

## PYTHON DERS NOTLARI

### Python Nedir?

- 1) Python **özgür ve ücretsiz** bir programlama dilidir.
- 2) **Guido Van Rossum** adlı Hollandalı bir programcı tarafından 90'lı yılların başında geliştirilmeye başlanmıştır.
- 3) Adı "**The Monty Python**" adlı bir İngiliz komedi grubunun, "**Monty Python's Flying Circus**" adlı gösterisinden esinlenerek konmuştur.
- 4) Guido Van Rossum 2005 ile 2012 yılları arasında **Google'da** çalışmıştır.
- 5) Python kelimesi Türkçe "**paytın**" şeklinde telaffuz edilir.

### Neden Python?

- 1) Dünya çapında büyük kuruluşlar (**Google, YouTube ve Yahoo! gibi**) bünyelerinde her zaman Python programcılarına ihtiyaç duyuyor.
- 2) Python programlama dili ile programları kendiniz yazabilirsiniz.
- 3) Python'ı kullanarak **masaüstü, oyun, mobil, web ve ağ** gibi programlama alanlarında çalışmalar yapabilirsiniz.
- 4) Python kodları **sade, basit ve hızlıdır**. Bu nedenle program yazmak başka dillere kıyasla çok kolaydır. Örneğin Python, C ve C++ gibi dillerin aksine derlenmeye ihtiyaç duymaz.
- 5) Python farklı işletim sistemleri ve platformları üzerinde çalışabilir. **Linux, Windows, Mac OS X, MS-DOS, iOS ve Android** gibi belki adını dahi duymadığınız pek çok ortamda Python uygulamaları geliştirebilirsiniz.
- 6) Herhangi bir ortamda yazdığınız bir Python programı, üzerinde hiçbir değişiklik yapılmadan veya ufak değişikliklerle başka ortamlarda çalışabilir.

### Python Sürümleri:

- 1) Piyasada iki çeşit Python sürümü vardır. Eğer bir Python sürümü 2 ile başlıyorsa (2.7.11 gibi) **Python 2.x**, şayet 3 ile başlıyorsa (3.6.2 gibi) Python 3.x serisine aittir.
- 2) Python3, Python2'ye göre çok daha güçlüdür ve de hatalardan arındırılmıştır.
- 3) Python2 ile yazılmış bir program Python3'te çalışmaz. Aynı şekilde Python3 ile yazdığınız bir program Python2'de çalışmaz.

### Python'u Kurmak:

- 1) Python'ı kurmak için <http://www.python.org/downloads> adresine tıklayın.
- 2) **Downloads** linkini tıkladığınızda '**Download Python 3.x.x**' ve '**Download 2.x.x**' yazan yan yana iki düğme göreceksiniz.
- 3) İlk düğme **Python3**, ikinci düğme ise **Python2** sürümünü içerir.
- 4) Python'ı hangi dizine kurduğumuzu bilmek önemlidir. Zira karşılaşacağımız bazı sorunlar bu dizine gitmemizi gerektirebilir.

### Python Dosyaları Nerede?

Windows kullanıcısı iseniz ve eğer siz farklı bir yere kaydetmemişseniz Python dosyaları genellikle **C:\Users\Kullanıcılar\AppData\Local\Programs\Python** dizini içindedir. Klasörler gözükmüyorsa gizli dosyaları göster seçeneğini aktifleştiriniz.

### Python Geliştirme Ortamları ve Editörler:

- 1) Yazdığımız kodları bir yere kaydedip saklamamız gerekir. İşte burada metin düzenleyiciler yani **editörler** devreye girer.
- 2) Bu kodlarının yazıldığı ortamlara geliştirme ortamları veya teknik olarak **etkileşimli kabuk** adı verilir. Herhangi bir editör kullanabilirsiniz. Ancak kodları ayırt edip renklendirebilen bir editör her zaman avantajlıdır.
- 3) Python kodlarınızı yazmak için Microsoft Word gibi, belgeleri ikili (binary) düzende kaydeden programlar uygun değildir. Kullanacağınız metin düzenleyici, belgelerinizi **düz metin (plain text)** biçiminde kaydedebilmelidir.

### Bazı Python Geliştirme Ortamları:

1) **IDLE**: Python.org web sitesinde yer alan ücretsiz program geliştirme ortamıdır.

2) **Wing IDE**: Wing IDE Professional, Wing IDE Personal ve Wing IDE 101. Bunlardan Wing IDE 101 ücretsizdir. <https://wingware.com/downloads/wingide-101> adresinden indirilebilir.

3) **Canopy**: <https://store.enthought.com/downloads/#default> ücretsiz indirilebilir.

