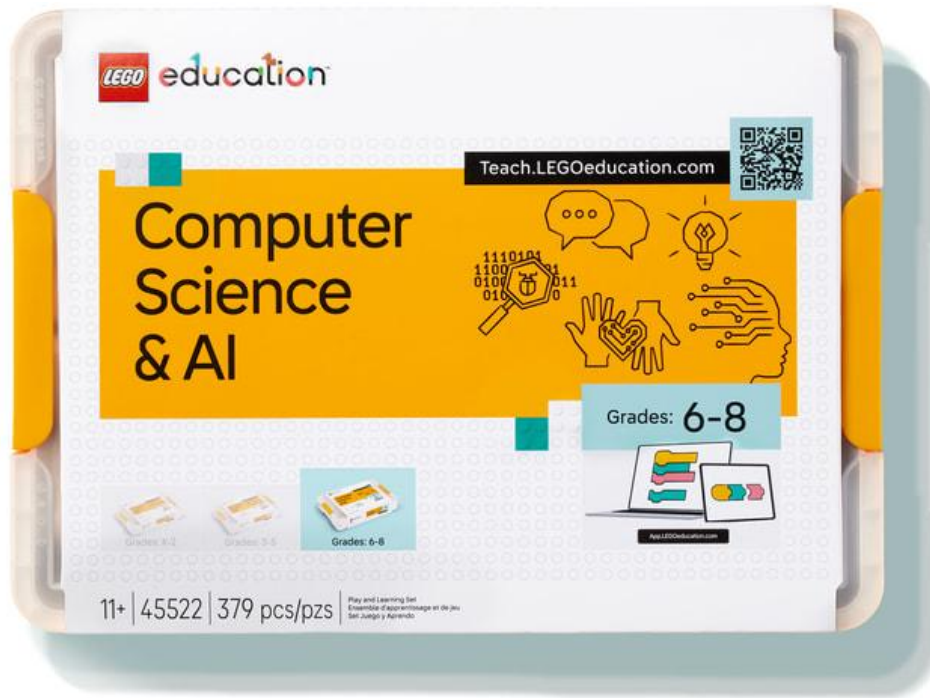




LEGO EDUCATION CS & AI LEARNING İLE PYTHON DÖKÜMANTASYONU





İçindekiler Tablosu

LEGO EDUCATION CS & AI KİTİ İLE PYTHON DOKÜMANTASYONU.....	2
1. Python'a giriş	2
2. Python'da sözdizimi.....	2
3. Çalışma Ortamı ve Veri Tipleri.....	4
4. LEGO Education Modülleri	5
5. Parametre Ekleme	6
6. Pseudo Code (Sözde Kod)	7
7. Asenkron Programlama nedir?.....	7
8. Kontrolcü (Controller) Kullanımı.....	8
9. Motor Kontrolü	9
10. Döngüler	11
11. Sensörler	12
12. TÜM KOD ÖRNEKLERİ REFERANS SAYFASI.....	13
KOD 1: Single Motor (Tekli Motor) Temel Kontrolü	13
KOD 2: Double Motor (Çiftli Motor) Senkronize Sürüş ve IMU Okuma	14
KOD 3: Color Sensor (Renk Sensörü) ile Asenkron Döngü.....	15
KOD 4: Combine 2 - Controller (Kontrolcü) ve Çiftli Motor Teleoperasyonu.....	15



LEGO EDUCATION CS & AI KİTİ İLE PYTHON DOKÜMANTASYONU

1. Python'a giriş

Python, yüksek seviyeli bir programlama dilidir ve genellikle okunabilir ve anlaşılabilir sözdizimi ile tanınır.¹ Guido Van Rossum tarafından geliştirilmeye başlanan Python, basit ve etkili bir şekilde kod yazmayı teşvik eder.¹ Geniş bir standart kütüphane içerir ve çeşitli uygulama alanlarında kullanılabilir.¹

Python'un temel felsefesi "İşleri basit ve açık bir şekilde ifade etmek" olarak özetlenebilir.¹ Bu felsefe, karmaşıklığı azaltmayı ve kodun okunabilirliğini artırmayı amaçlar. Bu nedenle, Python dilinde kod yazmak ve anlamak oldukça kolaydır.¹ Python'u uygulamaya başladığında aynı zamanda 1970'lerde BBC komedi dizisi olan "Monty Python's Flying Circus"un yayınlanmış senaryolarını da okuyan Guido Van Rossum; kısa, benzersiz ve biraz da gizemli bir isme ihtiyacı olduğunu düşündü ve bu dile Python adını vermeye karar verdi.¹

