

Python ile Programlamaya Giriş

DERS 2: PYTHON İLE İLGİLİ GENEL BİLGİLER

DR. HÜSEYİN BAHTİYAR

Ders 1 Özet

- ▶ Programlama nedir, kullanıcı programcı kimdir.
- ▶ Python nasıl bir dildir.
- ▶ Python'ı nasıl kurabilirim.
- ▶ Kurup test etmek.

Print Komutu

- ▶ Öncelikle konsolu açıp `print()` yazıp enter tuşuna basalım.
 - ▶ Ne olduğu hakkında konuşalım.
- ▶ Şimdi `print('merhaba')` yazalım ve çıktısı hakkında konuşalım.

Sabitler

- ▶ Sabit değerler : rakam, harf, harf katarları (string) sabitler olarak adlandırılırlar çünkü bu değerler değişmezler.
- ▶ Nümerik sabitler
 - ▶ `print (95.3)`
- ▶ Harf katarları için ise tek ve çift tırnak kullanılır
 - ▶ `print('merhaba ben python koduyum!')`

Değişkenler

- ▶ Veriyi kendi içinde depolayan birimlere değişken adı verilir.
- ▶ Değişkenin adını kendimiz seçebiliriz.
- ▶ Değişkenin içeriğini programın içerisinde değiştirebiliriz.

$x = 12.2$

$y = 14$

$x = 100$

x

~~12.2 100~~

y

14


```
x = 12.2
print(x)
x = x + 2
print(x)
y = x - 5
print(y)
x = x/y
print(x, y)
```

x:12.2

```
x = 12.2
print(x)
x = x + 2
print(x)
y = x - 5
print(y)
x = x/y
print(x, y)
```

x:12.2

x:14.2


```
x = 12.2
print(x)
x = x + 2
print(x)
y = x - 5
print(y)
x = x/y
print(x, y)
```

x:12.2

x:14.2

```
x = 12.2
print(x)
x = x + 2
print(x)
y = x - 5
print(y)
x = x/y
print(x, y)
```

x:12.2

x:14.2

y:9.2

```
x = 12.2
print(x)
x = x + 2
print(x)
y = x - 5
print(y)
x = x/y
print(x, y)
```

x:12.2

x:14.2

y:9.2

x:1.54347826087 y:9.2

Python Değişken İsimlendirme Kuralları

12

- ▶ Alt çizgi _ veya bir harf ile başlayabilir.
- ▶ Harf, altçizgi veya rakam içerebilir.
- ▶ Büyük/küçük harf duyarlılığı vardır.

```
Doğru:      spam      eggs      spam23      _speed
Yanlış:      23spam      #sign      var.12
Farklı:      spam      Spam      SPAM
```

Rezerve Edilmiş Kelimeler

13

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

Belleksel Değişken İsimleri

14

- ▶ Programcılar kendi değişken isimlerini rahatça seçebildiğinden dolayı bazen kendi ihtiyacımıza göre farklı değişkenler seçebiliyoruz.
- ▶ Bazı değişkenlerin isimlerini içerisinde ne olduğunu hatırlamamız için bu isimleri kullanırız.

```
x1q3z9ocd = 35.0
```

```
x1q3z9afd = 12.50
```

```
x1q3p9afd = x1q3z9ocd * x1q3z9afd
```

```
print(x1q3p9afd)
```

Bu Program ne yapıyor?

Belleksel Değişken İsimleri

15

- ▶ Programcılar kendi değişken isimlerini rahatça seçebildiğinden dolayı bazen kendi ihtiyacımıza göre farklı değişkenler seçebiliyoruz.
- ▶ Bazı değişkenlerin isimlerini içerisinde ne olduğunu hatırlamamız için bu isimleri kullanırız.

```
x1q3z9ocd = 35.0
```

```
x1q3z9afd = 12.50
```

```
x1q3p9afd = x1q3z9ocd * x1q3z9afd
```

```
print(x1q3p9afd)
```

```
a = 35.0
```

