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

In [1]:
```
a_str = 'Hello World'
print(a_str)
```

```
Hello World
```

In [2]:
```
print(a_str[0])
```

```
H
```

In [3]:
```
print(a_str[0:5])
```

```
Hello
```

Section 2.2 Set Data Types

Sets are unorderd collection of unique objects, there are two types of set:

```
1. Sets - They are mutable and new elements can be added once sets are d
efined
```

In [4]:
```
basket = {'apple', 'orange', 'apple', 'pear', 'orange', 'banana'}
print(basket)
```

```
{'pear', 'banana', 'orange', 'apple'}
```

In [5]:
```
a = set('abracadabra')
print(a)
```

```
{'c', 'r', 'b', 'a', 'd'}
```

In [6]:
```
a.add('z')
print(a)
```

```
{'c', 'r', 'z', 'b', 'a', 'd'}
```

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

In [7]:
```
b = frozenset('asdfagsa')
print(b)
```

```
frozenset({'g', 's', 'a', 'd', 'f'})
```

In [8]:
```
cities = frozenset(["Frankfurt", "Basel", "Freiburg"])
print(cities)
```

```
frozenset({'Freiburg', 'Basel', 'Frankfurt'})
```

Section 2.3: Numbers data type

Numbers have four types in python. int, float, complex, and long.

In [9]:
```
int_num = 10
float_num = 10.2
complex_num = 3.14j
long_num = 1234567
print(type(int_num))
print(type(float_num))
print(type(complex_num))
print(type(long_num))
```

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

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.

In [10]:
```
list = [123, 'abcd', 10.2, 'd']
list1 = ['hello', 'world']
print(list)
print(list[0:2])
print(list1 * 2)
print(list + list1)
```

```
[123, 'abcd', 10.2, 'd']
[123, 'abcd']
['hello', 'world', 'hello', 'world']
[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 accessed using square brackets[].

In [11]:
```
dic = {'name': 'red', 'age': 10}
print(dic)
print(dic['name'])
print(dic.values())
print(dic.keys())
```

```
{'name': 'red', 'age': 10}
red
dict_values(['red', 10])
dict_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 parenthesis() and cannot be updated. Tuples are immutable.

In [12]:
```
tuple = (123, 'hello')
tuple1 = ('world')
print(tuple)
print(tuple[0])
print(tuple, tuple1)
#tuple[1] = 'update' # error tuples are immutable
```

```
(123, 'hello')
123
(123, 'hello') world
```