COMPUTER PROGRAMMING I
Introduction To Python

BIL2205

Dokuz Eylul University, Faculty of Science,
Department of Statistics

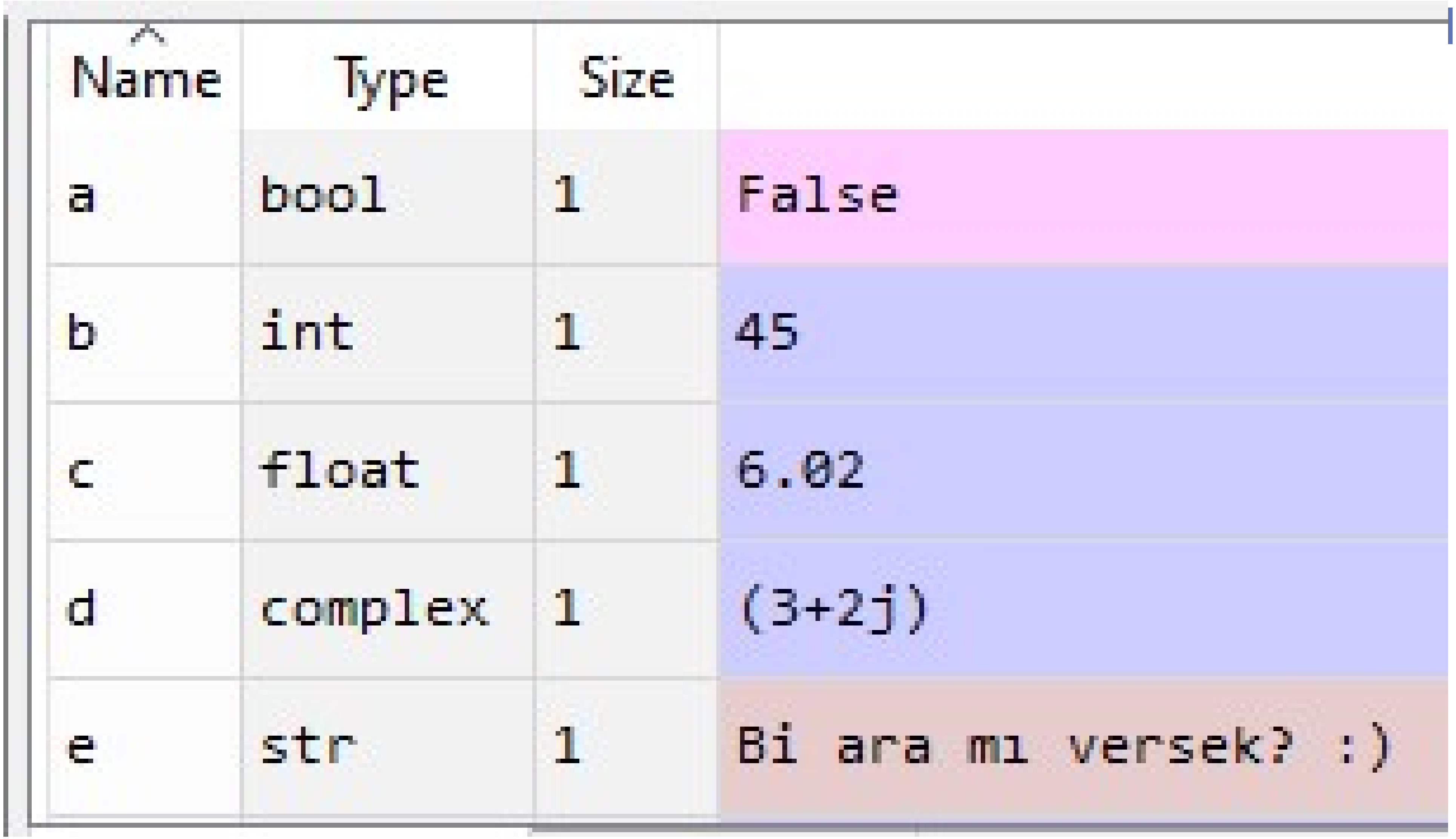
Data Types in Python

- *Basic Types:*
  - *Boolean (True or False)*
  - *Integer Numbers (47),*
  - *Floating Point Numbers (3.14),*
  - *Complex Numbers (3 + 2j),*
  - *Strings ("Alper", 'Computer Programming')*

