

2. DEĞERLER VE DEĞİŞKENLER

```
elif not (value == "#BLANK"):
    if (typeofFID == "REAL"):
        value = float(value)
        tempValue = str(round
            numOfdot = tempValue.
        if not (numOfdot == -
            tmpFormat = len(te
            tmpFormat = 15-tmp
        if (numOfdot == -1):
```

Bu bölümde;

- ✓ Python programla dilinde değerler ve değişkenler konusunda bilgi sahibi olacak,
- ✓ Veri türlerini tanıyacak,
- ✓ Farklı değişkenleri kullanan program yazabileceksiniz.

2.1. Tam Sayı ve Diziler

- Tam sayılar pozitif, negatif ya da sıfır değeri alabilir.
- Kesirli değerleri içermez.

```
print(4)
```

- `>>> 5 + 4`
- `9`
- `>>> 1+6+4+10`
- `21`
- `>>> print(1 + 6 + 4 + 10)`
- `21`
- İfadede tırnak (") işareti kullanılmadığına dikkat ediniz

2.1. Tam Sayı ve Diziler

- `>>> 16`
- `16`
- `>>> "16"`
- `"16"`
- `>>> '16'`
- `'16'`
- İlk satırda **tam sayı** olan 16 değeri tek ya da çift tırnak içine alındığında sayı olarak değil **metin** (alfasayısal kelime yani dizi) olarak işlem görmektedir.
- Bu ifadeler **karakter dizisi** ya da **dizi** olarak anılır.

2.1. Tam Sayı ve Diziler

- `>>> type(4)` # 4 sayısının veri tipini soruyoruz
- `<class "int">` # veri tipinin int yani tamsayı olduğunu söylüyor
- `>>> type("4")` # veri tipini soruyoruz
- `<class "str">` # veri tipinin str yani metin olduğunu söylüyor

```
>>> str(4) # tam sayı olan 4 metine dönüşüyor
"4"
>>> "5"
"5"
>>> int("5") # Metin olan 5 tam sayıya dönüşüyor
5
```

2.1. Tam Sayı ve Diziler

- Toplama amacıyla kullanılan + sembolü diziler için farklı biçimde çalışır.

```
>>> 5 + 10
```

```
15
```

```
>>> "5" + "10"
```

```
"510"
```

```
>>> "abc" + "xyz"
```

```
"abcxyz"
```

2.2. Değişkenler ve Atama

- Değişkenler, **değerleri korumak** için kullanılır.
- Bu değerler sayı, dizi gibi **farklı biçimlerde** olabilir

$x = 10$

- Bu ifade ile 10 değeri, x değişkenine **atanmaktadır**.
- Bir değişkene **birden fazla kez** atama yapılabilir.
- Değer atama için **=** kullanılır.
- Python dilinde ise atama (=) sembolünün sol tarafında yer alan ifade, sağ taraftaki ifadeyi üstlenir.

2.2. Değişkenler ve Atama

- Buradaki kullanım, matematikteki kullanımdan **farklı** olduğu için önemlidir.
- Matematikte eşitlik ve eşitliğin her iki yanında simetri söz konusudur yani matematiksel açıdan $x = 10$ ve $10 = x$ olacak biçimde her iki ifadede doğru iken
- Python kapsamında bu, simetri olmadığı için
- **$10 = x$** ifadesi **hatalı** olacaktır.

2.2. Değişkenler ve Atama

Bir değişkene defalarca farklı değerler atayabiliriz.

- `x = 10`
- `print(x)`
- `x = 20`
- `print(x)`
- `x = x+30`
- `print(x)`

2.2. Değişkenler ve Atama

```
>>x = 20
```

```
>>print("x = " + str(x))
```

```
x = 20
```

- Burada print fonksiyonu içerisinde kullanılan toplama + işlemi dizileri birleştirmek amacıyla kullanılmaktadır.
- Bu örnekte print fonksiyonu, iki parametre kabul etmektedir.

```
>>x = 20
```

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

```
x = 20
```

2.2. Değişkenler ve Atama

- Çoklu atama

```
>>x, y, z = 100, -45, 0
```

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

2.2. Değişkenler ve Atama

Çoklu atama, sol taraftaki çoklu öge grubu ile sağ taraftaki çoklu öge grubunun sayıları eşit ise gerçekleşir. Atanan değer bir değişken ismini bir nesneye bağlar.



Şekil 2.6: Değişkene değer atama

Örnek olarak $x = 2$ ifadesini inceleyelim (Şekil 2.6). Bir kutu değişkeni ifade eder ve bu, değişken ismi ile adlandırılır. Diğer kutudan nesneye doğru yönelen ok, değişkenin bağlandığı nesneyi gösterir. Bu durumda ok 2 değerini içeren başka bir kutuyu işaret eder. İkinci kutu 2 değerinin ikilik düzendeki karşılığını içeren hafıza yerini temsil eder.

Bilgisayar her bir program satırını işledikçe değişkenlerin değerlerinin nasıl değiştiğini gözleyelim.

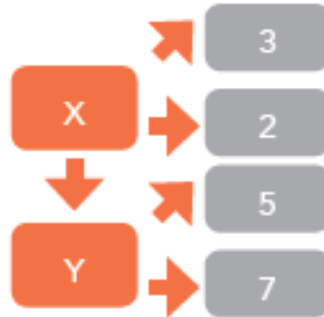
$x = 2$

$y = 5$

$x = 3$

$x = y$

$y = 7$



2.2. Değişkenler ve Atama

Programın çalışması sırasında bir değişkenin yalnızca değeri değil, türü de değişebilir.

```
a = 10
print("a değişkeninin ilk değeri", a, "ve tipi", type(a))
a = "ABC"
print("a değişkeninin yeni değeri", a, "ve tipi", type(a))
```

Bu ifade aşağıdaki çıktıyı oluşturur.

```
a değişkeninin ilk değeri 10 ve tipi <class "int">
a değişkeninin yeni değeri ABC ve tipi <class "str">
```

Değişkenin tipini değiştirmek sık yapılan bir işlem değildir.

