



---

# Python

Kaan ARI

---

1. HAFTA	2. HAFTA	3. HAFTA	4. HAFTA
<ul style="list-style-type: none"> <li>Giriş</li> <li>Python Nedir?</li> </ul>	<ul style="list-style-type: none"> <li>*Koşul İfadeleri</li> </ul>	<ul style="list-style-type: none"> <li>Dictionaries(Sözlük)</li> </ul>	<ul style="list-style-type: none"> <li>Akılda Kalan Sorular ve Hızlı Tekrar.</li> </ul>
<ul style="list-style-type: none"> <li>Hello World</li> <li>Yorum Satırı</li> </ul>	<ul style="list-style-type: none"> <li>*While Döngüsü</li> </ul>	<ul style="list-style-type: none"> <li>**Class/Object</li> </ul>	<ul style="list-style-type: none"> <li>Kodluyoruz!!!</li> </ul>
<ul style="list-style-type: none"> <li>*Değişkenler</li> </ul>	<ul style="list-style-type: none"> <li>*For Döngüsü</li> </ul>	<ul style="list-style-type: none"> <li>Kodluyoruz!!!</li> </ul>	<ul style="list-style-type: none"> <li>Ekstralar</li> </ul>
<ul style="list-style-type: none"> <li>Standart Input ve Output</li> </ul>	<ul style="list-style-type: none"> <li>*Fonksiyonlar</li> </ul>	<ul style="list-style-type: none"> <li>Inheritance(Kalıtım)</li> </ul>	<ul style="list-style-type: none"> <li>_____</li> </ul>
<ul style="list-style-type: none"> <li>Aritmetik Operatörler</li> </ul>	<ul style="list-style-type: none"> <li>Kütüphaneler(Modül)</li> </ul>	<ul style="list-style-type: none"> <li>Dates (Tarih kütüphanesi)</li> </ul>	<ul style="list-style-type: none"> <li>_____</li> </ul>
<ul style="list-style-type: none"> <li>Değişken Tipi Değiştirme (Type-Casting)</li> </ul>	<ul style="list-style-type: none"> <li>Kütüphane ve Fonksiyon Ekleme</li> </ul>	<ul style="list-style-type: none"> <li>JSON Nedir?</li> </ul>	<ul style="list-style-type: none"> <li>_____</li> </ul>
<ul style="list-style-type: none"> <li>STRING Özellikleri ve Metodları</li> </ul>	<ul style="list-style-type: none"> <li>*Listeler</li> </ul>	<ul style="list-style-type: none"> <li>Dosya İşlemleri</li> </ul>	<ul style="list-style-type: none"> <li>_____</li> </ul>
<ul style="list-style-type: none"> <li>*Karşılaştırma ve Mantıksal Operatörler</li> </ul>	<ul style="list-style-type: none"> <li>Kodluyoruz!!!</li> </ul>	<ul style="list-style-type: none"> <li>Try-Except Yapısı</li> </ul>	<ul style="list-style-type: none"> <li>Python Hackathon</li> </ul>

# Python Nedir?

Python, **Guido Van Rossum** adlı Hollandalı bir programcı tarafından yazılmış bir programlama dilidir. Geliştirilmesine 1990 yılında başlayan Python; C ve C++ gibi programlama dilleriyle kıyaslanırsa şöyle sonuçlar elde edebiliriz:

- Daha kolay öğrenilir.
- Program geliştirme sürecini kısaltır yani hızlı yazılır.
- Yukarıdaki verilen programlama dillerinin aksine ayrı bir derleyici ihtiyacı duymaz.
- Hem daha okunaklı, hem daha temiz kodsal söz dizimine sahiptir.

Python'ın bu ve buna benzer özelliklerinden dolayı, dünya çapında ün sahibi büyük kuruluşlar ( **Google**, **Yahoo!** ve **Dropbox** gibi) bünyelerinde her zaman Python programcılarına ihtiyaç duyuyor.



# Python'ın Çalıştığı Ortam

Programlama dili insanla makine arasında iletişim kurmak için kullanılır. Bu iletişimi kurma esnasında bazı işlemler yapılır.

Makine Dili ?

Makine diline çevirme iki şekilde olur bunlar derleme ve yorumlamadır. Biri “anında” diğeri ise “önceden” çevirme işlemidir. Şimdi bunlar nedir anlamaya çalışalım.



# Derleme (Compile)

Derleme(Compile) : Makine diline program çalışmadan önce çevirilir. Bu sayede program her açıldığında tekrar tekrar makine diline çevirmeye gerek yoktur. Program bu sayede daha hızlı çalışır. Bu işlemi Derleyici dediğimiz programlar yapar. Buna önceden çevirme diyebiliriz.



