

Second_Assignment

March 14, 2022

1 Python Chilla 10 hours video

1.1 Basics of Python

1.1.1 This file contains the basic concept of python by Ammar Bhai

1.1.2 1-indexing and Data Structure

```
[ ]: a = "Samosa Pakora"  
a
```

```
[ ]: 'Samosa Pakora'
```

```
[ ]: #length of indeces  
len(a)
```

```
[ ]: 13
```

```
[ ]: a[0]
```

```
[ ]: 'S'
```

```
[ ]: a[1]
```

```
[ ]: 'a'
```

```
[ ]: a[0:5]
```

```
[ ]: 'Samos'
```

```
[ ]: a[0:]
```

```
[ ]: 'Samosa Pakora'
```

```
[ ]: a[:-3]
```

```
[ ]: 'Samosa Pak'
```

```
[ ]: a[:]
```

```
[ ]: 'Samosa Pakora'
```

1.1.3 1a-String Method

```
[ ]: food ="Biryani"  
food
```

```
[ ]: 'Biryani'
```

```
[ ]: len(food)
```

```
[ ]: 7
```

```
[ ]: # Capitalize  
food.upper()
```

```
[ ]: 'BIRYANI'
```

```
[ ]: # Lower Case  
food.lower()
```

```
[ ]: 'biryani'
```

```
[ ]: # Replace  
food.replace("B","sh")
```

```
[ ]: 'shiryani'
```

```
[ ]: # Counting special character  
name ="Sartaj Ahmed is following the Chilla with Ammar Bhai"  
name
```

```
[ ]: 'Sartaj Ahmed is following the Chilla with Ammar Bhai'
```

```
[ ]: name.count("a")
```

```
[ ]: 5
```

1.1.4 1b-Finding an index in String

```
[ ]: name.find("w")
```

```
[ ]: 21
```

```
[ ]: # Splitting String  
food= "I love Samosa, Pakora, Biryani "  
food
```

```
[ ]: 'I love Samosa, Pakora, Biryani '
```

```
[ ]: food.split(',')
[ ]: ['I love Samosa', ' Pakora', ' Biryani ']
```

2 Basic Data Structure in Python

Tuple, List, Dictionaries, Set

1-Tuple

- Ordered collection of elements
- Enclosed in ()
- Different kind of elements can be stored
- Unmutable

```
[ ]: tup1 = (1,"python","True","10.2")
tup1
```

```
[ ]: (1, 'python', 'True', '10.2')
```

```
[ ]: print(type(tup1))
```

```
<class 'tuple'>
```

1a-Indexing in Tuple

```
[ ]: tup1[1]
```

```
[ ]: 'python'
```

```
[ ]: tup1[0:5]
```

```
[ ]: (1, 'python', 'True', '10.2')
```

```
[ ]: tup1[0:3]
```

```
[ ]: (1, 'python', 'True')
```

```
[ ]: len(tup1)
```

```
[ ]: 4
```

```
[ ]: #Concatination
tup2 =(2,"Baba",3.5,"False")
tup1+tup2
```

```
[ ]: (1, 'python', 'True', '10.2', 2, 'Baba', 3.5, 'False')
```

```
[ ]: tup1*2+tup2
```

```
[ ]: (1,  
      'python',  
      'True',  
      '10.2',  
      1,  
      'python',  
      'True',  
      '10.2',  
      2,  
      'Baba',  
      3.5,  
      'False')
```

```
[ ]: tup3 = (20,30,60,85)  
tup3
```

```
[ ]: (20, 30, 60, 85)
```

```
[ ]: min(tup3)
```

```
[ ]: 20
```