**Not**: Python programlama dilini öğrenmeye yeni başlayan Windows kullanıcıları için en uygun metin düzenleyici. **IDLE** aslında Python'ın etkileşimli kabuğudur. Dolayısıyla asıl kodları buraya yazmayacağız. Python programlama diline yeni başlayanların en sık yaptığı hatalardan biri de, kaydetmek istedikleri kodları bu ekrana yazmaya çalışmalarıdır. Unutmayın, bu ekrana yazdığınız kodlar kapattığınızda kaybolur. IDLE'ı açtığınızda sol üst köşede **File [Dosya]** menüsüne tıklayın ve menü içindeki **New Window [Yeni Pencere]** düğmesine basın. Beyaz bir ekranla karşılaşacaksınız. İşte Python kodlarımızı bu ekrana yazacağız. Dikkat ettiyseniz artık >>> sembolü yok.

**Not**: >>> işareti Python'da komut yazmaya hazır olduğu anlamına gelir.

**Not**: Python'a yeni başlayanların en sık yaptığı hatalardan biri >>> işareti ile komut arasında boşluk bırakmalarıdır. Eğer bırakırsanız kod hata verecektir. 3

**!!!NOT**: # işareti Python da açıklama (yorum) satırı olarak kullanılır. Program çalışırken # işaretinin yanında yazılan ifadeler programda gözükmez, sadece kullanıcıya bilgi vermek için kullanılır.!!!

## PRINT FONKSİYONU

**Print Fonksiyonu**: Ekranda görüntülenmesini istediğimiz tüm bilgi ya da sonuçları ekrana yazdırabilmek için kullanılır. Parantez içinde yazılan bilgilerin her birine parametre deriz. Birden fazla bilgi (parametre) yazacaksak aralarına virgül koymamız gerekir.

Örneğin:

```
print("Merhaba Dünya") # ekrana Merhaba Dünya yazısı yazdırıldı
print("Toplamı :", 10+23, "Çarpımı :", 10*23)
```

Ekranda metin yazdırmak istendiğinde print komutu kullanılır. Kullanımı aşağıda verilmiştir. Parantez içinde tek tırnak, çift tırnak veya üç tırnak kullanılabilir. Ayrıca Türkçe'ye özel harflerin (ı,ş,ç,ö,ğ,ü,ö) parantez içinde kullanılmasında bir sakınca yoktur. birden fazla bilgi yazdırılmak isteniyorsa her bir parametre arasına virgül (,) işareti konularak kullanılır. Değişken yazdırılacaksa, değişken ismi **kesinlikle** tırnak içine **alınmaz**. Kullanımı:

```
print("yazdırılacak bilgi")
```

Örnekler:

```
print('Merhaba Dünya') # tek tırnak Merhaba Dünya
print("Merhaba Dünya") # çift tırnak Merhaba Dünya
print("""Merhaba Dünya""") # üç tırnak Merhaba Dünya
```

Python'da karakter dizisi tanımlarken genellikle tek veya çift tırnak kullanıldığını unutmayın. O halde neden bazen üç tırnak kullanmamız gerektiğini bir örnek üzerinde göstereyim.

```
>>> ""Game Over Insert Coin!"" Game Over Insert Coin!
```

print komutunu ve üç tırnak işaretini kullanarak Game Over yazdım sonra enter tuşuna bastım, burada tek tırnak veya çift tırnak kullansaydım enter tuşuna bastıktan sonra program hata verecekti. Ancak üç tırnak kullandığım için hata vermedi. Bunun anlamı **yazmaya devam et** demektir. Yani **print fonksiyonu içinde alt satıra da bilgi yazdırmak istediğimiz zaman üç tırnak kullanılır**. Böylelikle çıktının devamı olan **İnsert Coin!** ifadesini yazdım ve üç tırnak ardından parantez ile fonksiyonumu tamamladım. Tekrar enter tuşuna bastığımda üst üste satırlardan oluşan çıktıyı elde ettim. Bu arada üç tırnak "" ya da "" "" şeklinde olabilir. İkisi de aynı işlevi görür. Peki, neden bazen tek tırnak bazen de çift tırnak kullanmamız gerekiyor? Bunu bir örnekle göstereyim. Örnek olarak aşağıdaki gibi bir çıktı elde etmek istediğimizi varsayalım.

### İstanbul'un 5 günlük hava durumu tahmini

Dikkat ederseniz yukarıdaki ifadede bir kesme işareti var. Yani tek tırnak. O halde kritik soru şu. İçinde tek tırnak işareti olan bir string'i tek tırnak içine almak mantıklı mı? Tabii ki değil çünkü programımız hata verecektir. Bu yüzden diğer bir alternatif olan çift tırnağı kullanmalıyız.

