Python Data Types

- In programming, data type is an important concept.
- Variables can store data of different types, and different types can do different things.
- Python has the following data types built-in by default, in these categories:

Text Type	Str
Numeric Types	int, float, complex
Sequence Types	list, tuple
Mapping Type	dict
Set Types	set
Boolean Type	bool

Text

Strings

Strings are sequences of character data. The string type in Python is called str.

A string in Python can contain as many characters as you wish. The only limit is your machine's memory resources. A string can also be empty:

Numeric Types

Integer

Integers can be of any length, it is only limited by the memory available.

Float

A floating-point number is accurate up to 15 decimal places. Integer and floating points are separated by decimal points. 1 is an integer, 1.0 is a floating-point number.

Complex Numbers

Complex numbers are specified as <real part>+<imaginary part>j. For example:

```
>>>
>>> 2+3j
(2+3j)
>>> type(2+3j)
<class 'complex'>
```

Sequence Type

Python List

List is an ordered sequence of items. It is one of the most used data type in Python and is very flexible. All the items in a list do not need to be of the same type.

Declaring a list is pretty straight forward. Items separated by commas are enclosed within brackets [ ].

Ex:→ a = [1, 2.2, 'python']

We can use the slicing operator [ ] to extract an item or a range of items from a list. The index starts from 0 in Python.

Ex:→ a = [5, 10, 15, 20, 25, 30, 35, 40]

```
# a[2] = 15

print ("a[2] = ", a[2])

# a[0:3] = [5, 10, 15]

print("a[0:3] = ", a[0:3])

# a[5:] = [30, 35, 40]

print("a[5:] = ", a[5:])
```

Output

```
a[2] =  15

a[0:3] =  [5, 10, 15]

a[5:] =  [30, 35, 40]
```

Lists are mutable, meaning, the value of elements of a list can be altered.

```
a = [1, 2, 3]

a[2] = 4

print(a)
```

Output

```
[1, 2, 4]
```

Python Tuple

Tuple is an ordered sequence of items same as a list. The only difference is that tuples are immutable. Tuples once created cannot be modified.

Tuples are used to write-protect data and are usually faster than lists as they cannot change dynamically.

It is defined within parentheses () where items are separated by commas.

```
t = (5,'program', 1+3j)
```

We can use the slicing operator [] to extract items but we cannot change its value.

```
t = (5,'program', 1+3j)

# t[1] = 'program'

print("t[1] = ", t[1])


# t[0:3] = (5, 'program', (1+3j))

print("t[0:3] = ", t[0:3])

# Generates error

# Tuples are immutable

t[0] = 10
```

Output

```
t[1] =  program

t[0:3] =  (5, 'program', (1+3j))

Traceback (most recent call last):

  File "test.py", line 11, in <module>

    t[0] = 10

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

Mapping Type

Python Dictionary

Dictionary is an unordered collection of key-value pairs.

It is generally used when we have a huge amount of data. Dictionaries are optimized for retrieving data. We must know the key to retrieve the value.

In Python, dictionaries are defined within braces {} with each item being a pair in the form key:value. Key and value can be of any type.

```
>>> d = {1:'value','key':2}

>>> type(d)

<class 'dict'>
```

Set Types

Python Set

Set is an unordered collection of unique items. Set is defined by values separated by comma inside braces { }. Items in a set are not ordered.

```
a = {5,2,3,1,4}


# printing set variable

print("a = ", a)


# data type of variable a

print(type(a))
```

Output

```
a =  {1, 2, 3, 4, 5}

<class 'set'>
```

Boolean Type

Boolean

Python 3 provides a Boolean data type. Objects of Boolean type may have one of two values, True or False:

>>>

>>> type(True)

<class 'bool'>

>>> type(False)

<class 'bool'>