

## Strings - String methods

Ex:-

a = "String 1"

b = "String 2"

c = "String 3"

d = "String1"

e = 'String1'

f = """String1"" "



print (a, "\n", "\n", b, c, d, e, f) ]

Output :-

String 1

String 2 String3 String1, String1

length of a string :-

a = "12345678"

length (a) ]

Output :- 8 ]

## Capital letters :-

\* a = "this is string 1111"

a.capitalize()

Output : 'This is string 1111'

\* a.upper()

Output : 'THIS IS STRING 1111'

## Count :-

\* a.count("i")

Output : 3

\* a.title()

Output : 'This Is String'

\* print('#'\*10)

Output : ##### ##### #####

index() :- Is a built-in function in Python, which searches for a given element from the start of the list & returns the lowest index where the element appears.

Ex:- `list1 = [1, 2, 3, 2, 4, 1, 1]`  
`print(list1.index(1))`

### Indexing & Slicing

Indexing :- Referring to an element of an iterable by its position

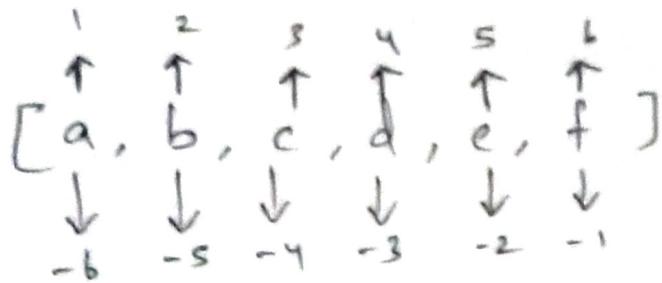
Slicing :- Getting a subset of elements from an iterable based

### Creation of list

`my_list = [- for - in 'abcdefghijkl']`

`my_list`

\* `my_list = [1, 2, 3, 4, 5]`  
`print(my_list[3])` ↴



### Slicing :-

my-list [start : stop]

~~start~~ :- is the index of the first element to include  
start :- is the index of the first element to include  
stop :- " " " " to item to stop

Ex:- my-list [1:5] ↴

[b, c, d, e]

0 1 2 3 4 5 6 7 8  
 [a, b, c, d, e, f, g, h, i]

\* my-list [5:] ↴

output: [f, g, h, i]

\* my-list [:4] ↴

[a, b, c, d]

- \* str. upper()
- str. capitalize()
- str. count()
- str. find("")

strip :-

- \* str1 = "hello string"

str1



output: 'hello string'

- \* str1. lstrip()

output: 'hellostring'

- \* str1. rstrip()

output: 'hellostring'

- \* str1 = str1. strip()

str1



output: 'hello string'

\* str = "Hello World"

str.replace = ("Hello", "Hi")

Output: "Hi World"

split :-

\* st. split()

['Hello', 'World']

\* str = "apple, mango, guava, banana"

str1. split( , )

Output: ['apple', 'mango', 'guava', 'banana']

Extended slicing:-

+ Reverse extended slicing offers to put a "step" field as  
[start:stop:step]

[ -1 : :-1 ]

↑      ↓  
start    end

Ex:-

\* `inputStr = 'Extended slice effect to put'`  
`print(inputStr[-1::-1])`

Output: tup of sreffe ecils ded netxE

\* `input_str = 'peter pipee picked a peck of pickles'`  
`input_str.split()`  
`' '.join([i[::-1] for i in input_`