Python, çeşitli programlama paradigmalarını destekler, bunlar arasında nesne yönelimli, işlevsel ve prosedürel programlama bulunur.¹ Ayrıca, dinamik tiplendirme özelliği sayesinde değişken türlerini belirtmeniz gerekmez.¹ Python'un geniş bir topluluğu bulunmaktadır ve bu topluluk sürekli olarak yeni kütüphaneler ve araçlar geliştirerek dilin işlevselliğini genişletmektedir.¹ Python betik dosyaları (.py uzantılı) kullanılarak yazılır ve yorumlanır, bu da kodun hızlıca test edilip çalıştırılmasını sağlar.¹

2. Python'da sözdizimi

Sözdizimi nedir? Sözdizimi, bir programlama dilinin nasıl yazılması gerektiğini belirleyen kurallar bütünüdür.¹ Bir dilin sözdizimi, programcıların kodu yazarken takip etmeleri gereken dilbilgisel yapıları ve düzenleri tanımlar.¹ Temel amacı, insanların programları anlayabilir ve düzenleyebilir bir biçimde yazmalarını sağlamaktır.¹ Python'un sözdizimi de bu amacı taşır ve aşağıdaki temel hedefleri güder:¹

- Okunabilirlik: Python, açık ve anlaşılabilir bir sözdizim ile tasarlanmıştır. Kodu okumak ve anlamak diğer dillere göre daha kolaydır.¹
- Kodun Düzeni: Python'ın girinti tabanlı sözdizimi, kodun yapısını ve hiyerarşisini belirlemek için kullanılır.¹
- Hata Azaltma: Sözdizimi kuralları, geliştiricilerin hataları daha erken aşamalarda yakalamasını sağlar.¹
- Kod Konsistansı: Python, kodun nasıl yazılması gerektiği konusunda belirgin kurallar sunar.¹
- Dilin Anlaşılabilirliği: Python, dilin kendine özgü bir şekilde ifade edilmesine olanak tanır.¹



a. Girintiler: Python'da kod blokları oluşturmak için girinti (indentation) kullanılır.¹ Diğer programlama dilleri genellikle süslü parantezler veya başka sembollerle kod bloklarını belirtirken, Python'da bloklar girinti ile belirlenir.¹ Örneğin, bir "if" koşulu altında çalışacak kodu belirtirken, kodu içe gömerek yani girinti yaparak belirtiriz.¹

Örnek:

```
if x > 5:
    print("x 5'ten büyük")
```

Burada print fonksiyonuna dikkat edin gördüğünüz üzere aynı sütun da değiller.¹ Bunun sebebi print fonksiyonun if koşulunun bir parçası olduğunu göstermek içindir.¹

b. Değişkenler ve Atama: Python'da değişkenler, değerleri saklamak için kullanılan isimlendirilmiş bellek bölgeleridir.¹ Değişkenlere değer ataması yaparken = işaretini kullanırız.¹

Örnek:

```
x = 10
y = "Merhaba, Dünya!"
# Burada x ve y değişkenlerine sırasıyla bir
sayı ve bir metin atandı.
```

c. Yorumlar: Yorumlar, kodun anlaşılmasını ve açıklamasını kolaylaştırmak için kullanılır.¹ Yorum satırları, Python tarafından kodun çalıştırılmasında dikkate alınmaz.¹ Örnek:

```
# Bu bir yorum satırıdır
```



d. Fonksiyonlar ve Metodlar: Fonksiyonlar, belirli bir işlevi yerine getiren kod bloklarıdır. Python'da fonksiyonlar `def` anahtar kelimesi ile tanımlanır.¹

```
def toplama(a, b):  
    return a + b
```

Burada 'toplama' adında bir fonksiyon tanımlandı. Bu fonksiyon, iki parametre alır ve bu parametrelerin toplamını döndürür.¹

e. Koşullu İfadeler: Python'da koşullu ifadeler, belirli şartlara bağlı olarak farklı kod bloklarının çalıştırılmasını sağlar. `if`, `elif` (opsiyonel) ve `else` ifadeleri kullanılır.¹ Örnek:

```
if x > 0:  
    print("x pozitif")  
elif x == 0:  
    print("x sıfır")  
else:  
    print("x negatif")
```

Bu örnekte, "if" koşulu, x değerinin pozitif olup olmadığını kontrol eder. Eğer pozitifse, ilgili mesaj yazdırılır. Eğer değilse, başka bir koşul kontrol edilir.¹