`print("İstanbul'un 5 günlük hava durumu tahmini")` **İstanbul'un 5 günlük hava durumu tahmini**

`print('Firat', 'Dicle')` Çıktı: Firat Dicle # , iki ayrı stringi birleştirirken araya bir boşluk koyar!

`print ("bilgi"+"sayar")` Çıktı: bilgisayar # + iki ayrı stringi boşluk koymadan birleştirir!

`print(3* 'w')` Çıktı: www # \* stringi sayı adeti kadar yan yana yazdırır!

`print(2*'15')` Çıktı: 1515 #\* stringi sayı adeti kadar yan yana yazdırır!

`print(2*15)` Çıktı: 30# \* integer ifadelerde çarpma işlemi yapar!

**len()** komutu ile parametrelerin uzunluğunu öğrenebiliriz. Bir örnekle açıklayalım.

`print(len("Bilgisayar Bilimi"))` çıktı: **17**

**#"Bilgisayar Bilimi" stringinin uzunluğu 17 karakterdir. Burada boşluk karakterinin de sayıldığına dikkat edelim.**

`print(len("Bilgisayar Bilimi")+ len("Dersi"))` Çıktı: **22**

**NOT: len()** komutu çıktı olarak daima sayısal değer verir. Yani len() komutu ile bir **integer** veri türü elde etmiş oluruz.

**sep()** sep parametresi print() fonksiyonuna verilen parametreler birleştirilirken araya hangi karakterin gireceğini belirler.

`print('Bilgisayar', 'bilimi', 'dersi', sep='**')` çıktı: Bilgisayar\*\*bilimi\*\*dersi

**End()** end parametresi ise bu parametrelerin sonuna neyin geleceğini belirler.

`print("Python öğreniyorum", end="!")` çıktı: Python öğreniyorum!

`print("T","B","M","M",sep=".")` → T.B.M.M

`print("T","B","M","M",sep=".",end=".")` → T.B.M.M.

`print("1","2","3","4","14",sep=" mumdur")` → 1 mumdur 2 mumdur 3 mumdur 4 mumdur 14

`print("Merhaba", "Python",sep="+++")` → Merhaba+++Python

**BİLGİ : sep="\n" alt satıra yazdırır.**

`print("Merhaba", "Python",sep="\n")` → Merhaba

Python

`print("kodlama öğreniyorum \n" *2)`

Çıktı: kodlama öğreniyorum  
kodlama öğreniyorum

**BİLGİ : sep="\t" tab kadar boşluk bırakır. (8 karakter boşluk koyarak yazar)**

`print("Merhaba", "Python",sep="\t")` → MerhabaPython

**Özel karakterler (Escape Characters):**

`\n` : new line (alt satır)(imlecin alt satıra geçmesini sağlar)

`\t` : tab

`\'` : tek tırnak yazdırır

`\"` : çift tırnak yazdırır

`\\` : \ yazdırır

`print("Bu\ttab\tboşluğu")` → Bu tab boşluğu

`print("\" Merhaba \")` → "Merhaba"

\* print komutu için de yazılan karakteri harflere böler.

```
print(*"Linux", sep=".") ÇIKTI: L.i.n.u.x
```

### Python' da Dört İşlem

Dört işlem yaparken +, -, \*, / sembolleri kullanılır. Kullanım örneği aşağıda verilmiştir.

```
x=20+10 y=20-10 z=20*10 w=20/10 print (x,y,z,w) #Kodu çalıştırdığımızda ekranda 30,10,200,2 sonucu çıkacaktır.
```

### Python'da Üs Alma, Bölümden Kalanı Bulma (mod), Tamsayı Bölümü

```
a=3**20 # 3'ün 20.kuvveti bu şekilde alınır.
```

```
b=81%4 # 81'in 4'e bölümünden kalan bu şekilde alınır (mod)
```

```
c=26//4 # 26'nın içinde 4 kaç kere vardır (tam sayı bölümü) sonuç=6 dir.
```

+	Toplama	5 + 3
-	Çıkarma	5 - 3
*	Çarpma	5 * 3
/	Bölme	5 / 3
**	Üs alma	5 ** 3 = 125
//	Bölümün tamsayı kısmını verir.	5//2=2
%	Mod alma kalanı verir	15 % 2=1

Not: Matematikte olduğu gibi programlamada işlem önceliği geçerlidir!!

