiği cevap ekrana basılacaktır.

Yukarıdaki fonksiyonun tek işlevi kullanıcıdan aldığı isim bilgisini ekrana basmaktır. Aldığınız bu veriyi başka yerlerde kullanamazsınız. Bu fonksiyonu çağırdığınız anda kullanıcıya ismi sorulacak ve alınan cevap ekrana basılacaktır. Ancak siz, tanımladığınız fonksiyonların tek görevinin bir veriyi ekrana basmak olmasını istemeyebilirsiniz.

Örneğin yukarıdaki fonksiyon yardımıyla kullanıcıdan ismini aldıktan sonra, bu isim bilgisini başka bir karakter dizisi içinde kullanmak isteyebilirsiniz. Diyelim ki amacınız ismin_ne() fonksiyonuyla aldığınız ismi şu karakter dizisi içine aşağıdaki şekilde yerleştirmek:

```
Merhaba Fırat. Nasılsın?
```

Bildiğimiz yöntemi kullanarak bu amacımızı gerçekleştirmeye çalışalım:

```
print("Merhaba {}. Nasılsın?".format(ismin_ne()))
```

Buradan şöyle bir çıktı alıyoruz:

```
ismin ne? Fırat
Fırat
Merhaba None. Nasılsın?
```

Gördüğünüz gibi, istediğimiz şeyi elde edemiyoruz. Çünkü dediğimiz gibi, yukarıdaki fonksiyonun tek görevi kullanıcıdan aldığı çıktıyı ekrana basmaktır. Bu fonksiyondan gelen çıktıyı başka yerde kullanamayız. Eğer kullanmaya çalışırsak yukarıdaki gibi hiç beklemediğimiz bir sonuç alırız.

Kazanım

- Öğrenci bu bölümde fonksiyonları, parametre ve argümanları öğrenir.
- Uygulamalar ile birlikte konuyu pekiştirir.

✅ Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Fonksiyonlar konusunun anlatılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere Fonksiyonlar konusunun anlatılması ve uygulamalarla pekiştirilmesi amaçlanmaktadır.
2	☑ Uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen eşliğinde editör üzerinde uygulamaların yapılması amaçlanmaktadır.

PYTHON: GÖMÜLÜ FONKSIYONLAR

Müfredat Kodu / Ünite / Konu	İCT-301-1.12.1 / Python Fonksiyonlar / Gömülü Fonksiyonlar
Hedeflenen Süre	180 dakika

🚀 Gömülü Fonksiyonlar

Gömülü fonksiyonlar İngilizcede *builtin functions* olarak adlandırılır. Bu fonksiyonlar gerçekten de dile gömülü vaziyettedirler. Bildiğiniz gibi, bir fonksiyonu kullanabilmemiz için o fonksiyonu tanımlamamız gerekir. İşte gömülü fonksiyonlar, bizim tanımlamamıza gerek kalmadan, Python geliştiricileri tarafından önceden tanımlanıp dile gömülmüş ve hizmetimize sunulmuş faydalı birtakım araçlardır.

Bu bölümde elbette birtakım fonksiyonları salt art arda sıralamakla yetinmeyeceğiz. Python'daki gömülü fonksiyonları incelerken bir yandan da Python programlama dilindeki çok önemli bazı kavramları ele alacağız.

İlk olarak abs() adlı bir fonksiyonla başlıyoruz gömülü fonksiyonları incelemeye...

🔺 *abs() Fonksiyonu*

İngilizcede 'mutlak' anlamına gelen *absolute* adlı bir kelime bulunur. İşte bu fonksiyonun adı da bu kelimeden gelir. Fonksiyonumuzun görevi de isminin anlamına yakındır. abs() fonksiyonunu bir sayının mutlak değerini elde etmek için kullanıyoruz.

Peki 'mutlak değer' ne anlama geliyor. Esasında siz bu kavrama matematik derslerinden aşinasınız. Ama bilmeyenler veya unutmuş olanlar için tekrar edelim. 'Mutlak değer' bir sayının 0'a olan uzaklığıdır. Örneğin 20 sayısının 0 sayısına olan uzaklığı 20'dir. Dolayısıyla 20 sayısının mutlak değeri 20'dir. Aynı şekilde -20 sayısının da 0 sayısına uzaklığı 20'dir. Yani, -20 sayısının da mutlak değeri 20'dir.

İşte abs() fonksiyonu bize bir sayının mutlak değerinin ne olduğunu söyler:

```
>>> abs(-20)
20
>>> abs(20)
20
>>> abs(20.0)
20.0
```

🔺 *round() Fonksiyonu*

round() fonksiyonu bir sayıyı belli ölçütlere göre yukarı veya aşağı doğru yuvarlamamızı sağlar. Basit birkaç örnek verelim:

```
>>> round(12.4)
12
>>> round(12.7)
```

```
13
```

Gördüğünüz gibi bu fonksiyon, kayan noktalı sayıları en yakın tam sayıya doğru yuvarlıyor.

Ancak burada dikkat etmemiz gereken bir nokta var.

Şu örnekleri bir inceleyelim:

```
>>> round(1.5)
2
>>> round(12.5)
12
```

Gördüğünüz gibi, fonksiyonumuz 1.5 sayısını yukarı doğru, 12.5 sayısını ise aşağı doğru yuvarladı. Bunun sebebi, kayan noktalı bir sayının üst ve alt tam sayılara olan uzaklığının birbirine eşit olduğu durumlarda Python'ın çift sayıya doğru yuvarlama yapmayı tercih etmesidir. Mesela yukarıdaki örneklerde 1.5 sayısı hem 1 sayısına, hem de 2 sayısına eşit uzaklıkta bulunuyor. İşte Python bu durumda, bir çift sayı olan 2 sayısına doğru yuvarlamayı tercih edecektir.

▲ *all() Fonksiyonu*

All kelimesi Türkçede 'hepsi' anlamına gelir. Bu fonksiyonun görevi de bu anlamı çağrıştırır. all() fonksiyonunun görevi, bir dizi içinde bulunan bütün değerler True ise True değeri, eğer bu değerlerden herhangi biri False ise de False değeri döndürmektir.

Örneğin elimizde şöyle bir liste olduğunu varsayalım:

```
>>> liste = [1, 2, 3, 4]
```

Şimdi all() fonksiyonunu bu liste üzerine uygulayalım:

```
>>> all(liste)
True
```

Bildiğiniz gibi, 0 hariç bütün sayıların bool değeri True'dur. Yukarıdaki listede False değeri verebilecek herhangi bir değer bulunmadığından, all() fonksiyonu bu liste için True değerini veriyor.

▲ *any() Fonksiyonu*

Any kelimesi İngilizcede 'herhangi bir' anlamına gelir. İşte any() fonksiyonunun görevi de, bir dizi içindeki bütün değerlerden en az biri True ise True çıktısı vermektir.

Örneğin:

```
>>> liste = ['ahmet', 'mehmet', '']
>>> any(liste)

True
```

any() fonksiyonunun True çıktısı verebilmesi için listede yalnızca bir adet True değerli öğe olması yeterlidir. Bu fonksiyonun False çıktısı verebilmesi için dizi içindeki bütün öğelerin bool değerinin False olması gerekir.

Tıpkı all() fonksiyonunda olduğu gibi, any() fonksiyonunu da, bir grup nesnenin bool değerlerini denetlemek amacıyla kullanabilirsiniz.

ascii() Fonksiyonu

Bu fonksiyon, bir nesnenin ekrana basılabilir halini verir bize. Dilerseniz bu fonksiyonun yaptığı işi tanımlamak yerine bunu bir örnek üzerinden anlatmaya çalışalım:

```
>>> a = 'istihza'
>>> print(ascii(a))

'istihza'
```

Bu fonksiyonun, print() fonksiyonundan farklı olarak, çıktıya tırnak işaretlerini de eklediğine dikkat edin.

bool() Fonksiyonu

Bu fonksiyon bir nesnenin bool değerini verir:

```
>>> bool(0)

False

>>> bool(1)

True

>>> bool([])

False
```

bin() Fonksiyonu

Bu fonksiyon, bir sayının ikili düzendeki karşılığını verir:

```
>>> bin(12)

'0b1100'
```

Bu fonksiyonun verdiği çıktının bir sayı değil, karakter dizisi olduğuna dikkat etmelisiniz.

bytes() Fonksiyonu

Bu fonksiyon bytes türünde nesneler oluşturmak için kullanılır. Bu fonksiyonu 'bayt' adlı veri tipini incelerken ayrıntılı olarak ele almıştık. Gelin isterseniz burada da bu fonksiyona şöyle bir değinelim.

Dediğimiz gibi, bytes() adlı fonksiyon, bytes türünde veriler oluşturmaya yarar. Bu fonksiyon işlev olarak, daha önce öğrendiğimiz list(), str(), int() , set(), dict() gibi fonksiyonlara çok benzer. Tıpkı bu fonksiyonlar gibi, bytes() fonksiyonunun görevi de farklı veri tiplerini 'bayt' adlı veri tipine dönüştürmektir.

Bu fonksiyon, kendisine verilen parametrelerin türüne bağlı olarak birbirinden farklı sonuçlar ortaya çıkarır. Örneğin eğer bu fonksiyona parametre olarak bir tam sayı verecek olursanız, bu fonksiyon size o tam sayı miktarınca bir bayt nesnesi verecektir. Gelin isterseniz bu durumu örnekler üzerinde göstermeye çalışalım:

```
>>> bytes(10)
b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
```

Yukarıdaki komut bize, her bir öğesinin değeri 0 olan 10 baytlık bir veri döndürdü:

```
>>> a = bytes(10)
>>> a
b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
>>> a[0]
0
>>> a[1]
0
>>> a[2]
0
```

Gördüğünüz gibi, bytes(10) komutuyla oluşturduğumuz a değişkeni içinde toplam 10 adet bayt var ve bu baytların her birinin değeri 0.

list() Fonksiyonu

Bu fonksiyon iki farklı amaç için kullanılabilir:

1. Liste tipinde bir veri oluşturmak
2. Farklı veri tiplerini liste adlı veri tipine dönüştürmek

Birinci amaç için bu fonksiyonu şu şekilde kullanıyoruz:

```
>>> l = list()
```

Böylece liste tipinde bir veri oluşturmuş olduk.

Dediğimiz gibi list() fonksiyonunu, farklı tipteki verileri listeye dönüştürmek için de kullanabiliriz. Örneğin:

```
>>> list('istihza')

['i', 's', 't', 'i', 'h', 'z', 'a']
```

Burada 'istihza' adlı karakter dizisini bir listeye dönüştürdük.

set() Fonksiyonu

set() fonksiyonu list() fonksiyonuna çok benzer. Bu fonksiyon da tıpkı list() fonksiyonu gibi, veri tipleri arasında dönüştürme işlemleri gerçekleştirmek için kullanılabilir. set() fonksiyonunun görevi farklı veri tiplerini kümeye dönüştürmektir:

```
>>> k = set()
```

Burada boş bir küme oluşturduk. Şimdi de mesela bir karakter dizisini kümeye dönüştürelim:

```
>>> i = 'istihza'
>>> set(i)

{'t', 's', 'z', 'a', 'i', 'h'}
```

tuple() Fonksiyonu

tuple() fonksiyonu da, tıpkı list(), set() ve benzerleri gibi bir dönüştürücü fonksiyondur. Bu fonksiyon farklı veri tiplerini demete dönüştürür:

```
>>> tuple('a')

('a',)
```

frozenset() Fonksiyonu

Bu fonksiyonu kullanarak farklı veri tiplerini dondurulmuş kümeye dönüştürebilirsiniz:

```
>>> s = set('istihza')
>>> df = frozenset(s)
>>> df
```

```
frozenset({'t', 's', 'a', 'z', 'i', 'h'})
```

complex() Fonksiyonu

Sayılardan söz ederken, eğer matematikle çok fazla içli dışlı değilseniz pek karşılaşmayacağınız, 'karmaşık sayı' adlı bir sayı türünden de bahsetmiştik. Karmaşık sayılar, bir gerçek, bir de sanal kısımdan oluşan sayılardır.

Karmaşık sayılar Python'da 'complex' ifadesiyle gösteriliyor. Mesela şu bir karmaşık sayıdır:

```
>>> 12+0j
```

İşte eğer herhangi bir sayıyı karmaşık sayıya dönüştürmeniz gerekirse complex() adlı bir fonksiyondan yararlanabilirsiniz. Örneğin:

```
>>> complex(15)
(15+0j)
```

Böyle bir kod yazdığımızda, verdiğimiz parametre karmaşık sayının gerçek kısmını oluşturacak, sanal kısım ise 0 olarak kabul edilecektir. Elbette isterseniz sanal kısmı kendiniz de belirleyebilirsiniz:

```
>>> complex(15, 2)
(15+2j)
```

float() Fonksiyonu

Bu fonksiyonu, sayıları veya karakter dizilerini kayan noktalı sayıya dönüştürmek için kullanıyoruz:

```
>>> float('134')
134.0
>>> float(12)
12.0
```

int() Fonksiyonu

Bu fonksiyon birkaç farklı amaç için kullanılabilir. int() fonksiyonunun en temel görevi, bir karakter dizisi veya kayan noktalı sayıyı (eğer mümkünse) tam sayıya dönüştürmektir:

```
>>> int('10')
10
>>> int(12.4)
12
```

Bunun dışında bu fonksiyonu, herhangi bir sayma sisteminde temsil edilen bir sayıyı onlu sayma sistemine dönüştürmek için de kullanabiliriz. Örneğin:

```
>>> int('12', 8)
10
```

Burada, sekizli sayma sistemine ait sayı değerli bir karakter dizisi olan '12'`yi onlu sayma sistemine dönüştürdük ve böylece `10 sayısını elde ettik.

int() fonksiyonunu sayma sistemleri arasında dönüştürme işlemlerinde kullanabilmek için ilk parametrenin bir karakter dizisi olması gerektiğine dikkat ediyoruz.

str() Fonksiyonu

Bu fonksiyonun, farklı veri tiplerini karakter dizisine dönüştürmek için kullanıldığını biliyorsunuz. Örneğin:

```
>>> str(12)
'12'
```

Burada 12 sayısını bir karakter dizisine dönüştürdük. Şimdi de bir baytı karakter dizisine dönüştürelim:

```
>>> bayt = b'istihza'
```

Bayt nesnemizi tanımladık. Şimdi bunu bir karakter dizisine dönüştürelim:

```
>>> kardiz = str(bayt, encoding='utf-8')
>>> print(kardiz)
istihza
```

Gördüğünüz gibi, bir baytı karakter dizisine dönüştürmek için str() fonksiyonuna encoding adlı bir parametre veriyoruz. Fonksiyonumuz, bu parametrede hangi kodlama biçimi belirtildiyse, baytları bu kodlama biçiminin kurallarına göre bir karakter dizisine dönüştürüyor.

oct() Fonksiyonu

Bu fonksiyon, bir sayıyı sekizli düzendeki karşılığına çevirmemizi sağlar:

```
>>> oct(10)

'0o12'
```

hex() Fonksiyonu

Bu fonksiyon, bir sayıyı onaltılı düzendeki karşılığına çevirmemizi sağlar:

```
>>> hex(305)

'Ox131'
```

Yalnız hem oct() hem de hex() fonksiyonlarında dikkat etmemiz gereken şey, bu fonksiyonların parametre olarak bir sayı alıp, çıktı olarak bir karakter dizisi veriyor olmasıdır.

dir() Fonksiyonu

Eğer dir() fonksiyonunu parametresiz olarak kullanırsak, mevcut isim alanındaki öğeleri bir liste halinde elde ederiz:

```
>>> dir()

['__builtins__', '__doc__', '__loader__', '__name__', '__package__']
```

Bu bakımdan dir() fonksiyonu globals() ve locals() fonksiyonlarına benzer. Ancak onlardan farkı, dir() fonksiyonunun çıktı olarak bir liste, globals() ve locals() fonksiyonlarının ise birer sözlük vermesidir.

Ayrıca dir() fonksiyonunu kullanarak nesnelerin metot ve niteliklerini içeren bir listeye ulaşabileceğimizi de biliyorsunuz. Örneğin bu fonksiyonu kullanarak farklı veri tiplerinin metot ve niteliklerini listeleyebiliriz:

```
>>> dir('')
>>> dir([])
>>> dir({})
```

exit() Fonksiyonu

Bu fonksiyon, o anda çalışan programdan çıkmanızı sağlar. Eğer bu komutu etkileşimli kabukta verirseniz o anda açık olan oturum kapanacaktır.

help() Fonksiyonu

help() fonksiyonu gömülü fonksiyonlar içinde en faydalı fonksiyonların başında gelir. Bu fonksiyonu kullanarak Python programlama diline ait öğelere ilişkin yardım belgelerine ulaşabiliriz. Örneğin:

```
>>> help(dir)
```

Bu komutu verdiğimizde dir() fonksiyonunun ne işe yaradığını gösteren İngilizce bir belgeye ulaşırız. Gördüğünüz gibi, hakkında bilgi edinmek istediğimiz öğeyi help() fonksiyonuna parametre olarak vererek ilgili yardım dosyasına erişebiliyoruz.

Eğer bu fonksiyonu parametresiz olarak kullanırsak 'etkileşimli yardım' denen ekrana ulaşırız:

```
>>> help()

Welcome to Python 3.3!  This is the interactive help utility.

If this is your first time using Python, you should definitely check out
the tutorial on the Internet at http://docs.python.org/3.3/tutorial/.

Enter the name of any module, keyword, or topic to get help on writing
Python programs and using Python modules.  To quit this help utility and
return to the interpreter, just type "quit".

To get a list of available modules, keywords, or topics, type "modules",
"keywords", or "topics".  Each module also comes with a one-line summary
of what it does; to list the modules whose summaries contain a given word
such as "spam", type "modules spam".

help>
```

Bu ekranda, hakkında bilgi edinmek istediğiniz öğeyi help> ibaresinden hemen sonra, boşluk bırakmadan yazarak öğeye ilişkin bilgilere ulaşabilirsiniz:

```
help> dir
```

Etkileşimli yardım ekranından çıkmak için 'q' harfine basabilirsiniz.

id() Fonksiyonu

Python'da her nesnenin bir kimliğinin olduğunu biliyorsunuz. Kimlik işleçlerini incelediğimiz konuda bundan bir miktar bahsetmiş ve orada id() adlı bir fonksiyondan söz etmiştik.

Orada şöyle bir örnek vermiştik:

```
>>> a = 50
>>> id(a)

505494576

>>> kardiz = "Elveda Zalim Dünya!"
>>> id(kardiz)
```

```
14461728
```

Orada söylediğimiz ve yukarıdaki örneklerden de bir kez daha gördüğünüz gibi, Python'daki her nesnenin kimliği eşşiz, tek ve benzersizdir.

format() Fonksiyonu

Bu gömülü fonksiyonun görevi, daha önce karakter dizilerini işlerken, karakter dizilerinin bir metodu olarak öğrendiğimiz format() metoduna benzer bir şekilde, karakter dizilerini biçimlendirmektir. Ancak format() fonksiyonu, daha önce öğrendiğimiz format() metoduna göre daha dar kapsamlıdır. format() metodunu kullanarak oldukça karmaşık karakter dizisi biçimlendirme işlemlerini gerçekleştirebiliriz, ama birazdan inceleyeceğimiz format() gömülü fonksiyonu yalnızca tek bir değeri biçimlendirmek için kullanılır.

Basit bir örnek verelim:

> ***>>>*** *format(12, '.2f')*
>
> *'12.00'*

Yukarıdaki ifadeyi daha önce gördüğümüz format() metodu ile şu şekilde yazabiliriz:

```
>>> '{:.2f}'.format(12)
'12.00'
```

hash() Fonksiyonu

Bu fonksiyon, belirli türdeki nesnelere bir karma değeri vermemizi sağlar. Örneğin:

```
>>> hash('istihza')
-840510580
>>> hash('python')
212829695
```

Ancak bu fonksiyonun ürettiği çıktı aynı nesne için bütün sistemlerde aynı olmayabilir. Yani örneğin yukarıdaki hash('istihza') komutu 32 bitlik ve 64 bitlik işletim sistemlerinde birbirinden farklı sonuçlar verebilir. Ayrıca bu fonksiyonun ürettiği karma değerlerinin birbiriyle çakışma ihtimali de yüksektir. Dolayısıyla bu fonksiyonu kullanarak, mesela parola girişleri için karma değeri üretmek doğru olmaz.

len() Fonksiyonu

Bu fonksiyon yardımıyla nesnelerin uzunluklarını hesaplayabileceğimizi biliyorsunuz:

```
>>> len('istihza')
7
```

```
>>> l = [1, 4, 5, 3, 2, 9, 10]
>>> len(l)

7
```

map() Fonksiyonu

Diyelim ki elimizde şöyle bir liste var:

```
>>> l = [1, 4, 5, 4, 2, 9, 10]
```

Amacımız bu liste içindeki her öğenin karesini hesaplamak. Bunun için şöyle bir yol izleyebiliriz:

```
>>> for i in l:
...     i ** 2
...
1
16
25
16
4
81
100
```

Böylece, istediğimiz gibi, bütün öğelerin karesini bulmuş olduk. Bu tür bir işlemi yapmanın bir başka yolu da map() adlı bir gömülü fonksiyondan yararlanmaktır. Dikkatlice bakın:

```
>>> def karesi(n):
...     return n ** 2
...
```

max() Fonksiyonu

max() gömülü fonksiyonunun görevi, bir dizi içindeki en büyük öğeyi vermektir. Bu fonksiyon birkaç farklı parametre alır ve verdiği çıktı, aldığı parametrelerin türüne ve sayısına bağlı olarak değişiklik gösterebilir.

Bu fonksiyonu en basit şu şekilde kullanabilirsiniz:

```
>>> max(1, 2, 3)

3
```

max() fonksiyonu yukarıdaki şekilde çalıştırıldığında, kendisine verilen parametreler arasında en büyük olanı bulacaktır. Yukarıdaki parametrelerden en büyüğü 3 olduğu için de yukarıdaki komut 3 çıktısı verecektir.

Yukarıdaki kodların sağladığı etkiyi şu şekilde de elde edebiliriz:

```
>>> liste = [1, 2, 3]
>>> max(liste)

3
```

max() fonksiyonu yukarıda gösterildiği gibi birtakım isimsiz parametrelerle birlikte key adlı isimli bir parametre de alır. Bu parametre yardımıyla m ax() fonksiyonunun 'en büyük' kavramını hangi ölçüte göre seçeceğini belirleyebiliriz.

min() Fonksiyonu

min() fonksiyonu max() fonksiyonunun tam tersini yapar. Bildiğiniz gibi max() fonksiyonu bir dizi içindeki en büyük öğeyi buluyordu. İşte min() fonksiyonu da bir dizi içindeki en küçük öğeyi bulur. Bu fonksiyonun kullanımı max() ile aynıdır.

open() Fonksiyonu

Bu fonksiyon herhangi bir dosyayı açmak veya oluşturmak için kullanılır. Eğer dosyanın açılması veya oluşturulması esnasında bir hata ortaya çıkarsa IOError türünde bir hata mesajı verilir.

Bu fonksiyonun formülü şudur:

```
>>> open(dosya_adi, mode='r', buffering=-1, encoding=None,
...     errors=None, newline=None, closefd=True, opener=None)
```

Gördüğünüz gibi, bu fonksiyon pek çok farklı parametre alabiliyor. Biz şimdiye kadar bu parametrelerin yalnızca en sık kullanılanlarını işlemiştik. Şimdi ise geri kalan parametrelerin ne işe yaradığını da ele alacağız.

Yukarıdaki formülden de görebileceğiniz gibi, open() fonksiyonunun ilk parametresi dosya_adi'dır. Yani açmak veya oluşturmak istediğimiz dosya adını bu parametre ile belirtiyoruz:

```
>>> open('falanca_dosya.txt')
```

Elbette eğer açmak istediğiniz dosya, o anda içinde bulunduğunuz dizinde değilse dosya adı olarak, o dosyanın tam adresini yazmanız gerekir. Mesela:

```
>>> open('/home/istihza/Desktop/dosya.txt')
```

Bu arada, dosya adresini yazarken ters taksim yerine düz taksim işaretlerini kullanmak daha doğru olacaktır. Bu taksim türü hem Windows'ta hem de GNU/Linux'ta çalışır. Ancak eğer ters taksim işaretlerini kullanacaksanız, dosya yolu içindeki sinsi kaçış dizilerine karşı dikkatli olmalısınız:

```
>>> f = open('test\nisan.txt')

Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
OSError: [Errno 22] Invalid argument: 'test\nisan.txt'
```

Burada problemin nisan.txt adlı dosyanın ilk harfi ile, bundan önce gelen ters taksim işaretinin birleşerek tesadüfen bir kaçış dizisi oluşturması olduğunu biliyorsunuz. Bu tür hatalara karşı ters taksim yerine düz taksim işaretlerini kullanabileceğiniz gibi r adlı kaçış dizisinden de yararlanabilirsiniz:

```
f = open(r'test\nisan.txt')
```

print() Fonksiyonu

print() fonksiyonunu artık gayet iyi tanıyoruz. Bu fonksiyonu, bildiğiniz gibi, kullanıcılarımıza birtakım mesajlar göstermek için kullanıyoruz.

Kullanımını daha önce ayrıntılı bir şekilde anlatmış olduğumuz bu fonksiyonu şu şekilde formüle edebiliriz:

```
print(deg1, deg2, deg3, ..., sep=' ', end='\n', file=sys.stdout, flush=False)
```

quit() Fonksiyonu

Bu fonksiyonu programdan çıkmak için kullanıyoruz. Eğer bu fonksiyonu etkileşimli kabukta verecek olursanız etkileşimli kabuk kapanacaktır.

range() Fonksiyonu

Bu fonksiyonu belli bir aralıktaki sayıları listelemek için kullanıyoruz. Yani mesela 0 ile 10 arası sayıların listesini almak istersek şöyle bir komut yazabiliriz:

```
>>> l = range(0, 10)
```

Ancak burada dikkat etmemiz gereken bir özellik var: Bu fonksiyon aslında doğrudan herhangi bir sayı listesi oluşturmaz. Yukarıda l değişkenine atadığımız komutu ekrana yazdırırsak bunu daha net görebilirsiniz:

```
>>> print(l)

range(0, 10)
```

Bir de bu verinin tipine bakalım:

```
>>> type(l)
<class 'range'>
```

Gördüğünüz gibi, elimizdeki şey aslında bir sayı listesi değil, bir 'range' (aralık) nesnesidir. Biz bu nesneyi istersek başka veri tiplerine dönüştürebiliriz. Mesela bunu bir listeye dönüştürelim:

```
>>> list(l)
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

reversed() Fonksiyonu

Diyelim ki elimizde şöyle bir liste var:

```
>>> isimler = ['ahmet', 'mehmet', 'veli', 'ayşe', 'çiğdem', 'ışık']
```

Eğer bu listedeki isimleri ters çevirmek, yani şöyle bir liste elde etmek isterseniz:

```
['ışık', 'çiğdem', 'ayşe', 'veli', 'mehmet', 'ahmet']
```

... ne yapmanız gerektiğini biliyorsunuz. Bu amaç için liste dilimleme yöntemlerinden yararlanabilirsiniz:

```
>>> isimler[::-1]
['ışık', 'çiğdem', 'ayşe', 'veli', 'mehmet', 'ahmet']
```

sorted() Fonksiyonu

Bu metot, daha önceki derslerimizden de bildiğiniz gibi, bir dizi içindeki öğeleri belirli bir ölçüte göre sıraya dizmemizi sağlıyor. Bununla ilgili çok basit bir örnek verelim:

```
>>> sorted('ahmet')
['a', 'e', 'h', 'm', 't']
```

Bu kodlar yardımıyla ahmet adlı karakter dizisi içindeki harfleri alfabe sırasına dizdik.

Elbette bu fonksiyonu sadece karakter dizileri üzerine uygulamıyoruz. sorted() adlı fonksiyon, dizi özelliği taşıyan her türlü nesne üzerine uygulanabilir. Mesela demetlerin ve listelerin bir dizi olduğunu biliyoruz. Dolayısıyla:

```
>>> sorted(('elma', 'armut', 'kiraz', 'badem'))

['armut', 'badem', 'elma', 'kiraz']

>>> sorted(['elma', 'armut', 'kiraz', 'badem'])
['armut', 'badem', 'elma', 'kiraz']
```

sorted() fonksiyonuna hangi türde bir veri tipi verirseniz verin, aldığınız çıktı her zaman bir liste olacaktır. Bunu unutmayın.

Kazanım

- Öğrenci bu bölümde Gömülü Fonksiyonları öğrenir ve uygulamalarla pekiştirir.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Gömülü Fonksiyonların anlatılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere Gömülü Fonksiyonlar konusunun anlatılması ve uygulamalarla pekiştirilmesi amaçlanmaktadır.
2	☑ Uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen eşliğinde editör üzerinde uygulamaların yapılması amaçlanmaktadır.

PYTHON: NESNE TABANLI PROGRAMLAMA (BÖLÜM 1)

Müfredat Kodu / Ünite / Konu	İCT-301-1.13.1 / Python Nesne Tabanlı Programlama / Nesne Tabanlı Programlama (Bölüm 1)
Hedeflenen Süre	90 dakika

🚀 Nesne Tabanlı Programlama (OOP)

Bu bölümde, programlama faaliyetlerimizin önemli bir kısmını oluşturacak olan nesne tabanlı programlama yaklaşımına bir giriş yaparak, bu yaklaşımın temel kavramlarından biri olan sınıflara değineceğiz. Bu bölümde amacımız, sınıflar üzerinden hem nesne tabanlı programlamayı tanımak, hem bu yaklaşıma ilişkin temel bilgileri edinmek, hem de etrafımızda gördüğümüz nesne tabanlı yapıların büyük çoğunluğunu anlayabilecek seviyeye gelmek olacaktır. Bu bölümü tamamladıktan sonra, nesne tabanlı programlamayı orta düzeyde bildiğimizi iddia edebileceğiz.

📌 Sınıflar

Nesne tabanlı programlamanın temelinde, yukarıdaki giriş bölümünde de adını andığımız 'sınıf' (*class*) adlı bir kavram bulunur. Bu bölümde, bu temel kavramı hakkıyla ele almaya çalışacağız.

Peki tam olarak nedir bu sınıf denen şey?

Çok kaba ve oldukça soyut bir şekilde tanımlayacak olursak, sınıflar, nesne üretmemizi sağlayan veri tipleridir. İşte nesne tabanlı programlama, adından da anlaşılacağı gibi, nesneler (ve dolayısıyla sınıflar) temel alınarak gerçekleştirilen bir programlama faaliyetidir.

'Hiçbir şey anlamadım!' dediğinizi duyar gibiyim. Çünkü yukarıdaki tanım, 'nesne' ne demek, 'sınıf' ne anlama geliyor gibi sorulara cevap vermiyor. Yani programcılık açısından 'nesne' ve 'sınıf' kelimelerini burada ne anlamda kullandığımızı, yukarıdaki tanıma bakarak kestiremiyoruz. Eğer siz de bu fikirdeyseniz okumaya devam edin...

🧐 Sınıflar Ne İşe Yarar?

Buraya gelene kadar Python'da pek çok veri tipi olduğunu öğrendik. Mesela önceki derslerimizde incelediğimiz listeler, karakter dizileri ve hatta fonksiyonlar hep birer veri tipidir. Bu tiplerin, verileri çeşitli şekillerde evirip çevirmemizi sağlayan birtakım araçlar olduğunu biliyoruz. İşte sınıflar da, tıpkı yukarıda saydığımız öteki veri tipleri gibi, verileri manipüle etmemizi sağlayan bir veri tipidir.

Peki bu bölümde ele alacağımız 'sınıf' (*class*) veri tipi ne işe yarar?

Dilerseniz bunu basit bir örnek üzerinde anlatmaya çalışalım.

Diyelim ki, kullanıcının girdiği bir kelimedeki sesli harfleri sayan bir kod yazmak istiyorsunuz. Bu amacı gerçekleştirebilmek için yazabileceğiniz en basit kod herhalde şu olacaktır:

