-Indexing

In [1]:
```
# make a string
a = "Samosa Pakora"
a
```

Out[1]:
```
'Samosa Pakora'
```

In [2]:
```
a
```

Out[2]:
```
'Samosa Pakora'
```

In [3]:
```
# length of indeces
len(a)
```

Out[3]:
```
13
```

In [4]:
```
a[0]
```

Out[4]:
```
'S'
```

In [5]:
```
a[1]
```

Out[5]:
```
'a'
```

In [6]:
```
a[3]
```

Out[6]:
```
'o'
```

In [7]:
```
a[12]
```

Out[7]:
```
'a'
```

In [8]:
```
a[0:5]
```

Out[8]:
```
'Samos'
```

In [9]:
```
# exclusive element is the last one/ last index is exclusive
a[0:13]
```

Out[9]:
```
'Samosa Pakora'
```

In [10]:
```
a[-6:13]
```

Out[10]:
```
'Pakora'
```

In [11]:
```
food = "biryani"
food
```

Out[11]:
```
'biryani'
```

string methods

In [12]:
```
food
```

Out[12]:
```
'biryani'
```

In [13]:
```
len(food)
```

Out[13]:
```
7
```

In [14]:
```
# capitalize every element
food.capitalize()
```

Out[14]:
```
'Biryani'
```

In [15]:
```
# uppercase letters
food.upper()
```

Out[15]:
```
'BIRYANI'
```

In [16]:
```
# lowecase letters
food.lower()
```

Out[16]:
```
'biryani'
```

In [17]:
```
food.replace("b","sh")
```

Out[17]:
```
'shiryani'
```

In [18]:
```
# counting a specific alphabet in a string
name = "baba_aammar with Dr Aammar Tufail"
name
```

Out[18]:
```
'baba_aammar with Dr Aammar Tufail'
```

In [19]:
```
name.count("a")
```

Out[19]:
```
8
```

- finding an index number in string

In [20]:
```
name = "baba_aammar with Dr Aammar Tufail"
name
```

Out[20]:
```
'baba_aammar with Dr Aammar Tufail'
```

In [21]:
```
name.find("T")
```

Out[21]:
```
27
```

In [22]:
```
### - How to split a string
food ="I love samosa, pakora, raita, biryani and rarahi"
food
```

Out[22]:
```
'I love samosa, pakora, raita, biryani and rarahi'
```

In [23]:
```
food.split(",")
```

Out[23]:
```
['I love samosa', ' pakora', ' raita', ' biryani and rarahi']
```

Basic data structure in Python

1- Tules

2- List

3- Dictionaries

4- Set

1- Tuple

- Ordered collection of elements
- Enclosed in () round baraces/ prenthesis
- Different kind of elements cab be stored
- Once elements are stored you can not change then (unmutatable)

In [24]:
```
tup1 = (1, "Python", True, 2.5)
tup1
```

Out[24]:
```
(1, 'Python', True, 2.5)
```

In [25]:
```
#type of a tuple
type(tup1)
```

Out[25]:
```
tuple
```

- Indexing in tuple

In [26]:
```
tup1[1]
```

Out[26]:
```
'Python'
```

In [27]:
```
tup1[2]
```

Out[27]:
```
True
```

In [28]:
```
#last element is exclusive
tup1[0:3]
```

Out[28]:
```
(1, 'Python', True)
```

In [29]:
```
#count of elemets in tuple
len(tup1)
```

Out[29]:
```
4
```

In [30]:
```
tup2 = (2,"babaAammar", 3.5, False)
tup2
```

Out[30]:
```
(2, 'babaAammar', 3.5, False)
```

In [31]:
```
#concatinate (to add tuple or >2)
tup1 + tup2
```

Out[31]:
```
(1, 'Python', True, 2.5, 2, 'babaAammar', 3.5, False)
```

In [32]:
```
#concatinate  + repeat
tup1*2+tup2
```

Out[32]:
```
(1, 'Python', True, 2.5, 1, 'Python', True, 2.5, 2, 'babaAammar', 3.5, False)
```

In [33]:
```
tup3 = (20, 50, 30, 60, 79,85)
tup3
```

Out[33]:
```
(20, 50, 30, 60, 79, 85)
```

In [34]:
```
#minimum
min(tup3)
```

Out[34]:
```
20
```

In [35]:
```
#maximum
max(tup3)
```

Out[35]:
```
85
```

In [36]:
```
tup3*2
```

Out[36]:
```
(20, 50, 30, 60, 79, 85, 20, 50, 30, 60, 79, 85)
```

2- List

- ordered collection of elemets
- enclosed in [] square braces/ brackets
- mutateables, you can change the values