## INPUT FONKSİYONU

**Input Fonksiyonu: klavyeden bilgi girişi için kullanılan fonksiyondur.** Klavyeden girilen bilgi input fonksiyonu ile istenilen değişkene string olarak atanır.

Örnek: Girilen isme merhaba diyen kod.

```
isim=input("Adınızı giriniz:")  
print("Merhaba", isim)
```

Ekran Çıktısı:

```
Adınızı giriniz: Mehmet  
Merhaba Mehmet
```

Eğer girilen bilgi üzerinde matematiksel işlemler yapılmak isteniyorsa; veri türünün tamsayı (integer) ya da ondalıklı (float) sayıya çevrilerek işlem yapılması gerekir. Klavyeden girilen girilen iki tamsayıyı (int) toplayıp sonucu ekrana yazdıran kod:

```
sayi1=int(input("1. Sayıyı giriniz:"))  
sayi2=int(input("2. Sayıyı giriniz:"))  
toplam=sayi1+sayi2  
print("Sonuç:",toplam)
```

Girilen iki ondalıklı (float) sayıyı toplayıp sonucu ekrana yazdıran kod: (girilen bilgi üzerinde matematiksel işlem yapılmadan önce tamsayı ya da ondalıklı sayıya çevrilmelidir.)

```
sayi1=input("1. Sayıyı giriniz:")  
sayi2=input("2. Sayıyı giriniz:")  
toplam=float(sayi1)+float(sayi2) #ondalıklı sayılar toplanıyor  
print("Sonuç:",toplam) # toplam değişkeni yazdırılıyor.
```

## TYPE FONKSİYONU: Verinin türünü belirler.

String: karakterler (harf, özel karakterler, rakamlar) string ifadeler tırnak içinde yazılır.

Integer: negatif ve pozitif tam sayılar

Float : ondalıklı sayılar

```
print(type("Ali")) →string
```

```
print(type(5)) →integer
```

```
print(type(1.7)) →float
```

```
print(type("5")) →string
```

```
print(type("abc06")) →string
```

```
print(type(abc)) → hata
```

## Değişkenler ve Değerinin Yazdırılması

Değişkenler programlama aşamasında hafızada bilgi tutmaya yarayan araçlardır. Değişkenlere bir isim verilerek gerektiğinde o isimlerle çağırılır. Değişkenlere atama yaparken = işareti kullanılır. Kullanım şekli aşağıdaki gibidir.

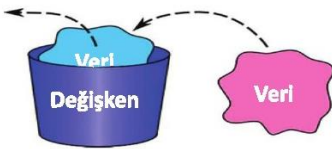
```
a=10 # (bu örnekte a isimli değişkene 10 değeri verilmiştir.)
```

```
print (a) # (değişkenin değerini yazdırmak istediğimizde tırnak kullanılmaz. Bu iki satır kod çalıştığında ekranda 10 görüntülenir.
```

```
Print("a") # çıktı a olarak gözükür. Çünkü a burada değişken olmaktan çıktı.
```

Değişkenlere isim verirken dikkat etmemiz gereken kurallar:

1. Değişken isimleri rakam ile başlamaz
2. Değişken isimleri iki kelimedenden oluşuyorsa arada boşluk kullanılmaz. (\_ kullanılır)
3. Değişken isimleri Türkçe karakterden (ç,ş,ı,ü,ö,ğ) oluşmaz.
4. Değişken isimleri verilirken programlamada kullanılan komutlar kullanılmaz. (for, if, elif, len, print, input...)
5. Değişken isimleri verilirken özel karakterler kullanılmaz. (\*,/, -,+, #,%,&,(,=,?,\$,[, { gibi...).
6. Değişkene içerdiği değer ile tutarlı isimler vermeliyiz.
7. Değişken isimlendirmede genellikle küçük harfle başlanır ve ikinci bir kelime yazılacaksa ilk kelimenin hemen ardından büyük harfle devam edilebilir.. Örnek: tcKimlikNo



### Değişken isimlerine örnekler:

sayi1, okulNo, soru\_2, tc\_kimlik, taksitSay, adSoyad

NOT: Değişkenlere atama yaparken = işaretini kullanınız. sayi1=15, ad='aslı' gibi

## Değişken Türleri

Değişkenler tuttukları verinin içeriğine göre türlere ayrılırlar. Aşağıda örnekleri verilmiştir.

```
a=10 # (a değişkeni tam sayı bir değer tuttuğu için integer türünde bir değişkendir.)
```

```
b=3.14 # (b değişkeni ondalıklı sayı bir değer tuttuğu için float türünde bir değişkendir.)
```

```
c="KAL-1977" # (c değişkeninin değeri tırnak içine yazıldığı için string türünde bir değişkendir.)
```

```
d="Ali" # (d değişkeninin değeri tırnak içine yazıldığı için ve karakterden oluştuğu için string türünde bir değişkendir.)
```

```
a=10
b=20

print(a+b) çıktı:30

x,y,z=100,-45,15

print(x,y,z) çıktı: 100,-45,15
```

```
x=10
x=20
x=30

print(x) çıktı:30
```

değişkenlerde en son atanan değer hafızada kaldığı için çıktı 30 olarak gözükür.