Data Types in Python

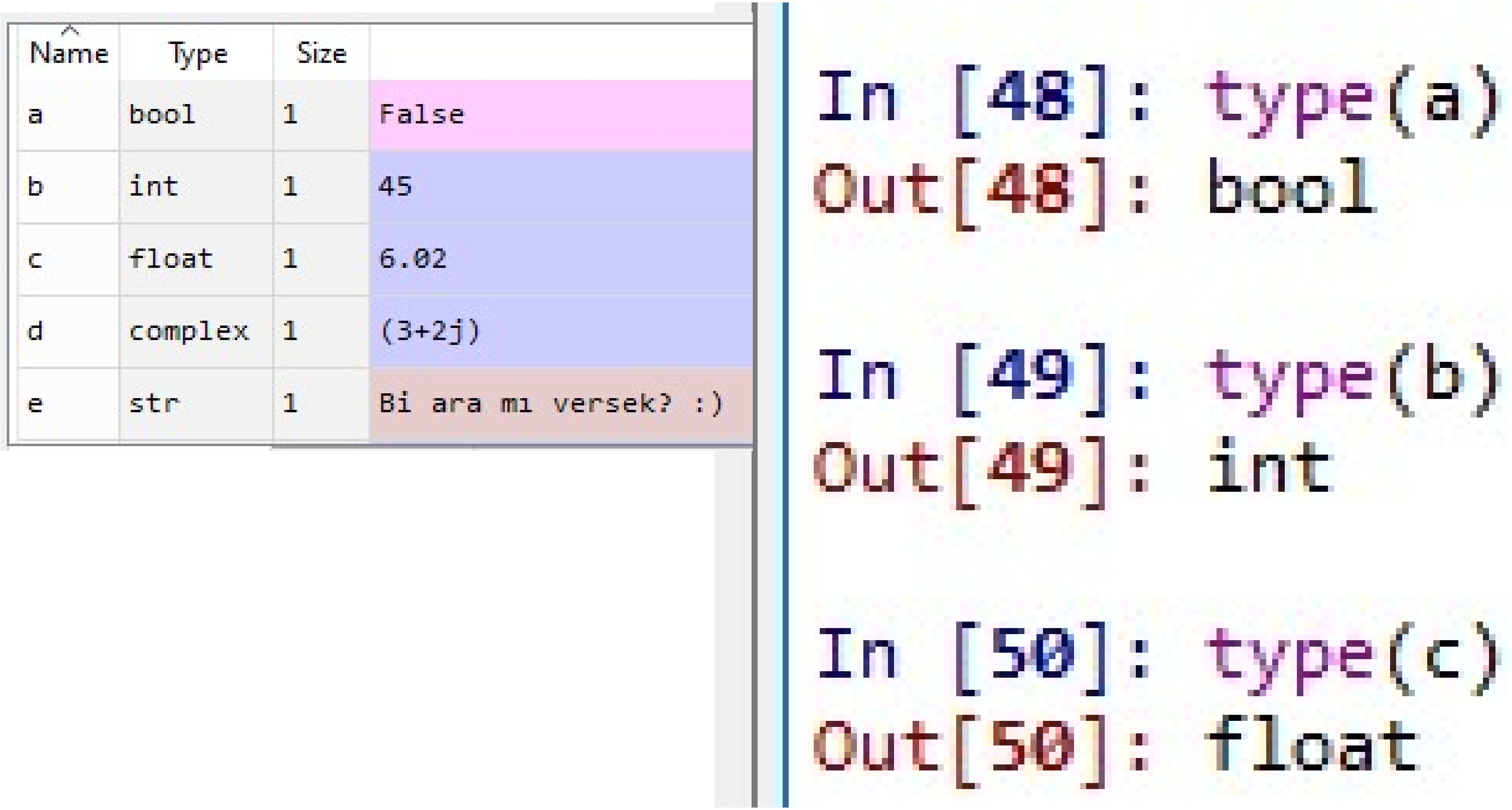
- *Boolean (True or False),*
  - In [1]: a = 45 < 23
- *Integer Numbers (47),*
  - In [2]: b = 45
- *Floating Point Numbers (3.14),*
  - In [3]: c = 6.02
- *Complex Numbers (3 + 2j),*
  - In [4]: d = 3 + 2j
- *Strings ("Alper", 'Computer Programming')*
  - In [5]: e = "Bi ara mı versek? :)"

Data Types in Python



Name	Type	Size	
a	bool	1	False
b	int	1	45
c	float	1	6.02
d	complex	1	(3+2j)
e	str	1	Bi ara mı versek? :)

