

## Built in Function (Dahili İşlevler)

Tek bir ilgili eylemi gerçekleştirmek için kullanılan bir dizi işleve sahip komutlara denilir.

Built-in Functions				
<b>abs()</b>	<b>delattr()</b>	<b>hash()</b>	<b>memoryview()</b>	<b>set()</b>
<b>all()</b>	<b>dict()</b>	<b>help()</b>	<b>min()</b>	<b>setattr()</b>
<b>any()</b>	<b>dir()</b>	<b>hex()</b>	<b>next()</b>	<b>slice()</b>
<b>ascii()</b>	<b>divmod()</b>	<b>id()</b>	<b>object()</b>	<b>sorted()</b>
<b>bin()</b>	<b>enumerate()</b>	<b>input()</b>	<b>oct()</b>	<b>staticmethod()</b>
<b>bool()</b>	<b>eval()</b>	<b>int()</b>	<b>open()</b>	<b>str()</b>
<b>breakpoint()</b>	<b>exec()</b>	<b>isinstance()</b>	<b>ord()</b>	<b>sum()</b>
<b>bytearray()</b>	<b>filter()</b>	<b>issubclass()</b>	<b>pow()</b>	<b>super()</b>
<b>bytes()</b>	<b>float()</b>	<b>iter()</b>	<b>print()</b>	<b>tuple()</b>
<b>callable()</b>	<b>format()</b>	<b>len()</b>	<b>property()</b>	<b>type()</b>
<b>chr()</b>	<b>frozenset()</b>	<b>list()</b>	<b>range()</b>	<b>vars()</b>
<b>classmethod()</b>	<b>getattr()</b>	<b>locals()</b>	<b>repr()</b>	<b>zip()</b>
<b>compile()</b>	<b>globals()</b>	<b>map()</b>	<b>reversed()</b>	<b>__import__()</b>
<b>complex()</b>	<b>hasattr()</b>	<b>max()</b>	<b>round()</b>	

Tablodaki komutlardan bazılarını ele alalım.

- **abs(X)**: x'in mutlak değerini alır .

ÖRNEK:

```
: x=-47
: abs(x)
```

ÇIKTI:

47

- **max(),min(),sum()**

ÖRNEK:

```
ARA_SINAV_NOTU=( [ 25,56,98,74,78,25,15,65,45,95,55,45,56, ] )
```

En düşük değer çıktısı:

```
min(ARA_SINAV_NOTU)
```

15

En yüksek değer çıktısı:

```
max(ARA_SINAV_NOTU)
```

98

Toplam değer çıktısı:

```
sum(ARA_SINAV_NOTU)
```

732

## DEĞİŞKENLER

Programa değişken atarken dikkat etmemiz gereken bazı noktalar var. Öncelikle bir değişken bir harf veya kelimeden değişken isimlendirilebilir. Başında sayı, özel karakter, kısa çizgi kullanılmasına program izin vermez. Değişken adı bir sayı ile başlayamaz. Değişkenlerde büyük küçük harf uyumu olmalıdır. (ad-Ad aynı değişken değildir.)

Değişken tanımlama nasıl örneğimizde inceleyelim.

### ÖRNEK:

```
: ad
  Soyad
  yaş
  ülke
  Kent
  ikinci ad
  uyruk
  Başkent
  _gelir # eğer kelimeyi değişken olarak kullanmak istiyorsak
  yıl_2021
  yıl2021
  current_year_2021
  doğum yılı
  kişi1
  sayı2
```

Değişkenler arasında boşluk bırakılmaz ve sayılardan değişken olmaz örneğin aşağıdaki gibi değişken tanımları geçersizdir.

```
gelir dağılımı
örnek1-1
23
|
```

Eğer değişkenimde birden fazla değişkenim varsa Python genel değişken kullanımı olarak kelimeler arası alt çizgi karakteri kullanılır.

### ÖRNEK:

```
büşra_akkaya_gelir|
```

```
ilk_ad='büşra'
soyad='akkaya'
evli_bekar='bekar'
ülke='Türkiye'
şehir='istanbul'
yaş='25'
kullanılan_programlar='R,MADlab,spss,Python,c++'
```

## Kaç karakter olduğunu öğrenmek için genel Python komutu

```
print(len('öğrenmek istediğimiz kelime'))
```

### ÖRNEK:

```
print(len('büşra'))
```

5

```
first_name='fatma büşra'  
print(len(first_name))
```

11

Yaşımı ismimi nerde yaşadığımı soran bir Python komutu yapmayalı deneyelim. Python'da soru olarak bana gelmesini istediğim metnin önüne input() komutu koyarak programın çalışmasını sağlayalım.

### ÖRNEK:

```
ilk_adın=input('adın ne:')  
yaş=input('Kaç yaşındasın?')  
şehir=input('hangi şehirde yaşıyorsun')
```

### ÇIKTI:

adın ne:

Sistem tek tek soru kutusu açmakta

```
adın ne: büşra  
yaşın: 25  
yaşadığın şehir: istanbul
```

## VERİ TÜRLERİNİ DÖNÜŞTÜRME

Daha önceki yazımızda veri türlerinin programlamadaki öneminden bahsetmiştik. Dönüşünde kullandığımız komutlar "int(), float(), str(), list()" olarak bilinir örnek üzerinde gösterim.

### list() komutu

İsmimin harflerini liste şekline getiren komutu yazalım

```
first_name='büşra'  
print(first_name)
```

büşra

```
first_name_to_list = list ( first_name )  
print ( first_name_to_list )
```

['b', 'ü', 'ş', 'r', 'a']

### float() komutu

Örneğin elimizdeki çıktı virgülden sonra 10 haneli fakat göremiyoruz tam sayıyı virgülden sonrasını görmek için aşağıdaki komutu kullanabiliriz.

```
num_int = 10  
print('num_int',num_int)  
num_float = float(num_int)  
print('num_float:', num_float)
```

num\_int 10  
num\_float: 10.0

Örneğin elimizde float şeklinde bir sayı dizisi var ve bunu yuvarlanmış görmek istiyorum aşağıdaki komutu kullanabiliriz.

```
pi_number= 3.14  
print(int(pi_number))
```

3

### str() komutu

Elimizdeki sayı dizisini metin halinde yazdıralım.

```
#numara_int = 10  
#print(numara_int)  
numara_str = str(numara_int)  
print(':',numara_str)
```

: 10

