

VERİ BİLİMİ DERSİ

# Python

# Temelleri ve Veri Bilimi Ekosistemi

Hafta 1 · Modül 1

**Dr. Murat Altun**

Veri Bilimi ve Yapay Zekâ Eğitimi · 2026

6

Saat

3

Notebook

∞

Potansiyel

1

## Veri Bilimi Ekosistemi

Veri bilimi nedir, neden önemlidir, yol haritası ve Google Colab tanıtımı

2

## Python Temelleri

Değişkenler, veri tipleri, operatörler, input/output işlemleri

3

## Veri Yapıları ve Kontrol Akışı

Listeler, sözlükler, tuple, set, döngüler ve koşullu ifadeler

4

## Fonksiyonlar ve GitHub

Fonksiyon tanımlama, lambda, GitHub temelleri, ödevler

## Tanım

Veri bilimi; istatistik, matematik, programlama ve alan bilgisini birleştirerek veriden anlamlı içgörüler çıkaran ve bu içgörülerini karar alma süreçlerine dönüştüren disiplinler arası bir alandır.

**150K+**

Türkiye'de yıllık  
veri bilimi iş ilanı

**₺85K**

Ortalama başlangıç  
maaşı (aylık brüt)

**%36**

Yıllık büyüme  
oranı (2023-2026)



Bu eğitimde 15 hafta boyunca tüm bu aşamaları kapsayan uygulamalı projelerle ilerleyeceğiz. Her hafta en az bir notebook ve bir mini proje tamamlayacaksınız.

## Google Colab Nedir?

- Tarayıcıda çalışan ücretsiz Jupyter ortamı
- Google Drive entegrasyonu
- Ücretsiz GPU/TPU erişimi
- Kurulum gerektirmez
- Paylaşım ve işbirliği kolay
- [colab.research.google.com](https://colab.research.google.com)

## Hücre Türleri

Kod Hücresi:

Python kodu yazıp çalıştırabilirsiniz  
Shift+Enter ile hızlı çalıştırma

Metin (Markdown) Hücresi:

Açıklama, formül, başlık yazabilirsiniz  
LaTeX matematik formülleri destekler

## Hızlı İpuçları

Ctrl+Enter → Hücreyi çalıştır | Shift+Enter → Çalıştır ve sonrakine geç | Ctrl+M B → Altına yeni hücre | Ctrl+M D → Hücreyi sil | !pip install paket → Paket yükle

Tip	Anahtar	Örnek	Açıklama
Metin	<code>str</code>	"Merhaba"	Karakter dizisi, tırnak içinde
Tam Sayı	<code>int</code>	42	Ondalık kısmı olmayan sayılar
Ondalıklı	<code>float</code>	3.14	Noktalı (ondalıklı) sayılar
Mantıksal	<code>bool</code>	True / False	Doğru veya yanlış değer

```
# Değişken tanımlama isim = "Ahmet" # stryas = 25 # intboy = 1.78
# floatogrenci_mi = True # boolprint(type(isim)) # <class 'str'>
```

## Aritmetik

+ Toplama  
- Çıkarma  
\* Çarpma  
/ Bölme  
// Tam bölme  
% Mod (kalan)  
\*\* Üs alma

## Karşılaştırma

== Eşit mi?  
!= Farklı mı?  
> Büyük mü?  
< Küçük mü?  
>= Büyük eşit

## Mantıksal

and Her ikisi doğru  
or En az biri doğru  
not Tersini al

Örnek:  
x > 5 and x < 10  
not ogrenci\_mi

## input() – Kullanıcıdan Veri Alma

Kullanıcıdan klavye ile veri alır.  
Dönen değer her zaman str tipindedir.  
Sayı almak için int() veya float() ile dönüştürün.

## print() – Ekrana Yazdırma

Değerleri ekrana yazdırır.  
f-string ile değişken gömme: f"..."  
sep ve end parametreleri ile özelleştirme.

```
# input() ile veri almaisim = input("Adınız: ")yas = int(input("Yaşınız: "))# f-string ile formatlı çıktıprint(f"Merhaba {isim}, {yas} yaşındasınız!")# sep ve end parametreleriprint("Python", "Veri", "Bilimi", sep=" | ")# Çıktı: Python | Veri | Bilimi
```

## Liste Nedir?

Sıralı, değiştirilebilir (mutable) ve farklı veri tiplerini barındırabilen koleksiyon yapısıdır.