```
sesli_harfler = 'aeıioöuü'
sayaç = 0

kelime = input('Bir kelime girin: ')

for harf in kelime:
    if harf in sesli_harfler:
        sayaç += 1
```

```
mesaj = '{} kelimesinde {} sesli harf var.'
print(mesaj.format(kelime, sayaç))
```

Düzgün bir şekilde çalışan, gayet basit kodlardır bunlar. Ayrıca amacımızı da kusursuz bir şekilde yerine getirir. Üstelik kodlardaki bütün öğeler tek bir isim/etki alanı (*namespace*, *scope*) içinde bulunduğu için, bunlara erişimde hiçbir zorluk çekmeyiz. Yani mesela sesli_harfler, sayaç, kelime, harf, mesaj değişkenlerine kodlar içinde her yerden erişebiliriz.

✍️ Sınıf Tanımlamak

Nesne tabanlı programlama yaklaşımı, özellikle birtakım ortak niteliklere ve davranış şekillerine sahip gruplar tanımlamak gerektiğinde son derece kullanışlıdır. Mesela şöyle bir örnek düşünün: Diyelim ki çalıştığınız işyerinde, işe alınan kişilerin kayıtlarını tutan bir veri tabanınız var. Bir kişi işe alındığında, o kişiye dair belli birtakım bilgileri bu veri tabanına işliyorsunuz. Mesela işe alınan kişinin adı, soyadı, unvanı, maaşı ve buna benzer başka bilgiler...

Çalışmaya başlayacak kişileri temsil eden bir 'Çalışan' grubunu, bu grubun nitelikleri ile faaliyetlerini tutacak yapıyı ve bu grubun bütün öğelerinin taşıyacağı özellikleri nesne tabanlı programlama yaklaşımı ile kolayca kodlayabilirsiniz.

Aynı şekilde, mesela yazdığınız bir oyun programı için, bir 'Asker' grubunu nesne tabanlı programlama mantığı içinde tanımlayarak, bu grubun her bir üyesinin sahip olacağı nitelikleri, kabiliyetleri ve davranış şekillerini kodlayabilir; mesela askerlerin sağa sola nasıl hareket edeceklerini, hangi durumlarda puan/enerji/güç kazanacaklarını veya kaybedeceklerini, bir asker ilk kez oluşturulduğunda hangi özellikleri taşıyacağını ve aklınıza gelebilecek başka her türlü özelliği tek tek belirleyebilirsiniz.

Amacınız ne olursa olsun, atmanız gereken ilk adım, ilgili sınıfı tanımlamak olmalıdır. Zira fonksiyonlarda olduğu gibi, bir sınıfı kullanabilmek için de öncelikle o sınıfı tanımlamamız gerekiyor. Mesela, yukarıda bahsettiğimiz işe uygun olarak, Çalışan adlı bir sınıf tanımlayalım:

```
class Çalışan:
    pass
```

Yukarıdaki, boş bir sınıf tanımıdır. Hatırlarsanız fonksiyonları tanımlamak için def adlı bir ifadeden yararlanıyorduk. İşte sınıfları tanımlamak için de class adlı bir ifadeden yararlanıyoruz. Bu ifadenin ardından gelen Çalışan kelimesi ise bu sınıfın adıdır.

Eğer arzu ederseniz, yukarıdaki sınıfı şu şekilde de tanımlayabilirsiniz:

```
class Çalışan():
    pass
```

Yani sınıf adından sonra parantez kullanmayabileceğiniz gibi, kullanabilirsiniz de. Her ikisi de aynı kapıya çıkar. Ayrıca sınıf adlarında, yukarıda olduğu gibi büyük harf kullanmak ve birden fazla kelimeden oluşan sınıf adlarının ilk harflerini büyük yazıp bunları birleştirmek adettendir. Yani:

```
class ÇalışanSınıfı():
    pass
```

Veya parantezsiz olarak:

```
class ÇalışanSınıfı:
    pass
```

Gördüğünüz gibi sınıf tanımlamak fonksiyon tanımlamaya çok benziyor. Fonksiyonları tanımlarken nasıl def deyimini kullanıyorsak, sınıfları tanımlamak için de class deyimini kullanıyoruz.

Sınıf Nitelikleri

Yukarıda, boş bir sınıfı nasıl tanımlayacağımızı öğrendik. Elbette tanımladığımız sınıflar hep boş kalmayacak. Bu sınıflara birtakım nitelikler ekleyerek bu sınıfları kullanışlı hale getirebiliriz. Mesela:

```
class Çalışan():
    kabiliyetleri = []
    unvanı = 'işçi'
```

Burada unvanı ve kabiliyetleri adlı iki değişken tanımladık. Teknik dilde bu değişkenlere 'sınıf niteliği' (*class attribute*) adı verilir.

Biraz önce, sınıf tanımlamayı öğrenirken sınıf tanımlamanın fonksiyon tanımlamaya çok benzediğini söylemiştik. Gerçekten de öyledir. Ancak fonksiyonlarla sınıflar arasında (başka farkların dışında) çok önemli bir fark bulunur. Bildiğiniz gibi, bir fonksiyonu tanımladıktan sonra, o fonksiyonun işlemeye başlaması için, o fonksiyonun mutlaka çağrılması gerekir. Çağrılmayan fonksiyonlar çalışmaz. Mesela yukarıdaki sınıfa benzeyen şöyle bir fonksiyon tanımladığımızı düşünün:

```
def çalışan():
    kabiliyetleri = []
    unvanı = 'işçi'

    print(kabiliyetleri)
    print(unvanı)
```

Bu fonksiyonun çalışması için, kodlarımızın herhangi bir yerinde bu fonksiyonu çağırmamız lazım:

```
çalışan()
```

Ancak sınıflar farklıdır. Bunu görmek için yukarıdaki fonksiyonu bir sınıf haline getirelim:

```
class Çalışan():
    kabiliyetleri = []
    unvanı = 'işçi'

    print(kabiliyetleri)
    print(unvanı)
```

Bu kodları mesela deneme.py adlı bir dosyaya kaydedip çalıştırdığınızda, unvanı ve kabiliyetleri değişkenlerinin değerinin ekrana basıldığını göreceksiniz.

Aynı şey, yukarıdaki kodların bir modül olarak içe aktarıldığı durumlarda da geçerlidir. Yani yukarıdaki kodların deneme.py adlı bir dosyada bulunduğunu varsayarsak, bu modülü şu komutla içe aktardığımızda, sınıfı kodlarımızın herhangi bir yerinde çağırmamış olmamıza rağmen sınıf içeriği çalışmaya başlayacaktır:

```
>>> import deneme

[]
işçi
```

Eğer sınıf niteliklerinin ne zaman çalışacağını kendiniz kontrol etmek isterseniz, bu nitelikleri sınıf dışında kullanabilirsiniz:

```
class Çalışan():
    kabiliyetleri = []
    unvanı = 'işçi'

print(Çalışan.kabiliyetleri)
print(Çalışan.unvanı)
```

Burada Çalışan() adlı sınıfın niteliklerine nasıl eriştiğimize dikkat edin. Gördüğünüz gibi, sınıf niteliklerine erişmek için doğrudan sınıfın adını parantezsiz bir şekilde kullanıyoruz. Eğer sınıf adlarını parantezli bir şekilde yazarsak başka bir şey yapmış oluruz. Bundan biraz sonra bahsedeceğiz. Biz şimdilik, sınıf niteliklerine erişmek için sınıf adlarını parantezsiz kullanmamız gerektiğini bilelim yeter.

Hatırlarsanız, bu bölüme başlarken, nesne tabanlı programlama yaklaşımının, özellikle birtakım ortak niteliklere ve davranış şekillerine sahip gruplar tanımlamak gerektiğinde son derece kullanışlı olduğunu söylemiştik. Gelin isterseniz yukarıdaki Çalışan() sınıfına birkaç nitelik daha ekleyerek bu iddiamızı destekleyelim:

```
class Çalışan():
    kabiliyetleri = []
    unvanı = 'işçi'
    maaşı = 1500
    memleketi = ''
    doğum_tarihi = ''
```

Burada belli kabiliyetleri, unvanı, maaşı, memleketi ve doğum_tarihi olan bir Çalışan() sınıfı tanımladık. Yani 'Çalışan' adlı bir grubun ortak niteliklerini belirledik. Elbette her çalışanın memleketi ve doğum tarihi farklı olacağı için sınıf içinde bu değişkenlere belli bir değer atamadık. Bunların birer karakter dizisi olacağını belirten bir işaret olması için yalnızca memleketi ve doğum_tarihi adlı birer boş karakter dizisi tanımladık.

Yukarıda tanımladığımız sınıf niteliklerine, doğrudan sınıf adını kullanarak erişebileceğimizi biliyorsunuz:

```
print(Çalışan.maaşı)
print(Çalışan.memleketi)
print(Çalışan.doğum_tarihi)
```

Eğer isterseniz bu sınıfa yeni sınıf nitelikleri de ekleyebilirsiniz:

```
Çalışan.isim = 'Ahmet'
Çalışan.yaş = 40
```

Gayet güzel...

Ancak burada şöyle bir sorun var: Biz yukarıdaki gibi doğrudan sınıf adını kullanarak öğelere eriştiğimizde kodlarımız tek kullanımlık olmuş oluyor. Yani bu şekilde ancak tek bir Çalışan() nesnesi ('nesne' kavramına ilerde değineceğiz), dolayısıyla da tek bir çalışan oluşturma imkanı elde edebiliyoruz. Ama biz, mantıken, sınıf içinde belirtilen özellikleri taşıyan, Ahmet, Mehmet, Veli, Selim, Selin ve buna benzer, istediğimiz sayıda çalışan oluşturabilmeliyiz. Peki ama nasıl?

Kazanım

- Öğrenci bu bölümde sınıfları, sınıf tanımlamayı ve sınıf niteliklerini öğrenir.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Nesne Tabanlı Programlama konusunun anlatılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere Nesne Tabanlı Programlama konusunun anlatılması ve uygulamalarla pekiştirilmesi amaçlanmaktadır.
2	☑ Uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen eşliğinde editör üzerinde uygulamaların yapılması amaçlanmaktadır.

PYTHON: NESNE TABANLI PROGRAMLAMA (BÖLÜM 2)

Müfredat Kodu / Ünite / Konu	İCT-301-1.13.2 / Python Nesne Tabanlı Programlama / Nesne Tabanlı Programlama (Bölüm 2)
Hedeflenen Süre	90 dakika

🚀 Sınıf Metotları

Nesne tabanlı programlamaya giriş yaptığımız geçen bölümde şunlara değindik:

1. Sınıflar (*classes*)
2. Örnekler (*instances*)
3. Sınıf nitelikleri (*class attributes*)

Bunlar nesne tabanlı programlamanın en temel kavramlarıdır. Bunları iyice öğrendiyseniz, etrafta gördüğünüz kodların büyük bölümünü anlayabilecek kıvama gelmişsiniz demektir.

Ama elbette nesne tabanlı programlama yalnızca bu temel kavramlardan ibaret değil. Nesne tabanlı programlamanın derinlerine indikçe, bunların dışında başka pek çok kavramla daha karşılaşacağız. Mesela sınıf metotları (*class methods*) bu kavramlardan biridir. İşte bu bölümde, nesne tabanlı programlamanın ileri düzey kavramlarının ilki olan bu sınıf metotlarından (*class methods*) söz edeceğiz.

Dilerseniz ne ile karşı karşıya olduğumuzu anlayabilmek için basit bir örnek üzerinden ilerleyelim:

```
class Çalışan():
    personel = []

    def __init__(self, isim):
        self.isim = isim
        self.kabiliyetleri = []
        self.personele_ekle()

    def personele_ekle(self):
        self.personel.append(self.isim)
        print('{} adlı kişi personele eklendi'.format(self.isim))

    def personeli_görüntüle(self):
        print('Personel listesi:')
        for kişi in self.personel:
            print(kişi)

    def kabiliyet_ekle(self, kabiliyet):
        self.kabiliyetleri.append(kabiliyet)

    def kabiliyetleri_görüntüle(self):
        print('{} adlı kişinin kabiliyetleri:'.format(self.isim))
        for kabiliyet in self.kabiliyetleri:
            print(kabiliyet)
```

Bu kodlarda, bir personel listesi oluşturmamızı, personele ekleme yapmamızı, personeli görüntülememizi, personele yeni kabiliyet eklememizi ve eklediğimiz kabiliyetleri görüntüleyebilmemizi sağlayan örnek metotları var.

@classmethod Bezeyicisi ve cls

Örnek metotlarını oluşturmak için self adlı bir kelimeden yararlanıyoruz. Tanımladığımız örnek metotlarının parametre listesinde ilk sıraya yerleştirdiğimiz bu kelimeyi kullanarak, sınıf içinde örnek metotlarına erişebiliyoruz. İşte sınıf metotları için de benzer bir işlem yapacağız.

Çok basit bir örnek verelim:

```
class Sınıf():
    sınıf_niteliği = 0

    def __init__(self, param1, param2):
        self.param1 = param1
        self.param2 = param2
        self.örnek_niteliği = 0

    def örnek_metodu(self):
        self.örnek_niteliği += 1
        return self.örnek_niteliği

    def sınıf_metodu(cls):
        cls.sınıf_niteliği += 1
        return cls.sınıf_niteliği
```

Burada örnek_metodu() ile sınıf_metodu() arasındaki fark, ilkinde self, ikincisinde ise cls kullanmamız. Ancak hatırlarsanız, self kelimesinin Python açısından bir zorunluluk olmadığını söylemiştik. Tıpkı self gibi, aslında cls kelimesi de Python açısından bir zorunluluk değildir. Yani cls yerine de istediğimiz kelimeyi kullanabilirdik. Bu metotlarda önemli olan, parametre listesinin ilk sırasını işgal eden kelimenin ne olduğudur. Dolayısıyla yukarıdaki örnekte Python açısından örnek_metodu() ile sınıf_metodu() arasında hiçbir fark bulunmaz. Python her iki metodu da birer örnek metodu olarak değerlendirir. Bu iki örnek metodu arasındaki fark, ilkinde sınıf örneklerini temsil edecek kelimenin self, ikincisinde ise cls olarak belirlenmiş olmasıdır. Python self veya cls kelimelerine özel bir önem atfetmez. Ama Python topluluğu içinde, örnek metotları için self, sınıf metotları için ise cls kullanmak çok güçlü bir gelenektir.

Sözün özü, sınıf_metodu() fonksiyonunun ilk parametresini cls yapmış olmamız bu metodun bir sınıf metodu olabilmesi için gereklidir, ama yeterli değildir. Python'da bir sınıf metodu oluşturabilmek için bir parçaya daha ihtiyacımız var:

```
class Sınıf():
    sınıf_niteliği = 0

    def __init__(self, param1, param2):
        self.param1 = param1
        self.param2 = param2
        self.örnek_niteliği = 0

    def örnek_metodu(self):
        self.örnek_niteliği += 1
        return self.örnek_niteliği
```

```
    @classmethod
    def sınıf_metodu(cls):
        cls.sınıf_niteliği += 1
        return cls.sınıf_niteliği
```

İşte Python'da bir sınıf metodunu örnek metodundan ayıran asıl öğe, yukarıdaki örnekte gördüğümüz @classmethod ifadesidir. Python'da isminin önünde @ işareti olan bu tür öğelere 'bezeyici' (*decorator*) adı verilir. Gördüğünüz gibi, @classmethod bezeyicisi, yukarıdaki örnekte bir fonksiyonu sınıf metoduna dönüştürme işlevi görüyor.

✍️ Alternatif İnşacılar

Sınıf metotlarının, işimize yarayabilecek bir başka özelliği ise, bunların bir 'alternatif inşacı' (*alternative constructor*) olarak kullanılabilecek olmasıdır. "Alternatif neyci?" diye sorduğunuzu rahatlıkla duyabiliyorum...

Gelin isterseniz 'alternatif inşacı' kavramını bir dizi örnek üzerinde kabataslak da olsa açıklamaya çalışalım.

İlk denememizi yapalım:

```
liste = [('9789753424080', 'Greenberg', 'Sana Gül Bahçesi Vadetmedim', 'Metis'),
         ('975872519X', 'Evren', 'Postmodern Bir Kız Sevdim', 'İthaki'),
         ('9789754060409', 'Nietzsche', 'Böyle Buyurdu Zerdüşt', 'Cem')]

def sorgula(ölçüt=None, değer=None):
    for li in liste:
        if not ölçüt and not değer:
            print(*li, sep=', ')

        elif ölçüt == 'isbn':
            if değer == li[0]:
                print(*li, sep=', ')

        elif ölçüt == 'yazar':
            if değer == li[1]:
                print(*li, sep=', ')

        elif ölçüt == 'eser':
            if değer == li[2]:
                print(*li, sep=', ')

        elif ölçüt == 'yayınevi':
            if değer == li[3]:
                print(*li, sep=', ')
```

Burada öncelikle kitap listemizi tanımladık. Daha sonra da sorgulama işlemini gerçekleştirecek sorgula() adlı bir fonksiyon yazdık.

Bu fonksiyon toplam iki parametre alıyor: ölçüt ve değer. Bu parametrelerin öntanımlı değerlerini None olarak belirledik. Böylece bu fonksiyonu herhangi bir argüman vermeden de çalıştırabileceğiz.

⚡ Statik Metotlar

Python'da örnek metotları ve sınıf metotları dışında bir de statik metotlar bulunur. Bildiğiniz gibi, örnek nitelikleri üzerinde işlem yapacağımız zaman örnek metotlarını kullanıyoruz. Aynı şekilde sınıf nitelikleri üzerinde işlem yapacağımız zaman ise sınıf metotlarından faydalanıyoruz. Örnek metotları içinde herhangi bir örnek niteliğine erişmek istediğimizde self kelimesini kullanıyoruz. Sınıf metotları içinde bir sınıf niteliğine erişmek için ise cls kelimesini kullanıyoruz. İşte eğer bir sınıf içindeki herhangi bir fonksiyonda örnek veya sınıf niteliklerinin hiçbirine erişmeniz gerekmiyorsa, statik metotları kullanabilirsiniz.

@staticmethod Bezeyicisi

Buraya gelene kadar öğrendiğimiz örnek ve sınıf metotlarını nasıl kullanacağımızı biliyorsunuz:

```
class Sınıf():
    sınıf_niteliği = 0

    def __init__(self, veri):
        self.veri = veri

    def örnek_metodu(self):
        return self.veri

    @classmethod
    def sınıf_metodu(cls):
        return cls.sınıf_niteliği
```

Burada örnek_metodu(), self yardımıyla örnek niteliklerine erişiyor. sınıf_metodu() ise cls yardımıyla sınıf niteliklerine erişiyor. Sınıf metodu tanımlamak için ayrıca @classmethod bezeyicisini de kullanıyoruz. İşte eğer sınıf içinde tanımlayacağınız fonksiyon herhangi bir örnek ya da sınıf niteliği üzerinde herhangi bir işlem yapmayacaksa şöyle bir şey yazabilirsiniz:

```
class Sınıf():
    sınıf_niteliği = 0

    def __init__(self, veri):
        self.veri = veri

    def örnek_metodu(self):
        return self.veri

    @classmethod
    def sınıf_metodu(cls):
        return cls.sınıf_niteliği

    @staticmethod
    def statik_metot():
        print('merhaba statik metot!')
```

Gördüğünüz gibi, statik metotları tanımlamak için @staticmethod bezeyicisini kullanıyoruz. Statik metotlar, ilk parametre olarak self veya cls benzeri kelimeler almaz. Çünkü bu tür sınıfların örnek veya sınıf nitelikleri ile herhangi bir işi yoktur.

Kazanım

- Öğrenci bu bölümde fonksiyonları, parametre ve argümanları öğrenir.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Nesne Tabanlı Programlama konusunun anlatılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere Nesne Tabanlı Programlama konusunun anlatılması ve uygulamalarla pekiştirilmesi amaçlanmaktadır.
2	☑ Uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen eşliğinde editör üzerinde uygulamaların yapılması amaçlanmaktadır.

HTML & CSS - JAVASCRİPT EĞITIM MÜFREDATI

HTML & CSS - JAVASCRİPT EĞİTİM MÜFREDATI														
EĞİTİM İÇERİĞİ / HAFTA	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.
HTML&CSS														
HTML: Tanımve Giriş														
HTML: Başlık Paragraflar														
HTML: Yazı														
HTML: Renk Kodları														
HTML: Link Ekleme														
HTML: Resim Ekleme														
HTML: Tablo Ekleme														
HTML: Liste Ekleme														
HTML: Elementler														
HTML: Class ve İD														
HTML: Form Yapısı														
CSS: Tanım ve Giriş														
CSS: Yazım Kuralları														
CSS: Kodları Ekleme Yöntemi														
CSS: Zemin Özellikleri														
CSS: Metin İşlemleri														
CSS: Yazı Tipi Özellikleri														
CSS: Linkler														
CSS: Listeler														
CSS: Tablolar														
CSS: Kutu Modeli														
CSS: Border														
CSS: Padding														
CSS: Boyutlar														
JAVASCRİPT														
Javascript: Tanıtım ve Programlamaya Giriş														
Javascript: Değişkenler ve Sabitler														
Javascript: Operatörler														
Javascript: Sayı Metotları														
Javascript: String Metotları														

Javascript: Koşul Durumları														
Javascript: Döngüler														
Javascript: Array														
Javascript: Fonsiyonlar														
Javascript: Tarih ve Zaman Kontrolleri														
Javascript: Nesne Yönelimli Programlama														

HTML & CSS EĞITIMI

Müfredat Kodu	İCT-301-2
Ders	HTML & CSS
Ders İçeriği	• HTML: Giriş • HTML: Etiketler • HTML: Açıklama Satırı • HTML: Başlık ve Paragraflar • HTML: Yazı • HTML: Renk Kodları • HTML: Link Ekleme • HTML: Resim Ekleme • HTML: Tablo Ekleme • HTML: Liste Ekleme • HTML: Elementler • HTML: Class ve id • HTML: Form Yapısı • CSS: Giriş • CSS: Yazım Kuralları • CSS: Kodları Ekleme Yöntemi • CSS: Zemin Özellikleri • CSS: Metin İşlemleri • CSS: Yazı Tipi Özellikleri • CSS: Linkler • CSS: Listeler • CSS: Tablolar • CSS: Tablo Elemanları • CSS: Kutu Modeli • CSS: Border • CSS: Padding • CSS: Boyutlar
Seviye	İCT-S1: BAŞLANGIÇ
Ön Koşullar	Teknoloji ve yazılımla ilgili olmak.
Amaç	Temel düzeyde bir web sitesi oluşturmak ve site içerisine metin, tablo, form vb. içeriklerin eklenmesi ile bunların anlamlı bir şekilde birbirine bağlanarak çalıştırılabilmesi amaçlanmaktadır.
Hedef	Öğrencilerin kod blokları, editör kullanımı, sayfa oluşturma, hizalama vs. gibi kodlama alanına ait kuralları tanımasını ve temel düzeyde web sitesi oluşturmayı öğrenebilmesini hedeflemektedir.
Gereksinimler	Bilgisayar ve Ortamda İnternet Ağı
Kazanım	• Öğrencilerin algoritma ve programlama becerileri gelişir.
Referanslar & Dokümantasyon	• Html'e Giriş. HyperText Markup Language (hypertext işaret dili) Nedir? - PDF Free Download (docplayer.biz.tr) • HTML - Vikipedi (wikipedia.org) • https://tr.wikipedia.org/wiki/HTML • Html Nedir? • https://www.webhostuzmani.com/html-etiketleritag-nedir-ne-ise-yarar/ • https://wmaraci.com/nedir/head-etiketi#:~:text=Head%20etiketi%2C%20bir%20HTML%20sayfas%C4%B1n%C4%B1n,botlar%C4%B1%20ve%20%C3%B6r%C3%BCmcekler%20taraf%C4%B1ndan%20kullan%C4%B1lmaktad%C4%B1r. • https://sefik.net/?mid=42&id=203#.Y7V4-nZBxPY • https://app.patika.dev/courses/html/en-cok-kullanilan-html-etiketleri-nelerdir

	• https://app.patika.dev/courses/html/aciklama-satirlari-olusturmak • https://www.tasarimveyazilim.com/html-aciklama-yorum-satiri-nedir-nasil-kullanilir/ • https://www.kodkampusu.com/html-yazi-formatlari/#close • https://www.natro.com/blog/htmlde-yazi-tipi-nasil-degistirilir/ • https://www.esiber.com/web/html-renk-kodlari/ • https://htmlcssdersleriogreniyorum.com/html_dersleri/6/html-renk-kodlari(color-codes) • https://gelecegiyazanlar.turkcell.com.tr/konu/egitim/201-html5-css/css-yazim-kurallari-syntax
Durum	BEKLIYOR / UYGULANIYOR / TAMAMLANDI / DEĞERLENDIRILDI
Hedeflenen Süre	720 Dakika
Eğitim Günleri	Başlama: .../.../.... Bitiş: .../.../....

HTML: GIRIŞ

Müfredat Kodu / Ünite / Konu	İCT-301-2.1.1 / Html & Css Tanım ve Giriş / Giriş
Hedeflenen Süre	10 Dakika

🚀 HTML Nedir ?

HTML web sayfalarını oluşturmak için kullanılan standart faktörünü ifade eden metin işaretleme dilidir. Bu terim programlama dili değildir.

HTML web sitelerinin okuduğu bir dildir. Bu sayede web sitelerini çalıştırır. HTML çok fazla derecede işe yarayan bir dildir. Dilin faydaları şunlardır:

- Yazı, görsel, video gibi ögelerin sayfa içerisine yerleştirilmesini sağlar.
- Konumlandırmayı gerçekleştirir.
- Bu sayfaların web tarayıcılarında düzgün bir şekilde görüntülenmesini sağlar.
- Arama motorlarına sayfa hakkında bilgi verilmesini gerçekleştirir.

🧐 HTML Nerelerde Kullanılır?

Bu terim web sitesi yaparken oluşturulan temeldir. Web tarayıcısı aracılığı ile bir web sitesine girdiğimizde ekranda gördüğümüz her şey HTML' dir.

Web sitelerinde görülen her şey HTML ile yapılıyor ve kodlanıyor. Bir web sitesi oluşturacaksanız hangi alanda çalışırsanız çalışın HTML bilmek zorundasınız.

HTML ile bir web sitesinde video, müzik, tablo, resim paylaşabilirsiniz. Oluşturulan sitelerde sağlık ile ilgili bilgilere, tablolara, verilere, yer verebilirsiniz. Aynı zamanda farklı alanlarda da web sitesi oluşturarak HTML ile dünyaya açılabilirsiniz.

📄 HTML Dosyalarının Açılımı

HTML dosyalarının çalıştırılması için oluşturduğunuz dosyaların sonuna **.html** uzantısı eklemeniz ve bir internet tarayıcısı üzerinde açmanız yeterli olacaktır. Böylelikle yazmış olduğunuz HTML kodları ile oluşturduğunuz web sitesini görüntüleyebilirsiniz.

💻 Geliştirme Ortamı

HTML öğrenebilmeniz için herhangi bir metin editörünü kullanmanız yeterlidir. Bu noktada size Sublime Text, Brackets, Visual Studio Code ve Notepad++ gibi bazı başarılı metin editörlerini önerebiliriz. Bu editörler, HTML kodlarını yazarak bir web sitesi inşa etmek için kullanılmaktadır. Bunun dışında, görsel olarak da bir web sitesi oluşturmak mümkündür. Bunun için Microsoft Expression Web, Adobe Dreamweaver ve BlueGriffon gibi farklı araçlar kullanılabilir.

⌨️ Kod Bloğu

```
<html>
  <head>
    <title>örnek başlık</title>
  </head>
  <body>
    <p style="font-size:10px"><a href="http://tr.wikipedia.org">Özgür Ansiklopedi</a></p>
  </body>
</html>
```

Kazanım

- Öğrencilere bu bölümde HTML kavramını öğretmek amaçlanır.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ HTML' in tanıtılması.	☑ Eğitmen ☐ Öğrenciler	Bu derste öğretmen öğrencilere HTML' i tanıtır.
2	☑ Editör keşfi.	☐ Eğitmen ☑ Öğrenciler	Öğrenci bu bölümde HTML ile tanışır ve seçilen editörü keşfeder.

Eğitmen Yol Haritası

1	Öğrencilere HTML tanıtılır
2	Öğrencilerin cihazlarına editör kurulumu yapılır.
3	Öğrencilerin editörleri keşfetmesi sağlanır.

HTML: ETIKETLER

Müfredat Kodu / Ünite / Konu	İCT-301-2.1.2 / Html & Css Tanım ve Giriş / Etiketler
Hedeflenen Süre	10 Dakika

🚀 Etiket Nedir?

Etiketler Html içinde bulunan ve önceden belirlenmiş bir kod dizinidir. Biz etiketleri tanımlayıp daha kullanışlı hale getirmek için etiketleri kullanmamız gerekir.

Etiketler “< “ ile başlayıp “>” ile biter.

Etiketlere; <html> , <head>,<title>,<body>,<input>, ,,<table>,<tr>,<th>,<td> gibi birçok örnekler verilebilir.

🔍 *<head> Etiketi*

Head etiketi, bir HTML sayfasının içerisinde yer alması gereken ancak bunları sayfa üzerine yazdırmamak ve ziyaretçilere göstermemek için kullanılan gizleme etiketidir. Head etiketi içerisinde kullanılan bilgiler internet tarayıcıları, arama motorlarının botları ve örümcekler tarafından kullanılmaktadır.

Head etiketi yardımıyla sayfanın içerisinde yer almasını istediğimiz fakat kullanıcıya gösterilmeyecek birçok elementi sayfanın içerisinde barındırabiliriz. Bu sayede çok daha aktif ve kullanışlı internet sayfaları meydana getirebiliriz.

Head etiketi altında bütün etiketleri kullanabilmeniz mümkün değil. Kullanabileceğiniz etiketler;

- <title> (Bu etiketi kullanmak şarttır)
- <meta>
- <style>
- <script>
- <noscript>
- <link>
- <base>

Bu etiketlere şöyle örnek verebiliriz:

```
<html>
  <head>
    <title> Sekmede Görülecek İsim </title>
    <meta name="Keywords" content="HTML,Kodluyoruz">
  </head>

  <body></body>

</html>
```

🔍 *<body> Etiketi*

Web sayfamızda görmek istediğimiz bütün içerikleri <body> etiketi altına yazıyoruz. <body> etiketini şöyle örnekleyebiliriz:

```
<html>
  <head>
    <title> Sekmede Görülecek İsim </title>
    <meta name="Keywords" content="HTML,Kodluyoruz">
  </head>

  <body>
    Site İçeriği
  </body>
</html>
```

<h> Etiketleri

H etiketleri başlık etiketleridir. Büyükten küçüğe sırasıyla;

<h1>

<h2>

<h3>

<h4>

<h5>

<h6>

HTML otomatik olarak Başlık etiketlerinin öncesine ve sonrasına satır atlatır.

<p> Etiketi

<p> etiketi paragraf etiketidir. Sayfa içerisinde oluşturacağımız metinleri <p> etiketi ile oluştururuz. Aşağıdaki örnekte hem başlık etiketi hem de paragrafa etiketini kullandım.

