

Python for Data Science

Day - 2



First Code

```
# First python output with 'Print' functions  
print('Hello World!')  
print('Hi, Python!')
```

Hello World!

Hi, Python!

Version control

```
# Python version check  
import sys  
print(sys.version) # version control
```

3.9.12 (main, Apr 4 2022, 05:22:27) [MSC v.1916 64 bit (AMD64)]

help() function

```
# The Python help function is used to display the documentation of  
modules, functions, classes, keywords, etc.  
help(sys) # here the module name is 'sys'
```

Output exceeds the size limit. Open the full output data in a text editor
Help on built-in module sys:

NAME
sys

MODULE REFERENCE

<https://docs.python.org/3.9/library/sys>

The following documentation is automatically generated from the Python source files. It may be incomplete, incorrect or include features that are considered implementation detail and may vary between Python implementations. When in doubt, consult the module reference at the location listed above.

DESCRIPTION

This module provides access to some objects used or maintained by the interpreter and to functions that interact strongly with the interpreter.

Dynamic objects:

argv -- command line arguments; argv[0] is the script pathname if known
path -- module search path; path[0] is the script directory, else "
modules -- dictionary of loaded modules

displayhook -- called to show results in an interactive session

...

FILE

(built-in)

Comment

• • •

```
# This is a comment, and to write a comment, '#' symbol is used.  
print('Hello World!') # This line prints a string.  
# Print 'Hello'  
print('Hello')
```

Hello World!

Hello

Errors

• • •

```
# Print string as error message  
frint('Hello, World!')
```

NameError Traceback (most recent call last)

Input In [13], in <cell line: 2>()
1 # Print string as error message
----> 2 frint('Hello, World!')

NameError: name 'frint' is not defined

• • •

```
# Built-in error message  
print('Hello, World!')
```

Input In [14]

print('Hello, World!')

^

SyntaxError: EOL while scanning string literal

• • •

```
# Print both string and error to see the running order  
print('This string is printed')  
frint('This gives an error message')  
print('This string will not be printed')
```

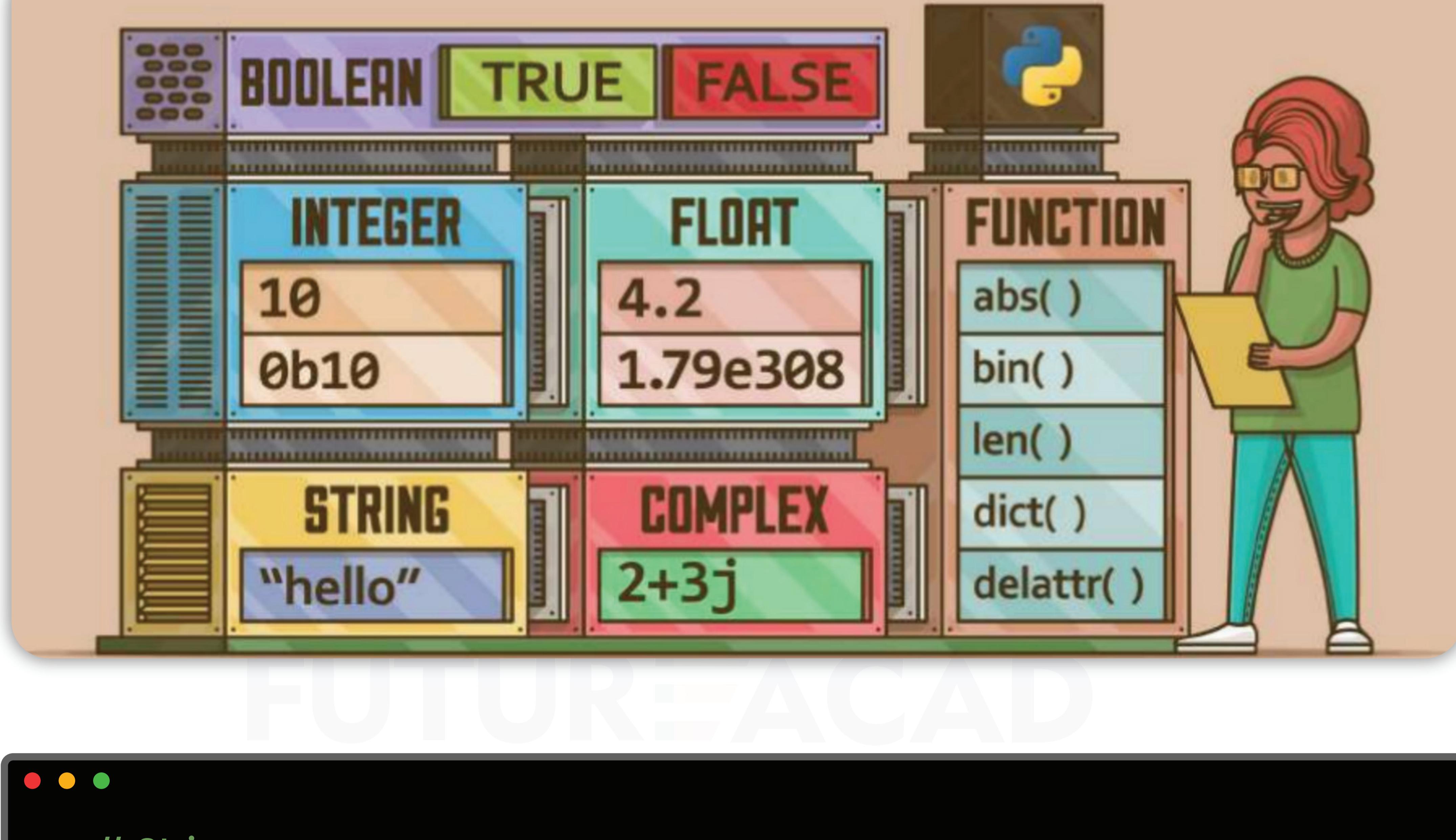
This string is printed

NameError Traceback (most recent call last)

