Data Types in Python

- A data type, in programming, is a classification that specifies which type of value a
  variable has and what type of mathematical, relational or logical operations can be
  applied.
- The data type defines which operations can safely be performed to create, transform
  and use the variable in another computation.
- Python variable assignment is different from some of the popular languages like c, c++
  and java.
- There is no declaration of a variable, just an assignment statement.
- When we declare a variable in C or alike languages, this sets aside an area of memory
  for holding values allowed by the data type of the variable.
- The memory allocated will be interpreted as the data type suggests.
- If it's an integer variable the memory allocated will be read as an integer and so on.
  When we assign or initialize it with some value, that value will get stored at that
  memory location.
- At compile time, initial value or assigned value will be checked.
- So we cannot mix types. Example: initializing a string value to an int variable is not
  allowed and the program will not compile.
- But Python is a dynamically typed language.
- It doesn't know about the type of the variable until the code is run.
- So declaration is of no use.
- What it does is, It stores that value at some memory location and then binds that
  variable name to that memory container.
- And makes the contents of the container accessible through that variable name.
- So the data type does not matter.
- As it will get to know the type of the value at run-time.

There are different data types in python as

- None
- Numeric -> int, float, bool, complex
- Sequence -> List, Tuple, Set, String, Range
- Dictionary

Consider below application which demonstrates concept of datatypes in Python

```
print("Demonstration of Data types")


no = None
print(no)
print(type(no))
print(id(no))


no = 11
print(no)
print(type(no))
print(id(no))


# Numeric datatype contains diffrent sub types as int,float, bool, complex
no = 11 # int
print(type(no))


no = 3.14 # float
print(type(no))


no = 6+3j # Complex
print(type(no))


no = True  # boolean
print(type(no))


# we can convert variable from one data type to another data type (typecasting)
no = 11
```

```
no = float(no)
print(no)

print(type(no))
print(id(no))

no = 3.14
no = int(no)
print(no)

print(type(no))
print(id(no))

a = 5
b = 6

no = complex(a,b)
print(no)

print(type(no))
print(id(no))

# Under Sequence there are different data types as List, Set, Tuple, Range

ListEx = [10,20,30,40]
print(ListEx)
print(type(ListEx))
```

```
SetEx = {10,20,30,40}
print(SetEx)
print(type(SetEx))

TupleEx = (10,20,30,40)
print(TupleEx)
print(type(TupleEx))
name = "Fork"
print(name)
print(type(name))

ex = list(range(1,10))
print(ex)
print(type(ex))

ex = list(range(10))
print(ex)
print(type(ex))

ex = list(range(1,20,2))
print(ex)
print(type(ex))

# Dictionary contains key and value
batches = {"PPA": "9000","LB":"7500","Python":"5000",
"Angular":"5000"}
print(batches)
print(type(batches))
print(batches.keys())
print(batches.values())
```

```
print(batches["Python"])
```