```
<body>
   <h2>Kodluyoruz</h2>
   <p>HTML Etiketleri</p>
</body>
</html>
```

*
 Etiketi*

 etiketi satır atlatma etiketidir ve kapatmaya ihtiyaç duymayan etiketlerden biridir. Atlatmak istediğiniz satır sayısı kadar <br> etiketi kullanabilirsiniz.

Bu etiketin <br>,
,
 şeklinde farklı kullanımlarını da görebilirsiniz.

```
<body>
   <h2>Kodluyoruz</h2> <br>
   <p>HTML <br> Etiketleri</p>
</body>
```

<a> Etiketi

<a> etiketinin en önemli özelliği href özelliğidir. Bu etiket ile sayfaları linkleyebiliriz. Etiket içerisine yazılan içerik sayfa üzerinde gösterilecek içeriktir. href içine yazılan ise tıklandığında gideceği URL'dir.

```
<body>
   <a href="https://www.kodluyoruz.org">Kodluyoruz</a>
</body>
```

<ul> - <ol> - <li> Etiketi

<ul> ve <ol> etiketleri liste oluşturma etiketleridir. Listeyi oluşturduktan sonra içeriğini oluşturmak için <li> etiketini kullanıyoruz.

<ul> = "unordered list" sırasız liste anlamına geliyor. <ol> = "ordered list" sıralı liste anlamına geliyor.

```
<body>

   <ul>
      <li>HTML</li>
      <li>CSS</li>
      <li>JavaScript</li>
   </ul>

   <ol>
      <li>HTML</li>
      <li>CSS</li>
      <li>JavaScript</li>
   </ol>

</body>
```

<hr> Etiketi

<hr> etiketi sayfaya yatay çizgi çizmek için kullanılan etikettir. Bu etiket diğer etiketlerden farklı olarak kapatılmaz. Size, width ve align parametreleri mevcuttur.

- size: çizilen çizginin kalınlığını belirlemek için kullanılır.
- width: çizginin sayfadaki uzunluğunu belirlemek için kullanılır.
- align: çizginin sayfada yer alacağı konumunu (sağa hizalı, sola hizalı, ortalanmış) belirlemek için kullanılır.

▲ *ve <b> Etiketi*

 etiketi bir metnin arama motorlarına önemli olduğunu bildirmek için kullanılır. Kullanıldığı zaman metini kalın yapar. Eğer sadece metini kalınlaştırmak isterseniz <b> etiketini kullanabilirsiniz.

```
<body>
   <h2><strong> Kodluyoruz </strong></h2>
   <p><b> HTML </b> Etiketleri </p>
</body>
```

▲ *<script> Etiketi*

<script> etiketi JavaScript kodlarını HTML içerisine yazabilmemizi sağlar.

```
<body>
   <script>
      document.write("Kodluyoruz")
   </script>
</body>
```

▲ *<button> Etiketi*

<button> etiketini buton oluşturmak için kullanırız. Buton üzerine yazmak istediğiniz içeriği etiketin içine yazmanız yeterlidir.

```
<body>
   <button> Buton </button>
</body>
```

▲ *<img> Etiketi*

Resim eklemek için<img> etiketini kullanıyoruz. <img src=”resim.jpg” alt=”açıklama yazısı” /> src="" kısmına eklemek istediğimiz görselin yolunu yani kaynağını yazmalıyız. Eğer görselimiz ve HTML dosyamız aynı klasörde ise görselin adını ve uzantısını yazmamız yeterlidir. alt="" kısmına görselin açıklamasını yazıyoruz fakat isterseniz boş bırakabilirsiniz. Bu etiket kapanmaya ihtiyaç duymaz.

```
<body>
   <img src=”görseller/kodluyoruz.jpg” alt=”Kodluyoruz Bootcamp” />
</body>
```

▲ *<iframe> Etiketi*

Belge içinde belge gösterebilmemizi sağlayan etikettir. Genelde başka bir sitedeki belgeyi kendi sayfamızda göstermek için kullanırız. *örn:* Youtube'dan bir videoyu sayfamızda göstermek istersek <iframe> kodlarını sayfamıza eklememiz yeterli.(video üzerinde sağ tıklayıp yerleştirme kodunu kopyala diyerek iframe kodunu kopyalayabiliriz.)

```
<body>
    <iframe width="1519" height="581" src="https://www.youtube.com/embed/BHPYQHnD_QA" frameborder="0"
        allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture"
        allowfullscreen></iframe>
</body>
```

Kazanım

- Öğrenciler bu bölümde etiket yapısını öğrenirler.

Eğitmen Yol Haritası

1	Öğrencilere etiket yapısının anlatılması.
2	Konu hakkında yapılacak uygulamaların tanıtılması.
3	Öğrencilerle birlikte editör üzerinde uygulamaların yapılması.

HTML: AÇIKLAMA SATIRI

Müfredat Kodu / Ünite / Konu	İCT-301-2.1.3 / Html & Css Tanım ve Giriş/ Açıklama Satırı
Hedeflenen Süre	10 dakika

🚀 Açıklama satırı

Html kodlarını yazarken kodlar arasına not almaya olanak veren aralıklara açıklama satırı denir. Web sayfasının kodlarını yazarken her bölüm arasına hangi bölüm olduğunu belirtiriz. Bu bizim kodlarımızın daha anlaşılır olmasını sağlar.

Açıklama satırı nasıl oluşturulur ?

Açıklama satırını HTML kodumuzda <!– ile başlatıp –> ile bitecek şekilde yazabiliyoruz. <!- ile başlatıp - - > işareti ile biten alan arasına yorumlarımızı ekleyebiliyoruz.

```
<!-yazdığımız kodla
ilgili not almak istediğimiz zaman kullanırız-->
```

Açıklama satırı web sayfasında görünür mü ?

Hayır, kullanıcılar web sayfanızda açıklama satırlarını görmezler. Ayrıca açıklama satırları web tarayıcıları tarafından herhangi bir işleme tabi tutulmazlar. Fakat kullanıcı sayfanızda sağ tıklayıp kaynağı görüntüle dediğinde açıklama satırlarınızı görebilir.

Kazanım

- Öğrenciler bu bölümde açıklama satırını öğrenirler.

Eğitmen Yol Haritası

1	Öğrencilere açıklama satırının anlatılması.
2	Konu hakkında yapılacak uygulamaların tanıtılması.
3	Öğrencilerle birlikte editör üzerinde uygulamaların yapılması.

HTML: BAŞLIK VE PARAGRAFLAR

Müfredat Kodu / Ünite / Konu	İCT-301-2.2 / Html & Css / Başlık ve Paragraflar
Hedeflenen Süre	20 Dakika

🚀 HTML Başlıklar

HTML'de en önemli kısım başlıklardır. <h1> Etiketi yazıların başlık şeklinde yazılması için kullanılır. Altı tane başlık çeşidi vardır. Bu başlık özellikleri <h1> den <h6> ya kadar kodlanmıştır. <h1> Etiketi en büyük fontu ve en çok önemi bildirirken <h6> ya doğru azalma göstermektedir.

```
<h1>H1 Başlık</h1>
<h2>H2 Başlık</h2>
<h3>H3 Başlık</h3>
<h4>H4 Başlık</h4>
<h5>H5 Başlık</h5>
<h6>H6 Başlık</h6>
```

⚠️ Başlıklar HTML dokümanlarda büyük önem arz eder. Belge yapısını göstermek için başlıkları kullanmak önemlidir. <h1>Başlıklar ana başlıklar için kullanılmalı, ardından başlıklar takip edilmeli <h3>, biraz daha az önemli <h4>vb.

🛫 HTML Paragraflar

HTML de enter işlevi yoktur. Bu yüzden eğer farklı farklı paragraflar oluşturmak istiyorsak <p> etiketini kullanırız. Bir paragraf her zaman yeni bir satırda başlar ve tarayıcılar bir paragrafın önüne ve arkasına otomatik olarak biraz beyaz boşluk (kenar boşluğu) ekler.

```
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
```

📘 Kazanım

- Öğrenciler bu bölümde başlık kullanımının önemini ve paragraf kullanımı öğrenirler.

👩‍🏫 Eğitmen Yol Haritası

1	Öğrencilere paragraf etiketinin anlatılması.
2	Öğrencilere başlık etiketinin anlatılması.
3	Konu hakkında yapılacak uygulamaların tanıtılması.
4	Öğrencilerle birlikte editör üzerinde uygulamaların yapılması.

HTML: YAZI

Müfredat Kodu / Ünite / Konu	İCT-301-2.3 / Html & Css / Yazı
Hedeflenen Süre	20 Dakika

🚀 HTML Yazı Stilleri

HTML kodlarıyla işlem yaparken çeşitli yazı kodları bulunmaktadır. Kodlar arasında bir tag olmadan yazı yazamayız. Yazı yazabilmemiz için çeşitli taglar bulunmaktadır.
HTML Yazı formatlarıyla yazılarımızda çeşitli değişiklikler yapmaktayız. Kalın yazma, italik yazma gibi türleri bu taglar sayesinde yapabilmekteyiz. Aynı zamanda çeşitli başlık tagları bulunmaktadır. Bu başlıklar büyükten küçüğe sıralı haldedir. Gelin şimdi yazı taglarını inceleyelim.

ℹ️ *Paragraf türünde yazı yazmak için:*

```
<p> Merhaba! Bu paragraf yazısıdır </p>
```

Merhaba! Bu paragraf yazısıdır

ℹ️ *Kalın yazı yazmak için:*

```
<b> Merhaba! Bu kalın yazıdır </b>
```

Merhaba! Bu kalın yazıdır

ℹ️ *İtalik yazı yazmak için:*

```
<em><i> Merhaba! Bu italik yazıdır </i></em>
```

Merhaba! Bu bir italik yazıdır

ℹ️ *Üstü çizili yazı:*

```
<del>Merhaba! Bu üstü çizili yazıdır </del>
```

~~Merhaba! Bu bir üstü çizili yazıdır~~

ℹ️ *Altı çizili yazı:*

```
<u>Merhaba! Bu altı çizili yazıdır</u>
```

Merhaba! Bu altı çizili yazıdır

<font-family>

Metnin yazı tipi ailesinin seçilmesini sağlar. Ancak kodların yazımında dikkat edilmesi gereken önemli noktaların başında, birden fazla sözcükten oluşan yazı tiplerinde tırnak (“”) kullanımı gelir. Örneğin; popüler yazı tiplerinden biri olan Times New Roman yazılırken tırnak içine alınır.

En Çok Kullanılan Yazı Stilleri Nelerdir?

- Arial
- Helvetica
- Times New Roman
- Trajan
- Roboto
- Times
- Courier
- Calibri
- Cambria

Bu yazı fontlarının kullanımına şöyle örnek verebiliriz;

```
<p style=”font-family:arial;”>Kullanmak istediğiniz herhangi bir metin</p>
```

Herhangi bir değişiklik yapılmadığı sürece yazı tipi her zaman Times New Roman olur.

Yazı Büyüklüğü Ayarlama

Yazı büyüklüğü ise font-size kodu ile değiştirilir. Burada ölçü birimi olarak punto, piksel, inç ya da yüzde kullanılabilir. Örneğin, 12 punto olmasını istediğiniz bir metin için aşağıdaki kod kullanılır:

```
<p style=”font-size:12pt”>kullanılmak istenen 12 puntoluk metin</p>
```

Yazı Rengini Değiştirme

Yazı renginin değiştirilmesi, font-color etiketi ile gerçekleşir. Örneğin;

```
<p style=”color:red”>kullanılmak istenen kırmızı renkteki metin</p>
```

Bu kod ile seçilen metin kırmızı renkte görünür.

Kazanım

- Öğrenci bu bölümde yazı stilini, yazının büyüklüğünü ve yazının rengini değiştirmeyi öğrenirler.

Eğitmen Yol Haritası

1	Öğrencilere yazı stilinin, büyüklüğünün ve renginin anlatılması.
2	Konu hakkında yapılacak uygulamaların tanıtılması.
3	Öğrencilerle birlikte editör üzerinde uygulamaların yapılması.

HTML: RENK KODLARI

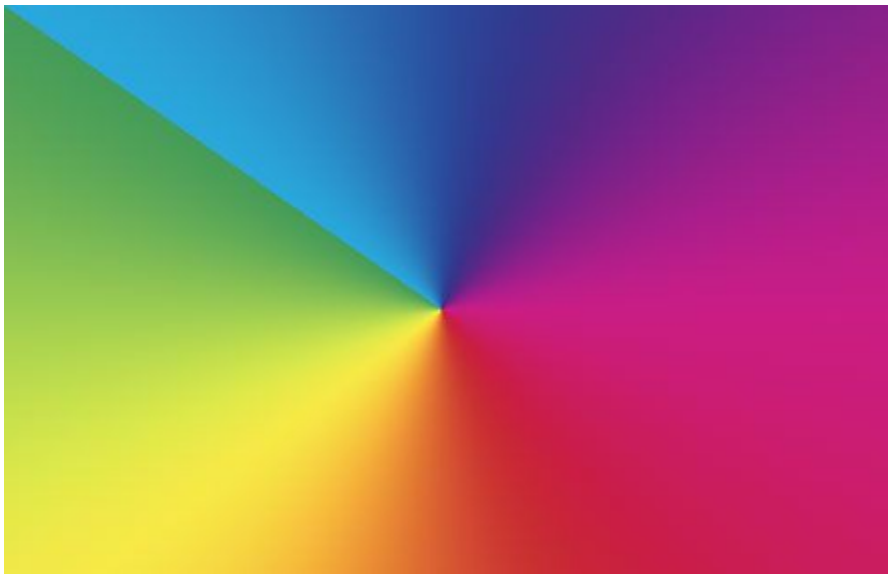
Müfredat Kodu / Ünite / Konu	İCT-301-2.4 / Html & Css / Renk Kodları
Hedeflenen Süre	20 Dakika

🚀 HTML Renk Kodları

Bütün renkler kırmızı, yeşil ve mavi ana renklerinin kombinasyonuyla elde edilir. Bir ışık kaynağından gelen rengin en düşük değeri: 0 (hex #00) , en yüksek değeri ise 255 (hex #FF) şeklindedir.

⚠️ Normalde oldukça fazla tonda renk çeşidi vardır. Yalnız tarayıcıların veya bilgisayar ekranları bütün renkleri aynı şekilde işlemez. Bu yüzden çalışmalarınızda standart dışı renk kullanmamaya özen göstermelisiniz. Ne kadar az renk skalası kullanırsanız içeriğiniz o kadar çok farklı cihazda hiç değişmeden aynı biçimde görünebilir. Diğer türlü içeriğinizin her ortamda aynı görünmemesi riskiyle karşı karşıya kalmış olursunuz.

Bu renk kodlarına ulaşmak kolay. Yapmanız gereken internette renk kodlarını aratmak ve çıkan yüzlerce renk çeşidinden birini seçmek. Seçtiğiniz rengin kodu karşınıza iki türlü çıkacaktır;



1. HEX: 6 haneli ve biraz daha çeşitliliği az olan renk kodlarıdır.

```
<p style="background-color:#00FF00">Yeşil metin</p><br>
```

2. RGB: Kırmızı. yeşil ve maviden oluşan üç rengin karışımından elde edilen renklerdir. Çok fazla çeşidi vardır.

```
<h2 style="background-color:rgb(205,0,0)">Koyu Kırmızı</h2>
```

Kazanım

- Öğrenciler bu bölümde renk kodlarını öğrenirler.

Eğitmen Yol Haritası

1	Öğrencilere renk kodlarının anlatılması.
2	Konu hakkında yapılacak uygulamaların tanıtılması.
3	Öğrencilerle birlikte editör üzerinde uygulamaların yapılması.

NOTLAR

HTML: LINK EKLEME

Müfredat Kodu / Ünite / Konu	İCT-301-2.5 / Html & Css / Link Ekleme
Hedeflenen Süre	20 Dakika

🚀 HTML de Link Ekleme

Bir HTML dosyası içerisinde herhangi bir sayfaya, sayfa içine, bir websitesine, e-mail, telefon adreslerine ya da dosya yolu belirtilen herhangi bir dosyaya (resim, video, zip vb.) erişmek için kullanılan HTML etiketidir. Genel yapısı <a></a> şeklindedir.

```
<a href="gitmek istenilen yer">Gidilecek yer için isim tanımlaması</a>
```

💡 *Sayfalar arası kullanımı:*

HTML sayfalar arası geçiş yapmak için linklerden şu şekilde faydalanırız. Şu an üzerinde çalıştığımız klasörün içerisinde bir linksayfasi.html adında dosya oluşturduk ve içerisine bir şeyler yazdık. Şimdi HTML'de sayfalar arası linkleme ile yeni oluşturduğumuz sayfaya gitmek için link oluşturacağız:

```
<a href="linksayfasi.html">İkinci sayfaya git</a>
```

Bunu oluşturduğumuzda tarayıcı ekranımızda Diğer sayfaya geç diye bir link oluşacak (üzerine gelindiğinde mouse'un el işaretine dönmesinden anlaşılabilir)

💡 *Sayfa içerisinde kullanımı:*

Sayfa içerisinde herhangi bir başlığa ya da bölüme gitmek için linkler kullanılabilir. Aynı sayfada işlem yapacağımız için birden fazla satır ekleyeceğim ve en aşağıya scroll ile kaydırmak yerine link yardımı ile gideceğim. Bunun için de gidilecek bölüm için a tagine name özelliği verilmelidir:

```
<a href="#sonbolum">Aşağı Git</a>
<a name="sonbolum"></a>
```

💡 *Website yönlendirmesinde kullanımı:*

a etiketinde href özelliğine verilen herhangi bir websitesi adresine kolayca gidilebilir. Burada target özelliğini göreceğiz. Bu özellik, gitmek istediğimiz bağlantının geçerli pencerede mi yoksa yeni bir pencerede mi açılması için kullanılır. _self özelliği geçerli pencerede açılması içindir. Varsayılan olarak böyledir. _blank özelliği ise yeni bir pencerede açılması içindir.

💡 *Mail ve telefon yönlendirmesinde kullanımı:*

a etiketinin href özelliğine verilen mailto: ve tel: özellikleri sayesinde direkt olarak herhangi bir e-mail adresine posta gönderilebilir ya da geçerli bir telefon numarası aranabilir.

```
<a href="mailto:info@kodluyoruz.org">Kodluyoruz'a mail atınız.</a>
```

Kazanım

- Öğrenciler bu bölümde link eklemeyi öğrenirler.

Eğitmen Yol Haritası

1	Öğrencilere link etiketinin anlatılması.
2	Konu hakkında yapılacak uygulamaların tanıtılması.
3	Öğrencilerle birlikte editör üzerinde uygulamaların yapılması.

HTML: RESIM EKLEME

Müfredat Kodu / Ünite / Konu	İCT-301-2.6 / Html & Css / Resim Ekleme
Hedeflenen Süre	20 Dakika

🚀 Resim Ekleme

HTML belgemize resim eklemek için <img> kodundan yararlanırız.

▲ *src="" Kullanımı*

Kod bloğundaki src="" özelliğine görselin URL ya da dosya adresi belirtilerek resim HTML sayfasına çağırılır. (Resmi webden çağırmak için resmin URL'ini src="..." parametesine eklemek yeterlidir.)

```
<img src="ornek.jpg"/>
```

⚠️ Yukarıdaki örnekte resim HTML dosyasıyla aynı dizinde(klasörde) olduğu için direkt adını ve uzantısını yazmak yeterlidir. Burada img uzantısına dikkat etmek önemli, HTML dosyaları nasıl .html ile bitiyorsa tüm resim dosyalarının sonu da .xbm, .gif, .png veya .jpg ile veya başka bir resim formatıyla bitmelidir.

▲ *alt="" Kullanımı*

Alt textlerin temel amacı, görüntüleri göremeyen kullanıcılar için metinler sunmaktır. Kullanıcı görseli görüntüleyemez ise (Yavaş bağlantı, src özelliğinde hata vb.) alt özelliği görüntü için alternatif bilgilendirici bir metin içerir.

```
<img src="../images/kedi.jpg" alt="Yavru Kedi"/>
```

▲ *title="" Kullanımı*

Title özelliği kullanıcıyı bilgilendirme amacı taşır. Cursor(mouse imleci) ile görselin üzerine gelince bu özelliğe verilen text mesajı görünür. Ek açıklama gerektirecek resimlerde kullanabiliriz. Bilgilendirme amacı taşır.

```
<img src="../images/kedi.jpg" title="image"/>
```

📘 *Title ve Alt parametreleri SEO açısından önem taşımaktadır*

🔍 *width ve height Kullanımı*

Width yatay genişlik, Height ise dikey uzunluk için kullanılır.

Şöyle örnek verebiliriz;

```
<img src="resim.jpg" width="300" />
<img src="resim.jpg" height="400" />
<img src="resim.jpg" width="300" height="600" />
<img src="resim.jpg" height="100%" />
```

border Kullanımı

Görseli belirtilen kalınlıkta çerçeve içine alır. Daha gelişmiş CSS Border özelliği bunun yerine kullanılabilir.

Aşağıda border örneği: Görsele 3 pixel kalınlıkta border verir.

```
<img src="resim.jpg" border="3" />
```

align Kullanımı

Web sayfasında resmin gözükeceği pozisyonu belirlemede align özelliği kullanılır. Bu özelliğe verilebilecek değerler şunlardır: left, right. Görselin sağa veya sola yaslı çıkmasını sağlar.

```
<img src="resim.jpg" align="right" />
```

Görsele Link Vermek

Görsele link vermek için img tag'i a taginin içerisinde kullanılır. Yönlendirilmek istenen yerin URL'i a taginin href özelliğine yazılır. Görsel ise a taginin açıklama kısmına eklenir.

Yani şöyle:

```
<a href="default.asp">
  <img src="smiley.gif" alt="HTML tutorial" style="width:42px;height:42px;">
</a>
```

Kazanım

- Öğrenciler bu bölümde resim eklemeyi öğrenirler.

Eğitmen Yol Haritası

1	Öğrencilere resim eklemenin anlatılması.
2	Konu hakkında yapılacak uygulamaların tanıtılması.
3	Öğrencilerle birlikte editör üzerinde uygulamaların yapılması.

HTML: TABLO EKLEME

Müfredat Kodu / Ünite / Konu	İCT-301-2.7 / Html & Css / Tablo Ekleme
Hedeflenen Süre	25 Dakika

🚀 Tablo Oluşturma

Html dilinde tablo oluşturma <table></table> etiketi arasına yazılır. Satır oluşturmak için tr, sütun oluşturmak için td etiketi kullanılır. Aşağıdaki tabloyu html dilinde oluşturalım.

	Beden	**Türkçe**
Sinem	95	80
Ahmet	100	65

```
<table border="1">
  <body>
    <tr>
      <td> </td>
      <td>Beden</td>
      <td>Türkçe<br />
      </td>
    </tr>
    <tr>
      <td>Sinem</td>
      <td>95</td>
      <td>80</td>
    </tr>
    <tr>
      <td>Ahmet</td>
      <td>100</td>
      <td>65</td>
    </tr>
  </body>
</table>
```

⚠️ <table> yanındaki border="1" tablonun dış kenar çizgi kalınlığını belirler. Farklı bir değer kullanabilirsiniz. İlk hücre boş olduğu için   yazdık. Eğer yazmasaydık kenar çizgileri gözükmezdi.

✏️ *Tablo içinde kullanılabilecek özellikler*

- border : Dış kenar çizgi kalınlığını belirler.
- width : Tablonun piksel olarak genişliğini belirler.
- heigth : Tablonun piksel olarak yüksekliğini belirler.

- bordercolor : Tablonun çizgi rengini belirler.
- bgcolor : Tablonun arka plan rengini belirler
- align : Veriyi yatay olarak hizalamak için kullanılır. left: sola , right: sağa , center: ortaya hizalar.

Şu şekilde yazılırlar;

```
<element_name align="left">

<element_name align="right">

<element_name align="center">
```

- valign : Veriyi dikey olarak hizalamak için kullanılır. top: üste , bottom: alta , middle: ortaya hizalar.

Şu şekilde yazılırlar;

```
<td valign="bottom">Şubat</td>

<td valign="top">Şubat</td>

<td valign="middle">Şubat</td>
```

- colspan: Sütun üzerinde kaç tane hücrenin birleştirileceğini belirtir.

```
<td colspan=" sayı "></td>
```

- rowspan: Sütun birleştirmek için kullanılır.

```
<td rowspan=" sayı "></td>
```

Kazanım

- Öğrenciler bu bölümde tablo oluşturmayı öğrenir ve tablo özelliklerini kavrarlar.

Eğitmen Yol Haritası

1	Öğrencilere tablonun nasıl olacağının anlatılması.
2	Konu hakkında yapılacak uygulamaların tanıtılması.
3	Öğrencilerle birlikte editör üzerinde uygulamaların yapılması.

HTML: LISTE EKLEME

Müfredat Kodu / Ünite / Konu	İCT-301-2.8 / Html & Css / Liste Ekleme
Hedeflenen Süre	25 Dakika

🚀 Liste Ekleme

Listeleri iki ana başlık altında listeleyebiliriz;

1. Sıralı Listeler
2. Sırasız Listeler

📏 *Sıralı Listeler*

Sıralı listeler ardışık liste numaraları vermek için kullanılır. Sıralı listelerden yararlanmak için <ol> etiketi kullanılır.

```
<p> Gerçek <b>tereyağı</b> nasıl anlaşılır ?</p>
  <ol>
    <li>Oda sıcaklığında tamamen erimez</li>
    <li>Suyun içinde tek parça halinde çözünür</li>
    <li>Tereyağı rengi sarı yada beyaz olabilir</li>
  </ol>
```

Ekran çıktısı şu şekilde olacaktır:

Gerçek tereyağı nasıl anlaşılır?

1. Oda sıcaklığında tamamen erimez

2.Suyun içinde tek parça halinde çözünür

3.Tereyağı rengi sarı yada beyaz olabilir

⚠️ Liste başındaki sıralandırmayı rakamdan başka roma rakamı veya alfabetik şeklinde de yapabiliriz. Bunun için type özelliğini kullanmamız gerekir.

```
<ol type="I">
  <li>Javascript</li>
  <li>C#</li>
  <li>Php</li>
</ol>
```

Ekran çıktısı şu şekilde olacaktır:

I. Javascript

II. C#

III. Php

📏 *Sırasız Listeler*

Sırasız listeler numaralandırma olmadan oluşturduğumuz listeleredir. Her bir liste elemanı bir satırı kaplayacak şekilde yani blok etiket şeklinde oluşturulur.

```
<ul>
  <li>Çay</li>
  <li>Türk Kahvesi</li>
  <li>Süt</li>
</ul>
```

Ekran çıktısı şu şekilde olacaktır:

Çay

Türk Kahvesi

Süt

📘 Liste elemanlarının başındaki içi dolu daireyi değiştirebilir veya silebiliriz.

❗ *Başında bulunan işareti silmek için;*

```
<ul style="list-style-type:none">
 <ul>
   <li>Kahve</li>
   <li>Çay</li>
   <li>Meyve Suyu</li>
</ul>
```

Ekran çıktısı şu şekilde olacaktır:

Kahve

Çay

Meyve Suyu

📘 Kazanım

- Öğrenciler bu bölümde liste eklemeyi ve liste kavramlarını öğrenir, uygulamalarla öğrendiklerini pekiştirir.

👩‍🏫 Eğitmen Yol Haritası

1	Öğrencilere liste konusunun anlatılması.
2	Konu hakkında yapılacak uygulamaların tanıtılması.
3	

	Öğrencilerle birlikte editör üzerinde uygulamaların yapılması.

NOTLAR

HTML: ELEMENTLER

Müfredat Kodu / Ünite / Konu	İCT-301-2.9 / Html & Css / Elementler
Hedeflenen Süre	30 Dakika

🚀 Elementler

HTML de iki element tipi bulunmaktadır. Bunlar blok düzeyi ve satır içi olarak tanımlanmaktadır. Blok düzeyi dediğimiz öğeler yeni bir satırdan başlar.

⚠️ Örnek= <h1>, <form>, <p>, <table>, vb.

Satır içi öğeler ise satır atlamaları olmadan görüntülenir.

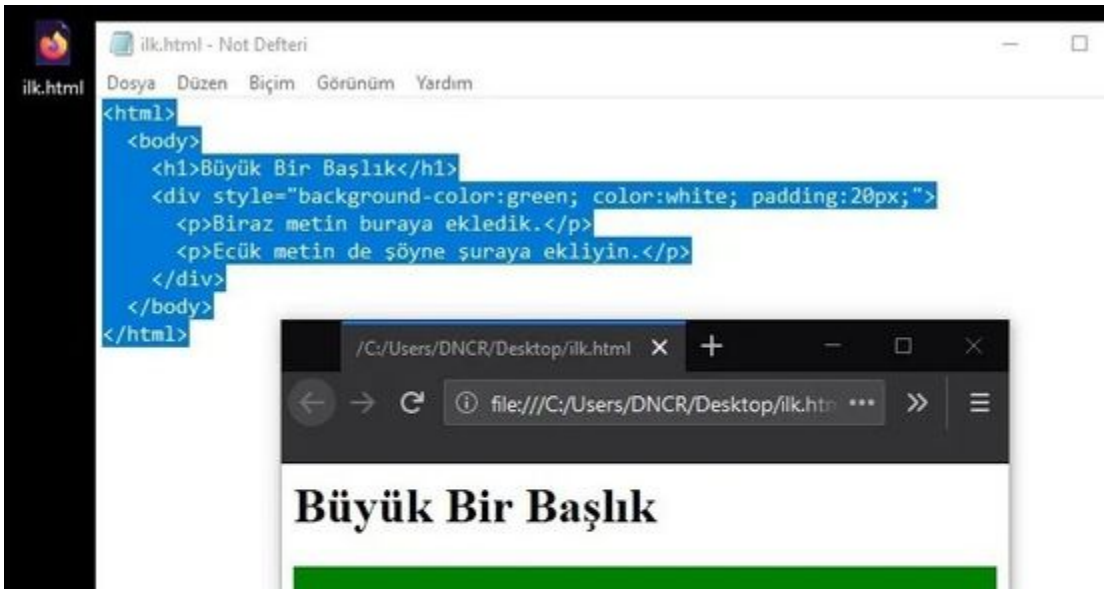
⚠️ Örnek = <b>, <a>, <img>, <input>, vb.

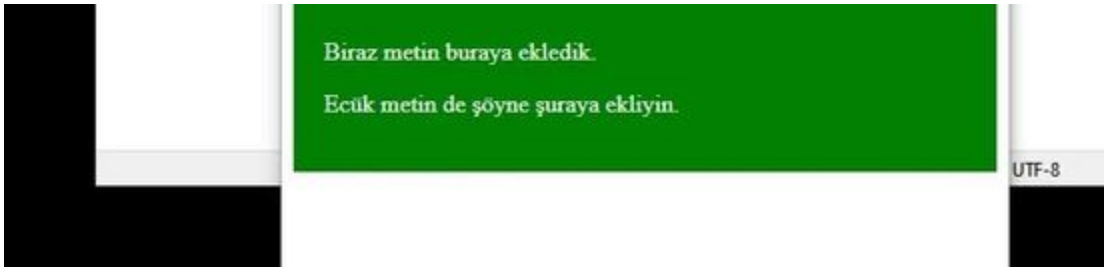
🔺 * *

Genellikle diğer HTML öğelerini kapsayıcısı olarak kullanılan blok düzeyi bir öğedir. Kullanımı çok fazladır. HTML öğelerini bir kutu içerisine almak gibi düşünebilirsiniz. Bazı CSS stilleriyle birlikte kullanıldığında öğesi içerik bloklarını biçimlendirmek için kullanılabilir.