3. Çalışma Ortamı ve Veri Tipleri

Eski nesil SPIKE Prime Hub'i, nispeten sınırlı belleğe sahip olduğundan MicroPython kullanmaktaydı.¹ Ancak yeni LEGO Education CS & AI Kiti (45522), kodun doğrudan kendi bilgisayarınızda çalıştığı "Bağlı Yürütme" (Tethered) mimarisini kullanır.² Kodu çalıştırmak için bilgisayarınıza standart Python 3 (minimum sürüm 3.14) kurmanız ve `pip install legoeducation` komutunu çalıştırmanız gerekir.²

Kod Düzenleyici'nin metin ve sayıları farklı renklerde gösterdiğini göreceksiniz çünkü bunlar farklı veri tipleridir.¹ Python ayrıca tam sayılar ve ondalık sayılar arasında ayrım yapar.¹ Tam sayılar, optimize edilmiş tam sayılar (Integer) veri tipi olarak da bilinir. Ondalık sayılar, optimize



edilmemiş float veri tipini kullanır, bu nedenle donanım modülleri hız ve derece gibi komutlarda bu veri tipinden kaçınır.¹ Bu nedenle, tam sayılar kullanmalı veya ondalık sayıları tanımlamak için farklı birimler kullanmalısınız.¹ Yarım saniye için 0.5 saniye yerine 500 milisaniye kullanabilirsiniz.

Diziler (stringler) 2 tırnak (") arasında veya iki kesme işareti (') arasında olması ile anlaşılabilir.¹

```
"Hello World" # Bu bir String veri tipli metindir.  
0.895         # Bu bir float tipi veridir.  
234           # Bu bir Integer tipi veridir.
```

Bu veri tiplerini kullanabileceğiniz bir örneği ise şöyle verebiliriz:¹

```
import time # python'un zaman kütüphanesi  
  
time.sleep(2) # print fonksiyonumuzun çalışmadan önce 2 saniye  
beklemesini sağlıyoruz  
print("2 saniye duruldu")
```

4. LEGO Education Modülleri

Donanım bileşenlerini kontrol etmek için legoeducation kütüphanesine ihtiyaç duyarsınız.² Modüller, ilgili kodu düzenlemek için kullanılır.¹ Her CS & AI bileşeni için bir sınıf vardır; örneğin motor sınıfı, motorları kontrol etmek için gereken kodu içerir.¹ Bir modülün işlevini kullanmak için öncelikle kütüphaneyi şu deyimle içe aktarın:¹



```
import legoeducation as le
```

İhtiyacınız olan modülleri, Python programınızın başında bir kez içe aktarın.¹ CS & AI kitlerinde donanım bilgisayarla Bluetooth (BLE) üzerinden eşleşir ve Bağlantı Kartlarının (Connection Card) renkleri/seri numaraları kullanılarak tanımlanır.²

5. Parametre Ekleme

Aşağıdaki örnek merhaba() fonksiyonumuzun parametresi bulunmamaktadır; bu nedenle her çağırıldığında konsolda "Merhaba!" yazar.¹ Daha dinamik hale getirmek için fonksiyon tanımının parantezlerinin içine bir parametre adı ekleyin.¹ Fonksiyon gövdesindeki kod bloğu, parametrenin değerine göre farklı bir şey yapmak için bu parametreyi kullanabilir.¹ Bu + operatörü stringleri uçlarından birbirine bağlamaya yarar.¹

Örnek 1:

```
def merhaba(): # dinamik olmayan  
    print("Merhaba!")
```

Örnek 2:

```
def merhaba(isim): # dinamik olan  
    print("Merhaba " + isim + "!")  
    # Parametrenin değişkenliği yüzünden bu fonksiyon  
    daha dinamiktir.
```



6. Pseudo Code (Sözde Kod)

Pseudo code, bir programın mantığını ve adımlarını anlamak ve ifade etmek için kullanılan, gerçek programlama diline ait olmayan bir yazım şeklidir.¹ Pseudo code, genellikle insana ve diğer programcılara programın genel yapısını ve işleyişini anlatmak için kullanılır.¹ Gerçek bir programlama diliyle yazılmış kod gibi çalışmayacaktır, ancak temel işlem adımlarını ve mantığı anlamak için kullanışlıdır.¹

Gerçek Python kodu:

```
sayi1 = 5
sayi2 = 10
toplam = sayi1 + sayi2
print("Toplam:", toplam)
```