# Yorumlama (Interpretation)

Derleme ile platform bağımlılığı dediğimiz olay ortaya çıkar. Platform bağımlılığı demek, örnek olarak siz programı Windows'a göre derlerseniz bu program Linux 'ta çalışmaz ya da siz programı Arm işlemciye göre derlerseniz bu program Intel işlemcide çalışmaz.

Yorumlama(Interpretation) : Anında çevirme dediğimiz yorumlama için geçerlidir. Program bir anda okunurken bir anda makine diline çevrilir. Program her açılışta yorumlandığından dolayı derlenebilen dillere göre yavaş olur. Bağımsız platform desteği sunar, hata ayıklaması kolay olur, programların boyutu küçük olur.

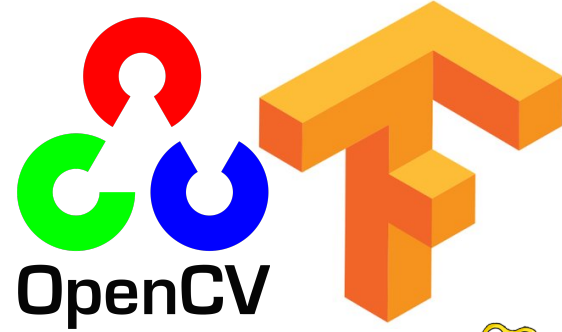


—

Peki...

Sizce **Python**'ı Nereelerde  
Kullanabilirsiniz?

Web Programlama, Server ve Socket Programlama, Oyun Geliştirme, Makine Öğrenmesi ve Yapay Zeka, Görüntü İşleme, Robotik, Haberleşme, Veri İşlemleri ve aklınıza gelebilecek birçok farklı alanda kullanım göstermektedir.





—

**Hocam,  
Nasıl Öğreneceğiz Bu Dili?**

# Cevap: Sorarak

\*Önünüze çıkan sorunları önce kendiniz aşmaya çalışmalısınız.

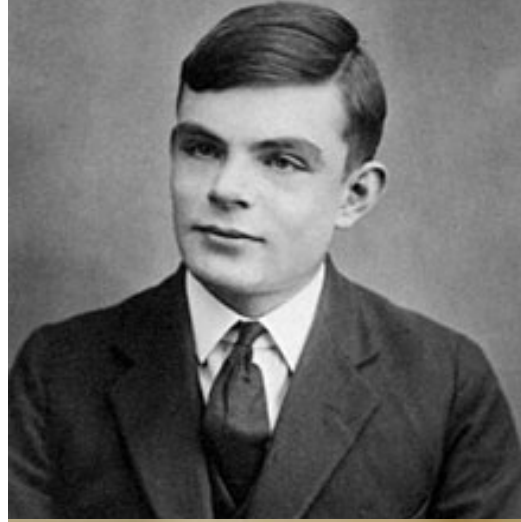
\*Yapamayacağınızı anladığınız zaman uzman yardımına başvurmalısınız.

\*Sormaktan çekinmeyin.

# Öğrenme Amacımız Nedir?

Amacımız, Bu zamana kadar  
yaşadığımız tüketici hayatını  
üretici hayatına  
dönüştürmektir.

Bu sebeple Python  
derslerindeki asıl amaç önceki  
slaytta anlatılanlardan çok  
insanları üretime  
yönlendirmektir.



---

# Başlamadan önce Sorularınız var mı?

(Çekineceğinizi biliyorum çekinmeyin!)

\*Soru sormadan Öğrenemezsiniz

—

**Örnek 1** 'e başlıyoruz.

—

**Ön bilgi : Yorum Satırı**

## Yorum Satırı Nedir?

Yorum satırları kodumuz çalıştığında hiç yokmuş gibi davranan, kodumuzun anlaşılabilirliğini arttıran bir özelliktir. Her programlama dilinde bulunur.