```
<html>
  <body>
    <h1>Büyük Bir Başlık</h1>
    <div style="background-color:green; color:white; padding:20px;">
      <p>Biraz metin buraya ekledik.</p>
      <p>Ecük metin de şöyne şuraya ekliyin.</p>
    </div>
  </body>
</html>
```

Ekran çıktısı şu şekilde olacak:





 öğesine verdiğimiz stiller ile altındaki tüm alan etkilendi.

**

Yine benzer şekilde öğesi, bazı metinler için kutu görevi gören satır içi bir öğedir. CSS ile birlikte kullanıldığında metnin sadece bir kısmının biçimlendirilmesi için kullanılabilir.

Bunun da örneğini şu şekilde verelim.

```
<html>
  <body>
    <h2>Bu Başlık
      <span style="color:red">Önemli</span>
    Başlık</h2>
  </body>
</html>
```

Ekran çıktısı şu şekilde olacak:

Son bir özet olarak, elementi blok düzeyinde bir bölüm/alan tanımlar. öğesi ise belgedeki satır içinde bir bölümü ifade etmektedir.

Kazanım

- Öğrenciler bu bölümde elementlerin kullanımını öğreniyorlar ve uygulamalarla pekiştiriyorlar.

Eğitmen Yol Haritası

1	Öğrencilere elementlerin anlatılması.
2	Konu hakkında yapılacak uygulamaların tanıtılması.
3	Öğrencilerle birlikte editör üzerinde uygulamaların yapılması.

HTML: CLASS VE ID

Müfredat Kodu / Ünite / Konu	İCT-301-2.10 / Html & Css / Class ve id
Hedeflenen Süre	30 Dakika

🚀 Class (Sınıf)

Class, HTML etiketinin hangi sınıfta olduğunu belirmemizi sağlar. Aynı sınıf adını birden fazla HTML etiketine verebiliriz.

```
<img src ="" class ="city">
<a href ="" class ="city" ></a>
```

⚠️ Burada görüldüğü gibi iki HTML etiketi de (img ve a) city sınıfında bulunmaktadır.

Sınıf oluşturmak için; bir nokta (.) karakteri ve ardından bir sınıf adı yazın. Ardından, süslü parantez {} içinde CSS özelliklerini tanımlayın:

```
.city {
  background-color: tomato;
  color: white;
  padding: 10px;
}
```

🛫 id (identity = kimlik)

id kullanılan etiketin kimliğini belirtir. Biriciktir*(unique)* yani dokümanın içindeki bir etikete yalnızca bir kez o id ismi verilebilir.

```
<a href ="https://www.facebook.com/" id ="facebook-URL" ><a>
```

Buradaki id yi kullanarak direkt olarak bu HTML etiketine ulaşabiliriz.

id biricik olduğu için istisna olarak id = facebook-URL şeklinde yazarak tırnak işareti kullanmayabiliriz.

id oluşturmak için; # karakteri ve ardından bir id ismi yazın. Ardından, süslü parantez {} içinde CSS özelliklerini tanımlayın:

```
#baslik {
        background-color: lightblue;
        color: black;
        padding: 40px;
        text-align: center;
    }
```

Bir id adı, sayfa içinde yalnızca bir HTML öğesi tarafından kullanılmalıdır

Kazanım

- Öğrenciler bu bölümde class ve id yapılarını öğrenirler.

Eğitmen Yol Haritası

1	Öğrencilere class ve id yapısının anlatılması.
2	Konu hakkında yapılacak uygulamaların tanıtılması.
3	Öğrencilerle birlikte editör üzerinde uygulamaların yapılması.

HTML: FORM YAPISI

Müfredat Kodu / Ünite / Konu	İCT-301-2.11 / Html & Css / Form Yapısı
Hedeflenen Süre	30 Dakika

🚀 Form Yapısı

Html form etiketlerinin görevi, <form> ... </form> etiketleri içinde bulunan html kontrolleri aracılığıyla kullanıcılardan toplanan bilgileri server'a göndermektir.

Örneğin bir internet sitesindeki üyelik formunu doldurduğumuzda girdiğimiz bilgileri form nesnesi tarayıcıdan alıp server'a taşır.

🧐 *Peki bilgiler neden server'a gider ?*

Server derken kastedilen, web sitesinin barındırıldığı, internet üzerinden kullanıcıların hizmetine açılan yüksek kapasiteli bilgisayarlardır.

Server bilgisayarlarında web sitelerinden gelen bilgileri işleyecek olan web programları vardır ve form nesnesi ile web programlarına gelen bilgiler çeşitli amaçlarla kullanılır. Örneğin bir hesaplama işlemi yada bilgilerin alınıp kalıcı bir veritabanı içine kayıt edilmesi gibi.

🔍 *<form>etiketi şu şekilde kullanılır:*

```
<form>

...

</form>
```

Form parametreleri şunlardır ;

🔺 readonly = yazıldığı form öğesini sadece "okunabilir" yapmaktadır ve değiştirilemez ama kopyalanabilir, seçilebilir.

```
<form>
  <label for="ad">Adınız :</label><br>
  <input type="text" id="ad" name="ad" value="Kodlama" readonly><br>
  <label for="soyad">Soyadınız :</label><br>
  <input type="text" id="soyad" name="soyad" value="Vakti">
</form>
```

🔺 disabled = yazıldığı form öğesinin değiştirilemez, seçilemez ve düzenlenemez olmasını sağlar.

```
<form>
  <label for="ad">Adınız :</label><br>
  <input type="text" id="ad" name="ad" value="Kodlama" disabled><br>
  <label for="soyad">Soyadınız :</label><br>
  <input type="text" id="soyad" name="soyad" value="Vakti">
</form>
```

size = yazıldığı form öğesinin karakter cinsinden görünür genişliğini belirtir. Varsayılan değer 20'dir.

```
<form>
  <label for="ad">Adınız :</label><br>
  <input type="text" id="ad" name="ad" size="50"><br>
  <label for="pin">PIN Kodu:</label><br>
  <input type="text" id="pin" name="pin" size="4">
</form>
```

maxlength = yazıldığı form öğesinin izin verilen maksimum karakter sayısını belirtir.

```
<form>
  <label for="ad">Adınız :</label><br>
  <input type="text" id="ad" name="ad" size="50"><br>
  <label for="pin">PIN Kodu:</label><br>
  <input type="text" id="pin" name="pin" max-length="4" size="4">
</form>
```

min ve max = yazıldığı form öğesinin minimum ve maksimum değerleri belirtir. Bu parametreler bazı input türleriyle çalışır: number, range, date, datetime-local, month, time ve week.

```
<label for="maxtarih">2000-01-01 tarihinden küçük bir değer giriniz : </label>
  <input type="date" id="maxtarih" name="maxtarih" max="1999-12-31"><br><br>
  <label for="mintarih">2000-01-01 tarihinden büyük bir değer giriniz : </label>
  <input type="date" id="mintarih" name="mintarih" min="2000-01-02"><br><br>
  <label for="miktar">Lütfen 1 ve 5 değerleri arasında bir sayı giriniz . </label>
  <input type="number" id="miktar" name="miktar" min="1" max="5">
</form>
```

multiple = yazıldığı form öğesinin birden fazla değer girmesine izin verir.

```
<form>
  <label for="dosyalar">Dosyaları Seçiniz :</label>
  <input type="file" id="dosyalar" name="dosyalar" multiple>
</form>
```

placeholder = yazıldığı form öğesinin beklenen değerini (örnek bir değer veya beklenen biçimin kısa bir açıklaması) açıklayan kısa bir ipucu belirtir.Kısa ipucu, kullanıcı bir değer girmeden önce giriş alanında görüntülenir. Placeholder parametresi ayrıca şu form girdi türleri ilede çalışır : text, search, url, tel, email, and password.

```
<form>
  <label for="telefon">Telefon Numaranız:</label>
  <input type="tel" id="telefon" name="telefon"
  placeholder="0553-xxx-xx-xx" >
</form>
```

required = yazıldığı form öğesinin doldurulmasının zorunlu olduğunu belirtir. Genellikle formlarda doldurulmasını kesin istediğimiz durumlarda kullanırız, kullanıcı bu alanı doldurmadı ise formu gönderemez.

```
<form>
  <label for="sifre">Şifreniz :</label>
  <input type="text" id="sifre" name="sifre" required>
</form>
```

İnput parametreleri şunlardır:

- <input type="button">
- <input type="checkbox">
- <input type="color">
- <input type="date">
- <input type="datetime-local">
- <input type="email">
- <input type="file">
- <input type="hidden">
- <input type="image">
- <input type="month">
- <input type="number">
- <input type="password">
- <input type="radio">
- <input type="range">
- <input type="reset">
- <input type="search">
- <input type="submit">
- <input type="tel">
- <input type="text">(varsayılan değer)
- <input type="time">
- <input type="url">
- <input type="week">

Eğer örnek verecek olursak:

```
<form>

    Ad:<br>
```

```
    <input type="text" name="firstname" value="Adınız"><br>

    Soyad:<br>
    <input type="text" name="lastname" value="Soyadınız"><br><br>

    Cinsiyet : <br>
    <input type="radio" name="gender" value="male" checked> Erkek<br>
    <input type="radio" name="gender" value="female"> Kadın<br>

    Şehir :<br>
    <select name="city">
      <option value="kocaeli">Kocaeli</option>
      <option value="istanbul">İstanbul</option>
      <option value="ankara">Ankara</option>
      <option value="izmir">İzmir</option>
   </select>
   <br>

   <input type="submit" value="Submit">

</form>
```

Kazanım

- Öğrenciler bu bölümde form yapısını ve form ve input parametrelerini öğrenirler.

Eğitmen Yol Haritası

1	Öğrencilere form yapısının ve form ve input parametrelerinin anlatılması.
2	Konu hakkında yapılacak uygulamaların tanıtılması.
3	Öğrencilerle birlikte editör üzerinde uygulamaların yapılması.

CSS: GIRIŞ

Müfredat Kodu / Ünite / Konu	İCT-301-2.12 / Html & Css / Giriş
Hedeflenen Süre	30 Dakika

🚀 CSS Nedir?

İngilizcesi “Cascading Style Sheets” olan CSS açılımı, “Basamaklanmış Stil Katmanları” anlamına gelmektedir. CSS, web sayfasını şekillendirmek için kullanılan bir kodlama türüdür. CSS kullanarak web sayfasının görünümünü ve düzenini değiştirebiliriz. Bir web sitesinin görünümünün masaüstü bilgisayarlar, tabletler ve mobil cihazlar gibi farklı ekranlarda nasıl değiştiğini de tanımlayabiliriz.

❌ CSS bir programlama dili değildir.

🔍 CSS Kullanım Şekilleri Nelerdir?

CSS yazılımı web sayfalarına ait özelliklerin tek tek veya toplu olarak kontrol edilmesini sağlayan oldukça esnek ve kullanışlı bir yapıya sahiptir. CSS’de; harici stil sayfası, dahili stil sayfası ve etiket içine yazılan stiller olmak üzere 3 farklı yöntemle web sayfalarınıza kodlarınızı ekleyebilirsiniz. Her bir yöntemin kullanım amacına ve yerine göre sahip olduğu çeşitli avantajlar vardır.

Sadece bir CSS sayfası ile birden çok web sayfanızın stil özelliklerini kontrol etmek istiyorsanız harici stil sayfası yöntemi ile CSS kodu ekleyebilirsiniz.

Tek bir web sayfasında yer alan ögeleri kontrol etmek ve gereken biçimlendirmeleri yapmak için dahili stil ekleme metodunu kullanabilirsiniz.

Etiket içine yazılan kodlar ise sadece ilgili etiketi kontrol eder ve gerekmedikçe kullanılması önerilmez.

🤖 Web sayfalarında sadece CSS ile yapılabileceğiniz işlemlerin dışında; CSS, HTML ve JavaScript dilinin gücünü birleştirerek oldukça etkileyici sonuçlar elde edebilirsiniz.

🧐 En Popüler CSS Kütüphaneleri Hangileridir?

- Bootstrap
- Foundation
- Bulma
- UIkit
- Semantic UI
- Materyalize CSS

⌨️ Kod Bloğu

```
h1 {
  color: red;
}
```

📘 Kazanım

- Öğrenciler bu bölümde CSS kavramını öğrenirler.

Eğitmen Yol Haritası

1	Öğrencilere CSS tanıtılır.
2	Öğrencilerin cihazlarına editör kurulumu yapılır.
3	Öğrencilerin editörleri keşfetmesi sağlanır.

CSS: YAZIM KURALLARI

Müfredat Kodu / Ünite / Konu	İCT-301-2.13 / Html & Css / Yazım Kuralları
Hedeflenen Süre	30 Dakika

🚀Yazım Kuralları

Bir CSS kural kümesi, bir seçici (selector) ve bir bildiri bloğundan (declaration block) oluşur. Selector, biçim (style) bilgilerini yükleyeceğiniz HTML elemanını işaret eder. Declaration block (bildiri bloğu) bir ya da daha fazla bildiriden oluşur. Değiştirilmek istenen birden fazla özellik varsa bildiriler arasına ";" sembolü konur. Her bildiri, bir özellik adı (property name) ve bir değer (value) içerir. Aralarına ":" sembolü yerleştirilir.

```
p {color:blue;text-align:left;}
```

🔺CSS kodunu daha okunabilir yapmak için her bildiri deyimi ayrı bir satıra konabilir:

```
p {
   color: blue;
   text-align: left;
}
```

✏️Aşağıda ki örneği tarayıcıda açtığımızda <p> etiketi içinde kalan yazılar mavi renkli görüntülenir.

```
<!DOCTYPE html>
<html>
   <head>
      <style>
         p {
            color: blue;
            text-align: left;
         }
      </style>
   </head>

   <body>
      <p>GELECEGİ YAZANLAR</p>
      <p>CSS STIL KODLAMASI ORNEGI</p>
   </body>
</html>
```

GELECEGİ YAZANLAR
CSS STIL KODLAMASI ORNEGI

🛩️CSS Yorum Satırları

Yorumlar yazılan kod üzerinde açıklama eklemek için kullanılır. Ayrıca kodu geliştiren kişinin bile uzun zaman geçtikten sonra unutabileceği bazı kritik noktaları kendisine hatırlatır. Yorum satırları tarayıcı tarafından dikkate alınmaz.

⚠️ Bir CSS yorumu, "/*" sembolleriyle başlar ve "*/" sembolleriyle sona erer. Birden çok satıra da yayılabilir.

```
p {
   color: blue;
   /* paragraf mavi renkte yapılıyor */
   text-align: left;
/*metin sola yanaşık*/
}
```

💡 CSS Seçiciler (selectors)

CSS seçicileri, HTML elemanlarını seçmenizi ve onlar üzerinde işlem yapmanızı sağlar. CSS seçicileri, HTML elemanlarını, id'leri, sınıfları (class), tipleri, özellikleri (attribute) ya da özellik değerlerini kullanarak bulur.

▲ *Eleman Seçiçi*

Eleman seçici, elemanları etiket adlarına göre seçer. Sayfadaki tüm <p> elemanlarını etkileyecek bir değişikliği bu şekilde gerçekleştirebilirsiniz.

Örnek verecek olursak:

```
p {
   text-align: left;
   color: blue;
}
<!DOCTYPE html>
<html>
   <head>
      <style>
         p {
            text-align: left;
            color: blue;
         }
      </style>
   </head>

   <body>
      <p>HER PARAGRAF STILDEN ETKILENECEKTIR</p>
      <p id="para1">BU PARAGRAF DA ETKILENIR</p>
      <p>BU DA ISTISNA DEGILDIR</p>
   </body>
</html>
```

HER PARAGRAF STILDEN ETKILENECEKTIR

BU PARAGRAF DA ETKILENIR

BU DA ISTISNA DEGILDIR

id Seçici

id seçici, belirli bir elemanı bulmak için, HTML etiketinin id özelliğini kullanır. id, sayfa içinde tekil (unique) olmalıdır. id seçicisini, sayfa içinde belirli tek bir elemanı bulmak söz konusu olduğu zaman kullanmalısınız. id ile elemanı bulmak için aranan elemanın id'sinin önüne "#" sembolü koymalısınız.

Örnek verecek olursak:

```
#parag1 {
   text-align: left;
   color: blue;
}
<!DOCTYPE html>
<html>
   <head>
      <style>
         #parag1 {
            text-align: left;
            color: red;
         }
      </style>
   </head>

   <body>
      <p id="parag1">GELECEGI YAZANLAR</p>
      <p>BU PARAGRAF STILDEN ETKILENMIYOR</p>
      <p>BU PARAGRAF DA STILDEN ETKILENMIYOR</p>
   </body>
</html>
```

GELECEGI YAZANLAR

BU PARAGRAF STILDEN ETKILENMIYOR

BU PARAGRAF DA STILDEN ETKILENMIYOR

▲ *Sınıf (class) Seçici*

Sınıf seçici aradığı elemanları belirli bir sınıf bilgisine göre arar. Sınıf seçici, HTML sınıf özelliğini kullanır. Belirli bir sınıfa ait elemanları bulmak için, bir nokta karakterinden sonra sınıfın adı yazılır.

```
.lefty {
   text-align: left;
   color: red;
}
/* Yukarıdaki örnekte class="lefty" özelliğine (attribute) sahip olan elemanlar aranmaktadır.*/
<!DOCTYPE html>
<html>
   <head>
      <style>
         .lefty {
            text-align: left;
            color: red;
         }
      </style>
   </head>

   <body>
      <h1 class="lefty">KIRMIZI VE SOLA YANASIK BASLIK</h1>
      <p class="other">STILIN ETKISI YOK</p>
      <h1 class="lefty">KIRMIZI VE SOLA YANASIK BASLIK</h1>
   </body>

</html>
```

KIRMIZI VE SOLA YANASIK BASLIK

STILIN ETKISI YOK

KIRMIZI VE SOLA YANASIK BASLIK

▲ *Seçicileri gruplama*

```
h1 {
   text-align: left;
   color: red;
}

h2 {
   text-align: left;
   color: red;
```

```
}

p {
    text-align: left;
    color: red;
}
```

Kodu küçültmek için gruplama yapabilirsiniz. Burada hem h1 ve h2 başlıkları hem de p paragrafı aynı stilde tanımlanmak isteniyor. Buna göre h1, h2 ve p'yi virgüllerle ayırarak yukarıdaki uzun ifadeleri aşağıdaki gibi kısa ve tek bir ifade haline getirebiliriz:

```
h1, h2, p {
    text-align: left;
    color: red;
}
```

Kazanım

- Öğrenciler bu bölümde CSS de yazım kurallarını, yorum satırını ve seçicileri öğrenirler.

Eğitmen Yol Haritası

1	Öğrencilere yazım kurallarını, yorum satırını ve seçicilerin anlatılması.
2	Konu hakkında yapılacak uygulamaların tanıtılması.
3	Öğrencilerle birlikte editör üzerinde uygulamaların yapılması.

CSS: KODLARI EKLEME YÖNTEMI

eMüfredat Kodu / Ünite / Konu	İCT-301-2.14 / Html & Css / Kodları Ekleme Yöntemi
Hedeflenen Süre	30 Dakika

🚀 Kodları Ekleme Yöntemi

Bir biçim (style) sayfasını eklemenin üç farklı yolu mevcuttur: Dış stil sayfası (External Style Sheet), İç stil sayfası (Internal Style Sheet), Çevrimiçi (Inline) stil.

🔺 *Dış Stil Sayfası*

Stil çok sayıda sayfaya uygulanacaksa, dış stil sayfası kullanmak uygun bir yöntemdir. Dış stil sayfasıyla, bütün web sayfasının görünümünü sadece bir tek dosyayla oynayarak gerçekleştirebilirsiniz. Her sayfa, <link> etiketiyle, stil sayfasına gönderen bir bağlantı içermelidir; <link> etiketi head bölümünün içinde olmalıdır.

```
<head>

<link rel="stylesheet" type="text/css" href="stil.css">

</head>
```

⚠️ Dış stil sayfası herhangi bir metin düzenleyicisinde (text editor) yazılabilir. CSS dosyası herhangi bir HTML etiketi içermemelidir ve .css uzantısı ile saklanmalıdır.

```
"stil.css":

body {
   background-color: lightgreen;
}

h1 {
   color: red;
   padding-left: 15px;
}
```

🔺 *İç Stil Sayfası*

Bir iç stil sayfası, tek bir belgenin kendine özgü tek bir biçimde stil tanımı olacaksa kullanılmalıdır. İç stiller, HTML sayfasının head kısmında yer almalıdır. Burada <style> etiketi içine yerleştirilir.

```
<head>

<style>
```

```
body {
   background-color: white;
}

h1 {
   color: blue;
   padding-left: 30px;
}
</style>
</head>
<!DOCTYPE html>
<html>
   <head>
      <style>
         body {
            background-color: white;
         }

      h1 {
         color: blue;
         padding-left: 30px;
      }

      </style>
   </head>

   <body>
      <h1>BASLIK-STIL TANIMI VAR</h1>
      <p>PARAGRAF-STIL TANIMI YOK</p>
   </body>
</html>
```

BASLIK-STIL TANIMI VAR

PARAGRAF-STIL TANIMI YOK

🔺 *Inline Stiller*

Bu yöntem CSS dosyalarının avantajını ortadan kaldıran bir yöntemdir; içerik ile stil karıştırılır. Bu nedenle kullanılması tavsiye edilmez.

```
<h1 style="color:red;padding-left:40px;">BASLIK </h1>
```

💡 *Çoklu Stil Sayfaları (Multiple Style Sheets)*

Şayet bazı özellikler aynı seçici için farklı stil sayfalarına yerleştirilmişse, değerler en özel stile göre oluşturulur. Örneğin, h1 seçicisi için aşağıdaki özellikler bir dış stil sayfasına yüklenmiş olsun:

```
h1 {
   color: green;
   padding-left: 30px;
}
```

Ayrıca, bir iç stil sayfası da h1 için aşağıdaki atamaları yapmış olsun:

```
h1 {
   color: blue;
}
```

Şayet iç stil sayfası olan web sayfası aynı zamanda dış stil sayfasına bağlantı veriyorsa, h1 elemanına atama aşağıdaki gibi olacaktır:

```
color: blue;
padding-left: 30px;
```

⚠ Sol marjin dış stil sayfasından, renk iç stil sayfasından miras alınır.

```
<!DOCTYPE html>
<html>
   <head>
      <link rel="stylesheet" type="text/css" href="stil.css">
         <style>
            h1 {
               color: navy;
            }
         </style>
   </head>
   <body>
      <h1>BASLIK</h1>
      <p>DIS-IC STIL KOMBINASYONU</p>
   </body>
</html>
```

BASLIK

DIS-IC STIL KOMBINASYONU

Farklı ortamlarda tanımlanan stil özelliklerinin önceliği

Stiller,

- Bir HTML5 elemanının içinde,
- Bir HTML5 sayfasının başlığı içinde,
- Bir dış CSS dosyasında tanımlanabilir.

Burada önemli soru şudur:

Bir HTML elemanında birden fazla stil tanımlanmışsa hangi stil kullanılacaktır?

Bu konuda aşağıdaki sırayı bilmek önemlidir:

1. Web tarayıcı varsayılan arama.
2. Dış stil sayfası
3. İç stil sayfası (head bölümü içinde)
4. Inline stil. (HTML elemanı içinde)

Sonuçta, inline stil (HTML elemanı içinde) en yüksek önceliğe sahiptir ve diğerlerinin önüne geçecektir.

Kazanım

- Öğrenciler bu bölümde bir biçim sayfasına kodların nasıl ekleneceğini öğrenirler.

Eğitmen Yol Haritası

1	Öğrencilere kodları ekleme yöntemlerinin anlatılması.
2	Konu hakkında yapılacak uygulamaların tanıtılması.
3	Öğrencilerle birlikte editör üzerinde uygulamaların yapılması.

CSS: ZEMIN ÖZELLIKLERI

Müfredat Kodu / Ünite / Konu	İCT-301-2.15 / Html & Css / Zemin Özellikleri
Hedeflenen Süre	30 Dakika

🚀Zemin Özellikleri

CSS zemin özellikleri, bir elemanın zemin özel tanımlamak için kullanılır. Bu özellikler;

- Zemin rengi,
- Zemin görseli,
- Zemin tekrarı,
- Zemin ekleri,
- Zemin pozisyonudur.

🔺 *Zemin rengi*

Bir elemanın zemininin rengini belirler. Sayfanın zemin rengi, body kısmında tanımlanır.

```
body {

    background-color: #c0d5fe;
}
```

CSS ile zemin rengi aşağıdaki biçimlerden biriyle belirlenir:

- Bir HEX değeri: "#de0420"
- Bir RGB değeri: "rgb(100,50,220)"
- Bir renk adı: red, blue vb.

Bir başka örneğe bakalım:

```
<!DOCTYPE html>
<html>
   <head>
      <style>
         body {
            background-color: #c2c5ff;
         }
      </style>
   </head>

   <body>
      <h1>CSS SAYFASI</h1>
      <p>GELCEGI YAZANLAR</p>
   </body>

</html>
```

▲ *Zemin resmi*

Zemin resmi özelliği, elemanın zeminini oluşturacak görseli belirler. Normal olarak görsel zeminde tekrarlanır yani tüm zemini kaplar. Zemin resmi aşağıdaki gibi atanır:

```
body {
   background-image: url("gy.png");
}
<!DOCTYPE html>
<html>
   <head>
      <style>
         body {
            background: #ff0ccc url("https://gelecegiyazanlar.turkcell.com.tr/sites/default/files
/icerik/gy_maraton.jpg") no-repeat left top;
            padding-right: 150px;
         }
      </style>
```

```
    </head>

    <body>
        <h1>TURKIYE CUMHURIYETI</h1>
        <p>ISTANBUL</p>
        <p>ANKARA</p>
    </body>
</html>
```

No-repeat özelliği kaldırılınca zemin resmi sayfa boyunca tekrarlanır. Aşağıdaki kodda no-repeat özelliğinin kaldırılmasını görebilirsiniz:

```
<!DOCTYPE html>
<html>
    <head>
        <style>
            body {
                background: #ff0ccc url("https://gelecegiyazanlar.turkcell.com.tr/sites/default/files
/icerik/gy_maraton.jpg")  left top;
                padding-right: 150px;
            }
        </style>
    </head>

    <body>

        <h1>TURKIYE CUMHURIYETI</h1>
        <p>ISTANBUL</p>
        <p>ANKARA</p>

    </body>
</html>
```

Zemin resminin yatay ve dikey tekrarı

Varsayılan olarak zemin resmi özelliği (background-image property) bir görseli sayfa içinde hem yatay hem de dikey olarak tekrarlar. Bazı resimlerin sadece yatay ya da dikey olarak tekrarlanması istenebilir.

Eğer zemin resmi yatay olarak tekrarlanacaksa aşağıdaki gibi bir kod yazılmalıdır:

```
body {
   background-image: url("https://gelecegiyazanlar.turkcell.com.tr/sites/default/files/icerik/mt.jpg");
   background-repeat: repeat-x;
}
```

Zemin resminin pozisyonu ve tekrarı

Zemin resmi pozisyonu background-position özelliği ile belirlenir. Aşağıdaki kod örneğinde, zemin resmi mt.png nin tekrar ve pozisyon örneklerini görebilirsiniz:

```
body {

   background-image: url("https://gelecegiyazanlar.turkcell.com.tr/sites/default/files/icerik/mt.jpg");

   background-repeat: no-repeat;

}
```

```
body {

   background-image: url("https://gelecegiyazanlar.turkcell.com.tr/sites/default/files/icerik/mt.jpg");

   background-repeat: no-repeat;

   background-position: right top;

}
```

🔺 *Zemin kısayol (background shorthand) özelliği*

Yukarıda bahsedilen çeşitli özellikleri sadece background sözcüğü ile kısa ve toplu bir şekilde ifade etmek de mümkündür:

```
body {
   background: #fffeee url("https://gelecegiyazanlar.turkcell.com.tr/sites/default/files/icerik/mt.jpg") no-repeat left top;
}
```

Shorthand özelliği kullanıldığı zaman, özellik değerlerinin sırası aşağıdaki gibidir:

1. background-color
2. background-image
3. background-repeat
4. background-attachment
5. background-position

Aşağıdaki tabloda zemin özelliklerinin tümü listelenmiştir:

Özellik	Açıklama
background	Bütün zemin özelliklerini tek bir bildiride gerçekleştirir.
background-attachment	Zemin resmi sabit mi yoksa sayfanın diğer kısmında hareket edecek mi bunu belirler.
background-color	Bir elemanın zemin rengini belirler.
background-image	Bir eleman için zemin resmini belirler.
background-position	Zemin resminin başlangıç pozisyonunu belirler.
background-repeat	Zemin resminin nasıl tekrarlanacağını belirler.

Kazanım

- Öğrenciler bu bölümde zemin özelliklerini öğrenirler.

Eğitmen Yol Haritası

1	Öğrencilere zemin özelliklerinin anlatılması.
2	Konu hakkında yapılacak uygulamaların tanıtılması.
3	Öğrencilerle birlikte editör üzerinde uygulamaların yapılması.

CSS: METIN İŞLEMLERI

Müfredat Kodu / Ünite / Konu	İCT-301-2.16 / Html & Css / Metin İşlemleri
Hedeflenen Süre	30 Dakika

🚀 Metin İşlemleri

Metinlerin özellikleri CSS kullanılarak değiştirilebilir.

🔺 *Metin rengi (Text Color)*

Renk özelliği, bir metnin rengini atamak için kullanılır. CSS'te renk atama,

- Bir HEX değerle: "#ff0ee0",
- Bir RGB (red-green-blue, kırmızı-yeşil-mavi) değerle: "rgb(200,10,50)",
- Ya da bir renk adı belirtilerek gerçekleştirilebilir.

⚠️ Bir sayfa için varsayılan renk, body seçicisi ile tanımlanır.

```
body {
   color: green;
}

h1 {
   color: #00ffec;
}

h2 {
   color: rgb(255,110,210);
}
<!DOCTYPE html>
<html>
   <head>
      <style>
         body {
            color: pink;
         }

      h1 {
         color: #0effcd;
      }

      p.ex {
         color: rgb(110,220,155);
      }

      </style>
   </head>
```

```
    <body>
        <h1>ILGINC BIR RENK, HEXADECIMAL ATANMIS</h1>
        <p>RGB KARMASI BIR RENK</p>
        <p class="ex">BAKALIM HANGI RENK</p>

    </body>
</html>
```

ILGINC BIR RENK, HEXADECIMAL ATANMIS

RGB KARMASI BIR RENK

BAKALIM HANGI RENK

Metnin yerleştirilmesi

Text-align özelliği, metnin yatay yerleşimi için kullanılır. Metin ortalanabilir (centered), sola ya da sağa yanaşık olabilir veya ayrılan yere göre kendini ayarlayabilir.

