##INDEXING

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

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

```
In [2]: a

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

```
In [3]: a[0] #0 means first letter of the word i.e s

Out[3]: 'S'
```

```
In [4]: a[1]

Out[4]: 'a'
```

```
In [5]: a[2]

Out[5]: 'm'
```

```
In [6]: a[7]

Out[6]: 'P'
```

```
In [7]: #We can aslo calculate number of letters or length of indexes as in ait is 13
        len(a)

Out[7]: 13
```

```
In [8]: a[13] #we will get error as here counting starts from 0, if we say 0 it means fir

---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
C:\Users\FAIZAF~1\AppData\Local\Temp/ipykernel_7228/1040558728.py in <module>
----> 1 a[13] #we will get error as here counting starts from 0, if we say 0 it
means first word. Similarly, 12 means 13th word i.e a

IndexError: string index out of range
```

```
In [9]: a[12]

Out[9]: 'a'
```

```
In [10]: a[0:5] #the word for to in python is ration i.e : o to 5= 0:5

Out[10]: 'Samos'
```

In [11]:
```
a[0:6] # Although samosa comes under 5 when start from 0, The last index is exclu
```
Out[11]:
```
'Samosa'
```

In [12]:
```
a[-4] # when minus is added, it works in reverse. 4 word from opposite side is 'k
```
Out[12]:
```
'k'
```

In [13]:
```
a[-6:-1] #when range is considered always move from left to right a[-1:-6] isn't
```
Out[13]:
```
'Pakor'
```

In [14]:
```
a[-6:13] #to inclue last word from other side we can't write 0  but the  13, That
```
Out[14]:
```
'Pakora'
```

In [15]:
```
food="biryani"
food
```
Out[15]:
```
'biryani'
```

**##String Methods**

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

In [17]:
```
len(food)
```
Out[17]:
```
7
```

In [18]:
```
#CAPITALELLEMENT
food.capitalize() #after. press tab and click the option capital #fist letter of
```
Out[18]:
```
'Biryani'
```

In [19]:
```
#UPPERCASE
food.upper() #Use similar method and click upper, all the letter became in upperc
```
Out[19]:
```
'BIRYANI'
```

In [20]:
```
#LOWERCASE
food.lower()
```
Out[20]:
```
'biryani'
```

In [21]:
```
#REPLACE LETTERS
food.replace("b", "sh")
```
Out[21]:
```
'shiryani'
```

```
In [101]: food.casefold()
          food

Out[101]: 'I love samosa, pakora, raita and coke'
```

```
In [103]: food.center()
          food

          ---------------------------------------------------------------------------
          NameError                                 Traceback (most recent call last)
          C:\Users\FAIZAF~1\AppData\Local\Temp/ipykernel_7228/388722259.py in <module>
          ----> 1 food.center(pakora)
                2 food

          NameError: name 'pakora' is not defined
```

```
In [22]: #COUNTING A SPECIFIC ALPHABET IN STRING
         name= "Baba_Aamar_Tufail_Python"
         name

Out[22]: 'Baba_Aamar_Tufail_Python'
```

```
In [24]: name.count("a") # a is present 5 times, only small a

Out[24]: 5
```

```
In [25]: name.count("A")

Out[25]: 1
```

**#Finding an index number in string**

```
In [26]: name.find("T") #means at which number T comes i.e 11 #remember it starts from 0

Out[26]: 11
```

```
In [27]: name.find("B")

Out[27]: 0
```

**SPLIT A STRING

```
In [28]: food="I love samosa, pakora, raita and coke"
         food

Out[28]: 'I love samosa, pakora, raita and coke'
```

```
In [29]: food.split(",") # it means split string  based on comma. #So, different strings

Out[29]: ['I love samosa', ' pakora', ' raita and coke']
```

**DATA STRUCTURE**

1- Tuple 2- List 3- Dictionaries 4- Set

##1- TUPLE

-ordered collection of elements

-enclosed in()

-Different kind of elements can be stored

-Once elements are stored you can't change them(unmutable)

```
In [33]: tup1=(1, "python", True, 2.5) #True has T as capital
         tup1

Out[33]: (1, 'python', True, 2.5)
```

```
In [34]: #Type of a Tupl
         type(tup1)

Out[34]: tuple
```

###-Indexing in tuple

```
In [36]: tup1[1]

Out[36]: 'python'
```

```
In [38]: tup1[0]

Out[38]: 1
```

```
In [40]: tup1[0:3]

Out[40]: (1, 'python', True)
```

```
In [41]: len(tup1)

Out[41]: 4
```

```
In [45]: tup2=(2, "babaAamar", 3.7, False)
         tup2

Out[45]: (2, 'babaAamar', 3.7, False)
```

```
In [46]: #Concatinate which means to add tuples
         tup1+tup2

Out[46]: (1, 'python', True, 2.5, 2, 'babaAamar', 3.7, False)
```

```
In [47]: tup1*2 + tup2 #tup1 will be 2 times

Out[47]: (1, 'python', True, 2.5, 1, 'python', True, 2.5, 2, 'babaAamar', 3.7, False)
```

```
In [48]: tup3= [10,20,20,40,50]
         tup3

Out[48]: [10, 20, 20, 40, 50]
```

```
In [51]: min(tup3)

Out[51]: 10
```

```
In [52]: max(tup3)