Pseudo Code:¹

- sayi1 değişkenini 5 olarak tanımla
- sayi2 değişkenini 10 olarak tanımla
- toplam değişkenini sayi1 + sayi2 olarak hesapla
- "Toplam:" yazısını ve toplam değişkenini ekrana yazdır

7. Asenkron Programlama nedir?

Asenkron programlama, bir programın işlemlerini sırayla ve beklemeksizin gerçekleştirmek amacıyla kullanılan bir programlama yaklaşımıdır.¹ Bu yaklaşım, özellikle uzun süren işlemlerin paralel olarak çalışmasını sağlamak için kullanılır.¹ Böylece bir işlem tamamlanırken diğer işlemler engellenmez ve programın daha hızlı ve verimli çalışmasına olanak tanır.¹

Yeni nesil CS & AI (45522) kiti, bilgisayardan Bluetooth (BLE GATT) üzerinden yönetildiği için asenkron programlamayı Geri Çağırma (Callback) fonksiyonları ile gerçekleştirir.³ Kodunuz kitlenip beklemesiz; donanımdan veri geldiğinde sistem otomatik olarak daha önce belirlediğiniz fonksiyonu tetikler.³



```
import legoeducation as le
import time

# Asenkron olarak tetiklenecek fonksiyon (Callback)
def bildirim_geldi(data):
    parsed_items = le.device_notification_parser(data)
    for parsed_item in parsed_items:
        if isinstance(parsed_item, le.MotorNotification):
            if parsed_item.motorBitMask == le.MOTOR_BITS_RIGHT:
                print(f"Sağ Motor Konumu: {parsed_item.position}")

doublemotor = le.DoubleMotor()
doublemotor.connect(card_color=le.LEGO_COLOR_AZURE,
                    card_serial='3683')
# Asenkron dinlemeyi başlat
doublemotor.set_notification_callback(bildirim_geldi)

# Ana program kilitlenmeden devam eder
while True:
    time.sleep(1)
```

8. Kontrolcü (Controller) Kullanımı

SPIKE Prime setlerinde yer alan matris ekranların aksine, CS & AI kitlerinde dışarıdan müdahale edebileceğiniz fiziksel bir Kontrolcü (Controller) bulunur.² Bu bölümde robotumuzu manuel olarak nasıl yöneteceğimizi öğreneceğiz. Kontrolcü üzerinde sol ve sağ olmak üzere iki joystick bulunur.²

Kontrolcünden iki tür okuma yapabiliriz:

- Açı (Angle): Joystick'in hangi yöne (kaç derece) itildiğini gösterir.²
- Yüzde (Percentage): Joystick'in ne kadar kuvvetle/derine itildiğini yüzde olarak gösterir.²

Örnek Kullanım:



```
import legoeducation as le
import time

controller = le.Controller()
controller.connect(card_color=le.LEGO_COLOR_AZURE,
card_serial='9999')

while True:
    sol_itme_yuzdesi = controller.get_left_lever_percentage()
    print(f"Sol Kol İtme Gücü: %{sol_itme_yuzdesi}")
    time.sleep(0.1)
```

9. Motor Kontrolü

CS & AI kiti içinde Single Motor (Tekli) ve Double Motor (Çiftli) olmak üzere iki eyleyici birim bulunur.²

Bölüm I: Sürekli Çalıştırma En kolay motor kontrol yöntemidir. Motora sadece dönüş yönü ve hız veririz. Ancak motor durdurulmazsa sonsuza kadar ilerler.¹

```
import legoeducation as le
import time

singlemotor = le.SingleMotor()
# Bağlantı varsayılmıştır
singlemotor.run(direction=le.MOVEMENT_DIRECTION_FORWARD, speed=50)

time.sleep(2)
singlemotor.stop(end_state=le.END_STATE_BRAKE) # Motoru durdurur
```

Bölüm II: Dereceyle Kontrol Etmek Çift Motor modülünü kullanarak robotumuzu belirli bir derece



sürdürebiliriz.³

```
import legoeducation as le

doublemotor = le.DoubleMotor()
# Robotu geriye doğru %10 hızla 180 derece sür
doublemotor.movement_move_for_degrees(
    180,
    direction=le.MOVEMENT_MOVE_DIRECTION_BACKWARD,
    speed=10
)
```

Bu fonksiyon parametrelere ayrıştırıldığında:¹

1. Parametre (180): Motorun kaç derece gideceğini belirtiyoruz.¹
2. Parametre (direction): Motorun dönüş yönü (FORWARD / BACKWARD).¹
3. Parametre (speed): Ne kadar hızlı gideceği (0-100 arası yüzde).¹