```
h1 {
    text-align: center;
}

p.date {
    text-align: left;
}

p.main {
    text-align: justify;
}
<!DOCTYPE html>
<html>
    <head>
        <style>
            h1 {
                text-align: center;
            }

        p.date {
            text-align: left;
        }

        p.main {
            text-align: justify;
        }

        </style>
    </head>
```

```
    <body>
        <h1>CSS METIN YERLESTIRME</h1>
        <p class="date">EYLUL, 2014</p>
        <p class="main">YUMUSAK YONETILEN ULKENIN HALKI YALINDIR
        SERT YONETIM HALKI KURNAZLASTIRIR</p>
        <p><b>Not</b> KONFUCYUS'UN BU UNLU SOZU BUGUN DE GECERLI DEGIL MI?'</p>
    </body>
</html>
```

CSS METIN YERLESTIRME

EYLUL, 2014

YUMUSAK YONETILEN ULKENIN HALKI YALINDIR SERT YONETIM HALKI KURNAZLASTIRIR

Not KONFUCYUS'UN BU UNLU SOZU BUGUN DE GECERLI DEGIL MI?'

Metin dekorasyonu

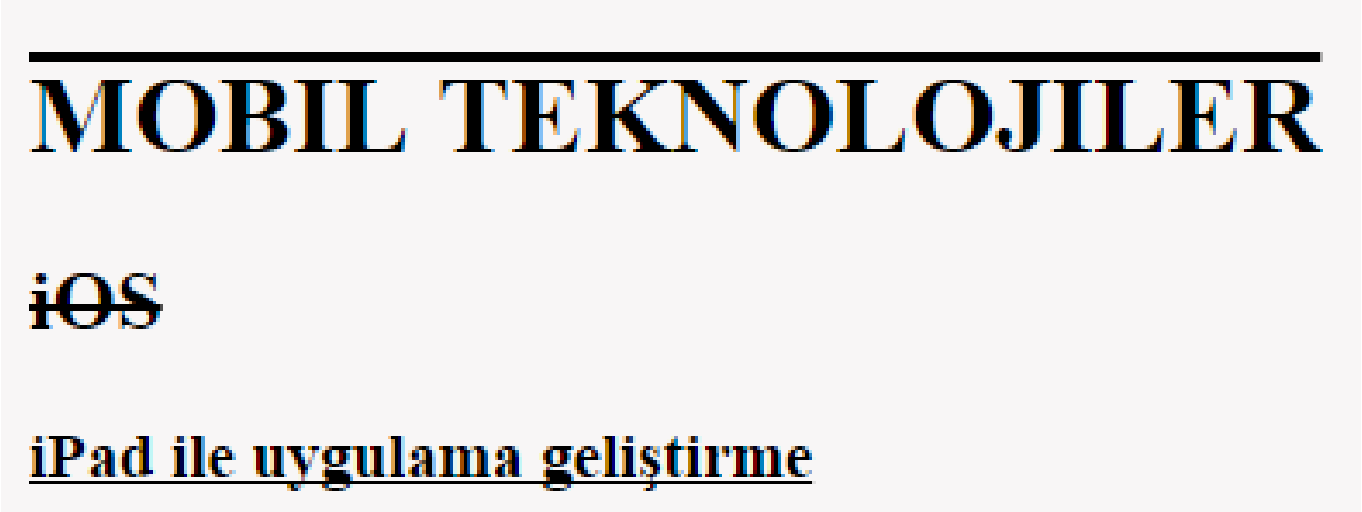
Text-decoration özelliği, metin içine bir dekorasyon ekleme ya da metinden bir dekorasyonu uzaklaştırma amacıyla kullanılır. text-decoration özelliği, genelde tasarım amacıyla bağlantılardan alt çizgileri uzaklaştırmak için kullanılır.

```
a {
    text-decoration: none;
}
```

Ayrıca metni dekore etmek için de kullanılabilir:

```
h1 {
    text-decoration: overline;
}

h2 {
    text-decoration: line-through;
}

h3 {
    text-decoration: underline;
}
<!DOCTYPE html>
<html>
    <head>
        <style>

        h1 {
            text-decoration: overline;
        }

        h2 {
            text-decoration: line-through;
```

```
        }

        h3 {
            text-decoration: underline;
        }
        </style>
    </head>

    <body>

        <h1>MOBIL TEKNOLOJILER</h1>
        <h2>iOS</h2>
        <h3>iPad ile uygulama geliştirme</h3>
    </body>
</html>
```



Metin dönüşümü

Metin dönüşüm (text-transform) özelliği, metin içinde büyük harf ve küçük harf karakterleri belirlemek için kullanılır. Ya karakterler büyük harfe veya küçük harfe çevrilir ya da her kelimenin ilk harfi büyük harf yapılır.

```
p.uppercase {
    text-transform: uppercase;
}

p.lowercase {
    text-transform: lowercase;
}

p.capitalize {
    text-transform: capitalize;
}
<!DOCTYPE html>
<html>
    <head>
        <style>
```

```
        p.uppercase {
          text-transform: uppercase;
        }

        p.lowercase {
           text-transform: lowercase;
        }

        p.capitalize {
           text-transform: capitalize;
        }
        </style>
    </head>

    <body>
        <p class="uppercase">ankara</p>
        <p class="lowercase">ANKARA</p>
        <p class="capitalize">ankara</p>
    </body>
</html>
```

Metin girintileme (Text Indentation)

Text-indent özelliği, metnin ilk satırının ne kadar içeriden yazılacağını belirler.

```
p {
   text-indent: 40px;
}
<!DOCTYPE html>
<html>
   <head>
      <style>
         p {
            text-indent: 50px;
         }

      </style>
   </head>

   <body>

      <p>Stefan Zweig. Musevi kökenli Avusturyalı yazar. 20. yüzyıl başında </br>Avrupa
entelektuel dünyasının önemli isimlerinden. Avusturya aşığı bu büyük edebiyatçı Hitler zulmünden
kaçtı; </br>Güney Amerika'ya sığındı. 1. Dünya Savaşı sona ermeden intihar etti.</p>

   </body>
</html>
```

Stefan Zweig. Musevi kökenli Avusturyalı yazar. 20. yüzyıl başında
Avrupa entelektuel dünyasının önemli isimlerinden. Avusturya aşığı bu büyük edebiyatçı Hitler zulmünden kaçtı;

Güney Amerika'ya sığındı. 1. Dünya Savaşı sona ermeden intihar etti.

Metin özellikleri

Özellik	Anlamı
color	Metnin rengini atama işlemini gerçekleştirir.
direction	Metnin yazma yönünü belirler.
letter-spacing	Metindeki karakterler arasındaki boşlukları artırır ya da azaltır.
line-height	Satır yüksekliğini atama işlemini gerçekleştirir.
text-align	Metnin yatay konumlandırılmasını belirler.
text-decoration	Metne eklenecek dekorasyonu belirler.
text-indent	Bir metin bloğundaki ilk satırın ne kadar içerden başlayacağını belirler.
text-shadow	Metne eklenecek gölge etkisini belirler.
text-transform	Metnin büyük-küçük harf durumunu belirler.
vertical-align	Bir elemanın düşey yerleşimini belirler.
white-space	Eleman içinde alfabe dışı karakterlerin nasıl işlem göreceğini belirler.
word-spacing	Bir metinde kelimeler arasında bulunan boşluk sayısını artırır ya da azaltır.

Kazanım

- Öğrenciler bu bölümde metin işlemlerini öğrenirler.

Eğitmen Yol Haritası

1	Öğrencilere metin işlemlerinin anlatılması.
2	Konu hakkında yapılacak uygulamaların tanıtılması.
3	Öğrencilerle birlikte editör üzerinde uygulamaların yapılması.

CSS: YAZI TIPI ÖZELLIKLERI

Müfredat Kodu / Ünite / Konu	İCT-301-2.17 / Html & Css / Yazı Tipi Özellikleri
Hedeflenen Süre	30 Dakika

🚀Yazı Tipi Özellikleri

CSS yazıtipi (font) özellikleri, yazıtipi ailesini, boyutu, kalın/italik olup olmama durumunu ve metnin biçemini (stil) tanımlamak için kullanılır.

🔺 *Yazı tipi Aileleri*

CSS'de, yazıtipi aileleri iki temel gruba ayrılır:

- Jenerik aile (generic family): Benzer görünüşteki yazıtipi ailesi ("Serif" ile "Monospace" gibi).
- Yazıtipi ailesi (font family): Spesifik bir yazıtipi ailesi ("Times New Roman" veya "Courier" gibi).

Jenerik aile	Yazıtipi ailesi	Açıklama
Serif	Times New Roman Georgia	Serifli yazıtipleri karakterlerin sonunda yer alan küçük çizgi ve kuyruklarla sonlanır.
Sans-serif	Arial Verdana	"Sans" Fransızca bir sözcük olup "olmaksızın" anlamına gelir. Bu aile karakter sonlarında çizgiler ve kuyruklar içermez.
Monospace	Courier New Lucida Console	Tüm monospace karakterler aynı genişliktedir.

🔺 *Yazı tipi ailesi*

Bir metnin yazıtipi ailesi, font-family özelliğiyle atanır. font-family özelliği, çeşitli yazıtipi adlarını bir yedek sistem gibi tutmalıdır. Tarayıcı ilk yazıtipini desteklemiyorsa bir sonraki yazıtipi (font) devreye sokulmalıdır. Yazıtipi ailesi birden fazla sözcükten oluşuyorsa çift tırnak sembolleri arasına alabilirsiniz.

⚠️ Birden fazla yazıtipi ailesi varsa, virgüllerle ayrılmalıdır

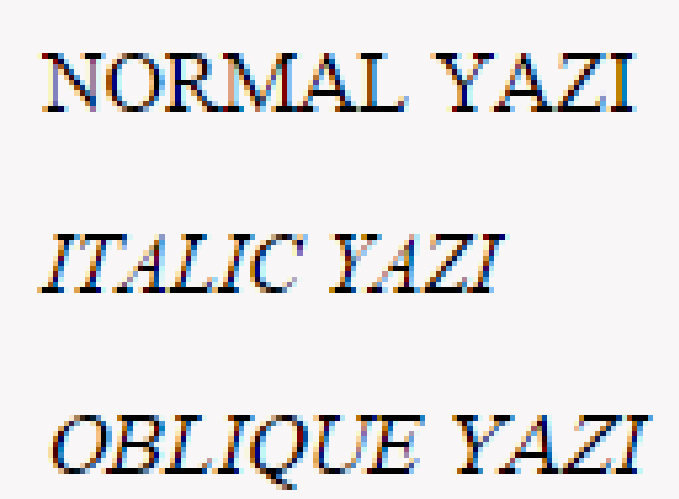
```
p {
   font-family: "Times New Roman", Times, serif;
}
<!DOCTYPE html>
<html>
   <head>
      <style>
         p.serif {
            font-family: "Times New Roman", Times, serif;
         }

      p.sansserif {
         font-family: Arial, Helvetica, sans-serif;
      }

      </style>
   </head>

   <body>
      <h1>CSS FONT AILESI</h1>
      <p class="serif">Times New Roman fontu ILE YAZILDI</p>
      <p class="sansserif">Arial fontu ILE YAZILDI.</p>
   </body>

</html>
```

▲ *Yazı tipi stili*

Font-style özelliği genellikle eğik bir metin oluşturmak için kullanılır. Bu özelliğin üç farklı değeri vardır:

1. normal: Metin normal görünür.
2. italic: Metin italik (eğik yazı) olarak görünür.
3. oblique: Metin biraz eğilmiş (italiğe çok benzer) olarak görünür.

```
p.normal {
   font-style: normal;
}

p.italic {
   font-style: italic;
}
p.oblique {
   font-style: oblique;
}
<!DOCTYPE html>
<html>
   <head>
      <style>
      p.normal {
            font-style: normal;
      }

      p.italic {
         font-style: italic;
      }


      p.oblique {
         font-style: oblique;
      }

      </style>
   </head>

   <body>
      <p class="normal">NORMAL YAZI</p>
      <p class="italic">ITALIC YAZI</p>
      <p class="oblique">OBLIQUE YAZI</p>
   </body>
</html>
```



Yazı tipi boyutu

Font-size özelliği, metnin harf boyutunu belirler.

> Bir sayfa içindeki <p> elemanlarını font-size ile <h> elemanlarına dönüştürmeyiniz ya da tersini yapmayınız. Başlıklar için daima HTML <h1>...<h6> etiketlerini, paragraflar içinse <p> etiketlerini kullanınız.

Font-size, mutlak (absolute) ya da göreceli (relative) olabilir.

Mutlak boyut:

- Metni belirtilen boyuta dönüştürür.
- Kullanıcının metnin boyutunu her tarayıcıda değiştirilmesine izin vermez.
- Çıktının fiziksel boyutu biliniyorsa faydalıdır.

Göreceli boyut:

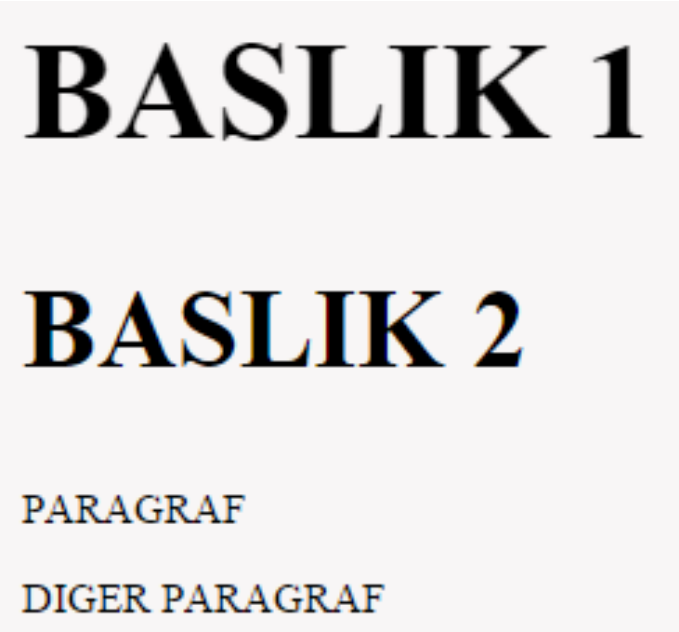
- İlgili elemanların boyutunu mutlak olarak değil göreceli olarak belirler.
- Kullanıcıya, tarayıcılarda metnin boyutunu değiştirmesi için izin verir.

Yazı tipi boyutunu piksel cinsinden atamak

Metin boyutunu piksel (px) cinsinden atarsanız, metin boyutu üzerinde tam kontrol sağlayabilirsiniz.

```
h1 {
   font-size: 50px;
}

h2 {
   font-size: 40px;
}

p {
   font-size: 15px;
}
<!DOCTYPE html>
<html>
   <head>
      <style>
      h1 {
         font-size: 50px;
      }

      h2 {
         font-size: 40px;
      }

      p {
         font-size: 15px;
      }
```

```
        </style>

    </head>

    <body>
        <h1>BASLIK 1</h1>
        <h2>BASLIK 2</h2>
        <p>PARAGRAF</p>
        <p>DIGER PARAGRAF</p>
    </body>
</html>
```

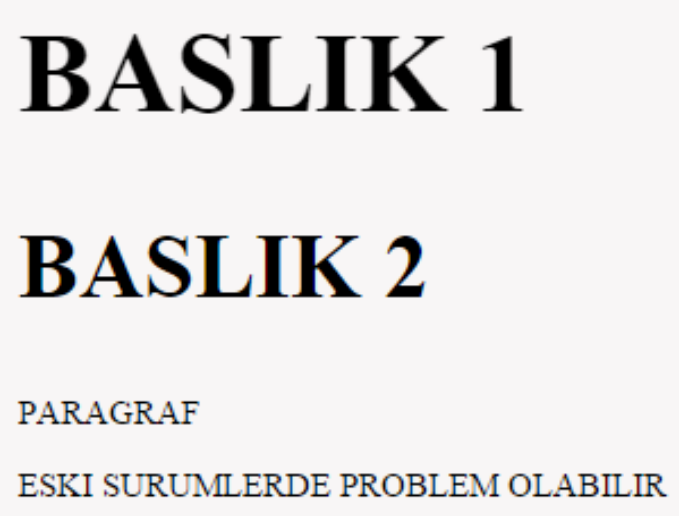


Yazı tipi boyutunu em birimi kullanarak atamak

Kullanıcıların, kullandıkları tarayıcının menüsünden metni yeniden boyutlandırabilmelerine (resize) olanak sağlamak amacıyla, birçok yazılım geliştirici piksel yerine em birimini kullanmayı tercih eder. em boyut birimi, W3C tarafından da önerilmektedir. Tarayıcılarda varsayılan metin boyutu 16px'tir. Buna göre, 1 em=16px olarak kullanılır.

```
h1 {
    font-size: 3.125em; /* 50px/16=2.5em */
}

h2 {
    font-size: 2.5em; /* 40px/16=2.5 */
}

p {
    font-size: 0.9375em; /* 15px/16=0.9375em */
}
<!DOCTYPE html>
<html>
    <head>
```

```
        <style>

        h1 {
            font-size: 3.125em; /* 50px/16=2.5em */
        }

        h2 {
           font-size: 2.5em; /* 40px/16=2.5 */
        }

        p {
           font-size: 0.9375em; /* 15px/16=0.9375em */
        }

        </style>
    </head>

    <body>
        <h1>BASLIK 1</h1>
        <h2>BASLIK 2</h2>
        <p>PARAGRAF</p>
        <p>ESKI SURUMLERDE PROBLEM OLABILIR</p>

    </body>
</html>
```



Yüzde ve em ikilisiyle boyut ayarı

<body> elemanına konacak yüzde türünde varsayılan yazıtipi (font) boyutu, tarayıcılarda boyut değiştirme konusunda karşılaşılan problemleri çözer.

```
body {
   font-size: 100%;
}
```

```
h1 {
   font-size: 3.125em; /* 50px/16=2.5em */
}

h2 {
   font-size: 2.5em; /* 40px/16=2.5 */
}

p {
   font-size: 0.9375em; /* 15px/16=0.9375em */
}
```

Özellik	Açıklama
font	Tek bir bildirimde tüm yazıtipi özelliklerinin atanmasını sağlar.
font-family	Bir metin için yazıtipi ailesini belirler.
font-size	Metnin yazıtipi boyutunu belirler.
font-style	Metin için yazıtipi stilini belirler.
font-variant	Bir metnin küçük veya büyük harflerle görüntüleneceğini belirler.
font-weight	Yazıtipinin ağırlığını belirler.

Kazanım

- Öğrenciler bu bölümde yazı tipi özelliklerini öğrenirler.

Eğitmen Yol Haritası

1	Öğrencilere yazı tipi özelliklerinin anlatılması.
2	Konu hakkında yapılacak uygulamaların tanıtılması.
3	Öğrencilerle birlikte editör üzerinde uygulamaların yapılması.

CSS: LINKLER

Müfredat Kodu / Ünite / Konu	İCT-301-2.18 / Html & Css / Linkler
Hedeflenen Süre	45 Dakika

🚀 Linkler

Web bağlantılarına (link) herhangi bir CSS özelliği kullanılarak (color, font-familiy, background vb) stil verilebilir. Bunun dışında linkler, hangi durumda (state) bulunduklarına göre de şekillendirilebilirler. Dört bağlantı durumu aşağıdaki gibidir:

- a:link – Normal ziyaret edilmemiş bağlantı.
- a:visited – Kullanıcının ziyaret ettiği bir bağlantı.
- a:hover – Kullanıcının faresinin üzerinde gezindiği bağlantı.
- a:active – Tıklanmış olan bağlantı.

```
/* ziyaret edilmemiş bağlantı */
a:link {
   color: #FF0000;
}

/* ziyaret edilmiş bağlantı */
a:visited {
   color: #00FF00;
}

/* fare gezdirilen bağlantı */
a:hover {
   color: #FF00FF;
}

/* seçilmiş link */
a:active {
   color: #0000FF;
}
<!DOCTYPE html>
<html>
   <head>
      <style>
         /* ziyaret edilmeyen bağlantı */
         a:link {
            color: #FF0000;
         }

      /* ziyaret edilen bağlantı */
      a:visited {
         color: #00FF00;
      }

      /* fare gezinen bağlantı */
```

```
        a:hover {
            color: #FF00FF;
        }

        /* seçilmiş bağlantı */
        a:active {
            color: #0000FF;
        }
        </style>
    </head>

    <body>
        <p><b><a href="sec.html" target="_blank">TÜGVA İCATHANE</a></b></p>
    </body>
</html>
```

TÜGVA İCATHANE

Çeşitli bağlantı (link) durumları için stil tanımı yaptığınızda, bazı sıra kuralları mevcuttur:

- a:hover, a:link ve a:visited'ten sonra gelir.
- a:active, a:hover'dan sonra gelir.

▲ *Yaygın bağlantı stilleri*

Yukardaki örnekte, duruma göre bağlantının renk değiştirdiğini gördük. Bağlantılara stil vermek için kullanılan diğer yaygın yöntemleri de aşağıda vereceğiz:

▲ *Metin dekorasyonu*

Text-decoration özelliği, yaygın olarak linklerden alt çizgi karakterini uzaklaştırmak için kullanılır.

```
a:link {
    text-decoration: none;
}

a:visited {
    text-decoration: underline;
}

a:hover {
    text-decoration: none;
}

a:active {
   text-decoration: underline;
```

```
}
<!DOCTYPE html>
<html>
  <head>
    <style>
      a:link {
        text-decoration: none;
      }


    a:visited {
      text-decoration: none;
    }

    a:hover {
      text-decoration: underline;
    }

    a:active {
      text-decoration: underline;
    }
    </style>
  </head>

  <body>
    <p><b><a href="sec.html" target="_blank">This is a link</a></b></p>
    <p><b>Not</b> a:hover, a:link ve a:visited tan sonra gelmeli</p>
    <p><b>Not</b> a:active, a:hover'dan sonra gelmeli</p>
  </body>
</html>
```

This is a link

Not a:hover, a:link ve a:visited tan sonra gelmeli

Not a:active, a:hover'dan sonra gelmeli

Zemin rengi

Background-property özelliği, bağlantılar (link) için zemin rengini belirler.

```
a:link {
  background-color: #B2EE94;
}

a:visited {
  background-color: #FFFF89;
}

a:hover {
```

```
    background-color: #FA704D;
}

a:active {
    background-color: #FF704D;
}
<!DOCTYPE html>
<html>
    <head>
        <style>
            a:link {
                background-color: #B2FF99;
            }

        a:visited {
            background-color: #FFFF85;
        }

        a:hover {
            background-color: #FF704D;
        }

        a:active {
            background-color: #FF704D;
        }

        </style>

    </head>

    <body>
        <p><b><a href="sec.html" target="_blank">Millet Kütüphanesi Link </a></b></p>
        <p><b>Not</b> a:hover, a:link ve a:visited tan sonra gelmeli</p>
        <p><b>Not</b> a:active, a:hover dan sonra gelmeli</p>
    </body>

</html>
```

Millet Kütüphanesi Link

Not a:hover, a:link ve a:visited tan sonra gelmeli

Not a:active, a:hover dan sonra gelmeli

Kazanım

- Öğrenciler bu bölümde linkler ile çalışmayı öğrenirler.

Eğitmen Yol Haritası

1	Öğrencilere link özelliklerinin anlatılması.
2	Konu hakkında yapılacak uygulamaların tanıtılması.
3	Öğrencilerle birlikte editör üzerinde uygulamaların yapılması.

CSS: LISTELER

Müfredat Kodu / Ünite / Konu	İCT-301-2.19 / Html & Css / Listeler
Hedeflenen Süre	45 Dakika

🚀 Listeler

CSS listesi özellikleri, sıralı listeler için farklı liste elemanı işaretçileri, sırasız listeler için farklı liste elemanı işaretçileri, liste elemanı işaretçisi olarak da bir resim atama işlemlerini gerçekleştirir.

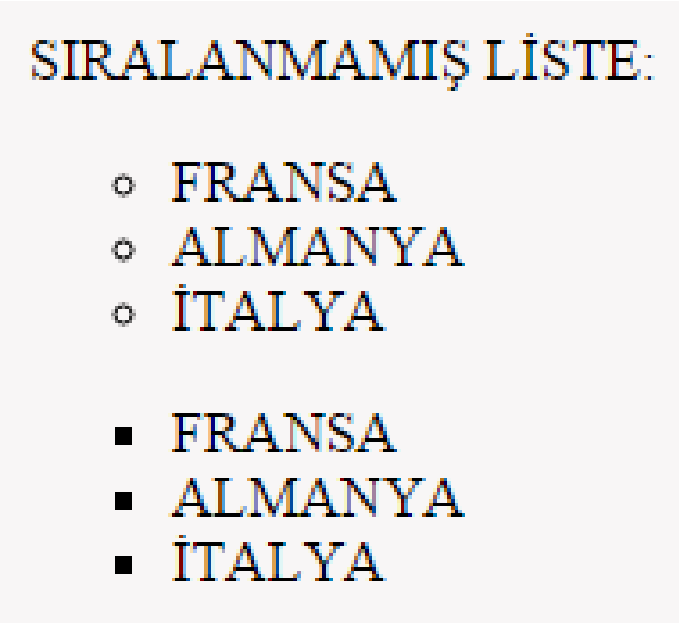
HTML'de iki tip liste vardır:

1. Sırasız liste: Liste elemanları "**.**" (bullet) ile işaretlenir.
2. Sıralı liste: Liste elemanları harf ya da rakamla işaretlenir.

Liste elemanı işaretçisinin tipi, list-style-type özelliğiyle belirlenir.

```
ul.a {
   list-style-type: circle;
}

ul.b {
   list-style-type: square;
}

ol.c {
   list-style-type: upper-roman;
}

ol.d {
   list-style-type: lower-alpha;
}
<!DOCTYPE html>
<html>
   <head>
      <style>
         ul.a {
            list-style-type: circle;
        }

      ul.b {
         list-style-type: square;
      }

      ol.c {
         list-style-type: upper-roman;
      }

      ol.d {
```

```
        list-style-type: lower-alpha;
      }

      </style>
    </head>
    <body>
      <p>SIRALANMAMIŞ LİSTE:</p>
      <ul class="a">
        <li>FRANSA</li>
        <li>ALMANYA</li>
        <li>İTALYA</li>
      </ul>

      <ul class="b">
        <li>FRANSA</li>
        <li>ALMANYA</li>
        <li>İTALYA</li>
      </ul>

      <p>SIRALANMIŞ LİSTE:</p>

      <ol class="c">
        <li>FRANSA</li>
        <li>ALMANYA</li>
        <li>İTALYA</li>
      </ol>

      <ol class="d">
        <li>FRANSA</li>
        <li>ALMANYA</li>
        <li>İTALYA</li>
      </ol>
    </body>
  </html>
```



SIRALANMIŞ LİSTE:

I. FRANSA
II. ALMANYA
III. İTALYA

a. FRANSA
b. ALMANYA
c. İTALYA

Bir liste elemanı işaretleyicisi olarak bir resim kullanma

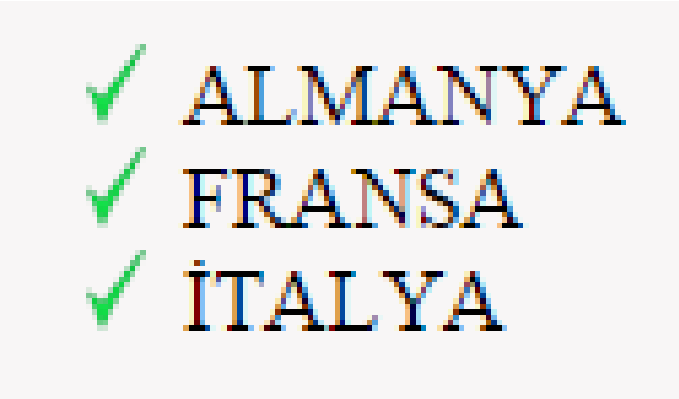
Bir resmi bir liste elemanı işaretleyicisi olarak kullanmak için list-style-image özelliği kullanılır.

```
ul {

  list-style-image: url('xx.gif');

}
```

Yukarıdaki resim tüm tarayıcılarda aynen görüntülenmez; Internet Explorer ve Opera tarayıcıları; Firefox, Chrome ve Safari'ye göre resmi biraz daha büyük görüntüler.

```
<!DOCTYPE html>

<html>
  <head>
    <style>

      ul {
        list-style-image: url('mit.bmp');
      }

    </style>
  </head>

  <body>
    <ul>
      <li>ALMANYA</li>
      <li>FRANSA</li>
      <li>İTALYA</li>
```

```
        </ul>
    </body>
</html>
```



Liste Özellikleri	Açıklama
list-style	Bütün liste özelliklerini tek bir bildirim içinde atar.
list-style-image	Bir resmi liste elemanı işaretleyicisi olarak atar.
list-style-position	Liste elemanı işaretleyicisinin içerik akışının içinde mi yoksa dışında mı olacağını belirler.
list-style-type	Liste elemanı işaretleyicisinin tipini belirler.

📘 **Kazanım**

- Öğrenciler bu bölümde listelerin özelliklerini ve işlevlerini öğrenirler.

👩‍🏫 Eğitmen Yol Haritası

1	Öğrencilere liste özelliklerinin anlatılması.
2	Konu hakkında yapılacak uygulamaların tanıtılması.
3	Öğrencilerle birlikte editör üzerinde uygulamaların yapılması.

CSS: TABLOLAR

Müfredat Kodu / Ünite / Konu	İCT-301-2.20 / Html & Css / Tablolar
Hedeflenen Süre	45 Dakika

🚀 Tablolar

CSS'in sağladığı araçlarla HTML'in tablo oluşturma olanakları büyük ölçüde iyileştirilmiştir. Bu eğitim içeriği ve alt içeriklerinde tabloları nasıl daha güzel ve kullanışlı tasarlayacağınızı göstereceğiz.

🔺 *Tablo sınırları*

CSS'te tablo sınırlarını belirlemek için border özelliğini kullanmalısınız.

```
table, th, td {
  border: 2px solid black;
}
```

⚠️ Aşağıdaki örnek, tablo, th ve td elemanları için siyah sınırlar belirliyor:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      table, th, td {
        border: 2px solid black;
      }
    </style>
  </head>
  <body>

    <table>
      <tr>
        <th>ÜLKE</th>
        <th>NÜFUS</th>
      </tr>
      <tr>
        <td>AZERBAYCAN</td>
        <td>10,14 Milyon</td>
      </tr>
      <tr>
        <td>ARNAVUTLUK</td>
        <td>2,812 milyon</td>
      </tr>
    </table>
```

```
    </body>
</html>
```

ÜLKE	NÜFUS
AZERBAYCAN	10,14 Milyon
ARNAVUTLUK	2,812 milyon

Gördüğünüz gibi sınırları ayarlamak hem daha kolay hem daha esnetik oldu.

⚠ Yukarıdaki örnek tabloda çift sınır çizgisi mevcuttur. Bunun sebebi, hem tablo hem de th/td elemanlarının ayrı sınırlara sahip olmasıdır. Tabloda tek bir sınır göstermek için border-collapse özelliğini kullanınız.

▲ *Sınırları daraltma*

Border-collapse özelliği, tablo sınırlarının tek bir sınıra mı indirgeneceği ya da ayrılacağı konusunu belirlemek üzere atama yapar. Şimdi bunu örneğimizle görelim:

```
table {
    border-collapse: collapse;
}

table, th, td {
    border: 1px solid black;
}

ÖRNEK HTML5 KODU:
<!DOCTYPE html>
<html>
    <head>
        <style>
            table {
                border-collapse: collapse;
            }

        table, td, th {
            border: 2px solid black;
        }
        </style>
    </head>
    <body>

        <table>
            <tr>
                <th>ULKE</th>
                <th>NUFUS</th>
            </tr>
            <tr>
```

```
            <td>ALMANYA</td>
            <td> 83,2 Milyon </td>
        </tr>
        <tr>
            <td>AZERBAYCAN</td>
            <td> 10,14 Milyon </td>
        </tr>
    </table>

    <p><b>EK BILGI:</b> !DOCTYPE IFADESI KULLANILMAZSA, border-collapse OZELLIGI
<br>IE8 VE ONCEKİ SURUMLERDE BEKLENMEYEN SONUCLAR URETEBILIR.</p>

  </body>
</html>
```

ULKE	NUFUS
ALMANYA	83,2 Milyon
AZERBAYCAN	10,14 Milyon

EK BILGI: !DOCTYPE IFADESI KULLANILMAZSA, border-collapse OZELLIGI
IE8 VE ONCEKİ SURUMLERDE BEKLENMEYEN SONUCLAR URETEBILIR.