## Temel Özellikler

- İndeks 0'dan başlar
- Negatif indeks: -1 son eleman
- Dilimleme (slicing): liste[1:4]
- İç içe listeler

```
# Liste oluşturmameyveler = ["elma", "armut", "kiraz", "muz"]# Eleman ekleme ve silmemeyveler.append("portakal")
# Sona eklemeyveler.insert(1, "çilek") # İndekse eklemeyveler.remove("armut") # Değere göre
sil# Dilimleme (Slicing)print(meyveler[1:3]) # ["çilek", "kiraz"]print(meyveler[::-1]) #
Ters çevir
```

## Sözlük (Dictionary)

Anahtar-değer (key-value) çiftlerinden oluşur.  
Anahtarlar benzersiz olmalıdır.

```
ogrenci = { "ad": "Ayşe", "yas": 22,  
           "bolum": "Bilgisayar"}#  
Erişimprint(ogrenci["ad"])ogrenci["not"] =  
3.5
```

## Tuple (Demet)

Sıralı ve değiştirilemez (immutable).  
Parantez ile tanımlanır.

```
# Tuple oluşturmakoordinat = (41.01,  
28.97)renkler = ("kırmızı", "mavi")#  
Unpackingx, y = koordinatprint(f"Enlem:  
{x}")# Değiştirilemez!# koordinat[0] = 40 #  
HATA!
```

## Set Nedir?

- Sırasız ve tekrarsız eleman koleksiyonu
- Matematikteki küme kavramına karşılık gelir
- Hızlı üyelik testi (in operatörü) sağlar

## Küme İşlemleri

union() → Birleşim ( $A \cup B$ )  
intersection() → Kesişim ( $A \cap B$ )  
difference() → Fark ( $A - B$ )

```
# Set oluşturmaA = {1, 2, 3, 4, 5}B = {4, 5, 6, 7, 8}# Küme işlemleriprint(A | B)          # Birleşim: {1,2,3,4,5,6,7,8}print(A & B)        # Kesişim:  {4, 5}print(A - B)        # Fark:      {1, 2, 3}# Tekrarları kaldırmasehirler = ["Ankara", "İstanbul", "Ankara", "İzmir"]print(set(sehirler)) # {'Ankara', 'İstanbul', 'İzmir'}
```

for döngüsü, bir koleksiyon (liste, string, range vb.) üzerinde iterasyon yapar. Python'da en sık kullanılan döngü yapısıdır. range(), enumerate() ve list comprehension ile güçlü kombinasyonlar oluşturur.

```
# range() ile döngüfor i in range(5):    print(f"Adım {i}")# enumerate() – indeks + değerdersler = ["Python", "İstatistik", "ML"]for i, ders in enumerate(dersler, 1):    print(f"{i}. {ders}")# List Comprehension – tek satırda listekareler = [x**2 for x in range(1, 11)]cift_kareler = [x**2 for x in range(1, 11) if x % 2 == 0]print(kareler) # [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

## while Nedir?

Koşul doğru olduğu sürece çalışır.  
break ile döngüden çıkılabilir.  
continue ile sonraki iterasyona geçilir.

## Dikkat!

Sonsuz döngü riski var!  
Koşulun bir noktada False olacağından  
emin olun veya break kullanın.

```
# Sayı Tahmin Oyunu
import random
gizli = random.randint(1, 100)
hak = 7
while hak > 0:
    tahmin = int(input("Tahmininiz: "))
    if tahmin == gizli:
        print("Tebrikler!")
        break
    elif tahmin < gizli:
        print("Daha büyük!")
    else:
        print("Daha küçük!")
        hak -= 1
```

## if

Koşul doğruysa çalışır



## elif

Önceki yanlışsa bunu kontrol et



## else

Hiçbiri değilse çalışır

## Örnek: Vücut Kitle İndeksi (VKİ) Hesaplayıcı

```
# VKİ Hesaplayıcı
kilo = float(input("Kilonuz (kg): "))
boy = float(input("Boyunuz (m): "))
vki = kilo / (boy ** 2)
if vki < 18.5:
    print(f"VKİ: {vki:.1f} – Zayıf")
elif vki < 25:
    print(f"VKİ: {vki:.1f} – Normal")
elif vki < 30:
    print(f"VKİ: {vki:.1f} – Fazla Kilolu")
else:
    print(f"VKİ: {vki:.1f} – Obez")
```

## Fonksiyon Nedir?

Fonksiyonlar, tekrar eden kod bloklarını adlandırarak yeniden kullanılabilir hale getirir. def anahtar kelimesi ile tanımlanır. Parametreler alabilir ve return ile değer döndürebilir. Kodun okunabilirliğini, test edilebilirliğini ve bakımını kolaylaştırır.

