Built-In Basic Data Types:

Programming for Data Science with Python

Overview

Python: Everything is an Object

In Python, everything is an object

- All values are objects
- Anything which can be used as a value (int, str, float, functions, etc.) are implemented as objects.

Built-In Basic Data Types

- NUMERIC:

  > - Integers: int
  > - Floating point numbers: float
  > - Complex numbers: complex

- LOGICAL/BOOLEAN: Boolean values: bool The focus: the integers and the floats

---

1. Numeric Data Types

1.1 Integers: int

****Run the following code:****

In [15]:
```
x = 3

y = x

print ("Data type of x: ", type(x), '\n')
print ("Data type of y: ", type(y), '\n')
```

```
Data type of x:  <class 'int'>

Data type of y:  <class 'int'>
```

In [16]:
```
a = 8

b = a
```

```
c = b

print ("Data type of a: ", type(a), '\n')
print ("Data type of b: ", type(b), '\n')
print ("Data type of c: ", type(c), '\n')
```

```
Data type of a:  <class 'int'>

Data type of b:  <class 'int'>

Data type of c:  <class 'int'>
```

1.2 Floating-Point Numbers: float

Run the following code:

In [17]:
```
PI = 3.14

y = PI

print ("Data type of PI: ", type(PI), '\n')
print ("Data type of y: ", type(y), '\n')
```

```
Data type of PI:  <class 'float'>

Data type of y:  <class 'float'>
```

1.3 Complex Numbers: complex

A complex number is represented by "x + yj".

- Python converts the real numbers x and y into complex using the function complex (x,y).
- The real part can be accessed using the function real() and imaginary part can be represented by imag(). ### **Run the following code:**

In [18]:
```
x = 5

y = 3

aComplex = complex(5,3)

print ("aComplex is a complex number: ", aComplex, '\n')
print ("Data type of aComplex: ", type(aComplex), '\n')
```

```
aComplex is a complex number:  (5+3j)

Data type of aComplex:  <class 'complex'>
```

In [19]:
```
a = 45

b = 39

aComplex = complex(45,39)
bComplex = complex(39, 45)
```

```
print ("aComplex is a complex number: ", aComplex, '\n')
print ("Data type of bComplex: ", type(aComplex), '\n')
print ("aComplex is a complex number: ", bComplex, '\n')
print ("Data type of bComplex: ", type(bComplex), '\n')
```

```
aComplex is a complex number:  (45+39j)

Data type of bComplex:  <class 'complex'>

aComplex is a complex number:  (39+45j)

Data type of bComplex:  <class 'complex'>
```

---

2. Logical Data Types/Boolean Values: bool

Run the following code:

In [20]:
```
boolVar = True
print ("boolVar is a boolean variable: ", boolVar, '\n')
print ("Data type of boolVar: ", type(boolVar), '\n')
```

```
boolVar is a boolean variable:  True

Data type of boolVar:  <class 'bool'>
```

In [21]:
```
boolVar = False
print ("boolVar is a boolean variable: ", boolVar, '\n')
print ("Data type of boolVar: ", type(boolVar), '\n')
```

```
boolVar is a boolean variable:  False

Data type of boolVar:  <class 'bool'>
```

IMPORTANT NOTES:

Any values that are ***NOT 0*** or null can be used the "True" Boolean value in Python.

Run the following 2 code blocks:

In [22]:
```
boolVar = 5

if (boolVar):
    print ("Data type of boolVar: ", type(boolVar), '\n')
```

```
Data type of boolVar:  <class 'int'>
```

In [23]:
```
boolie = 45

if (boolie):
    print ("Data type of boolVar: ", type(boolVar), '\n')
```

```
Data type of boolVar:  <class 'int'>
```

In [24]:
```
boolVar = False
print ("boolVar is a boolean variable: ", boolVar, '\n')
print ("Data type of boolVar: ", type(boolVar), '\n')
```

```
boolVar is a boolean variable:  False

Data type of boolVar:  <class 'bool'>
```

In [25]:
```
boolVar = True
print ("boolVar is a boolean variable: ", boolVar, '\n')
print ("Data type of boolVar: ", type(boolVar), '\n')
```

```
boolVar is a boolean variable:  True

Data type of boolVar:  <class 'bool'>
```

IMPORTANT NOTES:

Any zero values like ***0 or null*** can be used the "False" Boolean value in Python.

In [26]:
```
boolVar=0

if (boolVar):
    print ("Data type of boolVar: ", type(boolVar), '\n')
```

IMPORTANT NOTES:

In the above code, the value 0 can be used as "False." Therefore, nothing is printed out when the code in the above cell is executed.

3. Character Data Types

NOTES about character data types

- Python does not support character data type (char).
- It supports string and the characters as string of length one.

Run the following code:

In [27]:
```
aChar = 'a'
print ("aChar is a String variable, NOT a Character variable: ", aChar, "\n")
print ("ata type of aChar:",type(aChar),'\n')
```

```
aChar is a String variable, NOT a Character variable:  a

ata type of aChar: <class 'str'>
```

In [28]:
```
zChar = 'z'
print ("aChar is a String variable, NOT a Character variable: ", zChar, "\n")
print ("ata type of aChar:",type(zChar),'\n')
```

```
aChar is a String variable, NOT a Character variable:  z

ata type of aChar: <class 'str'>
```