🛫Genişlik ve Yükseklik

Bir tablonun ebatları, width (genişlik) ve height (yükseklik) özellikleriyle belirlenir. Aşağıdaki örnek kod ile bunlarla ilişkili atamalar yapacağız:

```
table {
  width: 100%;
}

th {
  height: 60px;
}
<!DOCTYPE html>
<html>
  <head>
    <style>
      table, td, th {
        border: 2px solid red;
      }
    table {
      width: 100%;
    }
    th {
      height: 60px;
    }
    </style>
  </head>
  <body>
    <table>
      <tr>
```

```
            <th>ÜLKE</th>
            <th>NÜFUS</th>
            <th>DİL</th>
        </tr>
        <tr>
            <td>Almanya</td>
            <td>80</td>
            <td>Almanca</td>
        </tr>
        <tr>
            <td>Fransa</td>
            <td>55</td>
            <td>Fransızca</td>
        </tr>
        <tr>
            <td>Azerbaycan</td>
            <td>8</td>
            <td>Almanca, Fransızca, İtalyanca, Romanca</td>
        </tr>
    </table>
  </body>
</html>
```

Kazanım

- Öğrenciler bu bölümde tablo kullanımını öğrenirler.

Eğitmen Yol Haritası

1	Öğrencilere tablo özelliklerinin anlatılması.
2	Konu hakkında yapılacak uygulamaların tanıtılması.
3	Öğrencilerle birlikte editör üzerinde uygulamaların yapılması.

CSS: TABLO ELEMANLARI

Müfredat Kodu / Ünite / Konu	İCT-301-2.20.1 / Html & Css Tablo / Tablo Elemanları
Hedeflenen Süre	45 Dakika

🚀 Tablo Elemanları

Tablo oluştururken bazı özelliklere ihtiyaç duyarız. Bunlar;

- Metin yerleştirme
- Tablo doldurma
- Tablo rengi

🔺 *Metin Yerleştirme*

Tabloya metin yerleştirmek için, text-align ve vertical-align özellikleri kullanılır. text-align özelliği, yatay yerleştirmeyi belirler; değerleri, left (sol), right (sağ) ya da center (merkez, ortala) olabilir.

text-align özelliği için şöyle örnek verebiliriz:

```
td {
   text-align: right;
}
```

Vertical-align özelliğiyse hücre içinde düşey doğrultudaki atamayı gerçekleştirir; değerler, top (üstte), bottom (altta) veya middle (ortada) olabilir.

```
td {
   height: 50px;
   vertical-align: bottom;
}
<!DOCTYPE html>
<html>
   <head>
      <style>
         table, td, th {
            border: 1px solid black;
         }

      td {
         height: 60px;
         vertical-align: bottom;
      }
      td {
         text-align: right;
      }
      </style>
```

```
    </head>
    <body>

      <table>
        <tr>
          <th>AD</th>
          <th>SOYAD</th>
          <th>MAAS</th>
        </tr>
        <tr>
          <td>ALI</td>
          <td>CAN</td>
          <td>2000</td>
        </tr>
        <tr>
          <td>AYSE</td>
          <td>OKAN</td>
          <td>1800</td>
        </tr>
        <tr>
          <td>MERT</td>
          <td>ALSAN</td>
          <td>3000</td>
        </tr>
        <tr>
          <td>BERNA</td>
          <td>CILEM</td>
          <td>2600</td>
        </tr>
      </table>

    </body>
  </html>
```

AD	SOYAD	MAAS
ALI	CAN	2000
AYSE	OKAN	1800

MERT	ALSAN	3000
BERNA	CILEM	2600

▲ *Tablo Doldurma*

Bir tablonun sınırı ile içerik arasındaki boşluğu ayarlamak için td ve th elemanları üzerinde padding özelliği kullanılır. Şimdi örneğimizle içerideki boşluğu 17 piksel (px) yapalım. Bunun için aşağıda gördüğünüz gibi, istediğimiz boşluk miktarını td parametresinin içinde padding özelliğine piksel cinsinden söyleyebiliriz:

```
td {
   padding: 17px;
}
```

Şimdi HTML dosyasında nasıl kullandığımıza bakalım:

```
<!DOCTYPE html>
<html>
   <head>
      <style>
         table, td, th {
            border: 2px solid black;
         }

      td {
         padding: 17px;
      }
      </style>
   </head>
   <body>

      <table>
         <tr>
            <th>AD</th>
            <th>SOYAD</th>
            <th>MAAS</th>
         </tr>
         <tr>
            <td>ALI</td>
            <td>OKAN</td>
            <td>2000</td>
```

```
        </tr>
        <tr>
          <td>VELI</td>
          <td>CAN</td>
          <td>3000</td>
        </tr>
        <tr>
          <td>AYSE</td>
          <td>BIRDAL</td>
          <td>2400</td>
        </tr>
        <tr>
          <td>BERNA</td>
          <td>SEN</td>
          <td>1500</td>
        </tr>
      </table>

    </body>
</html>
```

AD	SOYAD	MAAS
ALI	OKAN	2000
VELI	CAN	3000
AYSE	BIRDAL	2400
BERNA	SEN	1500

Tablo Rengi

Tablo renklerini istediğiniz gibi değiştirebilirsiniz. Bu zamana kadar göreceğiniz üzere sınır renkleri, metin ve zemin rengi atamaları, <th> elemanının içinde gerçekleştirilmektedir:

```
table, td, th {
   border: 2px solid yellow;
}

th {
   background-color: pink;
```

```
    color: white;
}
```

background-color ve color özellikleriyle arka zemin ve yazı renklerini kolayca ayarlayabilirsiniz. Dilerseniz, tablonun border özelliğini kullanarak çerçevenin kalınlığı ve renginin nasıl değiştirildiğini de deneyebilirsiniz.

Şimdi bir HTML belgesinde bu özelliklerin nasıl belirlendiğini görelim:

```
<!DOCTYPE html>
<html>
    <head>
        <style>
            table, td, th {
                border: 2px solid pink;
            }

        th {
            background-color: yellow;
            color: white;
        }
        </style>
    </head>

    <body>
        <table>
            <tr>
                <th>AD</th>
                <th>SOYAD</th>
                <th>MAAS</th>
            </tr>
            <tr>
                <td>ALI</td>
                <td>OKAN</td>
                <td>2000</td>
            </tr>
            <tr>
                <td>VELI</td>
                <td>CAN</td>
                <td>3000</td>
            </tr>
            <tr>
                <td>AYSE</td>
                <td>BIRDAL</td>
                <td>2400</td>
            </tr>
            <tr>
                <td>BERNA</td>
                <td>SEN</td>
                <td>1500</td>
            </tr>
        </table>
```

```
    </body>
  </html>
```

AD	SOYAD	MAAS
ALI	OKAN	2000
VELI	CAN	3000
AYSE	BIRDAL	2400
BERNA	SEN	1500

Kazanım

- Öğrenciler bu bölümde tabloya renk verme, tablonun içini doldurma ve metin yerleştirmelerini öğrenirler.

Eğitmen Yol Haritası

1	Öğrencilere tabloya yardımcı özelliklerinin anlatılması.
2	Konu hakkında yapılacak uygulamaların tanıtılması.
3	Öğrencilerle birlikte editör üzerinde uygulamaların yapılması.

CSS: KUTU MODELI

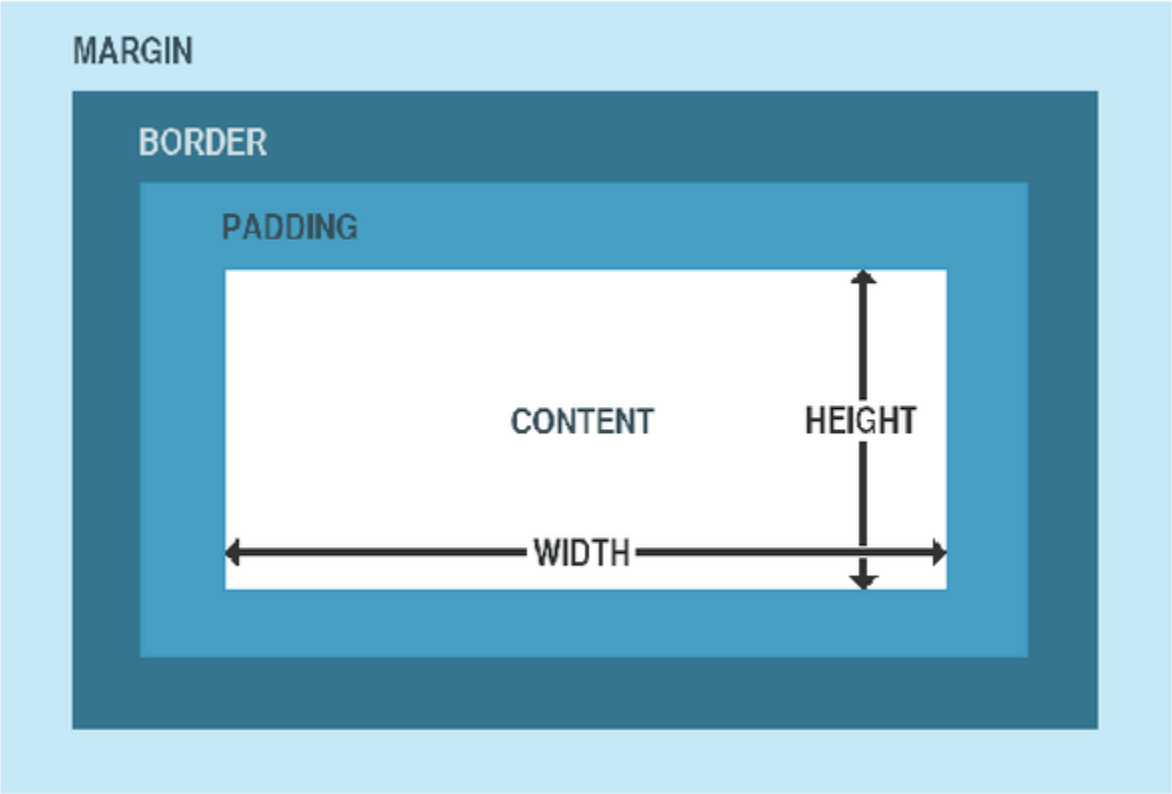
Müfredat Kodu / Ünite / Konu	İCT-301-2.21 / Html & Css / Kutu Modeli
Hedeflenen Süre	45 Dakika

🚀 Kutu Modeli

Tüm CSS elemanları birer kutu olarak düşünülebilir. CSS'te tasarım ve düzenden bahsediliyorsa, "Kutu Modeli" terimi kullanılır. CSS Kutu Modeli esasında HTML bileşenlerinin etrafını saran ve margin (kenar mesafesi), border (sınır), padding (boşluk) ve content (içerikten) oluşan elemanlardan ibarettir.

CSS Kutu Modeli, bize elemanların etrafına bir sınır ekleme (border) ya da elemanlar arasında bir boşluk (padding) tanımlama olanağı sağlar.

Aşağıdaki şema, CSS Kutu Modeli elemanlarını daha iyi kavramanıza yardımcı olacaktır:



Bu şemadaki elemanların açıklamalarına gelecek olursak;

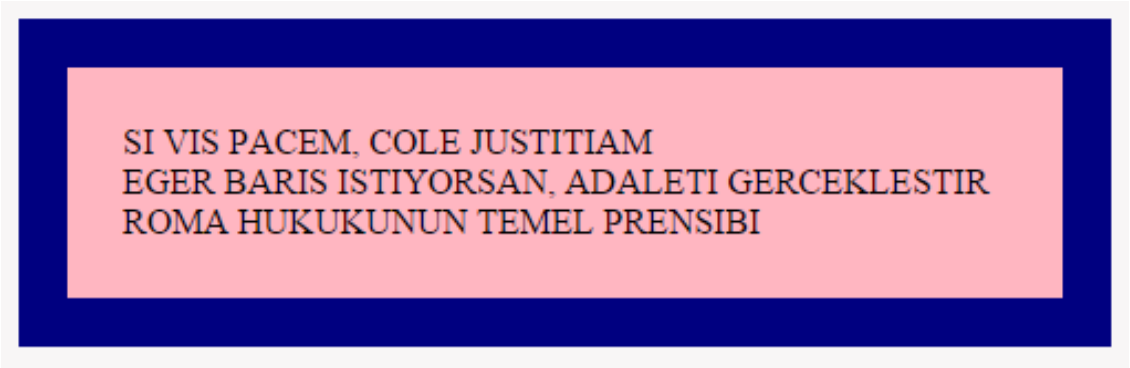
- **Content**: Resim ve metnin görüneceği kutunun içeriğidir.
- **Padding**: İçerik etrafında bir alanı temizleyerek boşluk oluşturur. Padding daima saydamdır.
- **Border**: Boşluk ve içeriğin etrafında yer alan sınır çizgisidir.
- **Margin**: Sınırın dışında kalan alandır. Daima saydamdır.

```
div {
   width: 200px;
   padding: 30px;
```

```
    border: 20px solid red;
    padding: 25px;
}
```

CSS Kutu Modeli elemanlarının bir HTML belgesi içinde nasıl kullandığını görmek için aşağıdaki kodu inceleyebilirsiniz:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      div {
        background-color: lightpink;
        width: 400px;
        padding: 30px;
        border: 22px solid navy;
        padding: 25px;
      }
    </style>
  </head>
  <body>

    <div>SI VIS PACEM, COLE JUSTITIAM<br>
    EGER BARIS ISTIYORSAN, ADALETI GERCEKLESTIR<br>
    ROMA HUKUKUNUN TEMEL PRENSIBI</div>

  </body>
</html>
```



Kazanım

- Öğrenciler bu bölümde kutu modelini öğrenirler.

Eğitmen Yol Haritası

1	Öğrencilere kutu modelinin anlatılması.
2	Konu hakkında yapılacak uygulamaların tanıtılması.
3	Öğrencilerle birlikte editör üzerinde uygulamaların yapılması.

NOTLAR

CSS: BORDER

Müfredat Kodu / Ünite / Konu	İCT-301-2.22 / Html & Css / Border
Hedeflenen Süre	30 Dakika

🚀 CSS (Sınır) Border

CSS Sınır (border) özellikleri, bir elemanın sınırlarına ait biçim (stil), boyut ve renkleri belirler.

🔺 *Sınır Biçimi (Border Style)*

Border-style özelliği, hangi tür sınır çizgisinin görüntüleneceğini belirler.

CSS kodu içindeki border-style değerlerini tanımlarken aşağıdaki özellikleri kullanabilirsiniz:

- none: Sınırı kaldırır.
- dotted: Noktalı bir sınır belirler.
- dashed: Kesikli çizgi şeklinde bir sınır belirler.
- solid: Katı sınır çizgisi belirler.
- double: İki sınır tanımlar. Her iki sınır çizgisi de border-width değeriyle belirlenir.
- groove: 3 boyutlu (3D) ve yivi dışa dönük sınır çizgisi tanımlar. Oluşacak etkinin derecesi, border-color özelliğine bağlıdır.
- ridge: 3D ortalanmış sınır çizgisi tanımlar. Etkinin derecesi, border-color değerine bağlıdır.
- inset: 3D içe dönük biçimde sınır çizgisi belirler. Etkinin derecesi, border-color değerine bağlıdır.
- outset: 3D tamamen dışa dönük sınır çizgisi tanımlar. Etkinin derecesi, border-color değerine bağlıdır.

🔺 *Sınır Genişliği*

CSS kodlamada border-width özelliği, sınır çizgisinin genişliğini atamak için kullanılır.

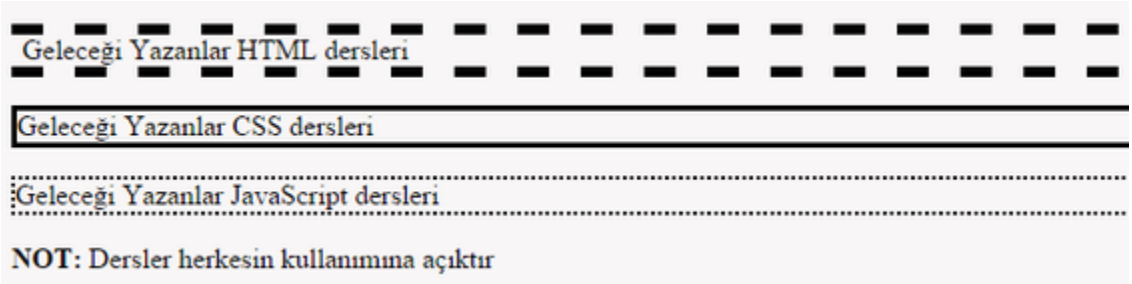
⚠️ Genişlik piksel cinsinden atanır ya da önceden tanımlanmış üç değerden biri olarak atanabilir. Bu üç değer, thin (ince), medium (orta) ve thick (kalın) şeklindedir.

```
p.one {
   border-style: dashed;
   border-width: 6px;
}

p.two {
```

```
    border-style: dashed;
    border-width: medium;
}
```

Şimdi bu tanımlamayı bir örnekle nasıl kullandığımıza bakalım:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      p.bir {
        border-style: dashed;
        border-width: 6px;
      }

    p.iki {
      border-style: solid;
      border-width: medium;
    }

    p.uc {
      border-style: dotted;
      border-width: 2px;
    }
    </style>
  </head>
  <body>

    <p class="bir">Geleceği Yazanlar HTML dersleri</p>
    <p class="iki">Geleceği Yazanlar CSS dersleri</p>
    <p class="uc">Geleceği Yazanlar JavaScript dersleri</p>
    <p><b>NOT: </b>Dersler herkesin kullanımına açıktır</p>

  </body>
</html>
```



▲ *Sınır Çizgisi Rengi*

CSS kodlamada border-color özelliği, sınır çizgisinin rengini atamak için kullanılır. Renk aşağıdaki yollardan biriyle atanır:

- name: renk adı (red,blue) kullanılarak renk atamak.

- RGB: Red-Green-Blue renk uzayı ile renk atamak: "rgb(255,111,30)" gibi.
- Hex: hexadecimal (onaltılık) değer ile renk atamak: "#ff0ec0" gibi.

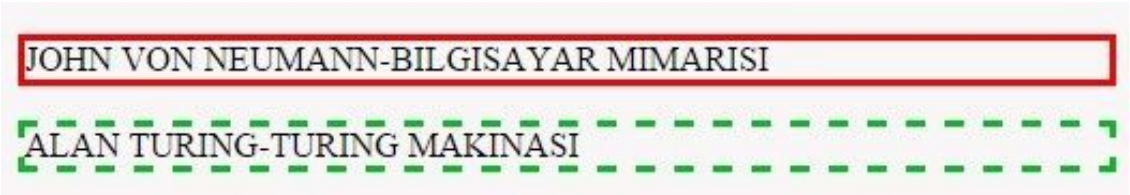
Sınır çizgisini dilerseniz şeffaf yani "transparent" olarak da atayabilirsiniz.

Örnek kullanımı aşağıdaki gibidir:

```
p.one {
   border-style: solid;
   border-color: blue;
}

p.two {
   border-style: dotted;
   border-color: #98bf21;
}
```

Şimdi de bir HTML içeriğinde bunu nasıl uygulayacağımızı görelim:

```
<!DOCTYPE html>
<html>
   <head>
      <style>
         p.bir {
            border-style: solid;
            border-color: red;
         }

      p.iki {
         border-style: dashed;
         border-color: #00bf21;
      }
      </style>
   </head>
   <body>

      <p class="bir">JOHN VON NEUMANN-BILGISAYAR MIMARISI</p>
      <p class="iki">ALAN TURING-TURING MAKINASI</p>
      <p><b>NOT:</b> ALBERT EINSTEIN GERCEK BIR DAHI</p>

   </body>
</html>
```



NOT: ALBERT EINSTEIN GERCEK BIR DAHI

Kazanım

- Öğrenciler bu bölümde border özelliklerini öğrenirler.

Eğitmen Yol Haritası

1	Öğrencilere border özelliklerinin anlatılması.
2	Konu hakkında yapılacak uygulamaların tanıtılması.
3	Öğrencilerle birlikte editör üzerinde uygulamaların yapılması.

CSS: PADDING

Müfredat Kodu / Ünite / Konu	İCT-301-2.23 / Html & Css / Padding
Hedeflenen Süre	30 Dakika

🚀 Padding

CSS padding özellikleri, elemanların etrafındaki boşluğu tanımlar.

Padding, bir eleman etrafındaki bir alanı siler (border'ın dışında). Padding'in bir zemin rengi yoktur ve tamamı ile saydamdır. Ayrı özellikler kullanılarak, üst (top), sağ (right), alt (bottom) ve sol (left) padding'ler bağımsız olarak değiştirilebilir. Ayrıca bir kısayol (shorthand) padding'i ile tümü birden de atanabilir.

🔍 *Mümkün değerler*

ÖZELLİK	AÇIKLAMA
auto	Tarayıcı bir padding hesaplar.
length	Bir padding piksel (px), punto (pt), santimetre (cm) ve diğer türlerde belirlenir. Öntanımlı değer 0 pikseldir.
%	Padding boyutunu ilgili eleman büyüklüğüne oranla yüzde cinsinden hesaplar.
inherit	Padding'in ebeveyn elemandan miras alacağını belirler.

🔺 *Ayrı Ayrı Atamalar*

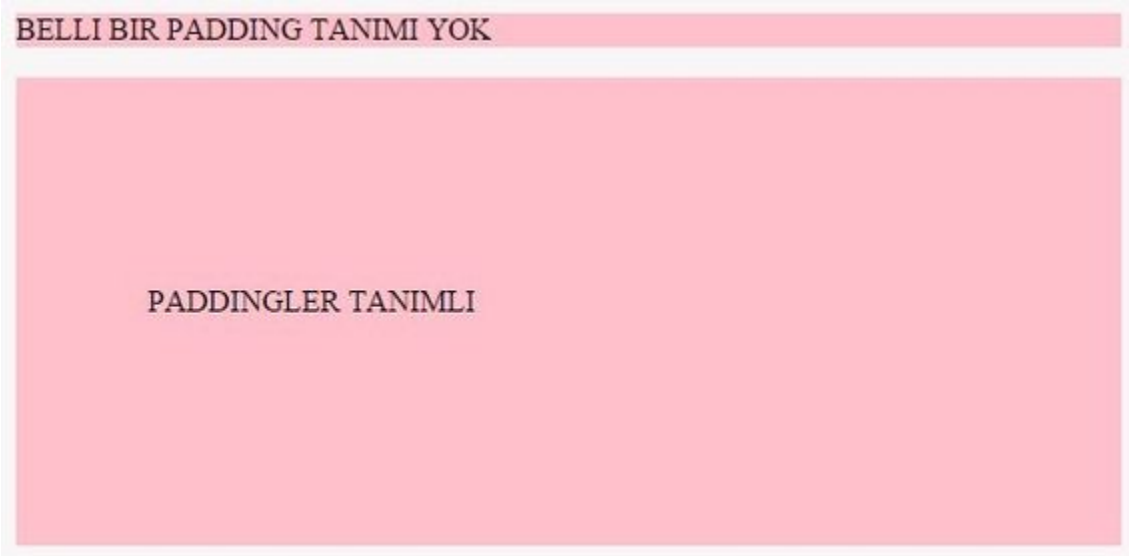
CSS'te, bir elemanın farklı kenarlarına farklı padding'ler atamak mümkündür. Şimdi kullanımını inceleyelim:

```
p {
   padding-top: 110px;
   padding-bottom: 120px;
   padding-right: 140px;
   padding-left: 70px;
}
```

Şimdi de bunu bir HTML belgesinde nasıl kullandığımıza bakalım:

```
<!DOCTYPE html>
<html>
   <head>
      <style>
         p {
            background-color: pink;
         }
```

```
        p.ex {
            padding-top: 110px;
            padding-bottom: 120px;
            padding-right: 140px;
            padding-left: 70px;
        }
        </style>
    </head>
    <body>

        <p>BELLI BIR PADDING TANIMI YOK</p>
        <p class="ex">PADDINGLER TANIMLI</p>
    </body>
</html>
```



Padding Özellikleri

ÖZELLİK	AÇIKLAMA
padding	Tek bir bildirimde tüm padding özelliklerini atayan kısayol özelliği.
padding-bottom	Bir elemanın alt paddingini atar.
padding-left	Bir elemanın sol paddingini atar.
padding-right	Bir elemanın sağ paddingini atar.
padding-top	Bir elemanın üst paddingini atar.

Kazanım

- Öğrenciler bu bölümde paddingi kavrar ve özelliklerini öğrenirler.

Eğitmen Yol Haritası

1	Öğrencilere padingin anlatılması.

2	Konu hakkında yapılacak uygulamaların tanıtılması.
3	Öğrencilerle birlikte editör üzerinde uygulamaların yapılması.

NOTLAR

CSS: BOYUTLAR

Müfredat Kodu / Ünite / Konu	İCT-301-2.24 / Html & Css / Boyutlar
Hedeflenen Süre	30 Dakika

🚀 Boyutlar

CSS boyut (dimension) özellikleri bir elemanın genişlik ve yüksekliğini kontrol etmenizi sağlar.

Konuyu bir örnekle anlatamaya başlamak daha doğru olacak:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
    p.ex {
      height: 120px;
      width: 120px;
    }
    </style>
  </head>
  <body>
      <p class="ex">BU PARAGRAF 120x120 BOYUTUNDA</p>
      <p>BU PARAGRAFTA YUKSEKLIK VE GENISLIK TANIMLANMADI</p>
      <p>BU PARAGRAFTA DA YUKSEKLIK VE GENISLIK TANIMLANMADI</p>
  </body>
</html>
```

BU PARAGRAF
120x120
BOYUTUNDA

BU PARAGRAFTA YUKSEKLIK VE GENISLIK TANIMLANMADI

BU PARAGRAFTA DA YUKSEKLIK VE GENISLIK TANIMLANMADI

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      p {
        max-width: 30%;
        background-color: pink;
      }
```

```
        </style>
    </head>
    <body>
        <p>BIRINCI SATIR<br>
        IKINCI SATIR<br>
        UCUNCU SATIR</p>
    </body>
</html>
```

BIRINCI SATIR
IKINCI SATIR
UCUNCU SATIR

Boyut Özellikleri

CSS boyut özelliklerini belirlerken aşağıdaki tabloda yer alan değerleri kullanabilirsiniz:

Özellik	Anlamı	Değerleri
heigh	Bir elemanın yüksekliğini atar.	auto length % inherit
max-height	Bir elemanın en büyük yükseklik değerini atar.	none length % inherit
max-width	Bir elemanın en büyük genişlik değerini atar.	none length % inherit
min-height	Bir elemanın en küçük yükseklik değerini atar.	length % inherit
min-width	Bir elemanın en küçük genişlik değerini atar.	length % inherit
width	Bir elemanın genişliğini atar.	auto

		length %

Kazanım

- Öğrenciler bu bölümde boyutları öğrenirler.

Eğitmen Yol Haritası

1	Öğrencilere boyutların anlatılması.
2	Konu hakkında yapılacak uygulamaların tanıtılması.
3	Öğrencilerle birlikte editör üzerinde uygulamaların yapılması.

JAVASCRIPT EĞITIMI

Müfredat Kodu	İCT-301-3
Ders	JavaScript
Ders İçeriği	• JavaScript: Tanıtım ve Programlamaya Giriş • Değişkenler • Sabitler • JavaScript: Operatörler • JavaScript: Sayı Metotları (Numbers) • JavaScript: String Metotları • Koşul Durumları: if-else • Koşul Durumları: Switch-Case Yapısı • Döngüler: For Döngüsü • Döngüler: While Döngüsü • Döngüler: Do While Döngüsü • JavaScript: Array (Diziler) • JavaScript: Fonksiyonlar • JavaScript: Tarih ve Zaman Fonksiyonları • Nesne Yönelimli Programlama (1.Bölüm) • Nesne Yönelimli Programlama (2.Bölüm)
Seviye	İCT-S2: ORTA
Ön Koşullar	Teknoloji ve yazılımla ilgili olmak.
Amaç	JavaScript, geliştiricilerin etkileşimli web sayfaları oluşturmak için kullandığı bir programlama dilidir. JavaScript işlevleri, sosyal medya akışlarını yenilemekten animasyonlar ve etkileşimli haritalar göstermeye kadar, bir web sitesi kullanıcısının deneyimini iyileştirmeyi amaçlamıştır.
Hedef	Sıradan web öğelerini alıp bunları etkileşime girilebilir hale getirmeyi hedefler. HTML ve CSS kodlarınızla birlikte kullanılabilir ve her web geliştirici için muazzam önemli bir araçtır.
Gereksinimler	Bilgisayar, Ortamda İnternet Ağı ve Temel Algoritma Bilgisi
Kazanım	• Dinamik web site tasarlayabilir, • Javascript ile en yeni web teknolojilerini kullanarak web site yapabilir, • Javascript ile oyun, dinamik içerikler, slaytlar, olaya duyarlı sayfalar geliştirebilir.
Referanslar & Dokümantasyon	• https://www.sadikturan.com/javascript-dersleri • https://yazilimdelisi.com/javascript-sabitler/ • https://zinzinzibidi.com/web_tasarim/jquery/javascriptte_operatorler • https://www.yusufsezer.com.tr/javascript-if-else/ • https://gelecegiyazanlar.turkcell.com.tr/konu/egitim/301-javascript/switch-case-yapisi • https://www.azkod.com/javascript/javascript-donguler • https://tr.javascript.info/array • https://talentgrid.io/tr/javascript-functions/ • https://www.tasarimveyazilim.com/javascript-nesne-yonelimli/
Durum	BEKLIYOR / UYGULANIYOR / TAMAMLANDI / DEĞERLENDIRILDI
Hedeflenen Süre	540 Dakika
Eğitim Günleri	Başlama: .../.../.... Bitiş: .../.../....

JAVASCRIPT: TANITIM VE PROGRAMLAMAYA GIRIŞ

Müfredat Kodu / Ünite / Konu	İCT-301-3.1 / JavaScript / Tanıtım ve Programlamaya Giriş
Hedeflenen Süre	45 dakika

🚀 JavaScript Nedir ?

JavaScript, ilk başta web belgelerine canlılık getirmek için oluşturulmuştur. Bu dilde yazılan kod kümelerine betik denir. Doğrudan HTML kodu içerisine yazılıp sayfa yüklendiğinde doğrudan çalışabilir. Komutlar herhangi bir derleme ve hazırlığa gereksinim duymazlar.

🔺 *Tarayıcı içerisindeki JavaScript neler yapabilir?*

- Sayfaya yeni HTML kodları ekleme veya öncekileri değiştirme, stilleri değiştirme veya ekleme.
- Kullanıcının eylemlerine karşılık verme. Tıklama veya fare imlecinin hareketine göre işlem yaptırabilme.
- Ağ üzerinden talep gönderebilme. Dosya yükleme veya indirebilme (buna AJAX ve COMET teknolojileri denir)
- Tarayıcıdaki çerezleri silme, ekleme veya düzeltme işlemlerinin yapılması.
- Kullanıcı tarafında girilen verilerin saklanması (“local storage”)

📘 Kazanım

- Öğrenci bu bölümde JavaScript ile tanışır ve JavaScript ile web üzerinde neler yapılabileceğini kavrar.