Data Types in Python



Name	Type	Size	
a	bool	1	False
b	int	1	45
c	float	1	6.02
d	complex	1	(3+2j)
e	str	1	Bi ara mı versek? :)

```
In [48]: type(a)
Out[48]: bool

In [49]: type(b)
Out[49]: int

In [50]: type(c)
Out[50]: float
```

Data Types in Python

- *Other Types:*
  - *Lists*
  - *Dictionaries*
  - *Tuples*
  - *Sets*

Strings in Python

- Strings are a basic data type in Python
- Indicated using pairs of single ' ' or double " " quotes.
- Multiline strings can be declared by using three quotes.
- *Ex:*

```
    metin = "I love Python"
    cümle = 'I love Python'
    paragraf = """This is a
                  multiline string
                  example."""
```

String Functions in Python

- len()
  - *returns the length of any Python variable that contains some sort of countable thing.*
  - *in the case of strings it is the number of characters in the string.*
- *Ex:*

```
metin = "bir"
print (len(metin))
metin = "I love Python lessons very much (!)"
print (len(metin))
```

String Functions in Python

- upper()
  - *returns the UPPERCASE of a string.*
- *Ex:*

```
metin = "I love Python lessons very much (!)"
print (metin.upper())
```

- lower()
  - *returns the lowercase of a string.*
- *Ex:*

```
metin = "I love Python lessons very much (!)"
print (metin.lower())
```

String Functions in Python

- title()
  - Converts The First Character Of Each Word To Upper Case.
- *Ex:*

```
metin = "I love Python lessons very much (!)"
print (metin.title())
```

- swapcase()
  - *lower case becomes upper case and vice versa.*
- *Ex:*

```
metin = "I love Python lessons very much (!)"
print (metin.swapcase())
```

String Functions in Python

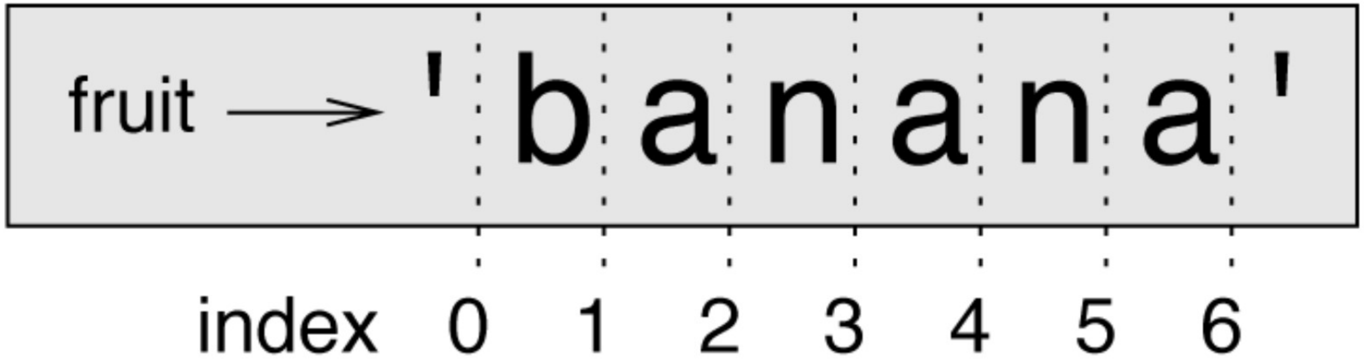
- Strings can be concatenated by " + " operator

```
ad = "Ali"
soyad = "Öztürk"
isim = ad + soyad
print (isim)
isim = ad + " " + soyad
print (isim)
```

String Functions in Python

- A string is a *sequence* of characters.
- You can access the characters one at a time with the bracket operator: [ ]
- Ex:

```
fruit = 'banana'
print (metin[1])
```



String Functions in Python

- A string is a *sequence* of characters.
- Write a string one character per line

```
metin = "Python is amazing"
for i in range(len(metin)):
    print (metin[i])
```

String Functions in Python

- A string is a *sequence* of characters.
- Write a string one character per line

```python
metin = "Python is amazing"
for harf in metin:
    print (harf)
```