In [37]:
```
list1 = [2,"babaAammar", False]
list1
```

Out[37]:
```
[2, 'babaAammar', False]
```

In [38]:
```
type(list1)
```

Out[38]:
```
list
```

In [39]:
```
len(list1)
```

Out[39]:
```
3
```

In [40]:
```
list1[2]
```

Out[40]:
```
False
```

In [41]:
```
list2 = [3, 5, "Aammar", "Codanics ", 478, 53.2, False]
list2
```

Out[41]:
```
[3, 5, 'Aammar', 'Codanics ', 478, 53.2, False]
```

In [42]:
```
list1 + list2
```

Out[42]:
```
[2, 'babaAammar', False, 3, 5, 'Aammar', 'Codanics ', 478, 53.2, False]
```

In [43]:
```
list1*2
```

Out[43]:
```
[2, 'babaAammar', False, 2, 'babaAammar', False]
```

In [44]:
```
list1.reverse()
list1
```

Out[44]:
```
[False, 'babaAammar', 2]
```

In [45]:
```
list3 = [20,30,35,50,40,12,15,11,10,356,56,886]
list3
```

Out[45]:
```
[20, 30, 35, 50, 40, 12, 15, 11, 10, 356, 56, 886]
```

In [46]:
```
list1.append("Codanics youtube channel")
list1
```

Out[46]:
```
[False, 'babaAammar', 2, 'Codanics youtube channel']
```

In [47]:
```
len(list3)
```

Out[47]:
```
12
```

In [48]:
```
#sorting a list
list3.sort()
list3
```

Out[48]:
```
[10, 11, 12, 15, 20, 30, 35, 40, 50, 56, 356, 886]
```

In [49]:
```
list3*3
```

Out[49]:
```
[10,
 11,
 12,
 15,
 20,
 30,
 35,
 40,
 50,
 56,
 356,
 886,
 10,
 11,
 12,
 15,
 20,
 30,
 35,
 40,
 50,
 56,
 356,
 886,
```

```
 10,
 11,
 12,
 15,
 20,
 30,
 35,
 40,
 50,
 56,
 356,
 886]
```

In [50]:
```
lists= list1 + list2
lists
```

Out[50]:
```
[False,
 'babaAammar',
 2,
 'Codanics youtube channel',
 3,
 5,
 'Aammar',
 'Codanics ',
 478,
 53.2,
 False]
```

Functions of list

In [51]:
```
#list append
cities = [ 'Karachi', 'Islamabad', 'Gilgit']
cities.append("Lahore")
cities
```

Out[51]:
```
['Karachi', 'Islamabad', 'Gilgit', 'Lahore']
```

In [52]:
```
#list clear
cities = [ 'Karachi', 'Islamabad', 'Gilgit']
cities.clear()
cities
```

Out[52]:
```
[]
```

In [53]:
```
#list copy
cities = [ 'Karachi', 'Islamabad', 'Gilgit']

x = cities.copy()
x
```

Out[53]:
```
['Karachi', 'Islamabad', 'Gilgit']
```

In [54]:
```
#list count
cities = [ 'Karachi', 'Islamabad', 'Gilgit']
```

```
x = cities.count("Gilgit")
x
```

Out[54]: 1

In [55]:
```
#list extend
cities = [ "Karachi", "Islamabad", "Gilgit"]
countries = ["Pakistan ", "China", "Germany"]
cities.extend(countries)
cities
```

Out[55]: ['Karachi', 'Islamabad', 'Gilgit', 'Pakistan ', 'China', 'Germany']

In [56]:
```
#list index
cities = [ "Karachi", "Islamabad", "Gilgit"]
x = cities.index("Islamabad")
x
```

Out[56]: 1

In [57]:
```
#list insert
cities = [ "Karachi", "Islamabad", "Gilgit"]
cities.insert(1, "Lahore")
cities
```

Out[57]: ['Karachi', 'Lahore', 'Islamabad', 'Gilgit']

In [58]:
```
#list pop
cities = [ "Karachi", "Islamabad", "Gilgit"]
cities.pop(1)
cities
```

Out[58]: ['Karachi', 'Gilgit']

In [59]:
```
#list remove
cities = [ "Karachi", "Islamabad", "Gilgit"]
cities.remove("Islamabad")
cities
```

Out[59]: ['Karachi', 'Gilgit']

In [60]:
```
#list reverse
countries = ["Pakistan ", "China", "Germany"]
countries.reverse()
countries
```

Out[60]: ['Germany', 'China', 'Pakistan ']