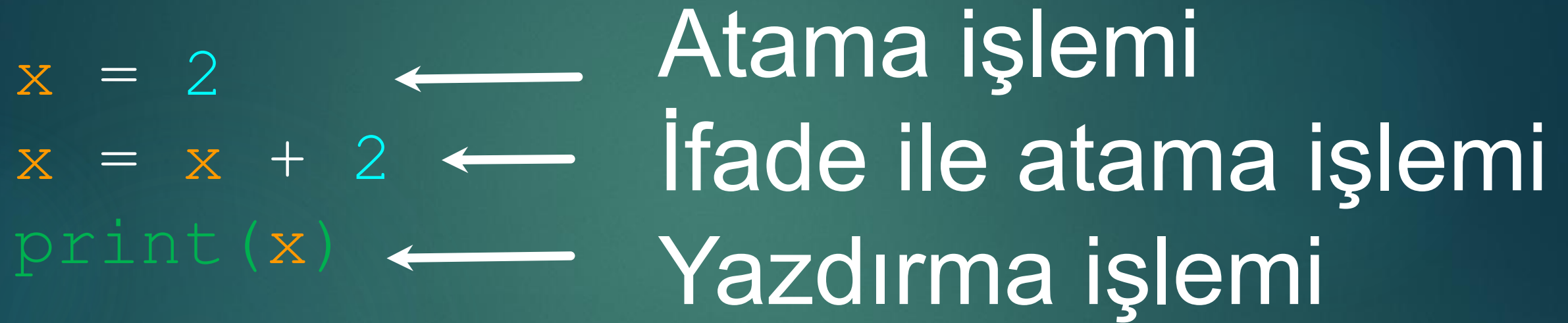
```
b = 12.50
```

```
c = a * b
```

```
print(c)
```

Bu Program ne yapıyor?

Satırlar ve Betikler



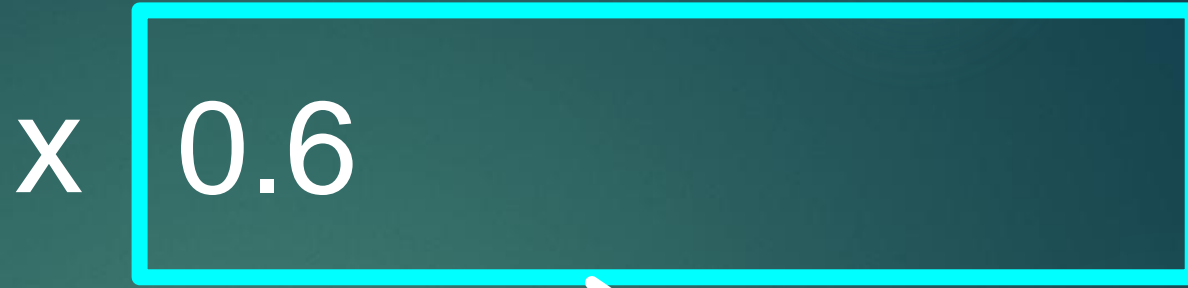
Değişken Operatör Sabit Fonksiyon

Atama İşlemleri

- ▶ Değişkene değer atama işlemi = işareti ile yapılır.
- ▶ Atama işleminde atamak istediğimiz ifade sağda değişken ise solda olacak şekilde yapılmaktadır.
- ▶ Şimdi aşağıdaki ifadeyi inceleyelim.

$$x = 3.9 * x * (1 - x)$$

Değişken aslında veriyi tutan bir hafıza noktasıdır. Kaydedilen değişken hafızadaki eski değeri (0.6) yeni değer ile değiştirir (0.936).



$$x = 3.9 * x * (1 - x)$$

0.6

0.6

0.4

0.936

Denklemin sağı bir ifadedir. İfade hesaplandığı zaman, sonuç değeri soldaki değişkene atanır.