Bölüm III: Zamana Göre Çalıştırma Motorumuzu zamana göre çalıştırmayı öğreneceğiz.¹
time.sleep() yerine motorun doğrudan zaman tabanlı metodu kullanılabilir.

```
# Çift motoru 1 saniye (1000 ms) boyunca ileri sür
doublemotor.movement_move_for_time(
    time_ms=1000,
    direction=le.MOVEMENT_DIRECTION_FORWARD,
    speed=50
)
```

Bölüm IV ve V: Mutlak ve Bağıl Konum ile Bitiş Durumları Motorlar içindeki encoderler sayesinde mutlak (absolute) ve bağıl (relative) konumlara gidebilirler.² Ayrıca motor durduğunda ne



yapacağı şu sabitlerle (constants) belirlenir:¹

- le.END_STATE_COAST: Motor boşa alınır, elektrik gitmez.¹
- le.END_STATE_BRAKE: Motor fren yapar, durduktan sonra da fren yapmaya devam eder.¹
- le.END_STATE_HOLD: Motor pozisyonunu PID ile korur.¹

10. Döngüler

Döngülerin sözdizimi de bir if koşulu gibidir. Döngü fonksiyonunun çağırımı ardından bir sütun boşlukla yazılır.¹

Bölüm I: While Döngüleri While döngüleri verilen veri doğru olduğu sürece kodu devam ettirir.¹ Örnek olarak while True sonsuza kadar gider.¹

```
a = 0
while a < 5:
    a = a + 1
```

Bu kod a'ya beşten yüksek olana dek 1 ekler ve 5 ten fazla olduğu zaman da döngü biter.¹

Bölüm II: For Döngüsü For döngüleri yineleme yapmak için kullanılır.¹ Örnek olarak bir listeniz var ve bu listenin içindeki her şeyi ekrana çıkartmak istiyorsunuz:

```
liste = [1, 2, 3, 4, "abc"]
for i in liste:
    print(i)
```

Buradaki "i" bir değişkendir. Bu değişken liste adlı listenin içeriklerinin verilerini alır her dönüşte ya da yinelemede.¹



Döngü Ekleri: "Break" ile başlarsak, bu fonksiyonu bir döngü içerisinde çağırırsak döngüden çıkmamızı sağlar.¹

```
for i in "ornek_string":
    if i == "k":
        break # k'ye geldiği zaman döngüden çıkar
    print(i)
```

"Continue" fonksiyonumuz çağrıldığında ise geri kalan kodu atlayarak döngüyü sonraki adımdan devam ettirir.¹ Son olarak "Range", 0'dan başlayarak x-1 sayısına kadar döngüyü döndürür.¹ range(6) yazarsak döngü altı kere döner (0,1,2,3,4,5).¹

11. Sensörler

Bu bölümde CS & AI kitinin özel donanımlarından bahsedeceğiz. Yeni mimaride sensörler de Bluetooth üzerinden haberleştiği için nesne yönelimli şekilde kullanılırlar.

Renk Sensörü (Color Sensor) Renk sensörünün birden fazla işlevi vardır.¹ Önüne koyulan renkli objenin rengini tanımlayabilir veya ışığın yansımaya seviyesini ölçebilir.¹

1. Işığın yansımaya seviyesini ölçmek:

```
import legoeducation as le

color_sensor = le.ColorSensor()
# Yansımaya seviyesini yüzdelik olarak verir
yansima = color_sensor.get_reflection()
```

2. Renk Tanımlama:



```
# 0 anda gördüğü rengin Kategorik LEGO  
Numarasını geri verir  
renk = color_sensor.get_color_number()
```

IMU - 6 Eksenli Sensör (Double Motor İçinde) CS & AI kitinin Çiftli Motor (Double Motor) modülü içine gizlenmiş, 3 eksenli ivmeölçer ve 3 eksenli cayroskop barındıran bir atalet ölçüm ünitesi (IMU) bulunur.²

```
import legoeducation as le  
  
doublemotor = le.DoubleMotor()  
# Bağlantı varsayıldı  
pusula_yonu = doublemotor.imu_device.yaw  
yokus_acisi = doublemotor.imu_device.pitch
```

(Not: SPIKE Prime setinde bulunan Kuvvet ve Mesafe sensörleri bu özel sete dahil değildir, ancak genel ekosistem mantığı gereği projelerde entegre edilebilirler).