In [61]:
```
#list sort
countries = ["Pakistan ", "China", "Germany"]
```

```
countries.sort()
countries
```

Out[61]: ['China', 'Germany', 'Pakistan ']

3- Dictionaries

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

In [62]:
```
#food and their prices
food1 = {"Samosa":30, "Pakora":20, "Salad":50, "Raita": 20, "Chicken rolls":30}
food1
```

Out[62]: {'Samosa': 30, 'Pakora': 20, 'Salad': 50, 'Raita': 20, 'Chicken rolls': 30}

In [63]:
```
type(food1)
```

Out[63]: dict

In [64]:
```
#extarct data
keys1 = food1.keys()
keys1
```

Out[64]: dict_keys(['Samosa', 'Pakora', 'Salad', 'Raita', 'Chicken rolls'])

In [65]:
```
values1 = food1.values()
values1
```

Out[65]: dict_values([30, 20, 50, 20, 30])

In [66]:
```
#adding new element
food1["Tikki"]=10
food1
```

Out[66]:
```
{'Samosa': 30,
 'Pakora': 20,
 'Salad': 50,
 'Raita': 20,
 'Chicken rolls': 30,
 'Tikki': 10}
```

In [67]:
```
# update the values
food1["Tikki"]=15
food1
```

Out[67]:
```
{'Samosa': 30,
```

```
 'Pakora': 20,
 'Salad': 50,
 'Raita': 20,
 'Chicken rolls': 30,
 'Tikki': 15}
```

In [68]:
```
food2 = {"Dates":50, "Choclates":200, "Swayyan": 1000}
food2
```

Out[68]:
```
{'Dates': 50, 'Choclates': 200, 'Swayyan': 1000}
```

In [69]:
```
#concatinate
food1.update(food2)
food1
```

Out[69]:
```
{'Samosa': 30,
 'Pakora': 20,
 'Salad': 50,
 'Raita': 20,
 'Chicken rolls': 30,
 'Tikki': 15,
 'Dates': 50,
 'Choclates': 200,
 'Swayyan': 1000}
```

Functions of Dictionaries

In [70]:
```
#Dictionary clear
cloths ={
    "brand": "Gussi",
    "Article": "Pajamas",
    "price": 1000
}
cloths.clear()
cloths
```

Out[70]:
```
{}
```

In [71]:
```
#Dictionary copy
phone ={
    "brand": "apple",
    "model": "13 pro max",
    "year": 2021
}
x = phone.copy()
x
```

Out[71]:
```
{'brand': 'apple', 'model': '13 pro max', 'year': 2021}
```

In [72]:
```
#Dictionary fromkeys
x = {"iPhone", "Samsung", "Huawei"}
y = 0
thisdict = dict.fromkeys(x,y)
thisdict
```

Out[72]:
```
{'iPhone': 0, 'Huawei': 0, 'Samsung': 0}
```

In [73]:
```
#Dictionary get
phone ={
    "brand": "apple",
    "model": "13 pro max",
    "year": 2021
}
x = phone.get("model")
x
```

Out[73]:
```
'13 pro max'
```

In [74]:
```
#Dictionary items
phone ={
    "brand": "apple",
    "model": "13 pro max",
    "year": 2021
}
x = phone.items()
x
```

Out[74]:
```
dict_items([('brand', 'apple'), ('model', '13 pro max'), ('year', 2021)])
```

In [75]:
```
#Dictionary keys
phone ={
    "brand": "apple",
    "model": "13 pro max",
    "year": 2021
}
x = phone.keys()
x
```

Out[75]:
```
dict_keys(['brand', 'model', 'year'])
```

In [76]:
```
#Dictionary pop
cloths ={
    "brand": "Gussi",
    "Article": "Pajamas",
    "price": 1000
}
cloths.pop("Article")
cloths
```

Out[76]:
```
{'brand': 'Gussi', 'price': 1000}
```

In [77]:
```
#Dictionary popitem
phone ={
    "brand": "apple",
    "model": "13 pro max",
    "year": 2021
}
```

```
phone.popitem()
phone
```

Out[77]:
```
{'brand': 'apple', 'model': '13 pro max'}
```

In [78]:
```
#Dictionary setdefault

car = {
  "brand": "Tesla",
  "model": " Model S",
  "year": 2012
}
x = car.setdefault("model", " Model S")
x
```

Out[78]:
```
' Model S'
```

In [79]:
```
#Dictionary update
car = {
  "brand": "Tesla",
  "model": " Model S",
  "year": 2012
}
car.update({"color": "Black"})
car
```

Out[79]:
```
{'brand': 'Tesla', 'model': ' Model S', 'year': 2012, 'color': 'Black'}
```