**Ör:**  $x = x + 5$  deyimini  $x += 5$  olarak kısaltılabilir. Bu ifade "x'i 5 arttır." anlamına gelir.

```
x=5
x+=5
print(x) çıktı: 10
```

```
a=20
a-=3 (a=a-3 ile aynı anlama gelir)
print(a) çıktı:17
```

int : Girilen karakter dizisini tamsayıya çevirir.

str : Girilen tam sayı veya ondalıklı sayıyı karaktere çevirir.

float : Girilen karakter dizisini ondalıklı sayıya çevirir.  
int("65") çıktı: 65

```
str(25) çıktı:"25"
```

```
float("65") çıktı: 65.0
```

örnek:

```
sayi1=7.897
```

```
int(sayi1) çıktı: 7
```

örnek

```
sayi2=9
```

```
float(sayi2) çıktı:9.0
```

örnek 1

```
ad = input("Adınızı öğrenebilir miyim") #klavyeden girilen isim ad değişkenine atandı ve ekrana yazdırıldı.
print("Adınız= ",ad)
```

```
yas=input("Yaşınızı öğrenebilir miyim")
```

```
print("Yaşınız= ",yas)
```

örnek 2

kenar=int(input("Karenin bir kenarının uzunluğunu giriniz")) # karenin kenar uzunluğu input fonksiyonu ile klavyeden istendi, girilen değer int parametresiyle tam sayıya çevildi ve kenar değişkenine atandı.

```
cevre=4*kenar # bilgisayara karenin çevresi hesaplatıldı
```

```
alan=kenar*kenar # bilgisayara karenin alanı hesaplatıldı
```

```
print("Karenin çevresi= ",cevre, "karenin alanı=",alan ) # ekrana karenin çevresi ve alanı yazıldı.
```

### UYGULAMA SORULARI

1.Ekrana adınızı yazdırınız.

2.a=10,b=5 değişkenlerini kullanarak 4 işlem yapınız, sonuçlarını ekrana yazdırınız.

3.a=5 a değişkeninin 50.kuvvetini alıp ekrana yazdırınız.

4.a=100 b=3 a'nın b'ye bölümünden kalanı ekrana yazdırınız.

5. a = 3

a = 5

```
print(a)=?
```

6. print(5 \* 'A') çıktı=?

```
7. sayi1 = '9'
sayi2 = '9'
topla = sayi1 + sayi2
print(topla)
```

8. sayi1 = 25 sayi2 = 12 print(sayi1+sayi2)=?

9. print('A','B','C',\_\_\_\_\_) ÇIKTI:A+B+C şeklinde olması için uygun python kodunu yazınız.

```
10. urun sayisi = 18
print("Ürün sayısı=", urun sayisi) çıktı=?
```

11. a, b = 3, 6

```
print("a değeri=", a)
print('b değeri=', b)
```

12. print('Python','Java','C#', sep = '-') çıktı:

13. print('Python','Java','C#', sep = '-', end='...') çıktı:

14. print("Python "+"Programlama "+"Dili") çıktı:

15. print("Python", "Programlama", "Dili") çıktı:

16. kelime = "Merhaba"

```
print(*kelime, sep="-") çıktı:
```

17. klavyeden girilen isim değişkeni için Merhaba isim yazan Python kodunu yazınız. Örnek: Merhaba Ayşe

18. klavyeden iki sayı girişi yaptırıp bu iki sayının ortalamasını hesaplayan Python kodunu yazınız.

19. x=5

y=3

Çıktının 8.0 olması için doğru python kodunu yazınız.

20. a=5

B=3

Çıktının 53 olması için doğru python kodunu yazınız.

21. kk=---(input("kısa kenar uzunluğunu giriniz"))

---=( input("uzun kenar uzunluğunu giriniz"))

alan=kk\*uk

-----("alan="-----

Yukardaki python kodunun doğru çalışabilmesi için boşluklara gelecek uygun komutları yazınız.

22. klavyeden girilen okul isminde bir değişken tanımlayarak uygun python kodunu yazınız.

Hazırlayan:  
Ayten TAŞKIN  
Bilişim Tek. Öğrt.