

Değişkenler: Değişebilen değerlere değişken denir. Değişken kullanarak daha sade, işlevsel ve zaman kazandırıcı işlemler yapılır. **Ör:**

```
a = "Trabzon"
print(a)
Çıktı: Trabzon
```

Burada a değişken, "Trabzon" ise string türünde bir veridir.
= işareti ise atama işlemi yapar. **Ör:**

```
a= 5
b=-4
c=30
print(a*b+c)
Çıktı:10
```

Not: = sembolü matematikte kullanıldığı şekliinden farklıdır. Matematikte bu sembol eşitlik sağlar fakat Python 'da simetri olmadığı için $5 = x$ gibi bir ifade hatalı olacaktır. **Ör:**

```
5=x
print(x)
Çıktı:hata
```

DEĞİŞKEN KURALLARI:

1-Değişken adları rakamla başlamaz.

```
3_kilo_elma = "10 tl"    ✗
kilo_elma_3 = "10 tl"   ✓
```

2-Değişken adları özel sembol veya noktalama işareti içermez (_ altçizgi hariç)

```
gelir?= "500 TL"        ✗
kullanici_adi= "admin"  ✓
```

3- Değişken adlarında boşluk kullanılmaz.

```
kullanici adi = "admin" ✗
kullanici_adi = "admin" ✓
```

Uyarı: Değişken adlarında Türkçe karakter kullanabiliriz. Ancak uyum sorunu ihtimaline karşı bundan kaçınınız.

4- Değişken adlarında bazı özel anlam ifade eden kelimeler kullanılmaz.

```
True=5    ✗
true=5    ✓    #Küçük büyük harf duyarlılığından hata oluşmaz.
```

and=8 ✘
 And=8 ✔ #Küçük büyük harf duyarlılığından hata oluşmaz.

Not: Python'da özel anlam ifade eden kelimeleri görmek için aşağıdaki kodları yazın.

```
import keyword
keyword.kwlist
```

Çıktı:

```
['False', 'None', 'True', 'and', 'as', 'assert', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else',
'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or',
'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']
```

Ör: Python'da özel anlam ifade eden kaç adet kelime olduğunu len fonksiyonu ile bulalım.

```
import keyword
a=keyword.kwlist
len(a)
```

Çıktı: 33

Ör: Bir değere iki veya daha fazla değişken atayabiliriz.

```
a=b=c=4
print(a*b+c)    #4*4+4
```

Çıktı: 20

Ör: Bir değişkene defalarca farklı değerler atayabiliriz. Her zaman son değer geçerlidir.

```
x = 10
x = 20
x = 30
print(x+5)    #30+5
```

Çıktı: 35

Ör: Çoklu atama işlemi

```
x, y, z = 4,3,2
print(x*y-z)    #4*3-2
```

Çıktı:10

Ör: Değişken Takası

```
x, y, z=4, 3, 2
x, y, z=y, z, x    #x artık y, y artık z, z artık x oldu.
print(z**y*x)    #4**2*3
```

Çıktı: 48

Ör: Değişken iptali için del komutunu kullanabiliriz.

```
a=2
del a    #a değişkeni artık yok yani iptal edildi.
print(a)    #a değişkenini iptal ettiğimiz için değişen ekrana yazdırılmayacaktır.
```

Not: Python operatör kullanırken kısaltmalar yapabilir. Örneğin $x = x + 5$ deyimini $x += 5$ olarak kısaltılabilir. Bu ifade “x’i 5 arttır.” anlamına gelir.

```
>>> x=5
>>> x+=7      #x=x+7 ile aynı anlama gelir.
>>> print(x)
12

>>> x -= 4    #x=x-4 ile aynı anlama gelir.
>>> print(x)
8

>>> x //= 3   #x=x//3 ile aynı anlama gelir.
>>> print(x)
2

>>> x *= 8    #x=x*8 ile aynı anlama gelir.
>>> print(x)
16

>>> x %= 6    #x=x%6 ile aynı anlama gelir.
>>> print(x)
4

>>> x**= 3    #x=x**3 ile aynı anlama gelir.
>>> print(x)
64
```