-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]
```

Out[3]: 's'

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

Out[4]: 's'

```
In [5]:  a[7]
```

Out[5]: 'P'

```
In [6]:  #length of indices
         len(a)
```

Out[6]: 13

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

Out[7]: 'a'

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

Out[8]: 'a'

```
In [9]:  #last index is exclusive
         a[0:6]
```

Out[9]: 'samosa'

```
In [10]:  a[0:6-7:12]
```

Out[10]: 's'

```
In [11]:
```

```
a[-6]
```

Out[11]: 'P'

In [12]:
```
a[-1:-6]
```

Out[12]: ''

In [13]:
```
a[-6:-2]
```

Out[13]: 'Pako'

In [14]:
```
a[-6:0]
```

Out[14]: ''

In [15]:
```
a[-6:1]
```

Out[15]: ''

In [16]:
```
a[-6:-1]
```

Out[16]: 'Pakor'

In [17]:
```
a[-6:13]
```

Out[17]: 'Pakora'

In [18]:
```
food = "Briyani"
food
```

Out[18]: 'Briyani'

In [19]:
```
len(food)
```

Out[19]: 7

In [20]:
```
food.upper()
```

Out[20]: 'BRIYANI'

In [21]:
```
food.lower()
```

Out[21]: 'briyani'

In [22]:
```
food.replace("Br", "Shr")
```

Out[22]: 'Shriyani'

In [23]:
```python
name = "baba_aamar with Dr Aamar tufail"
name
```

Out[23]: 'baba_aamar with Dr Aamar tufail'

In [24]:
```python
name.count("A")
```

Out[24]: 1

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

Out[25]: 2

-finding on index number in string

In [26]:
```python
name = "baba_aamar with Dr Aamar tufail"
name
```

Out[26]: 'baba_aamar with Dr Aamar tufail'

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

Out[27]: 5

In [28]:
```python
## -how to split a string
food = "I love samosa, pakora, raita, briyani and karahi"
```

In [29]:
```python
food.split()
```

Out[29]: ['I', 'love', 'samosa,', 'pakora,', 'raita,', 'briyani', 'and', 'karahi']

In [30]:
```python
food.splitlines()
```

Out[30]: ['I love samosa, pakora, raita, briyani and karahi']

In [31]:
```python
food.split(",")
```

Out[31]: ['I love samosa', ' pakora', ' raita', ' briyani and karahi']

Basic data Structure in Python

1-Tuples

2-List

3-Dictionaries

4-Set

1-Tuple

- ordered collection of elements
- enclosed in() round braces/paranthesis
- different kind of elements can be stored
- once elements are stored, cannot change them/unmutated

```
In [ ]:   Tupl =
```