

5. STRING MANIPULATION

It can be enclosed withing single quotation, double quotation, or triple quote

```
e.g.  'ipcsguide'
      "ipcsguide"
      """ipcsguide""" } 'ipcsguide'
```

Extract letter from String

```
Name = 'ipcsguide'
for x in Name:
    print(x , end=' ')          # i p c s g u i d e
```

Change the order of String (Reverse order)

```
x = input("Enter your name : ")    # abcd          d c b a
tot = len(x)
for x in range(-1, (-tot-1), -1):
    print(s1[x],end=" ")
```

Basic Operators in String

+ - concatenation of String (Addition two String)
* - Replication of String (Repetition of String)

```
"raj" + "kumar"          # 'rajkumar'
'ip' + "csguide"        # 'ipcsguide'
'2' + '2' + '2'         # '222'
'a' + 'b'               # 'ab'
'5' + 'xy'              # '5xy'
'2' + 3                 # Error
```

```
'how' * 3                # howhowhow
'how' * 3                # howhowhow
'2' * 3                  # 222
'5' * '4'                # Error
'5' + '4' * 3           # 5444
'*' * 4                  # ****d
```

Use of Membership operator in String

in & not in

```
s1 = 'education'
'h' in 'higher'         # True
'p' in 'higher'         # False
'a' in s1               # True
'b' in s1               # False
'cat' in s1             # True
'bat' in s1             # False
'D' in s1               # False

'u' not in 'higher'     # True
'g' not in 'higher'     # False
'cut' not in 'higher'   # True
'her' not in higher     # False
```

ASCII Value / Unicode Value

A – 65	a – 97	0 – 48
B – 66	b – 98	1 – 49
C – 67 and so on	c – 99	2 – 50
64 + character position	96 + character position	48 + number (0 - 9)
64 + 3 = 67 is for C	96 + 4 = 100 is for d	48 + 3 = 51 is for 2
'a' < 'A'	False	
'A' < 'D'	True	
'ABC' > 'AB'	True	
'abcd' > 'abcD'	True	

Find the ASCII Value for any character of vice versa

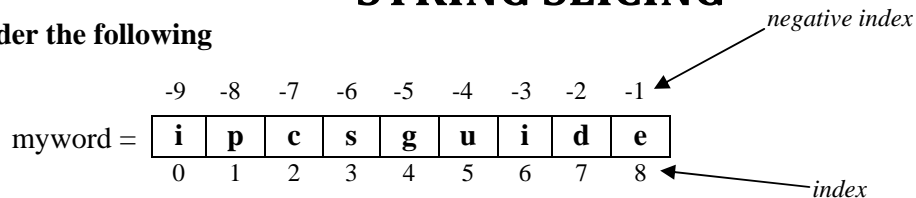
ord() operator – return the ASCII Code for any Character

chr() operator – Return the Character for any ASCII Value

ord('A')	# 65
ord('e')	# 101
ord('5')	# 53
chr(65)	# 'A'
chr(101)	# 'e'
chr(53)	# '5'

STRING SLICING

Consider the following



index - from left hand side
starts from 0 to size – 1

negative index - start right
from -1 to -(size)

myword	# ipcsguide
myword[0]	# i
myword[4]	# g
myword[-1]	# e
myword[8]	# p
myword[0]	# i
x= len(myword)	
x	# 9
myword[0:3]	# i p c
myword[4:7]	# g u i
myword[-7:-4]	# c s g
myword[:3]	# i p c (first 3 letters)
myword[6:]	# i d e (index 6 to last)
myword[:3] + myword[3:]	# ipcsguide

```
“ipcsguide”[3]      # s
“ipcsguide”[1]      # l
“ipcsguide”[-2]     # d
```

Every Second Word

```
"invitation"[1:8:2] # n i a i
"abcdefghij"[1:8:2] # b d f h
```

Every third letter

```
"abcdefghij"[1:10:3] # b e h
```

Every Second letter from right

```
"computer"[::-2]      # r t p o
```

Reverse a string

```
"computer"[::-2]      # r e t u p m o c
```

FUNCTIONS

a. capitalize()

Convert the first character of string into uppercase and all other into lower case

S1 = “ipcsguide is youtube CHANNEL”

```
S1.capitalize()      # 'Ipcsguide is youtube channel'
"computer".capitalize() # Computer
"god is great".capitalize() # God is great
"India IS GReat cOUNTry".capitalize() # India is great country
```

b. isalnum()

If the string is alphanumeric (alphabet or numbers), it will return True. String should have atleast one character.

```
"KVS123".isalnum()   # True
"KVS".isalnum()      # True
"123".isalnum()      # True
"123*".isalnum()     # False
" ".isalnum()        # False
"$".isalnum()        # False
```

c. isalpha()

If the string consists of alphabets, it will return True

```
"KVS123".isalpha()  # False
"KVS".isalpha()     # True
"123".isalpha()     # False
"123*".isalpha()    # False
" ".isalpha()       # False
"$".isalpha()       # False
```

d. isdigit()

If the string consists of numbers, it will return True

```
"KVS123".isdigit() # False
"KVS".isdigit()    # True
"123".isdigit()    # True
"123*".isdigit()   # False
" ".isdigit()      # False
"$".isdigit()      # False
```

e. islower()

If string consist has all characters in lower case, if character exists.

```
"KVS123".islower( )      # False
"KVS".islower( )         # False
"kvs".islower( )         # True
"kvs123".islower( )      # True
"kvs123*".islower( )     # True
"123".islower( )         # False
"123*".islower( )        # False
" ".islower( )           # False
"$".islower( )           # False
```

f. isupper()

If string consist has all characters in upper case, if character exists

```
"KVS123".isupper( )      # True
"KVS".isupper( )         # True
"kvs".isupper( )         # False
"kvs123".isupper( )      # False
"KVS123*".isupper( )     # True
"123".isupper( )         # False
"123*".isupper( )        # False
" ".isupper( )           # False
"$".isupper( )           # False
```

g. isspace()

If there is one or more space, returns true.

```
"".isspace( )            # False
" ".isspace( )           # False
"  ".isspace( )          # False
"my friend".isspace( )  # False
```

h. lower()

It return the string changed to lower case. Original string remains unchanged.

```
"IPCSGUIDE".lower( )    # ipcsguide
S1 = "IndiAn"
S1.lower( )              # indian
S1                        # IndiAn
```

i. upper()

It return the string changed to lower case. Original string remains unchanged.

```
"IpCsGuide".upper( )    # IPCSGUIDE
S1 = "IndiAn"
S1.uppper( )             # INDIAN
S1                        # IndiAn
```

- Q1. Write a program in python to print the number of upper case characters and lowers case characters present in a line of string
- Q2. Write a program in python to input a string and check whether it is a palindrome or not.
- Q3. Write a program in python to input a string and print the number of vowels present in that string.