Değişken aslında veriyi tutan bir hafıza noktasıdır. Kaydedilen değişken hafızadaki eski değeri (0.6) yeni değer ile değiştirir (0.936).



$$x = 3.9 * x * (1 - x)$$

0.6

0.6

0.4

0.936

Denklemin sağı bir ifadedir. İfade hesaplandığı zaman, sonuç değeri soldaki değişkene atanır.

Python da Matematik İşlemleri

20

Operator	İşlem
+	Toplama
-	Çıkarma
*	Çarpma
/	Bölme
**	Üs
%	Mod

```
print (4 + 2)
```

```
print (4 - 2)
```

```
print (4 * 2)
```

```
print (4 / 2)
```

```
print (4 ** 2)
```

```
print (4 % 2)
```

Nümerik ifadeler

21

```
>>> xx = 2
>>> xx = xx + 2
>>> print(xx)
4
>>> yy = 440 * 12
>>> print(yy)
5280
>>> zz = yy / 1000
>>> print(zz)
5.28

>>> jj = 23
>>> kk = jj % 5
>>> print(kk)
3
>>> print(4 ** 3)
64
```

Operator	İşlem
+	Toplama
-	Çıkarma
*	Çarpma
/	Bölme
**	Üs
%	Mod

Operatör Önceliği Kuralları

22

- ▶ Hesaplamalarımızda birden çok operatör kullanabiliriz. Eğer istediğimiz öncelikte yapmak istiyorsak Python'a hangisini önce yaptırmak istediğimizi söylemeyiz.
- ▶ Buna operatör önceliği denmektedir.
- ▶ Aşağıdaki işlemi inceleyelim sonucu kaç olur?

$$x = 1 + 2 * 3 - 4 / 5 ** 6$$

Operatör Önceliđi Kuralları

23

Operatör Önceliđi sıralaması:

- ▶ Parantezler
- ▶ Üsler
- ▶ Çarpım, Bölüm, Mod
- ▶ Toplama ve çıkarma
- ▶ Soldan sađa