Out[52]: 50
```

**##2- LIST**

- ordered collection of elements
- enclosed in []
- can be changesd-mutable

```
In [62]: list1 = [2, "babaAamar", False]
         list1

Out[62]: [2, 'babaAamar', False]
```

```
In [54]: type(list1)

Out[54]: list
```

```
In [55]: len(list1)

Out[55]: 3
```

```
In [56]: list2 = [3,5,"Aamar", "Sandhu", False, 4500, 35.8]
         list2

Out[56]: [3, 5, 'Aamar', 'Sandhu', False, 4500, 35.8]
```

```
In [57]: list1+list2

Out[57]: [2, 'babaAamar', False, 3, 5, 'Aamar', 'Sandhu', False, 4500, 35.8]
```

```
In [58]: list1*2

Out[58]: [2, 'babaAamar', False, 2, 'babaAamar', False]
```

```
In [61]: list1.reverse()
         list1                 #it got reverted in order

Out[61]: [False, 'babaAamar', 2]
```

```
In [63]: list1.pop()
         list1

Out[63]: [2, 'babaAamar']
```

```
In [64]: list1.append("Codanics Youtube") #Codanics yotube willb e added in the list 1
         list1

Out[64]: [2, 'babaAamar', 'Codanics Youtube']
```

```
In [66]: list1.count() #check on google that how it works
         list1

         ---------------------------------------------------------------------------
         TypeError                                 Traceback (most recent call last)
         C:\Users\FAIZAF~1\AppData\Local\Temp/ipykernel_7228/923197485.py in <module>
         ----> 1 list1.count()
               2 list1

         TypeError: list.count() takes exactly one argument (0 given)
```

```
In [70]: list3=[20,18,55,90.110,5,250]
         list3

Out[70]: [20, 18, 55, 90.11, 5, 250]
```

```
In [72]: list3.sort() #it will be arranged
         list3

Out[72]: [5, 18, 20, 55, 90.11, 250]
```

**##3_DICTIONARIES**

-An unordred collection of elements

-Key and Value

-Use curly brackets{}

-it is mutatable

```
In [73]: #Food and their prices
         food1={"samosa":30, "raita" :10, "pakora":100, "rolls": 50}
         food1

Out[73]: {'samosa': 30, 'raita': 10, 'pakora': 100, 'rolls': 50}
```

```
In [74]: type(food1) #it will show type of the data

Out[74]: dict

In [75]: #Extract data: we can separate key and values frm each other
         keys1= food1.keys()
         keys1

Out[75]: dict_keys(['samosa', 'raita', 'pakora', 'rolls'])

In [76]: values1= food1.values()
         values1

Out[76]: dict_values([30, 10, 100, 50])

In [78]: #if we want to add a new elemnt in the list of the data #Mutability i.e we can ch
         food1["Coke"]=90
         food1

Out[78]: {'samosa': 30, 'raita': 10, 'pakora': 100, 'rolls': 50, 'Coke': 90}

In [79]: #update the values, if roll becomes of 60
         food1["rolls"]=60  #LIST WILLBE UPDATED
         food1

Out[79]: {'samosa': 30, 'raita': 10, 'pakora': 100, 'rolls': 60, 'Coke': 90}

In [80]: food2= {"Dates":50, "choclates":40, "sawaiyan":500}
         food2

Out[80]: {'Dates': 50, 'choclates': 40, 'sawaiyan': 500}

In [82]: #Concatinate-means plus #different method of addition, add .after first food, the
         food1.update(food2)
         food1

Out[82]: {'samosa': 30,
          'raita': 10,
          'pakora': 100,
          'rolls': 60,
          'Coke': 90,
          'Dates': 50,
          'choclates': 40,
          'sawaiyan': 500}
```

In [95]:

```
food1.setdefault("2")
food1
```

Out[95]:

```
{'samosa': 30,
 'raita': 10,
 'pakora': 100,
 'rolls': 60,
 'Coke': 90,
 'Dates': 50,
 'choclates': 40,
 'sawaiyan': 500,
 '2': None}
```

In [99]:

```
food1.capitalize()
food1
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
C:\Users\FAIZAF~1\AppData\Local\Temp/ipykernel_7228/2050092856.py in <module>
----> 1 food1.capitalize(3)
      2 food1

AttributeError: 'dict' object has no attribute 'capitalize'
```

In [100]:

```
food1.fromkeys()
food1
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
C:\Users\FAIZAF~1\AppData\Local\Temp/ipykernel_7228/261988324.py in <module>
----> 1 food1.fromkeys()
      2 food1

TypeError: fromkeys expected at least 1 argument, got 0
```

**###SETS**

-Unordered collection of elements

-curly brackets used{}

-No duplicates allowed in it

-Boolean operators won't be printed e.g True, False etc

In [89]:

```
s1 = {1, 2.1, 5.2, "Sandhu", "Lahore", True}  #notice that True isn't printed
s1
```

Out[89]:

```
{1, 2.1, 5.2, 'Lahore', 'Sandhu'}
```

```
In [ ]: s1.add("Sandhu")
```

```
In [104]: s1
```

```
Out[104]: {' Sandhu', 1, 2.1, 5.2, 'Lahore', 'Sandhu'}
```

```
In [93]: s1.remove("Sandhu")
         s1
```

```
Out[93]: {' Sandhu', 1, 2.1, 5.2, 'Lahore'}
```