```
# Temel fonksiyon tanımlama
def selamla(isim, dil='tr'):    """Kullanıcıyı selamlar."""    if dil == 'tr':
    return f"Merhaba {isim}!"    return f"Hello {isim}!"# Birden fazla değer döndürme
def istatistik(sayilar):    ort =
sum(sayilar) / len(sayilar)    return min(sayilar), max(sayilar), ortmn, mx, ort = istatistik([10, 20, 30,
40])print(f"Min={mn}, Max={mx}, Ort={ort}") # Min=10, Max=40, Ort=25.0
```

## Lambda Nedir?

Tek satırlık anonim fonksiyondur.  
Kısa işlemler için idealdir.  
Söz dizimi: lambda parametre: ifade  
Pandas apply() ile sık kullanılır.

## map() ve filter()

map(): Her elemana fonksiyon uygular  
filter(): Koşula uyanları filtreler  
Her ikisi de lambda ile  
güçlü kombinasyonlar oluşturur.

```
# Lambda örnekleri  
kare = lambda x: x ** 2  
toplam = lambda a, b: a + b  
# map() – her elemana uygulasayilar = [1, 2, 3, 4, 5]  
kareler = list(map(lambda x: x**2, sayilar))  
print(kareler) # [1, 4, 9, 16, 25]  
# filter() – koşula göre  
filtrele  
ciftler = list(filter(lambda x: x % 2 == 0, sayilar))  
print(ciftler) # [2, 4]
```

## Hesap Aç

1

github.com'da ücretsiz hesap oluşturun

## Repo Oluştur

2

New Repository ile yeni proje başlatın

## Commit

3

Değişikliklerinizi kaydedin (snapshot)

## Push

4

Yerel değişiklikleri uzak sunucuya gönderin

```
# Terminal komutları  
git init  
dosyaları hazırla  
git commit -m "İlk commit"  
gönder
```

```
# Yeni repo başlat  
git add .  
# Değişiklikleri kaydet  
git push origin main
```

```
# Tüm  
# Sunucuya
```

1

## Notebook 1: Python Temel

01\_python\_temel.ipynb

Değişkenler, veri tipleri, operatörler, koşullu ifadeler ve döngülerle ilgili 15+ alıştırmaya. Her alıştırmada beklenen çıktı ve açıklama bulunur.

2

## Notebook 2: VKİ Hesaplayıcı

02\_vki\_hesaplayici.ipynb

Kullanıcıdan boy ve kilo alarak VKİ hesaplayan, sonucu kategorize eden ve sonuçları görselleştiren interaktif uygulama.

3

## Notebook 3: Sayı Tahmin Oyunu

03\_sayi\_tahmin\_oyunu.ipynb

Random modülü ile sayı üreten, while döngüsü ve if/else ile ipuçları veren, skor tablosu tutan eğlenceli bir konsol oyunu.

## Bu Hafta Yapılacaklar

- ✓ Google Colab hesabı açın
- ✓ 3 notebook'u tamamlayın:
  - python\_temel alıştırmaları
  - VKİ hesaplayıcı projesi
  - Sayı tahmin oyunu
- ✓ GitHub hesabı oluşturun
- ✓ İlk repo'nuzu açıp notebook'ları yükleyin

Teslim: Hafta 2 dersinden önce

## Önerilen Kaynaklar

### Kitaplar:

- Python Crash Course (E. Matthes)
- Automate the Boring Stuff

### Video:

- CS50P — Harvard (YouTube)
- Corey Schafer Python Tutorials

### Doküman:

- [docs.python.org/3/tutorial](https://docs.python.org/3/tutorial)
- [realpython.com](https://realpython.com)
- [w3schools.com/python](https://w3schools.com/python)

### Uygulama:

- HackerRank Python Challenge
- LeetCode Easy Problems

# Hafta 1 — Özet

- 1 Veri bilimi; istatistik, programlama ve alan bilgisini birleştiren disiplinler arası bir alandır.
- 2 Python'da 4 temel veri tipi (str, int, float, bool) ve 4 koleksiyon yapısı (list, dict, tuple, set) vardır.
- 3 for ve while döngüleri ile if/elif/else yapıları Python'un kontrol akışının temelini oluşturur.
- 4 Fonksiyonlar ve lambda ifadeleri kodun yeniden kullanılabilirliğini ve okunabilirliğini artırır.
- 5 GitHub ile versiyon kontrolü, profesyonel yazılım geliştirmenin vazgeçilmez bir parçasıdır.

*“Veriye dokunan, geleceği şekillendirir.”*