

# Python Ders Notları - 7

## Değişken Kuralları:

### 1-Değişken adları sayı ile başlamaz.

```
3_kilo_elma = "10 tl"    X
```

```
kilo_elma_3 = "10 tl"    ✓
```

### 2-Değişken adları özel sembol içermez ( \_ altçizgi hariç ).

```
gelir?= "500 TL"        X
```

```
kullanici_adi= "admin"  ✓
```

### 3- Değişken adlarında boşluk olmaz.

```
kullanici adi = "admin" X
```

```
kullanici_adi = "admin" ✓
```

**Not:** Değişken adlarında Türkçe karakter kullanabiliriz. Ancak uyum sorunu ihtimaline karşı bundan kaçınınız.

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

```
True=5    X
```

true=5 ✓ → Küçük büyük harf duyarlılığından hata oluşmaz.

```
and=8    X
```

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
```

```
['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:

```
>>> import keyword
```

```
>>> a=keyword.kwlist
```

```
>>> len(a) → Python'da özel anlam ifade eden kaç adet kelime var?
```

```
33
```

```
>>> type(a) → Özel anlam ifade eden kelime grubu hangi veri türüne girer?
```

```
<class 'list'>
```

**Not:** Bir değere iki veya daha fazla değişken atayabiliriz.

```
>>>a=b=c=4
```

```
>>>print(a*b+c) → 4*4+4
```

```
20
```

**Not:** Bir değişkene defalarca farklı değerler atayabiliriz:

```
>>>x = 10
```

```
>>>x = 20
```

```
>>>x = 30 → Değişkenin en son değeri
```

```
>>>print(x) → Değişkenin en son değeri geçerlidir.
```

```
30
```

### Çoklu atama:

```
>>> x, y, z = 4,3,2
```

```
>>> print(x*y-z) → 4*3-2
```

```
10
```

```
>>> x, y, z = 4,3,2
```

```
>>> print("x =", x, " y =", y, " z =", z)
```

```
x = 4 y = 3 z = 2
```

### Değişken Takası:

```
>>> x,y,z=4,3,2
```

```
>>> x,y,z=y,z,x → x artık y oldu, y artık z oldu, z artık x oldu.
```

```
>>> print(z**y*x) → 4**2*3
```

```
48
```

**Değişken İptali:** Değişken iptali için komut penceresini kapatıp açabiliriz ya da del komutunu kullanabiliriz.

```
>>>a=2
```

```
>>>print(a)
```

```
2
```

```
>>>del a → a değişkeni artık yok yani iptal edildi.
```

```
>>>print(a) → a değişkenini iptal ettiğimiz için kod hata verir. X
```

## Hata Çeşitleri:

### 1-Söz Dizimi Hataları (SyntaxError) :

**Ör1:** Hatalı atama işlemi;

```
>>>y = 5
```

```
>>>x + 2 = y
```

```
>>>print(x)
```

**doğrusu:**

```
y = 5
```

```
x=y-2
```

```
print(x)
```

```
3
```

**Ör2:** Eşleşmeyen parantez;

```
print(5+(3*4) hata
```

```
print(5+(3*4)) doğrusu
```

**Ör3:** Eşleşmeyen tırnak işareti;

```
print('hello") hata
```

```
print("hello") doğrusu
```

```
print('hello') doğrusu
```

**Ör4:** Hatalı girinti;

```
a=4
```

```
b=6
```

```
if a<b:
```

```
print("a, b'den küçüktür")
```

**Doğrusu:**

```
a=4
```

```
b=6
```

```
if a<b:
```

```
print("a, b'den küçüktür")
```

### 2-Çalışma Zamanı Hataları:

**Ör1:** Atanmayan değişken;

```
x=5
```

```
print(x+y) → y diye bir değişken yok, yani atanmamış.
```

**doğrusu:**

```
x=5
```

```
y=8
```

```
print(x+y)
```

```
13
```

**Ör2:** Sıfıra bölme işlemi;

```
32/0 # Bir sayı 0'a bölünemez.
```

**Ör3:** Sayıları string'e bölmek;

```
print(5/"a")
```

**Ör4:** Hiç gerçekleşmeyecek koşullar;

```
if 2>4:
```

```
print("Merhaba Uzaylı")
```

**3-Mantık Hataları (Anlam Bilimsel Hatalar) :** Program, genellikle hata mesajı vermeden çalışır ancak istenilen işlemi gerçekleştirmez. En zor hata ayıklama türüdür.