## Yorum Satırı kullanım örneği

```
c = 5 # Sayma değişkeni  
x = 2 # problem değişkeni  
bilinmeyen_fonk(c)  
# C nin karesini alan fonksiyon  
x = c**x # C nin karesini aldık.
```

---

**Örnek 2** 'ye başlıyoruz.

# KONU 1

## DEĞİŞKENLER

### Kazanımlar :

- Yazılımların temeli
- Değişkenlerin anlamı
- Değişkenlerin tanımlanması
- Değişkenlerin kullanılması

## Değişkenler nedir?

Değişken değerler atayabileceğiniz ve bu değerleri RAM üzerindeki Hafıza Hücrelerinde tutan, bütün yazılımların temelini oluşturan bir temel birimdir.

x=75;

┌  
└───→

2	75		
2	37	-	1
2	18	-	1
2	9	-	0
2	4	-	1
2	2	-	0
2	1	-	0
2	0	-	1

--- 1001011

-	-	0	1	0	0	1	0	1	1
---	---	---	---	---	---	---	---	---	---

↑

## Değişkenler neden vardır?

```
print(5+5)
```

```
Çıktı = 10
```

Peki 5+5 işleminin sonucunu bir sonraki adımda 2 ye bölmek istersem?

## Değişkenler neden vardır?

Bu sebeple değişkenleri kullanmaya ihtiyaç duyarız.

```
x=5  
y=5  
sonuc = x+y
```

```
print(sonuc)  
sonuc = sonuc/2  
print(sonuc)
```

# Değişken Tipleri



---

```
counter = 13      # Integer
miles   = 1023.0  # Float
name    = "HunrobotX" # String
boolean = True    # Boolean
```

```
(Değişken İsmi) = (Değeri)
```

Değişken tanımlamak için Python da değişkenin tipini belirtmenize **gerek yoktur.**

## Değişken Tipini Nasıl Öğreniriz?

Bunun için `type()` fonksiyonunu kullanırız.

---

**Örnek 3** 'ye başlıyoruz.

# KONU 2

Standard

Input and Output (I/O)

Kazanımlar :

- Bilgisayara ile ilk iletişim
- Bilgisayardan geri bildirim
- Basit Debug  
(HataAyıklama)

---

## `input()` Fonksiyonu

```
name = input()
```

Input fonksiyonu String tipinde değişkenler alır. Aldığımız bu değerleri matematiksel olarak kullanacaksak type-cast yapmamız gerekir.

## input() Fonksiyonu

```
name = input("İsminiz : ")
```

Bu şekilde kullanıldığında kullanıcıdan istenilen bilginin ne olduğunu ekrana bastırıp ardından input'u bekler.

## `print()` Fonksiyonu

```
print(name)
```

şeklinde kullanılır.

`input()` fonksiyonu gibi her hangi bir değişkene eşitlenmez.

## `print()` Fonksiyonu

```
print(name,surname,"suçludur.")
```

şeklinde kullanılırsa

Virgül ile ayrılan cümlelerin (`string`)

Ekranda boşlukla ayrılmış şekilde

yan yana yazılmasını sağlar.



## `print()` Fonksiyonu

`print()` fonksiyonunun bir diğer önemli özelliği ise Debug (Hata Ayıklama) işleminde en büyük yardımcımız olmasıdır.

---

**Örnek 4** 'e başlıyoruz.

# KONU 3

## Aritmetik Operatorler

### Kazanımlar :

- Bilgisayara ile ilk iletişim
- Bilgisayardan geri bildirim
- Basit Debug  
(HataAyıklama)

---

# Toplama

```
x = 7
```

```
y = 2
```

```
sonuc = x + y + 2
```

```
print(sonuc)
```

Ekrana ne yazılacak?

## Çıkarma

```
x = 21
```

```
y = 7
```

```
sonuc = x - y - 5
```

```
print(sonuc)
```

Ekrana ne yazılacak?

## Bölme

```
x = 15  
y = 2  
sonuc = (x / y) / 5  
print(sonuc)
```

Ekrana ne yazılacak?

**Paranteze dikkat edin!**  
**(İşlem önceliği)**

## Çarpma

```
x = 12
```

```
y = 5
```

```
sonuc = (x * y) * 2
```

```
print(sonuc)
```

Ekrana ne yazılacak?

## Mod Alma

`x = 18`

`y = 4`

`sonuc = (x % y) % 5`

`print(sonuc)`

Ekrana ne yazılacak?



## \*Bölüm

```
x = 18
y = 5
sonuc = (x // y) // 2
print(sonuc)
```

Ekrana ne yazılacak?

## Üst Alma

```
x = 9
y = 1/2
sonuc = (x ** y) ** 1/2
print(sonuc)
```

Ekrana ne yazılacak?

Dikkatli olun!

# KONU 4

## Değişken Tipi Değiştirme (Type-Casting)

### Kazanımlar :

- Değişken tipi değiştirme
  - Kullanıcıdan alınan input ile matematiksel işlem
  - Yararları
-

# Typecast Nedir?

Typecasting birbirlerine dönüştürülmesi  
mümkün olan değişkenlerin birbirine  
dönüştürülme işlemidir.

