Day - 2 Task

There are 4 built-in data structures in python - list, tuple, dictionary and set.

List- A list is a data structure that holds an ordered collection of items i.e. you can store a sequence of items in a list. A list is a mutable data type i.e. this type can be altered.

```
In [2]: my_list = ['apple','mango','banana']
```

```
In [3]: print(my_list)
```

```
['apple', 'mango', 'banana']
```

```
In [4]: my_list.append('orange')
```

```
In [5]: print(my_list)
```

```
['apple', 'mango', 'banana', 'orange']
```

```
In [6]: my_list.sort()
```

```
In [7]: print(my_list)
```

```
['apple', 'banana', 'mango', 'orange']
```

```
In [8]: del my_list[0]
```

```
In [9]: print(my_list)
```

```
['banana', 'mango', 'orange']
```

Tuple - Tuples are used to hold together multiple objects. Tuples are immutable i.e. we cannot modify tuples.

```
In [10]: animals = ('Tiger','Lion','elephant')
```

```
In [11]: print(animals)
```

```
('Tiger', 'Lion', 'elephant')
```

```
In [12]: animals.append('monkey')
```

```
---------------------------------------------------------------------------
--
AttributeError                            Traceback (most recent call las
t)
<ipython-input-12-73c020bea84b> in <module>()
----> 1 animals.append('monkey')

AttributeError: 'tuple' object has no attribute 'append'
```

Append Function doesn't work in tuple.

```
In [13]: new_animals = ('monkey','giraffe',animals)
In [14]: print(new_animals)

         ('monkey', 'giraffe', ('Tiger', 'Lion', 'elephant'))
```

```
In [15]: new_animals[2]

Out[15]: ('Tiger', 'Lion', 'elephant')
```

```
In [17]: new_animals[2][1]

Out[17]: 'Lion'
```

Dictionary - A dictionary is like an address-book where you can find the address or contact details of a person by knowing only his/her name. It has a 'key' with a 'value'. Key must be unique, immutable whereas Value is mutable.

```
In [29]: email={
             'Utkarsh':'utkarshgpt47@gmail.com',
             'Abhinav':'abhigarg2018@gmail.com',
             'Kushagra':'kushrajora88@gmail.com',
             'spam':'spam@spam.com'
         }
```

```
In [30]: email['Utkarsh']

Out[30]: 'utkarshgpt47@gmail.com'
```

```
In [31]: email['Abhishek']='abhishek305a@gmail.com'
```

```
In [32]: print(email)

         {'Utkarsh': 'utkarshgpt47@gmail.com', 'Abhinav': 'abhigarg2018@gmail.com
         ', 'Kushagra': 'kushrajora88@gmail.com', 'spam': 'spam@spam.com', 'Abhish
         ek': 'abhishek305a@gmail.com'}
```

```
In [33]: del email['spam']
```

```
In [34]: print(email)

         {'Utkarsh': 'utkarshgpt47@gmail.com', 'Abhinav': 'abhigarg2018@gmail.com
         ', 'Kushagra': 'kushrajora88@gmail.com', 'Abhishek': 'abhishek305a@gmail.
         com'}
```

Set- Sets are unordered collections of simple objects. These are used when the existence of an object in a collection is more important than the order or how many times it occurs.

```
In [42]: countries = set(['India','USA','Russia','Israel'])
         bric = set(['Britain','Russia','India','China'])
```

```
In [43]: 'India' in countries

Out[43]: True
```

```
In [44]: 'Pakistan' in countries

Out[44]: False
```

```
In [45]: countries.add('China')
In [46]: print(countries)

         {'India', 'USA', 'Russia', 'China', 'Israel'}
```

```
In [47]: countries.add('Nepal')
```

```
In [48]: print(countries)

         {'India', 'USA', 'Nepal', 'Russia', 'China', 'Israel'}
```

```
In [49]: countries & bric

Out[49]: {'China', 'India', 'Russia'}
```

Sequence-sequence is the generic term for an ordered set. There are several types of sequences in Python, like- list, strings, tuple. Slicing, concatenation are some operations of sequence.

Data Types

There are several built-in data types in python like Numeric (which contains int, float, complex) String, Boolean, List, Tuple.

```
In [53]: a = 5
         print(a, "is of type", type(a))

         a = 2.0
         print(a, "is of type", type(a))

         a = 1+2j
         print(a, "is complex number?", isinstance(1+2j,complex))

         a=('Python is great')
         print(a,"is of type", type(a))

         5 is of type <class 'int'>
         2.0 is of type <class 'float'>
         (1+2j) is complex number? True
         Python is great is of type <class 'str'>
```

Function

Function blocks begin with the keyword def followed by the function name and parentheses (( )).

Syntax

```
def functionname( parameters ):
   "function_docstring"
   function_suite
   return [expression]
```

Lambda Function

Syntax

```
lambda arguments: expression
```

Example

```
In [55]: g = lambda x: x*x*x
         print(g(3))
```

```
27
```