2.0.1 2-List

- Ordered collection of elements
- Enclosed in []
- Mutable

```
[ ]: lis = [1,"python","True","10.2"]  
lis
```

```
[ ]: [1, 'python', 'True', '10.2']
```

```
[ ]: type(lis)
```

```
[ ]: list
```

```
[ ]: len(lis)
```

```
[ ]: 4
```

```
[ ]: lis2 = [1,"python","True","10.2",12,13]  
lis2
```

```
[ ]: [1, 'python', 'True', '10.2', 12, 13]
```

```
[ ]: lis*2
```

```
[ ]: [1, 'python', 'True', '10.2', 1, 'python', 'True', '10.2']
```

```
[ ]: lis.append("Ahmed")  
lis
```

```
[ ]: ['Ahmed', '10.2', 'True', 'python', 1, 'Ahmed']
```

```
[ ]: lis.remove("Ahmed")
```

```
[ ]: lis
```

```
[ ]: ['Ahmed', '10.2', 'True', 'python', 1, 'Ahmed']
```

```
[ ]: lis.reverse()  
lis
```

```
[ ]: ['Ahmed', '10.2', 'True', 'python', 1, 'Ahmed']
```

```
[ ]: lis.count("Ahmed")
```

```
[ ]: 2
```

```
[ ]: lis.count("t")
```

```
[ ]: 0
```

```
[ ]: lis3 = [10,12,23,45,11,1,123,123]  
lis3
```

```
[ ]: [10, 12, 23, 45, 11, 1, 123, 123]
```

```
[ ]: lis3.sort()  
lis3
```

```
[ ]: [1, 10, 11, 12, 23, 45, 123, 123]
```

```
[ ]: lis2 + lis3
```

```
[ ]: [1, 'python', 'True', '10.2', 12, 13, 1, 10, 11, 12, 23, 45, 123, 123]
```

2.0.2 2-Dictionaries

- Unordered
- Enclosed in { }
- Keys and Values
- Mutable

```
[ ]: # Food and their prices  
D1 ={"Samosa":30,"Pakora":10,"Raita":5,"Chicken Roll":30}
```

```
D1
```

```
[ ]: {'Samosa': 30, 'Pakora': 10, 'Raita': 5, 'Chicken Roll': 30}
```

```
[ ]: type(D1)
```

```
[ ]: dict
```

```
[ ]: #Extract Data  
D1.keys()
```

```
[ ]: dict_keys(['Samosa', 'Pakora', 'Raita', 'Chicken Roll'])
```

```
[ ]: D1.values()
```

```
[ ]: dict_values([30, 10, 5, 30])
```

```
[ ]: D1["Kachori"] = 15
```

```
[ ]: D1
```

```
[ ]: {'Samosa': 30, 'Pakora': 10, 'Raita': 5, 'Chicken Roll': 30, 'Kachori': 15}
```

```
[ ]: D1["Raita"] = 8
```

```
[ ]: D1
```

```
[ ]: {'Samosa': 30, 'Pakora': 10, 'Raita': 8, 'Chicken Roll': 30, 'Kachori': 15}
```

```
[ ]: D2 = {"Dates": 50, "Chocolates": 200, "Sawyan": 1000}  
D2
```

```
[ ]: {'Dates': 50, 'Chocolates': 200, 'Sawyan': 1000}
```

```
[ ]: D1.update(D2)
```

```
[ ]: D1
```

```
[ ]: {'Samosa': 30,  
      'Pakora': 10,  
      'Raita': 8,  
      'Chicken Roll': 30,  
      'Kachori': 15,  
      'Dates': 50,  
      'Chocolates': 200,  
      'Sawyan': 1000}
```

2.0.3 4-Sets

- Unordered Elements
- Enclosed in { }
- No Duplicates are allowed

```
[ ]: s1 = {1,2.3,5.2,"Ammar","Codianics"}  
s1
```

```
[ ]: {1, 2.3, 5.2, 'Ammar', 'Codianics'}
```

```
[ ]: s1.add("Sartaj")
```

```
[ ]: s1
```

```
[ ]: {1, 2.3, 5.2, 'Ammar', 'Codianics', 'Sartaj'}
```

```
[ ]: s1.remove(1)
```

```
[ ]: s1
```

```
[ ]: {2.3, 5.2, 'Ammar', 'Codianics', 'Sartaj'}
```

```
[ ]: s2=[1,22,33,4,55,5,2]  
s2.sort()  
s2
```

```
[ ]: [1, 2, 4, 5, 22, 33, 55]
```

```
[ ]: s1.pop()  
print(s1)
```

```
{'Sartaj', 5.2, 'Ammar', 'Codianics'}
```

```
[ ]: thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]  
print(thislist[2:5])
```

```
['cherry', 'orange', 'kiwi']
```

```
[ ]: thislist[1] = "blackcurrant"  
print(thislist)
```

```
['apple', 'blackcurrant', 'cherry', 'orange', 'kiwi', 'melon', 'mango']
```

```
[ ]: del thislist[0]  
print(thislist)
```

```
['cherry', 'orange', 'kiwi', 'melon', 'mango']
```