Parantez
Üs
Çarpım
Toplam
Soldan sađa



```
>>> x = 1 + 2 ** 3 / 4 * 5
```

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

```
11.0
```

```
>>>
```

Parantez

Üs

Çarpım

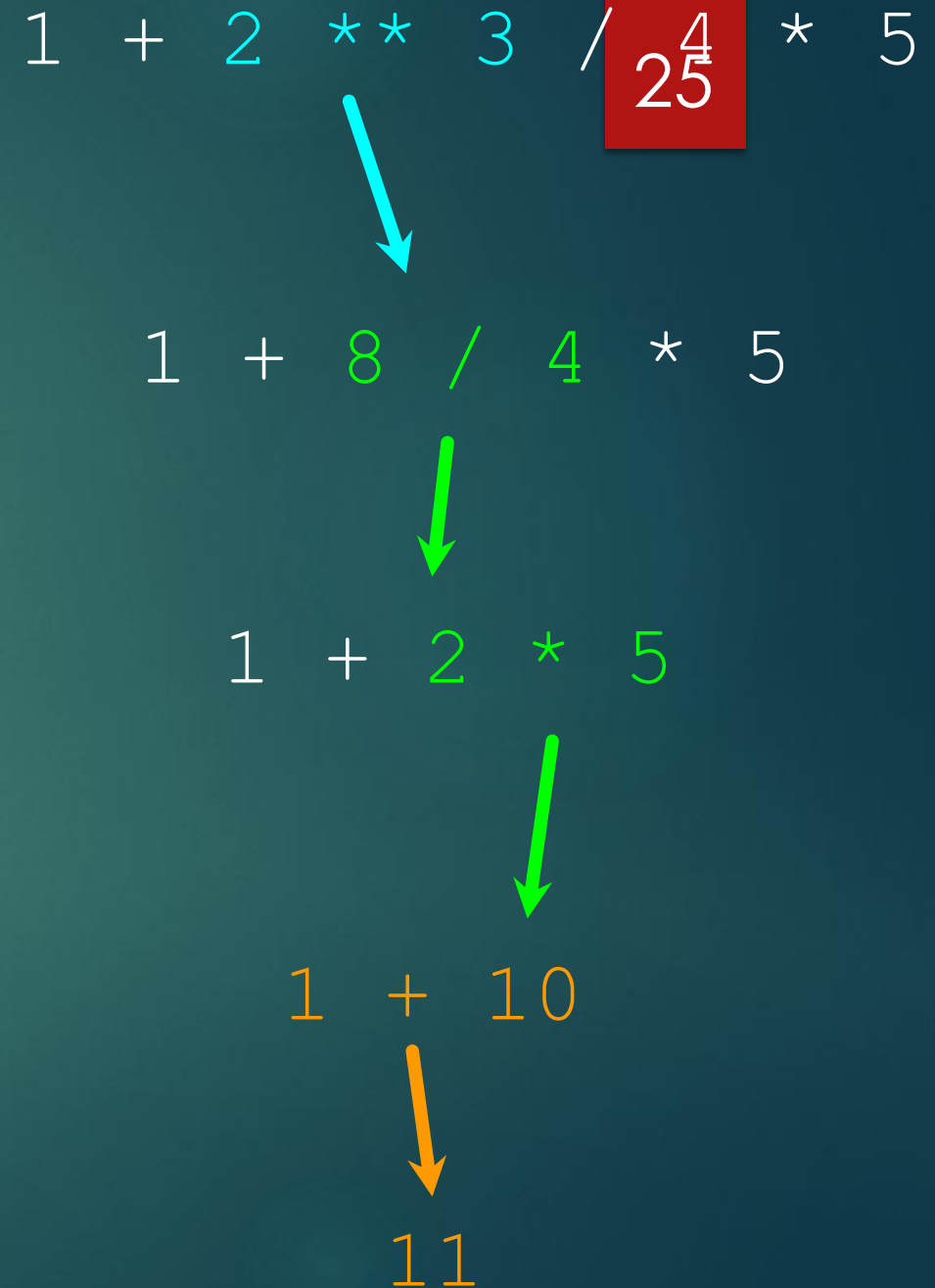
Toplam

Soldan sağa




```
>>> x = 1 + 2 ** 3 / 4 * 5
>>> print(x)
11.0
>>>
```

1 + 2 ** 3 / 4 * 5



1 + 8 / 4 * 5

1 + 2 * 5

1 + 10

11

Parantez
Üs
Çarpım
Toplam
Soldan sağa



Operatör Önceliđi Tavsiyeleri.

26

- ▶ Kuralları ezbere bilmelisiniz!
- ▶ Program yazarken parantez kullanın!
- ▶ Programın içinde matematiksel deđişkenleri kolayca anlaşılacak şekilde yazmanız anlamınızı (sonrasında okuyan başka birinin anlamasını) kolaylaştıracaktır.
- ▶ Uzun seri bir matematiksel işlemi küçük parçacalara ayırıp anlaşılması kolay şekilde yazmanız işleri çok kolaylaştırır.

Parantez
Üs
Çarpım
Toplam
Soldan sağa



Python da Matematik İşlemleri

27

- ▶ Şimdi ise farklı bir şey deneyelim;
`print ('4 + 2')` bize nasıl bir çıktı verdi?

`print ('4' + '2')` bize nasıl bir çıktı verdi?

Neden??



Python da Matematik İşlemleri

28

- ▶ Şimdi ise farklı bir şey deneyelim;
`print ('4 + 2')` bize nasıl bir çıktı verdi?

`print ('4' + '2')` bize nasıl bir çıktı verdi?

Neden??



STRING

Python da Veri Tipleri

29

- ▶ Şu an için tamsayı virgüllü sayı ve string olduğunu gördük şimdi ise bir örnek yapalım

```
>>> a = 99
```

```
>>> b = 100
```

```
>>> print (a /b)
```

```
0
```



????

Python da Veri Tipleri

30

- Şu an için tamsayı virgüllü sayı ve string olduğunu gördük şimdi ise bir örnek yapalım

```
>>> a = 99
>>> b = 100
>>> print (a /b)
0
```

```
>>> a = 99.0
>>> b = 100.0
>>> print (a /b)
0.99
```



????