✅ Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ JavaScript’in tanıtılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere JavaScript’in tanıtılması amaçlanmaktadır. Öğrencilere JavaScript’in tanıtılması ve web üzerinden JavaScript ile neler yapılabileceğinin örneklenmesi. Öğrencilerin editör üzerinden ilk JavaScript çıktılarını alması.
2	☑ İlk çıktıların alınması.	☐ Eğitmen ☑ Öğrenciler	Öğrencilerin basit bir örnekle ilk çıktılarını alması amaçlanmaktadır.

DEĞİŞKENLER

Müfredat Kodu / Ünite / Konu	İCT-301-3.2.1 / Değişkenler ve Sabitler / Değişkenler
Hedeflenen Süre	45 dakika

🚀 Değişkenler

⚡ Geçici olarak veri saklamak için oluşturduğumuz veri yapılarına değişken denir.

✍️ JavaScript Değişken Tanımlama

JavaScript de değişken tanımlamak için -var- ya da -let- komutu kullanılır.

```
var x = 10;
var y = 20;
var z = x + y;
var k;

consolo.log(x);  // 10
consolo.log(y);  // 20
consolo.log(z);  // 30
consolo.log(k);  // undefined
```

Tanımlamış olduğumuz x,y,z ve k değişkenleri bellekte tanımlanan geçici alanlardır. Uygulama her çalıştığında bellek üzerinde tanımlanan değerler silinir.

k değişkenine bir değer ataması yapmadığımızdan dolayı undefined değere sahip olur.

→ Değişkenleri var komutuyla tanımladıktan sonra içlerine değer ataması yapıyoruz ve en sona **;** (noktalı virgül) eklememiz gerekiyor.

```
var name = "Çınar";
```

→ Değişkenlere karakter ataması (string) yaparken tek tırnak ya da çift tırnak kullanmamız gerekir.

```
var name = "Çınar";
var surname = 'Turan';
```

⚠️ Değişkenlere sayısal bir atama yaparken tırnak kullanmamamız gerekiyor. Aksi halde string bir değişken tanımlaması yapmış oluruz.

```
var a = 50;   // sayısal olarak 50 değeri tanımladık
var b = "50"; // karakter (string) olarak 50 değeri tanımladık.
var toplam = a + b; // a + b' nin sonucu 5050 olur.
```

→ Eğer ki; a + b' nin sonucunun 100 olmasını istiyorsak bu durumda b değişkenini de sayısal olarak tanımlamamız gerekiyor. Yani tırnak kullanmadan değişken tanımlamamız lazım.

```
var a = 50;
var b = 50;
var toplam = a + b;
```

Bu durumda toplam değişkeninin içeriği 100 olur.

Değişken içerisinde var olan bir değeri yeni bir atamayla değiştirebiliriz.

```
var x = 10; // x içinde 10 değeri var.
var y = 20; // y içinde 20 değeri var.
x = 30; // x içinde bulunan 10 değeri silinir ve 30 değeri aktarılır.
```

📌 JavaScript Değişken Tanımlama Kuralları

Javascript de değişken isimlerini seçerken bazı kurallara uymamız gerekiyor.

1- Değişken isimleri rakam ile başlayamaz.

```
var 1yas; => hatalı
var yas1; => geçerli
var _yas; => geçerli
```

2- Komut isimleriyle tanımlama yapılamaz.

Örneğin; if ya da switch kelimesi değişken ismi olamaz.

- Büyük küçük harf duyarlılığı vardır.

```
var firstName = 'Sadık';
var FirstName = 'Çınar';
```

Burada tanımlanan 2 farklı değişken vardır.

3- Değişken isimlerinde türkçe karakter kullanmamalıyız. "ı" karakteri yerine "i" karakteri kullanmalıyız.

```
var kullaniciAdi = "sadikturan";
```

Kazanım

- Öğrenci bu bölümde değişkenleri tanır ve değişkenlere ait uygulamalarla öğrendiklerini pekiştirir.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Değişkenlerin anlatılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere değişkenlerin anlatılması hedeflenmektedir.
2	☑ Değişkenlere ait uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen eşliğinde öğrencilerin değişkenlere ait uygulamaları tamamlaması amaçlanmaktadır.

SABITLER

Müfredat Kodu / Ünite / Konu	İCT-301-3.2.2 / JavaScript Değişkenler ve Sabitler / Sabitler
Hedeflenen Süre	45 Dakika

🚀 Sabitler

✍️ *Sabit İsmi Verme Kuralları*

- Sabit tanımlaması yapılırken; sabit isimleri arasında boşluk bırakılamaz.
- Sabitler isimlendirilirken birden fazla kelime kullanılacaksa kelimeler arasına _ (alt çizgi) konur.
- Sabit isimleri rakamla başlayamaz. İlk karakterden sonra rakam ile devam edebilir. Aynı zamanda ilk karakter _ (alt çizgi) işareti olabilir.
- JavaScript için Reserved Words (Ayrılmış Kelimeler) vardır. Bu kelimeler herhangi bir sabit veya fonksiyon tanımlamasında kullanılamaz. Ayrılmış kelimeler şunlardır:

abstract, alert, arguments, Array, blur, boolean, Boolean, break, byte, callee, caller, captureEvents, case, catch, char, class, clearInterval, clearTimeOut, close, closed, confirm, const, contructor, continue, Date, debugger, default, defaultStatus, delete, do, document, double, else, enum, escape, eval, export, extends, false, final, finally, find, float, focus, for, frames Function, function, goto, history, home, if, implements, import, in, infinity, innerHeight, innerWidth, instanceof, int, interface, isFinite, isNan, java, length, location, locationbar, long, Math, menubar, moveBy, moveTo, name, NaN, native, netscape, new, null, Number, Object, open, opener, outerHeight, outerWidth, package, Packages, pageXOffset, pageYOffset, parent, parseFloat, parseInt, personalbar print, private, prompt, protected, prototype, public, RegExp, releaseEvents, resizeBy, resizeTo, return, routeEvent, scroll, scrollbars, scrollBy, scrollTO, self, setInterval, setTimeOut, short, static, status, statusbar, stop, String, super, switch, synchronized, this, throw, throws, unescape, while, window, var

---

📌 *Sabit Tanımlama*

Javascript sabitler konusu değişkenler kadar önemlidir. Javascript ortamında sabit değişken tanımlaması “const” anahtar kelimesi ile yapılır. const ile bir değişken sabit tanımlaması yapılırken mutlaka ilk değer ataması yapılmalıdır. Const anahtar kelimesinin standart kullanımı aşağıdaki gibidir.

const değişken_adi = ilk_deger;

const anahtar kelimesi ile sabit değişkenler tanımlayalım.

```
const pi_sayisi = 3.14;

const vize_carpan = 0.6;

const final_carpan = 0.4;
```

Yukarıdaki örnek olarak tanımlamış olduğumuz sabit değişkenler değiştirilemez değildir fakat bu sabit değişkenlere sonradan herhangi bir değer ataması durumunda bazı web browser’larda hatalar oluşacaktır. Aşağıdaki örneğimizi inceleyelim.

```
const pi_sayisi = 3.14;

const vize_carpan = 0.6;
```

```
const fnal_carpan = 0.4;

piSayisi = 3;

viz_carpan = 2;
```

Bu örnekte; sabit olarak tanımlamış olduğumuz değişkenlere sonradan değer değiştirme işlemini gerçekleştirdik. Bu değer değiştirme işlemi sonrası çalışan javascript kodlarımız bazı tarayıcılarda çalışmayacaktır. Örneğin Firefox ve Chrome gibi tarayıcılar yukardaki bir durumda javascript betiklerini çalıştırmaz. Doğal olarak yukarıdaki kodlarımız arasına document.write() ekleyip ekran çıktısını almak istediğimiz durumda boş bir sayfa ile karşılaşırız. Bu sebeple sabit olarak tanımadığımız bir değeri bir daha değiştirmeden kullanmamız kodlarımızın düzgün bir şekilde çalışması açısından önem arz etmektedir. Ayrıca sabit bir tanımlama yaparken ismin başına ya da sonun koyacağımız basit bir kelime ile bu değerin bir sabit olduğunu kolaylıkla fark etmemizi sağlar. Örnek vermek gerekirse;

Const pi_sayisi_sabiti=3.14;

Kazanım

- Öğrenci bu bölümde sabit tanımları öğrenir ve uygulamalarla öğrendiklerini pekiştirir.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Sabit tanımların anlatılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere sabit tanımların anlatılması hedeflenmektedir.
2	☑ Sabit tanımlara ait uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen eşliğinde öğrencilerin sabit tanımlara ait uygulamaları tamamlaması amaçlanmaktadır.

JAVASCRIPT: OPERATÖRLER

Müfredat Kodu / Ünite / Konu	İCT-301-3.3 / JavaScript / Operatörler
Hedeflenen Süre	45 dakika

🚀 Operatörler

Operatörler bir değişkene değer atamaya, bir koşulu ya da deyimi yerine getirmeye yarayan karakterlerdir.

JavaScript dilinde:

- Aritmetik Operatörler
- Atama Operatörleri
- Karşılaştırma Operatörleri
- Mantıksal Operatörler

olmak üzere 4 farklı operatör türü vardır.

💠 Aritmetik Operatörler

🔺 *Toplama Operatörü*

Toplama operatörü, sayısal ve mantıksal ifadeler üzerinde toplama işlemi yapmak için kullanılmaktadır.

```
<script type="text/javascript">

var sayi1=5;
var sayi2=10;

var toplam=sayi1+sayi2;

document.write(toplam);

</script>
```

⚠️ + işaretinin string ifadeleri yan yana yazdırmak içinde kullanıldığını unutmamalıyız.

🔺 *Çıkarma Operatörü (-)*

Çıkarma operatörü, çıkarma işlemi yapmak için kullanılır.

```
<script type="text/javascript">

var sayi1=37;
var sayi2=10;

var cikar=sayi1-sayi2;
```

```
document.write(cikar);

</script>
```

Çarpma Operatörü(*)

Çarpma operatörü, çarpma işlemi yapmak için kullanılmaktadır.

```
<script type="text/javascript">

var sayi1=7;
var sayi2=10;

var carpma=sayi1*sayi2;

document.write(carpma);

</script>
```

Bölme operatörü (/)

Bölme operatörü, bölme işlemi yapmak için kullanılmaktadır.

```
<script type="text/javascript">

var sayi1=1000;
var sayi2=5;

var bolme=sayi1/sayi2;

document.write(bolme);

</script>
```

Mod operatörü (%)

Mod alma operatörü, bölme işleminde kalanı bulmak için kullanılmaktadır. Bu işlem ile çift sayılar, tek sayılar, üçün katları vs bulunabilmektedir.

```
<script type="text/javascript">

var sayi1=1001;
var sayi2=5;

var modalma=sayi1%sayi2;

document.write(modalma);
```

```
</script>
```

Arttırma operatörü (++)

Arttırma operatörü, sayının değerini 1 arttırmak için kullanılmaktadır. Sayının sonuna ya da başına eklenmesi durumunda sayının değeri 1 artar.

```
<script type="text/javascript">
var sayi=36;
sayi++;
document.write(sayi);

</script>
```

Azaltma operatörü (--)

Azaltma operatörü, sayının değerini 1 azaltmak için kullanılır. Sayının sonuna ya da başına eklenmesi durumunda sayının değeri 1 azalır.

```
<script type="text/javascript">
var sayi=100;
sayi--;
document.write(sayi);
</script>
```

✍️ Atama Operatörleri

= operatörü

Eşittir operatörü atama işlemi yapmak için kullanılmaktadır. = operatörü, sağ taraftaki değeri değişkene atama işlemi yapmak için kullanılır. Yani herhangi bir değişkene değer girdirme işlemi için kullanılır.

```
<script type="text/javascript">
var deger=10;
document.write(deger);
</script>
```

+= Operatörü

+= operatörünün sağındaki değerle solundaki değişken değerini toplayıp tekrar operatörün solundaki değişkene atama işlemi yapmak için kullanılır.

```
<script type="text/javascript">

var x=10;
var y=5;

x+=y;

document.write(x);

</script>
```

-= Operatörü

-= operatörünün sağındaki değer ile eşitliğin solundaki değişken değerinden eksilterek tekrar operatörün solundaki değişkene atama işlemi yapmak için kullanılır.

```
<script type="text/javascript">

var x=10;
var y=5;

x-=y;

document.write(x);

</script>
```

**= Operatörü*

*= operatörünün sağındaki değerle solundaki değişken değeri çarpılıp tekrar operatörün solundaki atama işlemi yapmak için kullanılır.

```
<script type="text/javascript">

var x=10;
var y=5;

x*=y;

document.write(x);

</script>
```

🔺 /= Operatörü

/= operatörünün solundaki değişken değerini eşitliğin sağındaki değere bölerek tekrar operatörün solundaki değişkene atama işlemi yapar.

```
<script type="text/javascript">

var x=10;
var y=5;

x/=y;

document.write(x);

</script>
```

📌 Karşılaştırma Operatörleri

En az iki değeri karşılaştırıp sonuç olarak boolean yani true ya da false olarak bir değer üretmek için kullanılır. Akış denetimi yaparken sıklıkla kullanılmaktadır. Karşılaştırma operatörlerine şöyle bir örnek verebiliriz:

Doğum yılına göre yaşı 18 ve üstü alan bir kişi ehliyet alabilir. Dolayısıyla yaşını hesapladığımız bir kişinin ehliyet alıp alamayacağını (>=) büyük eşit operatörü ile kontrol edebiliriz.

```
var dogumYili = 1999;
var yas = 2020 - dogumYili;
var ehliyet = (yas >= 18)
```

✨ Mantıksal Operatörler

🔺 ve (&&) Operatörü

Birden fazla koşulu kontrol edeceksek ve tüm koşulların doğru olduğu anda işlem yapmasını istiyorsak ve bağlacını kullanırız.

Tüm koşullar doğru olduğunda true değeri döndürür.

```
<script type="text/javascript">

var x=10;
var y=100;

document.write(x>5 && y>99);

</script>
```

🔺 veya (||) Operatörü

Birden fazla koşulu kontrol edeceksek ve bu koşullardan herhangi biri doğru olduğu anda işlem yapmasını istiyorsak veya bağlacını kullanırız.

Koşullardan herhangi biri doğru olduğunda true değeri döndürür.

```
<script type="text/javascript">

var a=20;
var b=10;

document.write(a>15 || b<5);

</script>
```

Değil (!) Operatörü

Boolean (True- False) ifadenin tam tersini göstermek istediğimizde değil operatörünü kullanırız.

```
<script type="text/javascript">

var x=16;


document.write(!(x>19));

</script>
```

Kazanım

- Öğrenci bu bölümde operatörleri tanır ve operatör çeşitlerini kavrar.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Operatörlerin anlatılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere operatörlerin anlatılması hedeflenmektedir.
2	☑ Operatörlere ait uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen eşliğinde öğrencilerin operatörlere ait uygulamaları tamamlaması amaçlanmaktadır.

JAVASCRIPT: SAYI METOTLARI (NUMBERS)

Müfredat Kodu / Ünite / Konu	İCT-301-3.4 / JavaScript / Sayı Metotları (Numbers)
Hedeflenen Süre	30 dakika

🚀 Sayı Metotları (Numbers)

Javascript'de number nesnesini Number() metodu ile tanımlayabiliriz. Ancak tanımladığımız primitive tipteki (x=10) veriler de nesne olarak tanımlanırlar. Dolayısıyla;

```
var x = 10;
var y = new Number(10);
```

x ve y değişkenlerinin ikiside birer nesnedir ve iki satırda da yaptığımız iş aynıdır.

▲ *toString() Metodu*

Number türündeki veriyi string veri türüne çevirir.

```
var x = 10;
var y = 20;
var z = x.toString() + y.toString()  // 1020
```

▲ *toFixed() Metodu*

toFixed() metodu verilen parametreye göre veriyi yuvarlar.

```
var x = 10.439;
x.toFixed(0);        // 10
x.toFixed(2);        // 10.44
x.toFixed(4);        // 10.4390
x.toFixed(5);        // 10.43900
```

Ondalıklı kısım hangi sayıya yakınsa o sayıya yuvarlanır. Örneğin 10.439 yerine 10.501 değeri kullanılmış olsaydı ilk örnekte sonuç 11 olurdu.

Parametre olarak gönderdiğimiz sayı ondalıklı kısımdaki basamak sayısını belirtiyor.

▲ *toPrecision() Metodu*

toPrecision() metodu toFixed() metodu ile aynı iş için kullanılıyor ancak farkı toFixed() metodunda verilen parametre ondalıklı kısımdaki basamak sayısını belirtirken toPrecision() metodunda tam ve ondalıklı kısmın toplamındaki basamak sayısını belirtir.

```
var x = 10.439;
x.toPrecision(0);        // 10.439
```

```
x.toPrecision(2);        // 10
x.toPrecision(4);        // 10.44
x.toPrecision(5);        // 10.439
```

Kazanım

- Öğrenci bu bölümde sayı metotlarını tanır ve uygulamalarla öğrendiklerini pekiştirir.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Sayı metotlarının anlatılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere sayı metotlarının anlatılması hedeflenmektedir.
2	☑ Sayı metotlarına ait uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen eşliğinde öğrencilerin operatörlere ait uygulamaları tamamlaması amaçlanmaktadır.

JAVASCRIPT: STRING METOTLARI

Müfredat Kodu / Ünite / Konu	İCT-301-3.5 / JavaScript / String Metotları
Hedeflenen Süre	30 Dakika

🚀 String Metotları

String metotları, string türü veriler üzerinde işlem yaparken faydalandığımız metotlardır.

🔺 *concat() Metodu*

Javascript concat() metodu ile string concatenation yani string birleştirme işlemi yapabiliriz.

```
var adsoyad = ad.concat(' ',soyad);
```

🔺 *String indexof() ve lastindexof() Metodu*

String indexof() metodunu karakter dizisinde arama yapmak için kullanıyoruz.

string indexof() metodu parametre olarak gönderdiğimiz karakterin ilgili string dizisi içinde ilk geçtiği index bilgisini gönderir.

Eğer aranan ifade bulunamazsa -1 değeri gelir.

Karakter dizilerinin ilk karakteri 0' dan başlar.

```
var kurs = "Modern Javascript Dersleri: Baştan Sona Javascript Programlama";
var result = kurs.indexOf("Javascript"); // result: 7
```

⚠️ Javascript, karakter dizisi içerisinde ilk olarak 7.indeks de bulunur.

** lastindexof() metodu ise son bulunan karakterin indeks numarasını gönderir.

```
var kurs = "Modern Javascript Dersleri: Baştan Sona Javascript Programlama";
var result = kurs.lastindexOf("Javascript"); // result: 40
```

🔺 *search() Metodu*

String search() metodu da indexof() metodu gibi karakter dizisi içinde arama yapıp bulduğu karakterin indeks numarasını döndürür.

```
var kurs = "Modern Javascript Dersleri: Baştan Sona Javascript Programlama";
var result = kurs.search("Javascript"); // result: 7
```

🧐 search() ile indexof() metodunun farkı ise;

- search(), indexof() metodunda olduğu gibi aramaya başlanacak olan indeks numarasını almaz.
- indexof() metodu regular expression ifadelerini almaz.

🔺 *toUpperCase() ve toLowerCase() Metodu*

Javascript transform metotlarından toUpperCase() metodu ile karakter dizisini büyük harfe, toLowerCase() ile küçük harfe çevirebiliriz.

```
var str1 = "Merhaba Dünya!";
var str2 = str1.toUpperCase();  // MERHABA DÜNYA!
var str3 = str1.toLowerCase();  // merhaba dünya!
```

🔺 *trim() Metodu*

trim() metodu ile karakter dizisinin başındaki ve sonundaki boşluk karakterlerini sileriz.

```
var greeting="   Merhaba Dünya!    ";
alert(greeting.trim()); // Merhaba Dünya!
```

📌 String Parçalama Metotları

Javascript' de bir karakter dizisi içinde belli aralıktaki karakterleri alabiliriz. String parçamak için kullanabileceğimiz 3 tane string metodu vardır.

- slice() Metodu
- substring() Metodu
- substr() Metodu

String parçalama metotlarını sırasıyla öğrenelim.

■ *slice() Metodu*

slice() metoduna başlangıç ve bitiş indeksi vererek aralıktaki karakter dizisini alabiliriz.

```
var str = "Mazda,Togg,Toyota";
var result = str.slice(6, 10); // Togg
```

7.indeksten başlar ve 10' e kadar alır. Son indeks numarası dahil olmaz.

🔺 ** Başlangıç ve bitiş indekslerini negatif olarak da verebiliriz. En sağdaki karakter -1 ile başlar.

■ *substring() Metodu*

slice() metodun olduğu gibi substring() metoduyla başlangıç ve bitiş indeksi vererek aralıktaki karakter dizisini alabiliriz.

⚠️ slice() metodu ile substring() metodunun farkı; substring()'da negatif indeks kullanamıyoruz.

```
var str = "Mazda,Togg,Toyota";
var result = str.slice(6, 10); // Togg
```

** slice() metodunda olduğu gibi bitiş indeksini vermezsek sona kadar alır.

■ *substr() Metodu*

substr(), slice() ve substring() gibi string parçalama işlemlerinde kullanılır ancak diğer metotlardan farkı ikinci parametre başlangıçtan itibaren alınacak karakter sayısını temsil eder.

```
var str = "Mazda,Togg,Toyota";
var result = str.substr(6,4); // Togg
```

** ikinci parametre göndermezsek karakter dizisinin sonuna kadar alır.

Kazanım

- Öğrenci bu bölümde String metotlarını tanır ve uygulamalarla öğrendiklerini kavrar.

✅ Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ String metotlarının anlatılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere string metotlarının anlatılması hedeflenmektedir.
2	☑ String metotlarına ait uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen eşliğinde öğrencilerin string metotlarına ait uygulamaları tamamlaması amaçlanmaktadır.

KOŞUL DURUMLARI: IF-ELSE

Müfredat Kodu / Ünite / Konu	İCT-301-3.6.1 / Koşul Durumları/ if-else
Hedeflenen Süre	45 dakika

🚀 Koşul ifadeleri

JavaScript koşula göre komut çalıştırmak için koşul ifadeleri kullanılır.

- if koşul ifadesi, koşul doğruysa kod bloğu içerisindeki komutları çalıştırır.
- else koşul ifadesi, if koşul ifadesi yanlışsa kod bloğu içerisindeki komutları çalıştırır.
- else if koşul ifadesi, if koşul ifadesi yanlışsa yeni bir koşul ifadesi yazmamızı ve yazılan koşul doğruysa kod bloğu içerisindeki komutları çalıştırır.

🔺 *if Kullanımı*

if koşul ifadesi içerisindeki kod bloğunu koşul doğruysa çalışır.

```
if(koşul) {
   // koşul doğru-true ise çalıştırılacak kod bloğu
}
```

⚠️ if koşul ifadesi If veya IF olarak yazılırsa hata verecektir.

Örnekte görüldüğü gibi; saat 18'den önce ise "iyi günler" mesajını uyarı mesajı olarak ekrana yazacaktır.

```
<script>
 var tarih = new Date();
 var saat = tarih.getHours();
 if(saat < 18) {
    alert("iyi günler");
 }
</script>
```

🔺 *else Kullanımı*

if koşul ifadesindeki koşul yanlışsa else koşul ifadesi içerisindeki kod bloğu çalışır.

```
if(koşul) {
   // koşul doğru-true ise çalıştırılacak kod bloğu
} else {
   // koşul yanlış-false ise çalıştırılacak kod bloğu
}
```

Örneğe bakarsak; saat 18'den önce ise "iyi günler" değilse "iyi akşamlar" mesajını uyarı mesajı olarak ekrana yazacaktır.

```
<script>
 var tarih = new Date();
 var saat = tarih.getHours();
 if(saat < 18) {
    alert("iyi günler");
 } else {
    alert("iyi akşamlar");
 }
</script>
```

if - else Kullanımı

if koşul ifadesindeki koşul yanlışsa ve yeni koşul ifadesi yazmak istenirse if else koşul ifadesi kullanılır.

```
if(koşul) {
   // koşul doğru-true ise çalıştırılacak kod bloğu
} else if(koşul2) {
   // koşul yalış-false ise ve koşul2 doğru-true ise çalıştırılacak kod bloğu
} else {
   // koşul yanlış-false ise çalıştırılacak kod bloğu
}
```

Örnekte verildiği gibi; saat 10'dan önce ise "günaydın", saat 18'den önce ise "iyi günler "hiçbiri değilse "iyi akşamlar" mesajını uyarı mesajı olarak ekrana yazacaktır.

```
<script>
 var tarih = new Date();
 var saat = tarih.getHours();
 if(saat < 10) {
    alert("günaydın");
 } else if(saat < 18) {
    alert("iyi günler");
 } else {
    alert("iyi akşamlar")
 }
</script>
```

Saatin 12 olduğun varsayalım (saat = 12).

Saat 10'den önce olmadığı için if koşul ifadesindeki koşul (12 < 10) yanlış olacaktır.

Saat 18'den önce olduğu için else if koşul ifadesindeki koşul (12 < 18) doğru olacağından iyi günler mesajını verecektir.

Kazanım

- Öğrenci bu bölümde if-else koşullarını tanır ve uygulamalarla öğrendiklerini kavrar.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ if-else koşullarının anlatılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere if-else koşullarının anlatılması hedeflenmektedir.
2	☑ if-else koşullarınına ait uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen eşliğinde öğrencilerin if-else koşullarına ait uygulamaları tamamlaması amaçlanmaktadır.

KOŞUL DURUMLARI: SWITCH-CASE YAPISI

Müfredat Kodu / Ünite / Konu	İCT-301-3.6.2 / Koşul Durumları/ Switch-Case Yapısı
Hedeflenen Süre	30 dakika

🚀 Switch-Case Yapısı

Bir programda çok sayıda koşul kontrolü ve bunların sonucuna göre gerçekleştirilmesi gereken işlemler mevcutsa, if – else yapıları ile program akışının izlenmesinde güçlükler oluşabilir. Bu durumlar, genellikle, switch deyiminin kullanımının uygun olacağı durumlardır. Switch deyimi, tek bir ifadeyi kontrol ederek, ifadenin değerine göre, sınırsız sayıda çalışma yolu belirleme olanağı sağlayan bir deyimdir.

Switch sözcüğünden hemen sonra gelen ifade, parantezler içinde ve bir tam sayı ifade olmalıdır. Case sözcüklerini izleyen değerler de tam sayı sabit türünde olmalıdır.

Switch deyiminin çalışma prensibi basittir: Switch sözcüğünün yanındaki ifade hesaplanır; sonuç hangi case'i izleyen sabitle çakışıyorsa programın kontrolü o case'i izleyen kısma geçer. Bu case'den sonraki deyimler ise, bir kontrole tabi olmaksızın sonuna kadar çalıştırılır.

Case sabitlerinin hiçbiri ifade ile uyuşmuyorsa, programın akışı, default sözcüğünü izleyen kısma geçer (bu kısım mevcutsa). Default'un Switch-Caseyapısı içinde en sonda olması şart değildir fakat iyi bir programlama stilidir.

⚠️ İki case sabiti aynı değeri alamaz.

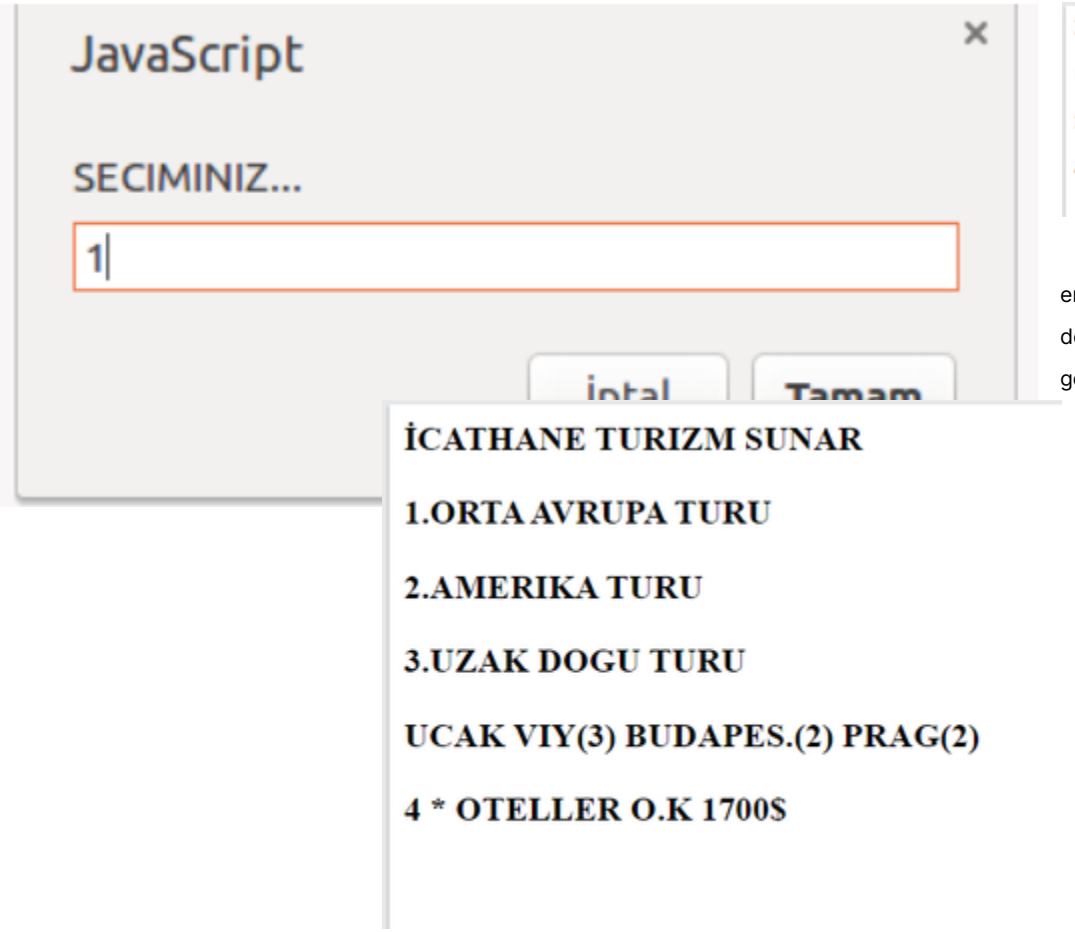
Switch-Case yapısında, programın kontrolünün ifadenin değerine göre, case seçeneklerinden sadece birini icra edip bundan sonra Switch-Case yapısını terketmesi isteniyorsa, bu durumda her case seçeneğinin sonuna bir break komutu yerleştirilmelidir.