String Functions in Python

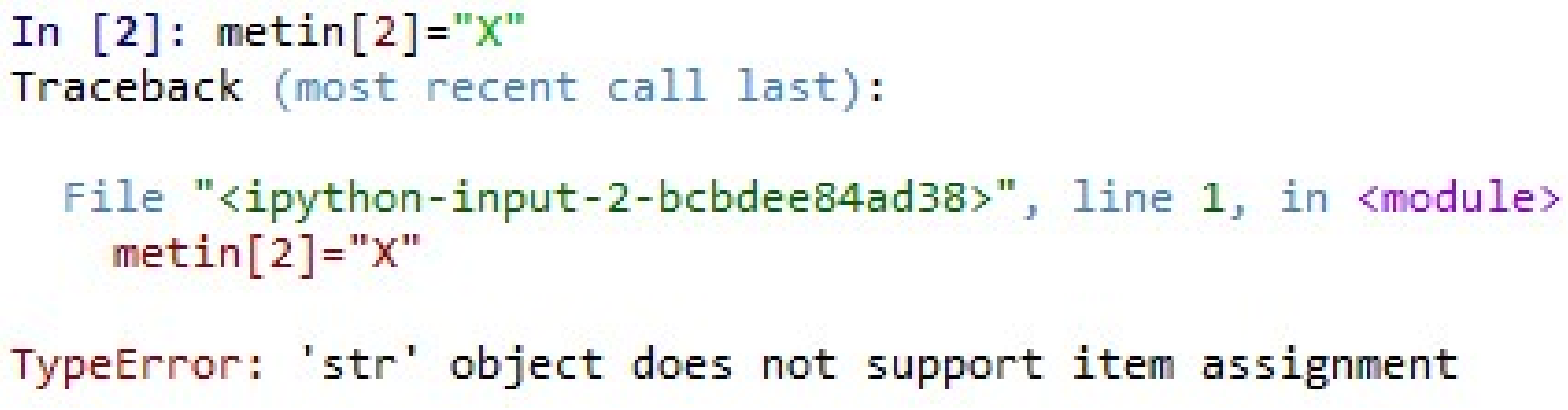
- A string is a *sequence* of characters.
- Reverse a string

```
metin = "Python is amazing"
for i in range(len(metin)-1, -1, -1):
    print (metin[i], end="")
```

String Functions in Python

- Strings are ***immutable!***

```
metin = "Python is amazing"
metin[2] = "X"
```



```
In [2]: metin[2]="X"
Traceback (most recent call last):

  File "<ipython-input-2-bcbdee84ad38>", line 1, in <module>
    metin[2]="X"

TypeError: 'str' object does not support item assignment
```

String Functions in Python

- Strings are ***sliceable***

```
metin = "Python is amazing"
print (metin[0])        # prints "P"
print (metin[0:5])      # prints "Pytho"
print (metin[7:9])      # prints "is"
print (metin[:6])       # prints "Python"
print (metin[10:])      # prints "amazing"
print (metin[-1])       # prints "g"
print (metin[-7:])      # prints "amazing"
print (metin[:-7])      # prints "Python is"
```

String Functions in Python

- `find(sth)`
  - finds the **first** occurrence of the specified value, returns -1 if not found.
- *Ex:*

```
metin = "Python is amazing"
print (metin.find("n"))
```

- `find(sth, from)`

```
print (metin.find("n",8))
```

- `find(sth, from, end)`

```
print (metin.find("n",8,12))
```

String Functions in Python

- rfind(sth)
  - finds the **last** occurrence of the specified value, returns -1 if not found.
- *Ex:*

```
metin = "Python is amazing"
print (metin.rfind("n"))
```

- rfind(sth, from)

```
print (metin.find("n",8))
```

- rfind(sth, from, end)

```
print (metin.find("n",8,12))
```

String Functions in Python

- `replace(old, new)`
- Replaces the "old" value with the "new" value in string.
- *Ex:*

```
    metin = "Python is amazing"
    print (metin.replace("n","X"))
```

```
txt = "one one was a race horse, two was one too."
print (txt.replace("one","forty"))
print (txt.replace("one","forty",2))
```

String Functions in Python

□ "in" Operator

- takes two strings and returns True if the first appears as a substring in the second
- *Ex:*

```
metin = "Python is amazing"
print ("n" in metin)
print ("Python" in metin)
print ("ama" in metin)
print ("Java" in metin)
```

Exercises

- □ *Print the common letters of a name and the surname.*

```
name = input("What is your Name? ")
surname = input("What is your Surname? ")
for letter in name:
    if letter in surname:
        print (letter)
```

Strings in Python

- Exercises:
  - Count the number of a given letter in a string.
  - Count the number of words in a string.
  - Convert "Ali Veli" to "Veli, ALİ"
  - Reverse each word in a string.

Strings in Python

- `metin = "Python"`
  - Print:

```
P         Python    P         P
Py        Pytho      y        yy
Pyt       Pyth        t       ttt
Pyth      Pyt          h      hhhh
Pytho     Py            o     ooooo
Python    P              n    nnnnnn

             Pnyoth
```