## `int(x)` Fonksiyonu

`x` değişkenini `INTEGER(tam sayı)`' ya çevirir.

Kullanım örnekleri:

```
sayi = "5"  
print(sayi)  
sayi2 = int(sayi)  
print(sayi2)
```

```
sayi = "10010"  
print(sayi)  
sayi2 = int(sayi,2)  
print(sayi2)
```

## `int(x)` Fonksiyonu

En çok kullandığı yer `input()` fonksiyonundan integer bir değer almak istediğimizdedir.

```
age = input("Yaş : ")  
print(age*2)  
age = int(age)  
print(age*2)
```

# `str(x)` Fonksiyonu

`x` değişkenini `STRING(karakter dizisi)`' e çevirmek için kullanılır

Kullanım örnekleri:

```
sayi = 12
print(sayi)
sayi2 = str(sayi)
print(sayi2)
```

```
sayi = 25.78
print(sayi)
sayi2 = str(sayi)
print(sayi2)
```

## `float(x)` Fonksiyonu

String veya integer tipinde değişkenleri FLOAT (ONDALIK) sayıya çevirir.

```
ort = input("Ortalama : ")
      print(ort*2)
      ort = float(ort)
      print(ort*2)
```



---

**Örnek 5** 'e başlıyoruz.

# KONU 5

## STRING (Karakter Dizisi) Metodları ve Özellikleri

### Kazanımlar :

- Metod nedir?
- Metod vs Fonksiyon
- Bazı STRING Methodları

---

# Metod Nedir?

**Metod** bir **Class**'a ait ve sadece onunla kullanılabilen bir **fonksiyondur**.

**String** ve **Integer** Python'da bir **Class**'tır.

# Metod vs Fonksiyon

Belirtilildiği gibi **Metod** sadece tanımlandığı **Class** tipinde kullanılır.

**Fonksiyon** ise tüm **değişkenlerde** ve **Classlarda** kullanılabilir.

## STRING için bazı önemli FONKSİYONLAR

```
len(my_string)
```

**Liste** tipindeki değişkenlerin  
eleman sayısını döndürür.

**\*STRING değişkeni aslında bir listedir.**

—

```
len(my_string)
```

```
my_string = "Python string"  
print(my_string)  
length = len(my_string)  
print(length)
```

**OUT : Python string**

**OUT : 13**

# —

## STRING için bazı önemli METODLAR

- `split()`
- `upper()`
- `lower()`
- `isalnum()`

Daha fazlası için

[https://www.w3schools.com/python/  
python\\_ref\\_string.asp](https://www.w3schools.com/python/python_ref_string.asp)

## `split()`

```
my_string.split("string")
```

`my_string` değişkenini istenilen diziden parçalamak için kullanılır.



## split()

```
my_string = "Python string"  
print(my_string)  
string2 = my_string.split(" ")  
print(string2)
```

**upper()**

```
my_string.upper()
```

`my_string` değişkeninin tüm karakterlerini  
büyük harf olarak değiştirir.

## upper()

```
my_string = "Python string"  
print(my_string)  
string2 = my_string.upper()  
print(string2)
```

---

# lower()

```
my_string.lower()
```

`my_string` değişkeninin tüm karakterlerini  
Küçük harf olarak değiştirir.

## lower()

```
my_string = "PytHon StRiNg"  
print(my_string)  
string2 = my_string.lower()  
print(string2)
```

## `isalnum()`

```
my_string.isalnum()
```

`my_string` değişkeninin tüm karakterlerinin  
Karater(a-z) veya sayı(0-9) olup olmadığını  
kontrol eder.

**True** veya **False** Boolean değişkeni döndürür.

## isalnum()

```
my_string = "Py+tHon123*"
print(my_string.isalnum())
```

```
my_string = "PytHon123"
print(my_string.isalnum())
```

# —

## STRING için bazı önemli ÖZELLİKLER

- + Toplama
- \* Çarpma
- [a:b] ?



—

+ (Toplama)

```
sonuc = string1 + string2
```

Python'da **STRING**'lerin toplanması  
yan yana birleştirilmesini ifade eder.

—

+ (Toplama)

```
name = "Dennis"  
surname = "Ritchie"  
fullname = name + " " + surname  
print(fullname)
```

—

**\* (Çarpma)**

```
sonuc = string1 * 2
```

Python'da **STRING**'in bir sayıyla çarpılması  
0 sayı kadar tekrar etmesini sağlar.

