-Indexing

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

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

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

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

```
In [ ]: #Length of Indexes
        len(a)
```

```
Out[ ]: 13
```

```
In [ ]: # Last index is exclusive
        a[0:5]
```

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

```
In [ ]: a[-1]
```

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

String Methods

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

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

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

```
Out[ ]: 7
```

```
In [ ]: food.upper()
```

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

```
In [ ]: food.lower()
```

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

```
In [ ]: food.capitalize()
```

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

```
In [ ]: food.replace('B','sh')
```

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

In [ ]:
```python
# counting a specific alphabelt in a string
name = "Dr Ammar Tufail"
name
```

Out[ ]:
```
'Dr Ammar Tufail'
```

In [ ]:
```python
name.count("A")
```

Out[ ]:
```
1
```

In [ ]:
```python
name.find("T")
```

Out[ ]:
```
9
```

In [ ]:
```python
food= "I am living in nespak, defence road, lahore, pakistan"
```

In [ ]:
```python
food.split(",")
```

Out[ ]:
```
['I am living in nespak', ' defence road', ' lahore', ' pakistan']
```

Basic Data Structure in Python

1- Tuple

2- List

3- Dictionaries

4- Sets

1-Tuple

- Ordered collection of elements
- Enclosed in () round braces/ parenthesis
- Different kind of elements can be stored
- Once elements are stored you can not change them (unmutatable)

In [ ]:
```python
tup1 = (1, "python", True, 2.5)
tup1
```

Out[ ]:
```
(1, 'python', True, 2.5)
```

In [ ]:
```python
#Type of tuple
type(tup1)
```

Out[ ]:
```
tuple
```

```
In [ ]: tup1[1:3]

Out[ ]: ('python', True)

In [ ]: tup1[1]

Out[ ]: 'python'

In [ ]: #count element in tuple
        len(tup1)

Out[ ]: 4

In [ ]: tup2 = (2, "Dr Ammar", 3.5, False)
        tup2

Out[ ]: (2, 'Dr Ammar', 3.5, False)

In [ ]: tup=tup1 +tup2
        tup

Out[ ]: (1, 'python', True, 2.5, 2, 'Dr Ammar', 3.5, False)

In [ ]: # Concatenation
        # Add more tuples
        tup1*2 + tup2

Out[ ]: (1, 'python', True, 2.5, 1, 'python', True, 2.5, 2, 'Dr Ammar', 3.5, False)

In [ ]: tup3 = (20 , 30 , 40 , 50 , 60)
        tup3

Out[ ]: (20, 30, 40, 50, 60)

In [ ]: min(tup3)

Out[ ]: 20

In [ ]: max(tup3)

Out[ ]: 60

In [ ]: tup3*3

Out[ ]: (20, 30, 40, 50, 60, 20, 30, 40, 50, 60, 20, 30, 40, 50, 60)

In [ ]: tup3[1]

Out[ ]: 30

In [ ]: tup3[-1]

Out[ ]: 60
```

In [ ]:

```
fruits = ("apple", "banana", "cherry")

(green, yellow, red) = fruits

print(green)
print(yellow)
print(red)
```

```
apple
banana
cherry
```

In [ ]:

```
for i in range(len(fruits)):
    print(fruits[i])
```

```
apple
banana
cherry
```

In [ ]:

```
fruits.count('apple')
```

Out[ ]:

```
1
```

In [ ]:

```
fruits.index('cherry')
```

Out[ ]:

```
2
```

2- List

- ordered collection of elements
- enclosed in [] square braces/brackets
- Mutateable, you can change values

In [ ]:

```
list1 = [2, "Dr Ammar", False]
list1
```

Out[ ]:

```
[2, 'Dr Ammar', False]
```

In [ ]:

```
type(list1)
```

Out[ ]:

```
list
```

In [ ]:

```
len(list1)
```

Out[ ]:

```
3
```

In [ ]:

```
list1[2]
```

Out[ ]:

```
False
```

In [ ]:

```
list2 = [3, 5, "Aammar", "Codanics", 478, False]
list2
```

Out[ ]:

```
[3, 5, 'Aammar', 'Codanics', 478, False]
```

In [ ]:

```
list1+ list2
```

Out[ ]:

```
[2, 'Dr Ammar', False, 3, 5, 'Aammar', 'Codanics', 478, False]
```

In [ ]:

```
list1.reverse()
list1
```

Out[ ]:

```
[False, 'Dr Ammar', 2]
```

In [ ]:

```
list1.append("codanics")
list1
```

Out[ ]:

```
[False, 'Dr Ammar', 2, 'codanics']
```

In [ ]:

```
list3 = [23,3, 43,65,7,8,43]
list3.sort()
list3
```

Out[ ]:

```
[3, 7, 8, 23, 43, 43, 65]
```

In [ ]:

```
fruits1 = ["apple", "banana", "mango", "cherry", "kiwi"]
fruits1
```