2.3. Reel Sayılar

- Pek çok hesaplamalı işlem, kesir parçası olan sayıları kullanır.
- Python ile programlama yaparken bu tanımlı yapmak için ***float*** kelimesi kullanılır.
- `>>> x = 5.62`
- `>>> x`
- `5.62`
- `>>> type(x)`
- `<class "float">`

2.3. Reel Sayılar

- Reel sayılardan farklı olarak tam sayılar kesirli ifadeleri içeremez.
- Reel bir sayıyı bir tam sayıya dönüştürmenin iki temel yolu vardır:
 - **Yuvarlama:** Reel sayıya en yakın tam sayıya ulaşmak için kesrin belirli bir miktarı eklenerek ya da çıkarılarak yuvarlama yapılır.
 - **Kesme:** Sayının kesirli kısmı tamamen göz ardı edilir.

2.3. Reel Sayılar

- Yuvarlama için **round()** kullanılır

```
>>> round(28.71)
```

```
29
```

- Kesme için **int()** kullanılır

```
>>> int(28.71)
```

```
28
```


2.3. Reel Sayılar

- **Round()** fonksiyonun da belirli sayıda basamağı koruyabiliriz

```
>>> x = 93.34836
```

```
>>> x
```

```
93.34836
```

```
>>> round(x)
```

```
93
```

```
>>> round(x, 2)
```

```
93.35
```

```
>>> round(x, 3)
```

```
93.348
```

2.4. Belirteçler

- Programcıların bundan uzun, anlamlı ve açıklayıcı değişken isimleri seçmesi gerekir.
- t, at, y ve s gibi isimler **yerine**, toplam, araToplam, yükseklik gibi değişkenler seçilmeli
- Değişken ismi program içerisindeki kullanım **amacına** uygun olmalıdır.

2.4. Belirteçler

- Değişken isimleri ne kadar doğru seçilirse program okuyan kişiler için o kadar çabuk **anlaşılır** ve **anamlı** olur.
- Python, büyük küçük harf duyarlıdır ve değişken isimleri için kesin kurallar kullanır.
- Bir değişken ismi belirteç için bir örnektir. Belirteç, öğeleri isimlendirmek için kullanılan kelimedir.

2.4. Belirteçler

Belirteçlerin özellikleri şöyle sıralanabilir:

- Bir belirteç en az bir karakter içermelidir.
- Belirtecin ilk karakteri harf (kucuk ya da buyuk) ya da alt çizgi olmalıdır (ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz_).
- Devam eden karakterler, harf (küçük ya da büyük), alt çizgi ya da sayı olabilir (ABCDEFGHIJKLabcd efghijklxyz_0123456789).
- Belirteçlerde boşluk dahil diğer özel karakterler kullanılamaz.
- Programlama diline ait ayrılmış kelimeler belirteç olarak kullanılamaz (Reserved Words).
- Son olarak print, int, str ya da type gibi özel kelimeler kullanılabilmesine rağmen iyi bir program için bunlar asla önerilmez.
- Türkçe karakterler kullanılamaz.

2.4. Belirteçler

Python için Ayrılmış Kelimeler			Python için Doğru Belirteçler	Python için Yanlış Belirteçler
and	del	from		<input type="checkbox"/> ara-toplam (- sembolü kullanılamaz)
as	elif	global	<input checked="" type="checkbox"/> x	<input type="checkbox"/> ilk değer (boşluk kullanılamaz)
assert	else	if	<input checked="" type="checkbox"/> a2	<input type="checkbox"/> 4ogrenci (sayı ile başlayamaz)
break	except	import	<input checked="" type="checkbox"/> Toplam	<input type="checkbox"/> *2 (* sembolü kullanılamaz)
class	False	in	<input checked="" type="checkbox"/> Toplam_Brut	<input type="checkbox"/> öğrenci (Türkçe karakter içeremez)
continue	finally	is	<input checked="" type="checkbox"/> Anahtar_10	<input type="checkbox"/> class (class ayrılmış kelime olduğu için kullanılamaz)
def	for	lambda		

Şekil 2.7: Python için özel durumlar

Açıklamalar

- Python'da kodlara açıklama eklerken # kullanılır.
- IDLE'de yeni bir sayfa açılınca kodları çalıştırmak için önce kaydetmeliyiz.
- Ardından F5 veya Run komutu kullanılır

Değişkene Değer Aktarma



File Edit Format Run Options Window Help

```
sayi=input("bir sayı giriniz:")
metin=input("bir metin giriniz:")
print("Girdiğiniz Sayı:",sayi)
print("Girdiğiniz Metin:",metin)
print(type(sayi))
print(type(metin))|
```

Örnek Kodlamalar

File Edit Format Run Options Window Help

```
birinci_Sayi=int(input("Birinci Sayıyı Giriniz: "))
ikinci_Sayi=int(input("İkinci Sayıyı Giriniz: "))
toplam=birinci_Sayi+ikinci_Sayi
carpim=birinci_Sayi*ikinci_Sayi
fark=birinci_Sayi-ikinci_Sayi
bolum=birinci_Sayi/ikinci_Sayi
print(type(birinci_Sayi))
print(type(ikinci_Sayi))
print("İki Sayının Toplamı= ",toplam)
print("İki Sayının Çarpımı= ",carpim)
print("İki Sayının Farkı= ",fark)
print("İki Sayının Bölümü= ",bolum)
```

|