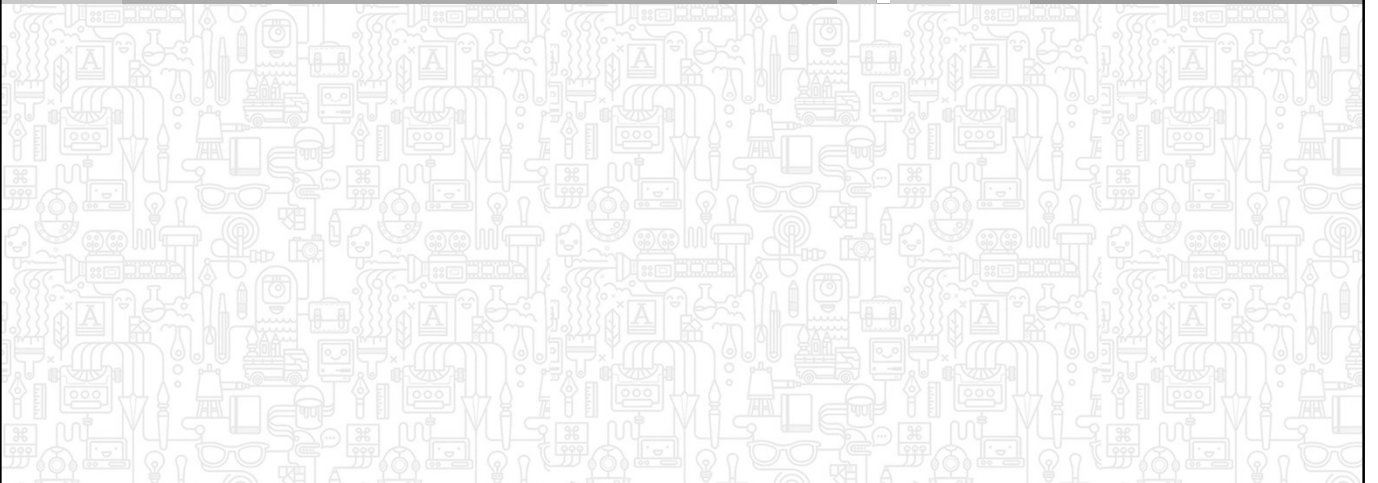


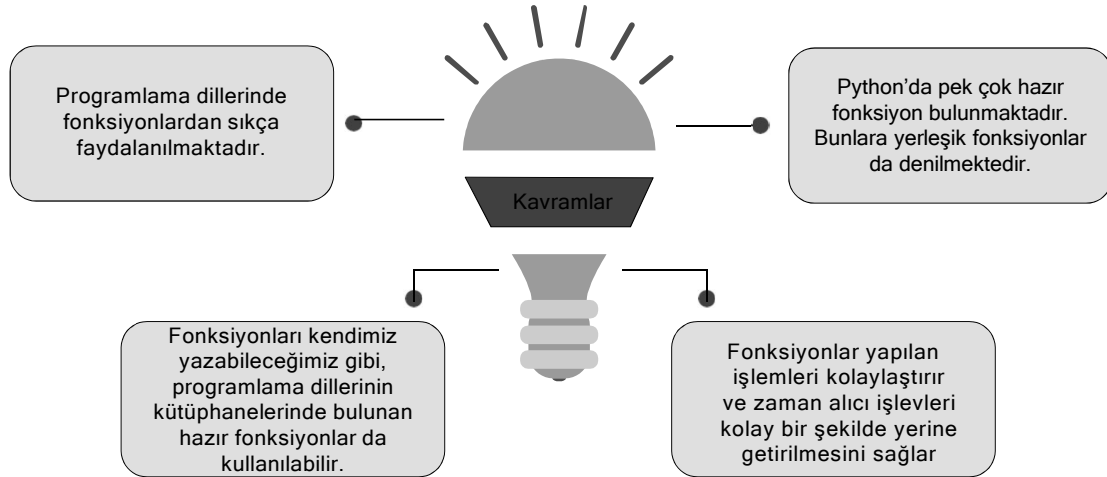
PYTHON



Python'da Temel Fonksiyonlar



Temel Fonksiyonlar



3

print() Fonksiyonu

print() fonksiyonu, konsola çıktı göndermek amacıyla kullanılır. Programların genellikle yapılan işlemler sonucunu kullanıcıya sunması gerekir. Programda veri print() fonksiyonu ile Python'daki konsolda görüntülenebilir. print() ile konsola gönderilecek değer bir metinsel ifade ise bu ifade çift tırnak, tek tırnak ya da üç tırnak içinde yazılmalıdır.