12. TÜM KOD ÖRNEKLERİ REFERANS SAYFASI

Aşağıda, dokümantasyonda işlenen yöntemlerin gerçek hayattaki Python karşılıkları listelenmiştir. Kodlar doğrudan kopyalanıp uygulanabilir yapıdadır.

KOD 1: Single Motor (Tekli Motor) Temel Kontrolü

```
import legoeducation as le  
import time
```

```
# Bağlantı Kartı değerlerini girin  
card_color = le.LEGO_COLOR_AZURE
```



```
card_serial = '1234'

# Motor objesini oluştur ve bağlan
singlemotor = le.SingleMotor()
singlemotor.connect(card_color=card_color, card_serial=card_serial)

if not singlemotor.connected:
    print('Tekli motora bağlanılamadı.')
    exit(1)

print('Motor 3 saniye boyunca çalıştırılıyor...')
# Motoru ileri yönde %75 hızla çalıştır
singlemotor.run(direction=le.MOVEMENT_DIRECTION_FORWARD, speed=75)

time.sleep(3)

# Motoru aktif frenleme (Brake) ile durdur
singlemotor.stop(end_state=le.END_STATE_BRAKE)
singlemotor.disconnect()
```

KOD 2: Double Motor (Çiftli Motor) Senkronize Sürüş ve IMU Okuma

```
import legoeducation as le
import time

doublemotor = le.DoubleMotor()
doublemotor.connect(card_color=le.LEGO_COLOR_AZURE, card_serial='3683')

if doublemotor.connected:
    # Senkronize Sürüş: Geriye doğru, tam 180 derece, %10 güç ile.
    doublemotor.movement_move_for_degrees(
        180,
        direction=le.MOVEMENT_MOVE_DIRECTION_BACKWARD,
        speed=10
    )

    time.sleep(1)

    # IMU Oryantasyon Kalibrasyonu ve Okuma
    doublemotor.imu_reset_yaw_axis(0)
    doublemotor.imu_set_yaw_face(yaw_face=le.DEVICE_FACE_LEFT)
```



```
# Anlık pusula yönünü konsola yazdır
print(f"Anlık Yaw (Sapma) Açısı: {doublemotor.imu_device.yaw}")

doublemotor.disconnect()
```

KOD 3: Color Sensor (Renk Sensörü) ile Asenkron Döngü

```
import legoeducation as le
import time

color_sensor = le.ColorSensor()
color_sensor.connect(card_color=le.LEGO_COLOR_AZURE, card_serial='5678')

try:
    while color_sensor.connected:
        renk_id = color_sensor.get_color_number()
        yansima = color_sensor.get_reflection()

        print(f"Renk: {renk_id} | Yansima: %{yansima}")

        if renk_id == le.COLOR_RED:
            print("KIRMIZI RENK ALGILANDI! GÖREV TAMAMLANDI.")
            break

        time.sleep(0.5)
except KeyboardInterrupt:
    color_sensor.disconnect()
```

KOD 4: Combine 2 - Controller (Kontrolcü) ve Çiftli Motor Teleoperasyonu

```
import legoeducation as le
import time

doublemotor = le.DoubleMotor()
controller = le.Controller()

# Cihazlara bağlantı sağlandığı varsayılmaktadır...
try:
    print("Uzaktan kumanda sürüş modu başlatıldı. Kapatmak için CTRL+C")
    while True:
        # Sol ve Sağ analog kolların ne kadar itildiğini yüzdelik olarak oku (0-100)
```



```
sol_hiz = controller.get_left_lever_percentage()
sag_hiz = controller.get_right_lever_percentage()

# Okunan analog yüzdeleri doğrudan motor hızlarına (speed) ata
doublemotor.motor.set_speed(sol_hiz)
doublemotor.motor.set_speed(sag_hiz)

time.sleep(0.02) # Saniyede 50 kere (50 Hz) yenileme
except KeyboardInterrupt:
    print("Sistem güvenli bir şekilde kapatılıyor...")
    doublemotor.stop(end_state=le.END_STATE_COAST)
```

Alıntılanan çalışmalar

1. LEGO SPIKE Prime Python Programming Guide
2. LEGO/LEGOEducation - GitHub, erişim tarihi Mayıs 9, 2026,
<https://github.com/LEGO/LEGOEducation>
3. LEGOEducation/doublemotor.md at main - GitHub, erişim tarihi Mayıs 9, 2026,
<https://github.com/LEGO/LEGOEducation/blob/main/doublemotor.md>