Python da Veri Tipleri

- ▶ Pythonda değişkenler, kelimeler ve sabitler vardır her birinin bir tipi vardır.
- ▶ Python tamsayı, ondalık sayı ve sözcük katarlarının farkını bilir.
- ▶ Örnek olarak + işareti sayılar için toplama işlemi iken stringleri ise birbirine bağlar.

```
>>> ddd = 1 + 4
```

```
>>> print(ddd)
```

```
5
```

```
>>> eee = 'merhaba ' + 'dünya'
```

```
>>> print(eee)
```

```
merhaba dünya
```


Tip Önemlidir



32

- ▶ Python değişkenin hangi tipte olduğunu önceden anlayabilir.
- ▶ Bazı yapılan işlemler yasaktır!
 - ▶ Mesela string değişkenine 1 ekleyemezsiniz
- ▶ Eğer emin değilsek Python'a değişkenin tipini `type()` fonksiyonunu kullanarak sorabiliriz.

```
>>> eee = 'hello ' + 'there'
>>> eee = eee + 1
Traceback (most recent call last):
  File "<stdin>", line 1, in
<module>TypeError: Can't convert 'int'
object to str implicitly
>>> type(eee)
<class'str'>
>>> type('hello')
<class'str'>
>>> type(1)
<class'int'>
>>>
```


Rakamların Tipleri

- ▶ Rakamların iki temel tipi vardır
 - ▶ **Tamsayılar: (int)**
 - ▶ 12, -2,4,0,23423423, ...
 - ▶ **Ondalık sayılar: (float)**
 - ▶ 0.25, -2.3, 0.0 , 1.92 ...
- ▶ Farklı tipler de vardır ancak bunlar ondalık ve tam sayı varyasyonlarıdır.

```
>>> xx = 1
>>> type (xx)
<class 'int'>
>>> temp = 98.6
>>> type (temp)
<class 'float'>
>>> type (1)
<class 'int'>
>>> type (1.0)
<class 'float'>
>>>
```

Tip deęişimleri

- ▶ Eęer tamsayı ile ondalık sayıyı bir işleme sokarsak python otomatik olarak sonucu ondalık sayı olarak verir.
- ▶ Bunu zorlama ile deęiştiremeyiz.

```
>>> print(float(99) + 100)
199.0
>>> i = 42
>>> type(i)
<class'int'>
>>> f = float(i)
>>> print(f)
42.0
>>> type(f)
<class'float'>
>>>
```

String Değişimleri

35

- ▶ Eğer string değişkeni nümerik karakterden oluşuyor ise, `int()` ve `float()` komutları ile yapısını değiştirebiliriz.
- ▶ Fakat bulunmuyor ise Python bize **hata mesajı** verir!

```
>>> sval = '123'
```

```
>>> type(sval)
```

```
<type 'str'>
```

```
>>> print (sval+1)
```

```
Traceback (most recent call last):
```

```
File "<stdin>", line 1, in <module>
```

```
TypeError: cannot concatenate 'str' and 'int' objects
```

```
>>> ival = int(sval)
```

```
>>> type(ival)
```

```
<type 'int'>
```

```
>>> print(ival+1)
```

```
124
```

```
>>> nival = "merhaba"
```

```
>>> nival = int(nival)
```

```
Traceback (most recent call last):
```

```
File "<stdin>", line 1, in <module>
```

```
ValueError: invalid literal for int() with base 10:
```

```
'merhaba'
```

```
>>>
```



Acknowledgements / Contributions

These slides are Copyright 2010- Charles R. Severance (www.dr-chuck.com) of the University of Michigan School of Information and made available under a Creative Commons Attribution 4.0 License. Please maintain this last slide in all copies of the document to comply with the attribution requirements of the license. If you make a change, feel free to add your name and organization to the list of contributors on this page as you republish the materials.

Initial Development: Charles Severance, University of Michigan School of Information

... Insert new Contributors and Translators here

Continue...