Örnek 1

Kod

```
print('yazdır')
print('yazdır')
print("yazdır")
print("""yazdır""")#Her üç kullanım da bize aynı çıktıyı verir.
```

Eğer konsola gönderilecek değer bir değişken ya da sayısal bir ifade ise tırnak içerisinde yazılmaz.

4

2

print() Fonksiyonu

Fonksiyonlar kullanılırken parantez içerisinde kullanılan değerlere argüman denilmektedir. Python, print() fonksiyonu argümanını kontrol ederek, belirtilen kurallara uyup uymadığını kontrol eder. Söz diziminin doğruluğu veya değişkenin tanımlanmış olması kontrol edilir. Kod, Python'un izin verdiği tanımlamalara uyuyorsa çalıştırılarak konsol üzerinden sonuç görüntülenir. Aksi durumda hata mesajı alınır.

Örnek 2

Kod

```
print(5)
```

Örnek 3

Kod

```
print("Merhaba, Python!")
```

5

print() Fonksiyonu

Aşağıdaki durumlarda eksik olan tanımlamalar neticesinde, hata mesajı ile karşılaşmıştır. Örnek 4'te tırnak kapatılmamış, örnek 5'te ise değişken tanımlanmadığı için hata vermiştir.

Örnek 4

Kod

```
print("python)
File "<ipython-input-11-89608cf8e227>", line 1
  print("python)
      ^
SyntaxError: EOL while scanning string literal
```

Örnek 5

Kod

```
print(a)
Traceback (most recent call last):
  File "<ipython-input-12-bca0e2660b9f>", line 1, in <module>
    print(a)
NameError: name 'a' is not defined
```

6

3

print() Fonksiyonu

Aşağıdaki karakter dizisi tek tırnak işareti ile başlamış, ancak kesme işareti olarak kullanıldığı yerde sonlanmış ve hata vermiştir. Burada çift tırnak ve tek tırnak beraber kullanılarak sorunun çözümüne gidilebilir.

Örnek 6

Kod

```
print('Türkiye'nin en kalabalık ili İstanbul'dur')  
SyntaxError: invalid syntax
```

Örnek 7

Kod

```
print("Türkiye'nin en kalabalık ili İstanbul'dur")
```

7

print() Fonksiyonu

print() fonksiyonu kullanılırken argüman değerleri arasında aritmetiksel işlemler yapılabilir.

Örnek 8

Kod

```
print(2+2)  
print(2-2)  
print(2*2)  
print(2/2)
```

8

print() Fonksiyonu ile Kullanılabilen Parametreler

Örnek 9'da ilk satırda çift tırnak işareti arasındaki "Merhaba," kelimesi karakter dizisi olarak algılandı ama sonrasında Python kelimesi çift tırnak içinde ve argümanlar arasında "," virgül olmadığı için hata verdi. İkinci satır için kaçış parametreleri (escape character) kullanılabilir.

Örnek 9

Kod

```
print("Merhaba," Python "kullanıyorum")
SyntaxError: invalid syntax
print('Bursa'nın iskenderi meşhurdur.')
SyntaxError: invalid syntax
```

9

Ters Taksim(\)

Aslında bu sorun, bir önceki bölümde çift tırnak ve tek tırnakla beraber kullanılarak çözülmüştü. Ancak burada ters taksim (\) işareti kullanılarak da bu sorun çözülebilir.

Örnek 10

Kod

```
print('Bursa\'nın iskenderi meşhurdur.')
```

10

Alt Satır Başı (\n)

`print()` fonksiyonu kullanılırken, karakter dizilerinde bazen alt satıra inme ihtiyacı duyulabilir.
Python'da en sık kullanılan kaçış parametresi `\n` parametresidir.