**\* (Çarpma)**

```
msg = "Özür dilerim hocam\n"  
full_msg = msg*200  
print(full_msg)
```

[a:b]

```
sonuc = string[2:5]
```

Python'da **STRING**'lerin a ve b ninci elemanların arasındaki **STRING**'i seçer.

—

[a:b]

```
      0 1 2 3 4 5 6 7 .....  
name = "Linus Torvalds"  
cut_name = name[2:7]  
print(cut_name)
```

# KONU 6

## Karşılaştırma ve Mantıksal Operatörleri

### Kazanımlar :

- Karşılaştırma Op. Nedir?
- Nerelerde Kullanılır
- Türleri Nedir?
- Ne döndürürler?

---

# Karşılaştırma Operatörleri

- `==` ( Eşitlik )
- `!=` ( Eşitsizlik )
- `>` ( Büyüklük )
- `<` ( Küçüklük )
- `>=` ( Büyüklük veya Eşitlik )
- `<=` ( Küçüklük veya Eşitlik )



== ( Eşitlik )

```
sonuc = num1 == num2
```

Python'da iki **değişkenin** birbirlerine **eşitliğini** kontrol edip **Boolean** sonuç döndüren bir fonksiyondur.

== ( Eşitlik )

```
num1 = 12
```

```
num2 = 12
```

```
sonuc = num1 == num2
```

```
print(sonuc)
```

—

**!= ( Eşitsizlik )**

```
sonuc = num1 == num2
```

Python'da iki değişkenin birbirlerine **eşitsizliğini** kontrol edip **Boolean** sonuç döndüren bir fonksiyondur.

—

**!= ( Eşitsizlik )**

```
num1 = 12
```

```
num2 = 12
```

```
sonuc = num1 != num2
```

```
print(sonuc)
```

> ( Büyüklük )

```
sonuc = num1 > num2
```

Python'da iki değişkenin birbirlerine göre **büyüklüğünü** kontrol edip **Boolean** sonuç döndüren bir fonksiyondur.

> ( Büyüklük )

```
num1 = 31
```

```
num2 = 5
```

```
sonuc = num1 > num2
```

```
print(sonuc)
```

< ( Küçüklük )

```
sonuc = num1 < num2
```

Python'da iki değişkenin birbirlerine göre **küçüklüğünü** kontrol edip **Boolean** sonuç döndüren bir fonksiyondur.

**< ( Küçüklük )**

```
num1 = 31
```

```
num2 = 5
```

```
sonuc = num1 < num2
```

```
print(sonuc)
```



**>= ( Büyüklük veya Eşitlik )**

```
sonuc = num1 >= num2
```

Python'da iki değişkenin birbirlerine göre **büyüklüğünü** veya **eşitliğini** kontrol edip **Boolean** sonuç döndüren bir fonksiyondur.

—

**>= ( Büyüklük veya Eşitlik )**

```
num1 = 7
num2 = 7
sonuc = num1 >= num2
print(sonuc)
```

**<= ( Küçüklük veya Eşitlik )**

```
sonuc = num1 <= num2
```

Python'da iki değişkenin birbirlerine göre **büyükliğini** veya **eşitliğini** kontrol edip **Boolean** sonuç döndüren bir fonksiyondur.

**<= ( Küçüklük veya Eşitlik )**

```
num1 = 9
num2 = 39
sonuc = num1 <= num2
print(sonuc)
```

# Mantıksal Operatörleri

- and ( Ve )
- or ( Veya )
- not ( Değil )

—

and ( Ve )

```
sonuc = bool1 and bool2
```

Python'da iki **boolean** değişkenin **True** olup olmadığını kontrol eder.  
**Boolean** sonuç döndürür.

—

and ( Ve )

```
bool1 = True
bool2 = True
sonuc = bool1 and bool2
print(sonuc)
```

—

or ( Veya )

```
sonuc = bool1 and bool2
```

Python'da iki **boolean** değişkenden herhangi birinin **True** olup olmadığını kontrol eder.

**Boolean** sonuç döndürür.



—

or ( Veya )

```
bool1 = True
bool2 = False
sonuc = bool1 or bool2
print(sonuc)
```

—

`not ( Değil )`

`sonuc = not bool1`

Python'da **boolean** değişkenin değerinin tersini almak için kullanılır.

**Boolean** sonuç döndürür.

—

`not ( Değil )`

```
bool1 = True
sonuc = not bool1
print(sonuc)
```

A	B	A AND B	A OR B	NOT A
False	False	False	False	True
False	True	False	True	True
True	False	False	True	False
True	True	True	True	False

---

**Örnek 6** 'e başlıyoruz.