Python for Scientific Computing

Venki Uddameri, Ph.D. P.E.

Python Data Types

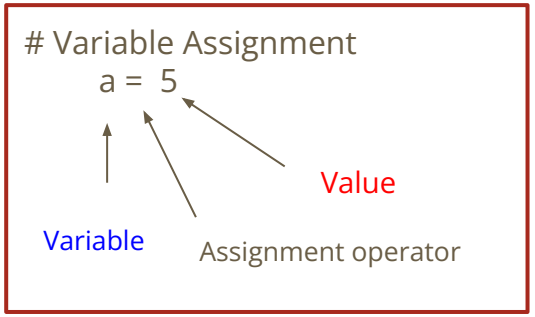
- What are data types?
- Why do we need them?
- Which data types are in-built in Python
- What are some non-primitive data types
- Some Python syntax

Understanding Data Types and Data Structures helps you use Python better

Variables and Objects

- Computer programming is all about data manipulation
  - We read in data, we can create data, we visualize data and store data

- Data in Python are stored in variables
  - We create a variable and assign value(s) to it

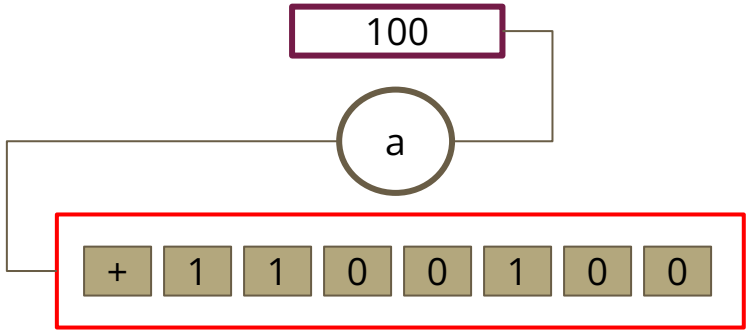


- A variable then serves many purposes
  - Allows us manipulate the data it is assigned
  - Handles how the data are handled or stored in the computer
  - Acts as an interface between what we want to do and where the data is stored on the computer

In Python Variables are also objects defined using a class (blueprint)

What are Data Types

- Data types are ways in which data used in Python is classified or categorized

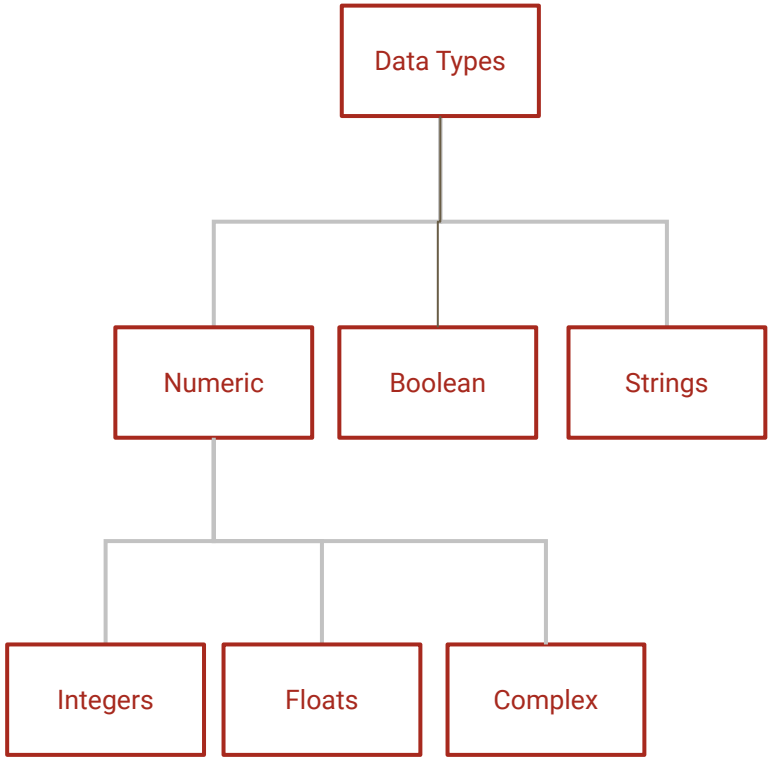


- Data types tell us:
  - They tell you what type of data is stored in the variable
  - How much memory is taken up by the variable
  - What types of calculations are possible with the variable

Data Types represent the most basic information related to the variable

Primitive Data Types

- Python offers 5 Primitive Data types
  - Numeric Values
  - Boolean Values
  - Text (strings)
- For numeric values we have 3 types
  - Integer - Used to stored signed and unsigned numbers
  - Float - Used to store decimal numbers typically accurate to 16 decimals
  - Complex - Used to store complex numbers
- For Boolean we use
  - True and False
    - Notice the capitalization
- Strings are used to store sequence of characters
  - Strings are defined using single or double-quotes
    - Do not mix and match
  - Strings are considered data type in Python as it does not have separate character data type



Examples of Data Types

- You can use type function to know the data type of the variable
- Python is object-oriented so every variable is also an object

```
# Defining Variables with Basic Data Types
# Numeric data
a = 65000000000000000000
b = 3.414
c = 2 + 3j
print(type(a))
print(type(b))
print(type(c))
```

```
<class 'int'>
<class 'float'>
<class 'complex'>
```

An unique feature of python is that it can store very very long integers extremely accurately.

Note Python uses ‘j’ instead of ‘i’ for imaginary numbers

Data Types Examples

```
# Boolean and String Data Types
x = True
y = 'I love Python'
print(type(x))
print(type(y))
```

```
<class 'bool'>
<class 'str'>
```

Do not put boolean value in quotes

Notice the capitalization

- int, float, complex and bool store only 1 number or value
- ‘str’ can store sequence of text. All text between quotes is treated as one value

Other Data Types

- Primitive Data Types are built-into Python
  - You have them whenever you need
- There are however many other Non-Primitive Data Types
  - These are specific to external libraries
  - You need to load the library before you can use them
- Date is one example from datetime module
    - Stores date in year-month-day format
- Some primitive types are modified to create new Data Types
  - Float64 in Numpy module uses 64 bits (instead of 32 to store decimal numbers)

Non-Primitive Data Types Examples

```
import datetime as dt
t = dt.date(2024,5,31)
print(t)
print(type(t))
```

```
2024-05-31
<class 'datetime.date'>
```

```
import numpy as np
f16 = np.float16(22/7)
f32 = np.float32(22/7)
f64 = np.float64(22/7)
print("The value of 22/7 as float16: ",f16)
print("The value of 22/7 as float32: ",f32)
print("The value of 22/7 as float64: ",f64)
type(f32)
```

```
The value of 22/7 as float16:  3.143
The value of 22/7 as float32:  3.142857
The value of 22/7 as float64:  3.142857142857143
numpy.float32
```

Precision of floats is important for numerical accuracy and computational speed

Concepts Covered

- What are variables
  - Objects
- What are data types
  - What role do they play
- What the primitive data types in Python
  - Numeric - Int, float, complex
  - Boolean - True and False
  - String - str
- Python also offers many non-primitive data types
  - Written in different libraries and modules
  - You need to load the library to use these data types
- Non-primitive data types
  - Allow us to manipulate special forms of data (e.g., dates)
  - Control the precision of the variables (e.g., float16, float64)

Use type(var) to get the data type

Use # for comment lines

Use of print statement

Use of import statement