

Chapter 2: Python Data Types

Data types are nothing but variables you use to reserve some space in memory. Python variables do not need an explicit declaration to reserve memory space. The declaration happens automatically when you assign a value to a variable.

Section 2.1: String Data Type

String are identified as a contiguous set of characters represented in the quotation marks. Python allows for either pairs of single or double quotes. Strings are immutable sequence data type, i.e each time one makes any changes to a string, completely new string object is created.

```
a_str = 'Hello World'
print(a_str)      #output will be whole string. Hello World
print(a_str[0])   #output will be first character. H
print(a_str[0:5]) #output will be first five characters. Hello
```

Section 2.2: Set Data Types

Sets are unordered collections of unique objects, there are two types of set:

1. Sets - They are mutable and new elements can be added once sets are defined

```
basket = {'apple', 'orange', 'apple', 'pear', 'orange', 'banana'}
print(basket)           # duplicates will be removed
> {'orange', 'banana', 'pear', 'apple'}
a = set('abracadabra')
print(a)                # unique letters in a
> {'a', 'r', 'b', 'c', 'd'}
a.add('z')
print(a)
> {'a', 'c', 'r', 'b', 'z', 'd'}
```

2. Frozen Sets - They are immutable and new elements cannot added after its defined.

```
b = frozenset('asdfagsa')
print(b)
> frozenset({'f', 'g', 'd', 'a', 's'})
cities = frozenset(["Frankfurt", "Basel","Freiburg"])
print(cities)
> frozenset({'Frankfurt', 'Basel', 'Freiburg'})
```

Section 2.3: Numbers data type

Numbers have four types in Python. Int, float, complex, and long.

```
int_num = 10      #int value
float_num = 10.2   #float value
complex_num = 3.14j #complex value
long_num = 1234567L #long value
```

Section 2.4: List Data Type

A list contains items separated by commas and enclosed within square brackets []. Lists are almost similar to arrays in C. One difference is that all the items belonging to a list can be of different data type.

```
list = [123, 'abcd', 10.2, 'd']    #can be an array of any data type or single data type.
list1 = ['hello', 'world']
print(list)    #will output whole list. [123, 'abcd', 10.2, 'd']
print(list[0:2])    #will output first two element of list. [123, 'abcd']
print(list1 * 2)    #will gave list1 two times. ['hello', 'world', 'hello', 'world']
print(list + list1)    #will gave concatenation of both the lists.
[123, 'abcd', 10.2, 'd', 'hello', 'world']
```

Section 2.5: Dictionary Data Type

Dictionary consists of key-value pairs. It is enclosed by curly braces {} and values can be assigned and accessed using square brackets [].

```
dic={'name':'red', 'age':10}
print(dic)    #will output all the key-value pairs. {'name':'red', 'age':10}
print(dic['name'])    #will output only value with 'name' key. 'red'
print(dic.values())    #will output list of values in dic. ['red', 10]
print(dic.keys())    #will output list of keys. ['name', 'age']
```

Section 2.6: Tuple Data Type

Lists are enclosed in brackets [] and their elements and size can be changed, while tuples are enclosed in parentheses () and cannot be updated. Tuples are immutable.

```
tuple = (123, 'hello')
tuple1 = ('world')
print(tuple)    #will output whole tuple. (123, 'hello')
print(tuple[0])    #will output first value. (123)
print(tuple + tuple1)    #will output (123, 'hello', 'world')
tuple[1]='update'    #this will give you error.
```