Örnek 11

Kod

```
print("1. satır\n2. satır\n3. satır")
```

11

Sekme(\t)

Klavyeden tab tuşuna basıldığında gibi belirli karakter boşluk bırakılmasını sağlayan bir parametredir.

Örnek 12

Kod

```
print("pazartesi \tsalı \tçarşamba")
```

12

Diğer Kaçış Dizileri

Kaçış Dizisi	Anlamı
\'	Karakter dizisi içinde tek tırnak işaretini kullanabilmemizi sağlar.
\"	Karakter dizisi içinde çift tırnak işaretini kullanabilmemizi sağlar.
\\	Karakter dizisi içinde \ işaretini kullanabilmemizi sağlar.
\n	Yeni bir satıra geçmemizi sağlar.
\t	Karakterler arasında sekme boşluğu bırakmamızı sağlar.
\u	UNICODE kod konumlarını gösterebilmemizi sağlar.
\U	UNICODE kod konumlarını gösterebilmemizi sağlar.
\W	Karakterleri UNICODE adlarına göre kullanabilmemizi sağlar.
\x	Onaltılı sistemdeki bir sayının karakter karşılığını gösterebilmemizi sağlar.
\a	Destekleyen sistemlerde, kasa hoparlöründen bir 'bip' sesi verilmesini sağlar.
\r	Aynı satırın başına dönülmesini sağlar.
\v	Destekleyen sistemlerde düşey sekme oluşturulmasını sağlar.
\b	İmlecin sola doğru kaydırılmasını sağlar
\f	Yeni bir sayfaya geçilmesini sağlar.
r	Karakter dizisi içinde kaçış dizilerini kullanabilmemizi sağlar.

13

end Parametresi

Bu parametre print() fonksiyonu ile ekrana gönderilen değerlerin sonunda hangi işlemin yapılacağını belirtmektedir.

Örnek 13

Kod

```
print("Merhaba!")
print("Python")
```

Örnek 13'teki gibi bir kullanımda kodların çıktısı alt alta çıkar. Ancak bazı durumlarda programın çıktısı birleştirilmek istenebilir. İlerleyen bölümlerde döngü konusunda, döngü değerini her seferinde konsola yazdırılmak istendiğinde bu değerler alt alta yazılacaktır. Döngünün büyük olduğu düşünülürse program sayfalar dolusu çıktı verebilir. İşte bu gibi durumlarda end parametresi çok işe yaramaktadır.

14

end Parametresi

Örnek 14'te görüldüğü üzere end parametresi içinde iki tırnak arasında boşluk karakteri kullanıldığı için program çıktısını birleştirerek araya boşluk eklendi. Aynı işlem virgül kullanılarak yapılabilir:

Örnek 14

Kod

```
print("Merhaba!", end=" ")
print("Python")
```

Örnek 15

Kod

```
print("Merhaba!", end=", ")
print("Python")
```

15

sep Parametresi

bir print() fonksiyonu birden fazla argüman alabilir. Her bir argümanın arasında farklı işlemler sep parametresi ile yapılabilir.

Örnek 16

Kod

```
print("pazartesi", "salı", "çarşamba", "perşembe", "cuma")
```

Örnek 17

Kod

```
print("pazartesi", "salı", "çarşamba", "perşembe", "cuma", sep="-")
```

Her bir argüman'ın arasına sep parametresi yardımıyla birer "-" koyulur.

16

Format() Metodu ile Biçimlendirme İşlemleri

Program yazarken bazı durumlarda bir string'in içinde daha önceden tanımlı string, float, int gibi farklı türden değerleri yerleştirmek isteyebiliriz. Böyle durumlar için Python'da format() metodu bulunmaktadır.

Örnek 18

Kod

```
a=5
b=6
c=9
print("girdiğiniz",a, b, "ve",c,"değerlerinin toplamı: ",a+b+c,"dir")
```

Örnekteki gibi bir kullanım ve hata yapmaya müsait bir kullanımdır. Python bu gibi durumlar için print() fonksiyonunda format metodunun kullanımına olanak sağlar.

