

## PYTHON NOTU

Python da ekrana yazdırma print ile yapılır.

```
Print("MERHABA DÜNYA")
```

```
Print(5+3)
```

### DEĞİŞKENLER

String (str) = metin demektir . Python da tırnak içinde yazılan herşey metindir.

-string üç şekilde yazılabılır; tek tırnak ('),çift tırnak (""),üç tırnak ("")

Üç tırnak = çok satırlı metinlerde kullanılır.

### String indeksleme (index)

Her karakterin bir sırası vardır . Sayım sıfır dan başlar.

Kelime = "merhaba"

```
Print(kelime[0]) m
```

```
Print(kelime[-1]) b
```

Not : negatif sayırlarda kullanılır. Ve kelimenin tersinden sıralamaya başlarız.

### String (slice)

Metnin belli bir bölümünü almak için

Yazi = "Python programlama"

```
Print(yazi[6:0]) çıktısı : python
```

Not: iki nokta üst üste den önceyse soldan saymaya başlarız,sonraysa sağdan başlarız.Ve p 0 diye baslıyo kelimeler arasındaki boşlukta numara alır (6) gibi .

```
Print(yazi[6:18]) "programlama "
```

### String tekrarı

```
Print("merhaba" * 3) merhabamerhabamerhaba
```

- Araya boşluk istersen print(("merhaba" \* 3))

merhaba merhaba merhaba

### String Metotları

Metin büyültme – küçültme (lower-upper)

Metin = "python"

```
Print( metin.lower()) #python
```

```
Print(upper.upper()) #PYTHON
```

## **if – elif – else**

Programı duruma göre yönlendirmeyi sağlar.

```
sayi = int(input("Sayı: "))
```

```
if sayı > 0:  
    print("Pozitif")  
elif sayı < 0:  
    print("Negatif")  
else:  
    print("Sıfır")
```

## **Fonksiyonlar**

Kod tekrarını azaltır.

Daha düzenli program yazmanı sağlar.

```
def selam():  
    print("Merhaba")
```

### **Parametreli fonksiyon**

Fonksiyona dışarıdan veri göndermemizi sağlar.

```
def topla(a, b):  
    print(a + b)
```

```
topla(3, 5)
```

### **Return**

Fonksiyonun dışarıya bir değer göndermesini sağlar.

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

fonksiyonun dışarıya değer döndürmesidir.(return)

```
sonuc = topla(3, 5)
```

## **Döngüler(Loops)**

Bir işlemi tekrar tekrar yapmamızı sağlar.Eller 100 kere yazmak yerine döngü otomatik tekrarlar.

for Döngüsü:

-Bir **liste**, **sayı aralığı**, **metin** veya **herhangi bir tekrar edilebilir veri** üzerinde sırayla gezinir.

Kaç kere edeceği **başından bellidir**.

```
for i in range(5):  
    print(i)
```

çıktı:

```
0  
1  
2  
3  
4  
5
```

```
meyveler = ["elma", "armut", "muz"]
```

```
for m in meyveler:  
    print(m)
```

-döngü listedeki her elemanı tek tek sayar.

## WHILE DÖNGÜSÜ

Bir şart **True olduğu sürece** çalışır.

Kaç kere edeceği **önceden belli değildir**, şart bitince durur.

```
x = 0
```

```
while x < 5:  
    Print(x)  
    x += 1
```

aynı çıktıyı üretir: 0-4

ama burada döngü şarta bağlıdır.

önemli: Şart hiç false olmazsa döngü sonsuz döngüye girer.

## Break

Döngüyü tamamen durdurur.

```
for i in range(10):  
    if i == 5:  
        break  
    print(i)
```

## **Continue**

Bu turu atlar, sonraki turdan devam eder.

```
for i in range(5):
    if i == 2:
        continue
    print(i)
```

## **Listeler Üzerinde Döngü Kullanımı**

```
sayilar = [2, 4, 6, 8]
```

```
toplam = 0
for s in sayilar:
    toplam += s
print("Toplam:", toplam)
```

## **Liste – Tuple – Dict – Set**

### **Liste**

Değiştirilebilir (mutable).

```
liste = [10, 20, 30]
liste.append(40)
liste.remove(20)
```

### **Tuple**

Değiştirilemez (immutable).

```
koordinat = (10, 20)
```

Avantajı: daha hızlı, veri değişimmeyecekse güvenli.

### **Dictionary (dict)**

Anahtar : Değer yapısıdır.

```
kisi = {
    "isim": "Ali",
    "yas": 25
}
print(kisi["isim"])
```

## **Set**

Sırasız ve benzersiz elemanlardan oluşur.

```
kume = {1, 2, 3, 3, 4}
```

```
print(kume) # {1,2,3,4}
```

Kullanımı: veri tekrarı engelleme, kümeler arası işlemler.

```
isimler = ["ali", "ayşe", "ali", "veli"]
```

```
benzersiz = set(isimler)
```

```
print(benzersiz)
```

## **Class Oluşturma**

```
class Araba:
```

```
    pass
```

## **init Metodu**

Nesne oluşturulurken çalışan kurucu metottur.

```
class Araba:
```

```
def __init__(self, marka, hiz):
```

```
    self.marka = marka
```

```
    self.hiz = hiz
```

## **Metot Tanımlama**

```
class Araba:
```

```
def __init__(self, marka, hiz):
```

```
    self.marka = marka
```

```
    self.hiz = hiz
```

```
def hizlan(self):
    self.hiz += 10
    print("Yeni hız:", self.hiz)
```

## **Nesne Oluşturma**

```
a1 = Araba("BMW", 120)
```

```
a1.hizlan()
```