Out[ ]:

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

Appending values to another List

In [ ]:

```

```

In [ ]:

```
fruits2=[]
for x in fruits1:
    fruits2.append(x)

fruits2
```

Out[ ]:

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

In [ ]:

```
fruits3= [x for x in fruits if "a" in x]
fruits3
```

Out[ ]:

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

3- Dictionaries

- An unordered collection of elements
- key and value
- curly braces or brackets{}
- Mutateable/ Change the values

In [ ]:

```
# Food and their prices
food1 = {"samosa":30, "Raita":20, "Pakora":30, "Salad":50, "chicken rolls":30}
food1
```

Out[ ]:

```
{'samosa': 30, 'Raita': 20, 'Pakora': 30, 'Salad': 50, 'chicken rolls': 30}
```

In [ ]:

```
type(food1)
```

Out[ ]:

```
dict
```

In [ ]:

```
# Extract data
keys1= food1.keys()
keys1
```

Out[ ]:

```
dict_keys(['samosa', 'Raita', 'Pakora', 'Salad', 'chicken rolls'])
```

In [ ]:

```
values1 =food1.values()
values1
```

Out[ ]:

```
dict_values([30, 20, 30, 50, 30])
```

In [ ]:

```
food1["Tikki"]=10
food1
```

Out[ ]:

```
{'samosa': 30,
 'Raita': 20,
 'Pakora': 30,
 'Salad': 50,
 'chicken rolls': 30,
 'Tikki': 10}
```

In [ ]:

```
# update the values
food1["Tikki"]=15
food1
```

Out[ ]:

```
{'samosa': 30,
 'Raita': 20,
 'Pakora': 30,
 'Salad': 50,
 'chicken rolls': 30,
 'Tikki': 15}
```

In [ ]:

```
food2 = {"Dates":50, "Chocolates":200, "Sawayyan":1000}
food2
```

Out[ ]:

```
{'Dates': 50, 'Chocolates': 200, 'Sawayyan': 1000}
```

In [ ]:

```
#concatenate
food1.update(food2)
food1
```

Out[ ]:

```
{'samosa': 30,
 'Raita': 20,
 'Pakora': 30,
 'Salad': 50,
 'chicken rolls': 30,
 'Tikki': 15,
 'Dates': 50,
 'Chocolates': 200,
 'Sawayyan': 1000}
```

In [ ]:

```
dict1 = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
```

In [ ]:

```
print(dict1['brand'])
```

```
Ford
```

In [ ]:

```
dict1.keys()
```

Out[ ]:

```
dict_keys(['brand', 'model', 'year'])
```

In [ ]:

```
for x in dict1:
    print(dict1[x])
```

```
Ford
Mustang
1964
```

4-Set

- Unordered and unindexed
- Curly braces are used{}
- No duplicates allowed

In [ ]:

```
s1 = {1, 2.2, 5.2, "Aammar", "Codanics", "Lahore", True}
s1
```

Out[ ]:

```
{1, 2.2, 5.2, 'Aammar', 'Codanics', 'Lahore'}
```

In [ ]:

```
s1.add("Aammar")
```

In [ ]:

```
s1.difference
```

Out[ ]:

```
{1, 2.2, 5.2, 'Aammar', 'Codanics', 'Lahore'}
```

In [ ]:

```
set1 = {"a", "b" , "c"}
set2 = {1, 2, 3}

set3 = set1.union(set2)
print(set3)
```

```
{1, 2, 3, 'c', 'b', 'a'}
```

In [ ]:

```
set2.clear()
set2
```

Out[ ]:

```
set()
```

In [ ]:

```
set1
```

Out[ ]:

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

```
In [ ]:
```