Input In [15], in <cell line: 3>()
1 # Print both string and error to see the running order
2 print('This string is printed')
----> 3 frint('This gives an error message')
4 print('This string will not be printed')

NameError: name 'frint' is not defined

Basic data types in Python



FUTURE ACAD
Joint initiative of Roaman & NSDC

● ● ●

```
# String
print("Hello, World!")
# Integer
print(12)
# Float
print(3.14)
# Boolean
print(True)
print(False)
print(bool(1)) # Output = True
print(bool(0)) # Output = False
```

Hello, World!
12
3.14
True
False
True
False

type() function

FUTURE ACAD
Joint initiative of Roaman & NSDC

● ● ●

```
type((True))
```

float

FUTURE ACAD
Joint initiative of Roaman & NSDC

● ● ●

```
# String
print(type('Hello, World!'))
# Integer
print(type(15))
print(type(-24))
print(type(0))
print(type(1))
# Float
print(type(3.14))
print(type(0.5))
print(type(1.0))
print(type(-5.0))
# Boolean
print(type(True))
print(type(False))
```

<class 'str'>
<class 'int'>
<class 'int'>
<class 'int'>
<class 'int'>
<class 'float'>
<class 'float'>
<class 'float'>
<class 'float'>
<class 'bool'>
<class 'bool'>

Converting an object type to another object type

```
# Let's convert the integer number 6 to a string and a float
number = 6
print(str(number))
print(float(number))
print(type(number))
print(type(str(number)))
print(type(float(number)))
str(number)
```

```
6
6.0
<class 'int'>
<class 'str'>
<class 'float'>
'6'
```

```
# Let's convert the float number 3.14 to a string and an integer
number = 3.14
print(str(number))
print(int(number))
print(type(number))
print(type(str(number)))
print(type(int(number)))
str(number)
```

```
3.14
3
<class 'float'>
<class 'str'>
<class 'int'>
'3.14'
```

```
# Let's find the data types of 9/3 and 9//4
print(9/3)
print(9//4)
print(type(9/3))
print(type(9//4))
```

```
3.0
2
<class 'float'>
<class 'int'>
```

Expression and variables

```
# Addition  
x = 56+65+89+45+78.5+98.2  
print(x)  
print(type(x))
```

```
431.7  
<class 'float'>
```

```
# Subtraction  
x = 85-52-21-8  
print(x)  
print(type(x))
```

```
4  
<class 'int'>
```

```
# Multiplication  
x = 8*74  
print(x)  
print(type(x))
```

```
592  
<class 'int'>
```

```
# Division  
x = 125/24  
print(x)  
print(type(x))
```

```
5.208333333333333  
<class 'float'>
```

```
# Floor division  
x = 5//2  
print(x)  
print(type(x))
```

```
2  
<class 'int'>
```

```
# Modulus  
x = 5%2  
print(x)  
print(type(x))
```

```
1  
<class 'int'>
```

```
# An example: Let's calculate how many minutes there are in 20  
hours?  
one_hour = 60 # 60 minutes  
hour = 20  
minutes = one_hour *hour  
print(minutes)  
print(type(minutes))
```

```
1200  
<class 'int'>
```

```
# An example: Let's calculate how many hours there are in 348 min-  
utes?  
minutes = 348  
one_hour = 60  
hours = 348/60  
print(hours)  
print(type(hours))
```

```
5.8  
<class 'float'>
```

```
# Mathematica expression  
x = 45+3*89
```

```
print(x)  
print(y)  
print(x+y)  
print(x-y)  
print(x*y)  
print(x/y)  
#print(x**y)  
print(x//y)  
print(x%y)
```

```
312  
4272  
4584  
-3960  
1332864  
0.07303370786516854  
0  
312
```

Variables

```
# Store the value 89 into the variable 'number'  
number = 90  
print(number)  
print(type(number))
```

```
90  
<class 'int'>
```

```
x = 25  
y = 87  
z = 5*x - 2*y  
print(z)  
t = z/7  
print(t)  
z = z/14  
print(z)
```

```
-49  
-7.0  
-3.5
```

```
x, y, z = 8, 4, 2 # the values of x, y, and z can be written in one line.  
print(x, y, z)  
print(x)  
print(y)  
print(z)  
print(x/y)  
print(x/z)  
print(y/z)  
print(x+y+z)  
print(x*y*z)  
print(x-y-z)  
print(x/y/z)  
print(x//y//z)  
print(x%y%z)
```

```
8 4 2  
8  
4  
2  
2.0  
4.0  
2.0  
14  
64  
2  
1.0  
1  
0
```