In [80]:
```
#Dictionary values
car = {
  "brand": "Tesla",
  "model": " Model S",
  "year": 2012
}
x = car.values()
x
```

Out[80]:
```
dict_values(['Tesla', ' Model S', 2012])
```

4- Set

- Unordered and unindexted collection of elements
- curly braces { }
- No duplicates allowed

In [81]:
```
s1 = {1,2.1, 5.2, "Hussain", "Python Ka Chilla", "Gilgit Baltistan", True}
s1
```

Out[81]:
```
{1, 2.1, 5.2, 'Gilgit Baltistan', 'Hussain', 'Python Ka Chilla'}
```

In [82]:

```
s1.add("Muhammad")
s1
```

Out[82]: {1, 2.1, 5.2, 'Gilgit Baltistan', 'Hussain', 'Muhammad', 'Python Ka Chilla'}

In [83]:
```
s1.remove("Hussain")
s1
```

Out[83]: {1, 2.1, 5.2, 'Gilgit Baltistan', 'Muhammad', 'Python Ka Chilla'}

Functions of set

In [84]:
```
#Add an element to the fruits set:
fruits ={"apple", "banana", "pears"}
fruits.add("oranges")
fruits
```

Out[84]: {'apple', 'banana', 'oranges', 'pears'}

In [85]:
```
#clear all elements from the fruits set:
fruits = {"apple", "banana", "cherry"}

fruits.clear()

print(fruits)
```

```
set()
```

In [86]:
```
#Copy the fruits set:
fruits = {"apple", "banana", "pears"}
x = fruits.copy()
x
```

Out[86]: {'apple', 'banana', 'pears'}

In [87]:
```
#difference

x = {"apple", "banana", "oranges"}
y = {"Python", "SPSS", "R"}

z = x.difference(y)
z
```

Out[87]: {'apple', 'banana', 'oranges'}

In [88]:
```
#difference update

x = {"apple", "banana", "oranges"}
y = {"Python", "SPSS", "R"}

x.difference_update(y)
x
```

Out[88]:
```
{'apple', 'banana', 'oranges'}
```

In [89]:
```
#discard Remove banana from the set

fruits = {"apple", "banana", "oranges"}
fruits.discard("banana")
fruits
```

Out[89]:
```
{'apple', 'oranges'}
```

In [90]:
```
#intersection
x = {"apple", "banana", "oranges"}
y = {"Python", "SPSS", "R"}

z = x.intersection(y)
z
```

Out[90]:
```
set()
```

In [91]:
```
#intersection_update

x = {"apple", "banana", "oranges"}
y = {"Python", "SPSS", "R"}

x.intersection_update(y)
x
```

Out[91]:
```
set()
```

In [92]:
```
#isdisjoint
x = {"apple", "banana", "oranges"}
y = {"Python", "SPSS", "R"}

z = x.isdisjoint(y)
z
```

Out[92]:
```
True
```

In [93]:
```
#issubset
x = {"Hamid", "Isha", "Rahim"}
y = {"a", "b", "c", "d", "e", "f", "g"}

z = x.issubset(y)
z
```

Out[93]:
```
False
```

In [94]:
```
#pop
fruits = {"apple", "banana", "oranges"}
```

```
fruits.pop
fruits
```

Out[94]:
```
{'apple', 'banana', 'oranges'}
```

In [95]:
```
#remove
fruits = {"apple", "banana", "oranges"}
fruits.remove("banana")
fruits
```

Out[95]:
```
{'apple', 'oranges'}
```

In [96]:
```
#symmetric_difference
x = {"apple", "banana", "oranges"}
y = {"Python", "SPSS", "R"}

z = x.symmetric_difference(y)
z
```

Out[96]:
```
{'Python', 'R', 'SPSS', 'apple', 'banana', 'oranges'}
```

In [97]:
```
#symmetric_difference_update
x = {"apple", "banana", "oranges"}
y = {"Python", "SPSS", "R"}

x.symmetric_difference_update(y)
x
```

Out[97]:
```
{'Python', 'R', 'SPSS', 'apple', 'banana', 'oranges'}
```

In [98]:
```
#union
x = {"apple", "banana", "oranges"}
y = {"Python", "SPSS", "R"}

z = x.union(y)
z
```

Out[98]:
```
{'Python', 'R', 'SPSS', 'apple', 'banana', 'oranges'}
```

In [99]:
```
#update
x = {"apple", "banana", "oranges"}
y = {"Python", "SPSS", "R"}

x.update(y)
x
```

Out[99]:
```
{'Python', 'R', 'SPSS', 'apple', 'banana', 'oranges'}
```