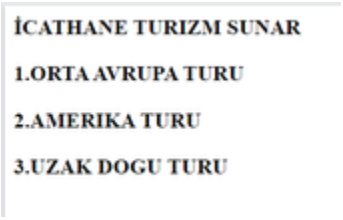
🔺 Switch-Case yapısı ile ilişkili örnek HTML-JavaScript kodu:

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JAVASCRIPT">
var a;
document.write("<H3>HOSOK TERE TURIZM SUNAR<H3>")
document.write("<H3>1.ORTA AVRUPA TURU<H3>")
document.write("<H3>2.AMERIKA TURU<H3>")
document.write("<H3>3.UZAK DOGU TURU<H3>")
a=window.prompt("SECIMINIZ...")
switch (parseInt(a))
{
case 1:
{
document.write("<H3>UCAK VIY(3) BUDAPES.(2) PRAG(2)<H3>");
document.write("<H3>4 * OTELLER O.K 1700$<H3>");
break;
}
case 2:
{
document.write("<H3>UCAK NEW YORK(4)(3) BOSTON.(2) LOS ANGELES(4)<H3>");
document.write("<H3>5 * OTELLER O.K 3700$<H3>");
break;
}
```

```
case 3:
{
document.write("<H3>UCAK BANG KONG(3) HONG KONG.(3) TOKYO(4)<H3>");
document.write("<H3>5 * OTELLER O.K 4000$<H3>");
break;
}
default:window.alert("SECIMINIZ HATALI...");
}
</SCRIPT>
</HEAD>

<BODY>
</BODY>
</HTML>
```

Yukarıdaki uygulamayı tarayıcınızda çalıştırdığınızda sizden seçim yapmanızı isteyecek bir pencereyle karşılaşacaksınız.





Listeden bir değer girdiğinizde, o değere karşılık gelen bloğun görüntülendiğini göreceksiniz.

Kazanım

- Öğrenci bu bölümde Switch-Case koşullarını tanır ve uygulamalarla öğrendiklerini kavrar.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Switch-Case koşullarının anlatılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere Switch-Case koşullarının anlatılması hedeflenmektedir.

2	☑ Switch-Case koşullarınına ait uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen eşliğinde öğrencilerin Switch-Case koşullarına ait uygulamaları tamamlaması amaçlanmaktadır.

DÖNGÜLER: FOR DÖNGÜSÜ

Müfredat Kodu / Ünite / Konu	İCT-301-3.7.1 / Döngüler / For Döngüsü
Hedeflenen Süre	15 Dakika

🚀 For Döngüsü

For döngüsü; belirlenen başlangıç değerinden itibaren belirtilen koşul sağlanana kadar içine yazıldığı kod parçasını ardı ardına çalıştıran bir döngü çeşididir.

```
for(başlangıç değeri;  koşul;  artış değeri)
{
//yapılacak işlemler
}
```

🖥️ Uygulama Bölümü

0'dan 10'a kadar ekrana for döngüsü ile yazan kod. Ekrana : 0123456789

```
<script>
for(var i=0; i<10; i++)
{
document.write(i);
}
</script>
```

0'dan 5'e kadar ekrana alt alta for döngüsü ile yazan kod.

```
<script>
for(var i=0; i<5; i++)
{
document.write(i+"<br>");
}
```

```
/* Ekrana
0
1
2
3
4  */
</script>
```

For döngüsü ile 100'den 0 a kadar azaltarak yazan kod.

```
<script>

for(var i=100; i>0; i--)

{

document.write(i);

}

</script>
```

Kazanım

- Öğrenci bu bölümde for döngüsünü tanır ve uygulamalarla öğrendiklerini kavrar.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ For döngüsünün anlatılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere for döngüsünün anlatılması hedeflenmektedir.
2	☑ For döngüsüne ait uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen eşliğinde öğrencilerin for döngüsüne ait uygulamaları tamamlaması amaçlanmaktadır.

DÖNGÜLER: WHILE DÖNGÜSÜ

Müfredat Kodu / Ünite / Konu	İCT-301-3.7.2 / Döngüler / While Döngüsü
Hedeflenen Süre	15 Dakika

🚀 While Döngüsü

JavaScript'te bir duruma bağlı olarak tekrarlayan işlemleri gerçekleştirmek için while döngüsünü kullanırız.

Kullanım şekli şöyledir:

```
while (koşul) {
  // çalıştırılacak kodlar.
}
```

While döngüsünde belirtilen koşul doğru (true) olduğu sürece blok içerisindeki kodlar işletilir eğer koşul false değer üretirse while döngüsünden çıkılır. Örnek olarak aşağıdaki kod bloğunu inceleyebiliriz:

```
let i = 1;

while (i <= 5) {
    console.log(i);
    i += 1;
}

// 1
// 2
// 3
// 4
// 5
```

i değerini başlangıçta 1' den başlatarak 5' e eşit ya da küçük olup olmadığını kontrol ediyoruz. Eğer i değeri 6 değerine gelirse while döngüsünün koşulu false değer getireceğinden dolayı döngü biter. Her döngü bloğu çalıştırıldığında i değerinin 1 arttırıldığına dikkat ediniz aksi halde sonsuz döngüye gireriz ve uygulama hata ile sonlanır.

⚠️ Kullanıcı negatif bir değer girerse while(false) olacağından dolayı while döngüsü sonlandırılır ve ekrana toplam değer yazdırılır.

📘 Kazanım

- Öğrenci bu bölümde while döngüsünü tanır ve uygulamalarla öğrendiklerini kavrar.

✅ Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ While döngüsünü anlatılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere while döngüsünü anlatılması hedeflenmektedir.

2	☑ While döngüsüne ait uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen eşliğinde öğrencilerin while döngüsüne ait uygulamaları tamamlaması amaçlanmaktadır.

NOTLAR

DÖNGÜLER: DO WHILE DÖNGÜSÜ

Müfredat Kodu / Ünite / Konu	İCT-301-3.7.3 / Döngüler / Do While Döngüsü
Hedeflenen Süre	15 Dakika

🚀 Do-While Döngüsü

While döngüsünde döngü bloklarının en az bir kere çalıştırılması için while koşulunun true getirmesi gerekir ancak do-while döngüsünde koşul false getirecek olsa bile while döngüsünün kod blokları en az bir kere işletilir çünkü koşul durumunun kontrolü sonda yapılır.

Do-while döngüsünün kullanım şekli şöyledir;

```
do {
  // çalıştırılacak kodlar.
}
while (koşul);
```

Kodlar yukarıdan aşağı doğru işletilir dolayısıyla while(koşul) false değer getirecek olsa bile döngü bunu bilemez ve blok içindeki kodları en az 1 kere çalıştırır.

Örnek:

```
let i = 1;

do {
    console.log(i);
    i++;
} while(i <= 5);

// 1
// 2
// 3
// 4
// 5
```

Başlangıçta tanımlanan i değeri hemen ekrana yazdırılır sonrasında 5' e eşit ya da küçük olup olmadığı kontrol edilir. While i değişkenin 5 e kadar true değeri getirmesinden dolayı işletilir ve ekrana 1-5 arasındaki sayılar yazdırılır.

Örnek:

```
let toplam = 0;
let sayi = 0;

do {
    toplam += sayi;
    sayi = parseInt(prompt('sayı giriniz: '));
```

```
} while(sayi >= 0)

console.log(`toplam: ${toplam}.`);

// sayı giriniz: 4
// sayı giriniz: 4
// sayı giriniz: -5
// toplam: 8
```

O halde yukarıdaki kod bloğuna dayalı olarak şunları söyleyebiliriz:

- Sayı başlangıçta 0 değerine eşitleniyor çünkü kullanıcı sayı girmeden önce etkisiz eleman ile toplam içeriğin toplanması gerekiyor.
- Kullanıcı sayı girdikten sonra while bloğunda girilen sayının pozitif kontrolü yapılıyor ve pozitif olduğu sürece toplam hesaplanıyor.
- Kullanıcı negatif değer girerse döngüden çıkılıyor.

Kazanım

- Öğrenci bu bölümde do-while döngüsünü tanır ve uygulamalarla öğrendiklerini kavrar.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Do-while döngüsünü anlatılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere do-while döngüsünü anlatılması hedeflenmektedir.
2	☐ Do-while döngüsüne ait uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen eşliğinde öğrencilerin do-while döngüsüne ait uygulamaları tamamlaması amaçlanmaktadır.

JAVASCRIPT: ARRAY (DIZILER)

Müfredat Kodu / Ünite / Konu	İCT-301-3.8 / JavaScript / Array (Diziler)
Hedeflenen Süre	25 dakika

🚀 Array (Diziler)

Objeler değerlerin anahtarlı bir şekilde koleksiyon halinde tutulmasını sağlar. Fakat bazı durumlarda sıralı koleksiyon tutmak gerekebilir, 1., 2. ve 3. elemente ihtiyaç olabilir. Örneğin kullanıcıların, ürünlerin, HTML elementlerinin liste halinde tutulmasını istediğinizde;

Obje kullanmak mantıklı değildir, çünkü bu objeler elemanların sırasını tutmaz ve var olanların "arasına" yeni bir özellik girilemez.

Bunun için özel bir veri yapısı vardır: Diziler.

Boş dizi oluşturmak için iki yöntem vardır:

```
let arr = new Array();
let arr = [];
```

Neredeyse her zaman ikinci yazım kullanılır. Başlangıç değerlerini köşeli parantez içinde verebilirsiniz:

```
let meyveler = ["Elma", "Portakal", "Erik"];
```

→ Diziler sıfır ile başlarlar.

```
let meyveler = ["Elma", "Portakal", "Erik"];

alert( meyveler[0] ); // Elma
alert( meyveler[1] ); // Portakal
alert( meyveler[2] ); // Erik
```

→ Elemanı değiştirmek mümkündür:

```
meyveler[2] = 'Armut'; // Şimdi ["Elma", "Portakal", "Armut"]
```

... Veya diziye yeni bir eleman eklemek mümkündür:

```
meyveler[3] = 'Limon'; // Şimdi ["Elma", "Portakal", "Armut", "Limon"]
```

→ Dizide bulunan elemanların boyutu length metodu ile öğrenilebilir:

```
let meyveler = ["Elma", "Portakal", "Erik"];

alert( meyveler.length ); // 3
```

→ Ayrıca alert kullanarak tüm dizinin gösterilmesi de mümkündür:

```
let meyveler = ["Elma", "Portakal", "Erik"];

alert( meyveler ); // Elma,Portakal,Erik
```

→ Dizi her türlü elemanı tutabilir.

```
// Karmaşık tipler
let arr = [ 'Elma', { isim: 'Ahmet' }, true, function() { alert('merhaba'); } ];

// Birinci indeksteki değeri al ve "isim" özelliğini görüntüle
alert( arr[1].name ); // John

// 3. indeksteki fonksiyonu al ve çalıştır.
arr[3](); // merhaba
```

→ Diziler objeler gibi virgül ile bitebilir:

```
let meyveler = [
  "Elma",
  "Portakal",
  "Erik",
];
```

“Sürekli virgül” stili yeni eleman ekleme veya çıkarma işlemlerini kolaylaştırır.

Kazanım

- Öğrenci bu bölümde dizileri tanır ve uygulamalarla öğrendiklerini kavrar.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Array (Diziler) konusunun anlatılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere Array (Diziler) konusunun anlatılması hedeflenmektedir.
2	☑ Array (Diziler) konusuna ait uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen eşliğinde öğrencilerin Array (Diziler) konusuna ait uygulamaları tamamlaması amaçlanmaktadır.

JAVASCRIPT: FONKSIYONLAR

Müfredat Kodu / Ünite / Konu	İCT-301-3.9 / JavaScript / Fonksiyonlar
Hedeflenen Süre	25 dakika

🚀 Fonksiyonlar

⚡ Fonksiyonlar, bir işlevi yerine getirmek için tasarlanmış kod bloklarıdır.

📌 *Fonksiyon Tanımlama*

Fonksiyonlar, function anahtar kelimesi ile tanımlanır. Fonksiyon isimlerinde harf, rakam, alt çizgi ve dolar işareti kullanılabilir. Birden fazla parametre olduğunda parantez içerisinde virgülle ayrılarak kullanılır. (parametre1, parametre2, parametre3, ...)
Fonksiyon kodları süslü parantez { . . . } içerisine yazılır.

♻️ *Fonksiyon Çağırma*

JavaScript ile tanımlanan fonksiyonların çalışması için fonksiyon çağırma veya fonksiyon çalıştırma işleminin yapılması gerekir.

JavaScript fonksiyon çalıştırmanın yolları;

- Bir olay meydana geldiğinde – kullanıcı HTML nesnesine tıkladığında,
- Fonksiyon çağırma işlemi yapıldığında,
- Otomatik – fonksiyon içinde fonksiyon çağırarak.

🧐 *Fonksiyon Değer Döndürme*

Fonksiyonlar “return” anahtar kelimesi ile değer döndürür.
JavaScipt fonksiyon kodları, return anahtar kelimesine geldiğinde çalışmayı durdurur. JavaScript fonksiyonun çağrıldığı yerden devam eder varsa kalan kodları işler.

✍️ *Fonksiyon Kullanımı*

JavaScript fonksiyon kullanımı ile bir defa kodu yazarak birden fazla kullanabiliriz.

JavaScript fonksiyonlarına parametre ekleyerek farklı değerler için farklı sonuçlar elde edebiliriz.

```
<script>
 function celsiusCevir(fahrenhayt) {
    return (5 / 9) * (fahrenhayt - 32);
 }
 alert(celsiusCevir(50));
 alert(celsiusCevir(104));
</script>
```

Örnekte farklı değerleri parametre olarak alan ve hesaplama yapan celsiusCevir isimli fonksiyon tanımlanmıştır.

Örnekte celsiusCevir(50) ile fonksiyon çağırma işlemi yapılmıştır.

Sadece fonksiyon ismi yazılırsa JavaScript fonksiyon kodlarını döndürecektir.

```
<script>
 var sonuc;
 function celsiusCevir(fahrenhayt) {
    return (5 / 9) * (fahrenhayt - 32);
 }
 alert(celsiusCevir);
</script>
```

JavaScript fonksiyon sonuçlarını değişkenlere atanabilir.

```
<script>
 var sonuc;
 function celsiusCevir(fahrenhayt) {
    return (5/9) * (fahrenhayt-32);
 }
 sonuc = celsiusCevir(50);
 alert(sonuc);
</antlr>
```

Fonksiyon sonucu ile işlem yapmayacaksanız değişkene atamadan kullanabilirsiniz.

Kazanım

- Öğrenci bu bölümde fonksiyonları tanır ve uygulamalarla öğrendiklerini kavrar.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Fonksiyonların anlatılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere fonksiyonların anlatılması hedeflenmektedir.
2	☑ Fonksiyonlara ait uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen eşliğinde öğrencilerin fonksiyonlara ait uygulamaları tamamlaması amaçlanmaktadır.

JAVASCRIPT: TARIH VE ZAMAN FONKSIYONLARI

Müfredat Kodu / Ünite / Konu	İCT-301-3.10 / JavaScript / Tarih ve Zaman Fonksiyonları
Hedeflenen Süre	20 dakika

🚀 Tarih ve Zaman Fonksiyonları

⚡ JavaScript tüm tarih işlemleri için Date nesnesi kullanılır.

✍️ *Tarih yazdırma*

Tarihi yazdırmak istediğimizde aşağıdaki gibi bir kod kullanabiliriz.

```
<script>
 alert(Date());
</script>
```

Tarih tanımlamanın çeşitli yöntemleri vardır.

- new Date();
- new Date(milisaniye);
- new Date(tarih);
- new Date(yıl, ay, gün, saat, dakika, saniye, milisaniye);

Tarih nesnesine parametre olarak bir şey yazılmadığında o anki tarih bilgisini verir.

```
<script>
 var tarih = new Date();
 alert(tarih);
</script>
```

Tarih nesnesi tarihi parametre alarak oluşturulursa parametre olarak girilen tarih bilgisini verir.

```
<script>
 var tarih = new Date("Jan 01, 2000 10:25:00");
 alert(tarih);
</script>
```

⚠️ JavaScript tarih nesnesi oluşturulurken geçerli tarih parametre olarak girilmediğinde hata verir.

Tarih nesnesi oluştururken tarihi milisaniye olarak girebiliriz.

```
<script>
 var tarih = new Date(86400000);
 alert(tarih);
</script>
```

Tarih milisaniye olarak girilirse başlangıç tarihi 1 Ocak 1970 00:00:00'den itibaren milisaniye süresi hesaplanarak tarih bilgisini verir.

Bir gün 86400000 milisaniyedir.

Tarih nesnesi oluştururken parametre olarak yıl, ay, gün, saat, dakika, saniye ve milisaniye değeri sırasıyla girilerek tarih bilgisine ulaşılabilir.

```
<script>
 var tarih = new Date(2001, 5, 24, 19, 21, 30, 0);
 alert(tarih);
</script>
```

Ekrana yazılan tarih pek anlaşılır ve okunaklı olmayabilir.

Daha okunaklı ve anlaşılır olması için toUTCString() veya toDateString() metotları kullanılabilir.

toUTCString() Metodu

```
<script>
 var tarih = new Date();
 alert(tarih.toUTCString());
</script>
```

toDateString() Metodu

```
<script>
 var tarih = new Date();
 alert(tarih.toDateString());
</script>
```

JavaScript ile tarih yazdırırken kod çalıştığı tarihin bilgisini verir. Tarih-saat bilgisini ise, bilgisayarın tarih-saat bilgisinden alır.

Tarih Formatları

JavaScript ile tarih ve saat belirlerken çeşitli tarih formatlarını kullanabiliriz.

Tarih formatı	Örnek
ISO formatı	"2001-03-25" (Uluslararası standart)
Kısa tarih formatı	"03/25/2001"
Uzun tarih formatı	"Mar 25 2001" veya "25 Mar 2001"

⚠️ ISO formatı diğer formatlara göre daha çok kullanılır.

ISO Formatı

ISO 8601 uluslararası tarih ve saat formatıdır ve yazım biçimi YYYY-AA-GG şeklindedir.

```
<script>
 alert(Date("2001-03-25"));
</script>
```

Tarihi ekrana yazdırdığımızda tarayıcı tarihi kullanıcı sistemindeki saat formatına çevirip yazdırır.

ISO formatı ay ve gün belirtmeden yazılabilir.

```
<script>
 alert(Date("2001"));
</script>
```

Yazım biçimi YYYY-AA-GGTSS-DD-SSZ şeklindedir.

```
<script>
 alert(Date("2001-03-25T12:00:00Z"));
</script>
```

Tarih ve saatin UTC formatında olduğunu bildirmek için ISO tarih formatının sonuna Z harfi eklenir.

Kısa Tarih Formatı

Kısa tarih formatı yazım biçimi AA/GG/YYYY şeklindedir.

```
<script>
 alert(Date("03/25/2001"));
</script>
```

Uzun Tarih Formatı

Uzun tarih formatı yazım biçimi genellikle AAA GG YYY şeklindedir.

```
<script>
 alert(Date("Mar 25 2001"));
</script>
```

Kazanım

- Öğrenci bu bölümde tarih ve zaman fonksiyonlarını tanır ve kavrar.
- Tarih ve zaman fonksiyonlarına ait metotları öğrenir.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Tarih ve zaman fonksiyonlarının anlatılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencilere tarih ve zaman fonksiyonlarının anlatılması hedeflenmektedir.
2	☑ Tarih ve zaman fonksiyonlarına ait uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen eşliğinde öğrencilerin tarih ve zaman fonksiyonlarına ait uygulamaları tamamlaması amaçlanmaktadır.

NESNE YÖNELIMLI PROGRAMLAMA (1.BÖLÜM)

Müfredat Kodu / Ünite / Konu	İCT-301-3.11.1 / JavaScript / Nesne Yönelimli Programlama (1.Bölüm)
Hedeflenen Süre	55 Dakika

🚀 Nesne Yönelimli Programlama

Object Oriented Programming (OOP) ingilizce kelimelerinin türkçesi Nesne Yönelimli Programlama'dır. Yapılan işlevin nesneler olarak soyutlandığı programlama şeklidir. Bu yapıya olanak veren yazılım dilleri yüksek seviye yazılım dilleri diye tanımlanır.

Nesne yönelimli programlama dillerine ait özellikleri şöyle inceleyebiliriz:

- Soyutlama (Abstraction): Nesnenin farklı özelliklerinin ayrı tanımlanması diyebiliriz. Örneğin insanın boy, kilo vb. özellikleri ile davranışlarının farklı tanımlanması gibi...
- Kapsülleme (Encapsulation) : Nesnelerin özelliklerini diğer nesnelerden yada dışarıdan saklanmasını sağlayan bir kavramdır.
- Kalıtım-Miras Alma (Inheritance) : Nesnenin başka bir nesneden oluştuğu durumlara kalıtım denir. Yani nesne türediği nesnenin özelliklerini taşır.
- Çok biçimlilik (Polymorphism) : Nesnelerin belirli bir işi farklı yollar ile yapmasıdır.
- Bütünleme (Aggregation) : Her nesnenin alt nesnelerden oluşan bir yapıya sahip olmasıdır.

♻️ Nesne Tanımlama ve Çağırma

Nesne tanımlarken iki yöntem vardır. İlk yöntemde nesne tanımlamanın mantığı değişken tanımlarken içeriğini süslü parantez içine alır ve özelliklerine atamalar yapılır. İkinci yöntem ise new terimi ile yapıyoruz.

Javascript nesne tanımlama örneği ilk yöntemle şöyle gösterilir:

```
var aracBilgi = { marka     : "toyota",
                  uretimYıl : 2015,
                  serisi    : "corolla",
                  vites     :"manuel"
                  };
```

Javascript nesne tanımlama örneği ikinci yöntemle(new terimi ile) şöyle gösterilir:

```
var aracBilgi = new Object();
aracBilgi.marka     = "toyota";
aracBilgi.üretimYil =  2015;
aracBilgi.serisi    = "corolla";
aracBilgi.vites     = "manuel";
```

▲ Bir Nesneyi Önceden Tanımlama ve İçeriği Atama

Nesne ismini yazıp ve değer girmeden süslü parantez açıp kapatarak nesneyi tanımlamış oluruz. Değerlerini örnekte verildiği gibi sonradan ekleyebiliriz.

```
var arac = {};
arac.marka     = "toyota";
arac.uretimYil = 2015;
arac. serisi   = "corolla";
document.write( arac.marka );
```

Daha Önce İçeriği ile Tanımladığımız Nesneye Özellik Ekleme

“NesneAdı.YeniÖzellik = deger ” şeklinde yazdıktan sonra eşittir atama operatörü ile yeni özellik ve değerini atayabiliriz.

```
var aracBilgi = { marka     : "toyota",
                  uretimYil : 2015,
                  serisi    : "corolla",
                  vites      : "manuel"  };
aracBilgi.renk = "mavi";// özellik ekleme
```

Fonksiyon Şeklinde Nesne Tanımlama ve Çağırma

Nesne tanımlarken değişkene direk new ile fonksiyona bağlarız ve fonksiyon içerisinde nesne içeriğini tanımlarız. Bu şekilde tanımlanan fonksiyonlara yapıcı fonksiyonlar (constructor) denir.
Örnek ile hem oluşturalım hemde çağıralım:

```
function myFunc(){
   this.marka     = "suzuki",
   this.uretimYil = 2016,
   this.serisi    = "swift",
   this.vites     = "otomatik"
}
var aracBilgi = new myFunc();
var arac = aracBilgi.marka + " " + aracBilgi.serisi;
document.write( arac );
```

Nesnelerde Dizi Mantığı ile Özelliklerine Ulaşmak

Dizilerde elemanlarına dizi ismi ve köşeli parantezler içine index değerini yazarak ulaştığımız gibi nesnelerinde özellik isimlerini yazıp değerlerine ulaşabiliriz.. Ama çok tercih edilen bir yöntem değildir.

```
var aracBilgi = { marka     : "opel",
                  uretimYil : 1997,
                  serisi    : "astra",
                  vites      : "manuel"}
```

```
var aracMarka = aracBilgi["marka"];
document.write(aracMarka);
```

▲ Nesnelerin Özelliğinin Değerini Değiştirme

Nesnenin istediğimiz özelliğine ulaşıp değerini değiştirebiliriz.

```
var aracBilgi = { marka     : "opel",
                  uretimYil : 1997,
                  serisi    : "astra",
                  vites     : "manuel"  };
document.write( aracBilgi.vites  + "<br/>");
aracBilgi.vites = "otomatik";
document.write( aracBilgi.vites );
```

▲ Nesnelerin İçinde Fonksiyon Tanımlama ve Çağırma

Nesnelerin içerisinde fonksiyon tanımlayıp işlem yaptırabiliriz. Fonksiyonu çağırırken dikkat edilmesi gereken husus parantez açıp kapatmayı unutmamak. Diğer özellikleri çağırırken kullanmadığımız için fonksiyonda da unutulabilir. Unutursak hata verir.

```
var kisi = {
      adi    : "Tasarım",
      soyadi : "Yazılım",
      dgmTrh : 2019,
      tamAdi : function () {
        return kisi.adi + " " + kisi.soyadi;
      }
}
document.write(kisi.tamAdi());
```

⚠️ Nesne içindeki fonksiyonlarda nesnenin kendisine nesne adı ile ulaşabildiğimiz gibi this değeri ile de ulaşabiliriz.

```
var kisi = {
      adi    : "Tasarım",
      soyadi : "Yazılım",
      dgmTrh : 2019,
      tamAdi : function () {
        return this.adi + " " + this.soyadi;
      }
}
document.write(kisi.tamAdi());
```

▲ İç İçe Nesne Tanımlama ve Çağırma

Bir nesnenin özelliği detayları varsa onu yeni bir nesne gibi nesnenin içinde tanımlarız. Bu durumda da içiçe nesne oluşmuş olur.

```
var kisi = {
      adi    : "Kubilay",
      soyadi : "Keskin",
      dgmTrh : 1994,
      vucutBilgisi:{
         boy     : 175,
         kilo    : 75,
         gozRenk : "kahve"   }
}
document.write(kisi.vucutBilgisi.gozRenk);
```

Nesnelerde Object.Create Yöntemi ile Nesne Oluşturma

Daha önce oluşturduğumuz bir nesne üzerinden create yöntemi kullanarak oluştururuz. Bir örnek ile yapalım.

```
const person = {
         name : "tasarım",
         surname : "yazılım",
         age : 2,
         fullname : function () {
            return this.name + " " + this.surname;
         }
      }
const person1 = Object.create(person);
person1.name = "kodlama";
person1.surname = "yazılım";
person1.age = 3;

console.log(person1);
console.log(person1.fullname());
```

Kazanım

- Öğrenci bu bölümde nesne ile tanışır ve nesne yönelimli programlamayı uygulamalarla öğrenir.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Nesne yönelimli programlama konusunun anlatılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencileri nesne ile tanıştırma ve nesne yönelimli programlama konusunun anlatılması hedeflenmektedir.
2	☑ Nesne yönelimli programlama konusuna ait uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen eşliğinde öğrencilerin nesne yönelimli programlama konusuna ait uygulamaları tamamlaması amaçlanmaktadır.

NESNE YÖNELIMLI PROGRAMLAMA (2.BÖLÜM)

Müfredat Kodu / Ünite / Konu	İCT-301-3.11.2 / JavaScript / Nesne Yönelimli Programlama (2.Bölüm)
Hedeflenen Süre	55 Dakika

🚀 Nesne Yönelimli Programlama (2.Bölüm)

⚡ *JavaScript Nesnelerde For İn Döngüsünün Kullanımı*

Nesnelerde for in döngüsünü kullanırken nesne içindeki her özellik sırayla döndürülür.

```
var aracBilgi = { marka     : "toyota",
                  uretimYil : 2015,
                  serisi    : "corolla",
                  vites      : "manuel"  };

for (const item in aracBilgi) {
   document.write( item + " : " + aracBilgi[item] + "<br/>" )
}
```

✍️ JavaScript Nesnelerde Kullanılan Metotlar

▲ JavaScript Nesnelerde Delete() Metodu Kullanımı

Nesnelerde delete() Metodu, bir nesnenin bir özelliğini silmek için kullanılır. Örnek:

```
var kisi = {
       adi    : "Kubilay",
       soyadi : "Keskin",
       dgmTrh : 1994    };

 delete kisi.dgmTrh;
 document.write(kisi.dgmTrh);
```

Yukarıdaki örnekte dgmTrh özelliğini sildiğimiz için tanımlanmamış değer (undefined) döndürür.

▲ Nesnelerde Prototype Özelliğinin Kullanımı

Nesnelerde tanımlama yaparken fonksiyon ile tanımlama yapıyorsak bu fonksiyonlara yapıcı fonksiyonlar diyoruz. Aşağıdaki örneğimizde de değerler fonksiyonu bu duruma örnektir.Değerler fonsiyonunun prototype inerek nesnemize ulaşıp işlemler yapabiliriz. Aşağıdaki örnekte olduğu gibi...

```
function degerler(sayi1, sayi2, sayi3) {
   this.say1 = sayi1;
   this.say2 = sayi2;
   this.say3 = sayi3;
```

```
 }
degerler.prototype.topla = function () {
   return this.say1 + this.say2 + this.say3;
 }
var sayilar = new degerler(12,23,34);
document.write(sayilar.topla());
```

JavaScript Nesnelerde hasOwnProperty() Metodu Kullanımı

hasOwnProperty() Metodu, kendisine değer olarak verilen değeri nesnenin özelliklerinde varsa true yoksa false sonucunu döndürür.

```
var aracBilgi = { marka     : "toyota",
                  uretimYil : 2015,
                  serisi    : "corolla",
                  vites     : "manuel"  };
document.write(aracBilgi.hasOwnProperty("marka")  + "<br/>" );
document.write(aracBilgi.hasOwnProperty("renk"));
```

JavaScript Nesnelerde propertyIsEnumerable() Metodu Kullanımı

propertyIsEnumerable() Metodu, kendisine değer olarak verilen değeri nesnenin özelliklerinde varsa ve numaralandırabilir ise true yoksa false sonucunu döndürür.

```
var aracBilgi = { marka     : "toyota",
                  uretimYil : 2015,
                  serisi    : "corolla",
                  vites     : "manuel"  };
document.write(aracBilgi.propertyIsEnumerable("marka")  + "<br/>" );
document.write(aracBilgi.propertyIsEnumerable("renk"));
```

JavaScript Nesnelerde isPrototypeOf() Metodu Kullanımı

isPrototypeOf() Metodu, nesnenin prototype yapısında yapıcı metot (constructor) varsa true yoksa false değeri döndürür. Yapıcı metotlar nesne tanımlanırken nesne içeriğinin fonksiyon içerisinde tanımlanması ile oluşturulur.

```
function myFunc() {
   this.nesneOzellik1 = "deger1",
   this.nesneOzellik2 = "deger2" }
var nesne = new myFunc();
document.write( myFunc.prototype.isPrototypeOf(nesne));
```

JavaScript Nesnelerde instanceof Operatörü Kullanımı

instanceof, nesnenin yapıcı metodu var mı diye kontrol eder varsa true sonucunu döndürür.

```
function myFunc() {
   this.nesneOzellik1 = "deger1",
   this.nesneOzellik2 = "deger2" }
var nesne = new myFunc();
document.write( nesne instanceof myFunc);
```

JavaScript Nesnelerde constructor Kullanımı

Nesnenin yapıcı metoduna (constructor) erişmek için kullanılır.

Örnek 1 : Aşağıdaki gibi yaptığımızda constructor içeriğini döndürür.

```
function myFunc() {
   this.nesneOzellik1 = "deger1",
   this.nesneOzellik2 = "deger2" }
var nesne = new myFunc();
document.write( nesne.constructor );
```

Örnek 2 : Aşağıdaki gibi yaptığımızda ise doğru ise true döndürür. Bu şekilde yaparak koşul ifadelerinde kullanılabilir.

```
function myFunc() {
   this.nesneOzellik1 = "deger1",
   this.nesneOzellik2 = "deger2" }
var nesne = new myFunc();
document.write(nesne.constructor == myFunc);
```

Kazanım

- Öğrenci bu bölümde nesne ile tanışır ve nesne yönelimli programlamayı uygulamalarla öğrenir.

Dersin Hedefleri

	Başlık	Hedef Kitle	Detaylar
1	☑ Nesne yönelimli programlama konusunun anlatılması.	☑ Eğitmen ☐ Öğrenciler	Öğrencileri nesne ile tanıştırma ve nesne yönelimli programlama konusunun anlatılması hedeflenmektedir.
2	☑ Nesne yönelimli programlama konusuna ait uygulamaların yapılması.	☐ Eğitmen ☑ Öğrenciler	Eğitmen eşliğinde öğrencilerin nesne yönelimli programlama konusuna ait uygulamaları tamamlaması amaçlanmaktadır.