17

Format() Metodu ile Biçimlendirme İşlemleri

Aşağıdaki örnekte print() fonksiyonunda kullanılan her bir { } ifadesine karşılık olarak format() metoduna bir adet argüman verilmelidir.

Örnek 19

Kod

```
print("çıkktı işlemi {} {} {}".format(1,2,3))
```

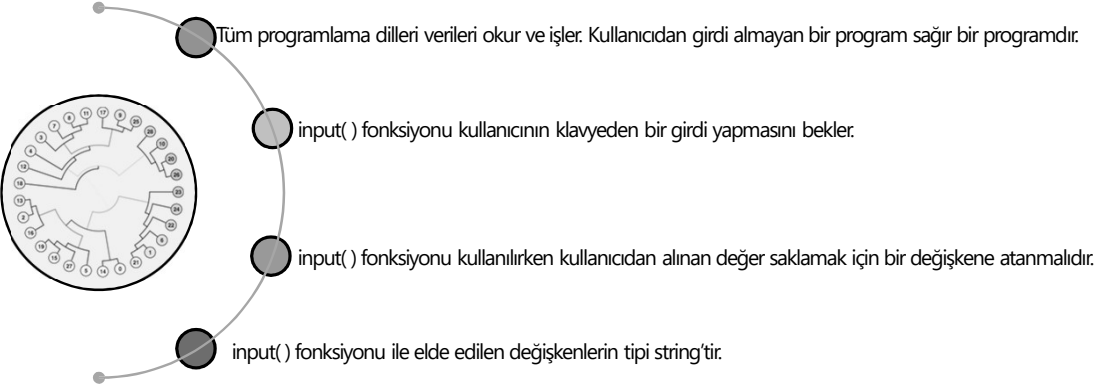
Örnek 20

Kod

```
a=5
b=6
c=9
print("girdiğiniz {}, {} ve {} değerlerinin toplamı = {} dir".format(a,b,c,a+b+c))
```

18

input() Fonksiyonu



21

input() Fonksiyonu

Örnek 22'de input() fonksiyonu ile kullanıcıdan bir karakter dizisi girmesi beklenmiş ve sonuç ekrana yazdırılmıştır.

Örnek 22

Kod

```
isim=input("isminizi giriniz: ")
print("merhaba! ",isim)
```

22

input() Fonksiyonu

Örnek 23'te görüldüğü üzere uygulama bize hatalı bir çıktı vermiştir. Bu kullanımda her iki değer bir karakter dizisi olarak algılanmış ve Python iki değer üzerinde toplama işlemi yapamadığı için yan yana yazarak birleştirmiştir. Örnek 24'te ise kullanım doğrudur.

Örnek 23

Kod

```
a=input("birinci sayıyı giriniz: ")
b=input("ikinci sayıyı giriniz: ")
print("girdiğiniz sayıların toplamı: ",a+b)
```

Örnek 24

Kod

```
a=input("adınızı giriniz: ")
b=input("soyadınızı giriniz: ")
print("Adı Soyadı: ",a+ " " +b)
```

23

input() Fonksiyonu

input() fonksiyonu kullanılırken girdi olarak sayısal ifadeler kullanılacağı zaman bu durumun Python'a bildirilmesi gerekmektedir.

Örnek 25'te görüldüğü üzere input() fonksiyonu, int() fonksiyonunun içerisine alınarak girdi sayısal ifadeye çevrilmiştir. Şimdi kullanım doğrudur.

Örnek 25

Kod

```
a=int(input("birinci sayıyı giriniz: "))
b=int(input("ikinci sayıyı giriniz: "))
print("girdiğiniz sayıların toplamı: ",a+b)
```

Görüldüğü üzere burada yapılan işlemle girdiler sayılara dönüştürülmüş ve toplama işlemi yapılmıştır. input() fonksiyonu kullanılırken sık yapılan hataların başında hatalı veri girişleri gelmektedir.

Yukarıdaki koddaki kullanıcı girişleri eğer string yaparsa yani "Ali" gibi bir değer girerse program hata verir.

24