

# Understanding Python Datatypes

In computer programming, data types specify the type of data that can be stored inside a variable. For example,

```
num = 24
```

Here, **24** (an integer) is assigned to the num variable. So the data type of num is of the int class.

## Python Data Types

Data Types	Class	Description
numeric	int, float, complex	Holds numeric values
string	str	Holds sequence of characters
sequence	list, tuple ,range	Holds collection of items
mapping	dict	Holds data in key-value pair form
boolean	bool	Holds either <code>True</code> or <code>False</code>
set	set	Holds collection of unique items

Since everything is an object in Python programming, data types are actually classes and variables are instances(object) of these classes.

## Python Numeric Data type

In Python, the numeric data type is used to hold numeric values.

Integers, floating-point numbers and complex numbers fall under Python numbers category. They are defined as int, float and complex classes in Python.

- **int** - holds signed integers of non-limited length.
- **float** - holds floating decimal points and it's accurate up to **15** decimal places.

We can use the `type()` function to know which class a variable or a value belongs to.

Let's see an example,

```
num1 = 55
```

```
num2 = 5.3
print(num1)
print(num2)
```

## Python List Data Type

A list is an ordered collection of similar or different types of items separated by commas and enclosed within brackets [ ]. For example,

```
languages = ["Python", "Dart", "Web", 23]
print(languages)
```

Here, we have created a list named languages with **3** string values inside it.

## Access List Items

To access items from a list, we use the index number (**0, 1, 2 ...**). For example,

```
languages = ["Python", "Dart", "Web", 23]
print(languages[1])
```

In the above example, we have used the index values to access items from the languages list.

- languages[0] - access the first item from languages i.e. Python
- languages[2] - access the third item from languages i.e. Web

## Python Tuple Data Type

A 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.

In Python, we use the parentheses () to store items of a tuple. For example,

```
products = ('XBox', 499.99, "Habibi", 23)
print(products)
```

Here, product is a tuple with a string value Xbox and integer value **499.99**.

## Access Tuple Items

Similar to lists, we use the index number to access tuple items in Python. For example,

```
products = ('XBox', 499.99, "Habibi", 23)
print(products[2])
```

## Python String Data Type

String is a sequence of characters represented by either single or double quotes. For example,

```
site_name = "Power Learn Project"
print(site_name)
```

In the above example, we have created string-type variables: name and message with values 'Python' and 'Python for beginners' respectively.

## Python Set Data Type

The Set is an unordered collection of unique items. Set is defined by values separated by commas inside braces { }. For example,

```
student_ids = {112, 114, 117, 113}
print(student_ids)
```

Here, we have created a set named student\_info with 5 integer values.

Since sets are unordered collections, indexing has no meaning. Hence, the slicing operator [] does not work.

## Python Dictionary Data Type

Python dictionary is an ordered collection of items. It stores elements in key/value pairs.

Here, keys are unique identifiers that are associated with each value.

Let's see an example,

```
capital_city = {"Kenya": "Nairobi", "Nigeria": "Lagos"}
```

```
print(capital_city)
```

**More Resources:**

1. <https://www.geeksforgeeks.org/python-data-types/>
2. <https://realpython.com/python-data-types/>
3. <https://www.digitalocean.com/community/tutorials/python-data-types>
4. <https://jakevdp.github.io/PythonDataScienceHandbook/02.01-understanding-dat>