10/10/22

Strings in Python

- A string is a collection of characters which are enclosed by single or double quotes.
  It can be a letter, a number, special character, white space or a backslash.

- Empty String → s = ""
  print (s)
  >>

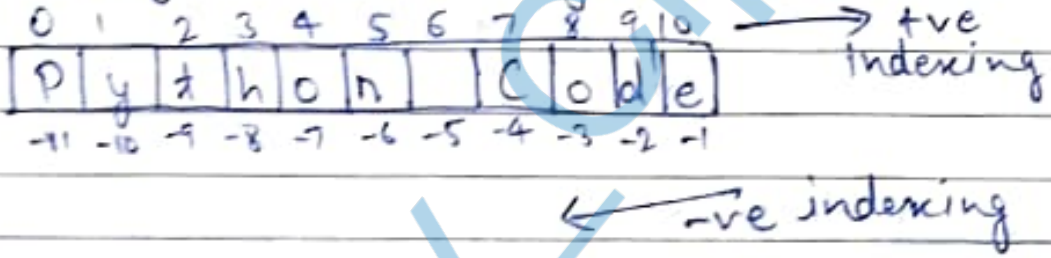
- Backslash (\) → i) s = 'Python \n code'
  print (s)
  >> Python
  >> code

  ii) s = 'Python \t code'
  print (s)
  >> Python code.

Indexing :-

The individual character of a string can be used by subscript value or index value, which is known as indexing. Each character has 2 index values.

e.g. → s = 'Python Code'



Traversing a String :-

It means accessing each character one after another by iterating through for or while loop.

for loop :-

```
s = "Alex"
for i in s:
    print (i)
```

O/P →
A
l
e
x

while loop:-

```
i = 0
s = "Alex"
while i < len(s):
    print(s[i])
    i = i + 1
```

- **len(s)** → length of characters / no. of characters in the string.

e.g. → `s = "Computer"`
`print(len(s))`

O/P = 8

Note:- len(s) always counts from 1 not from 0.

Q1) WAP to input a string and also input a character and count its occurance.

```
c = 0
s = input("Enter a string")
ch = input("Enter a character")
for i in s:
    if i == ch
        c = c + 1
print("Total no. of occurance of ", ch, "is", c)
```

Q2) WAP to input a string and display it in reverse order.

```
= s = input ("Enter a string")
  i = -1
  for i in range (-1, -len(s)-1, -1):
      print (s[i])
```

Special String Operation

There are various string operation that can be performed such as;

i) Concatenation
ii) Repetition
iii) Membership
iv) Comparison
v) Slicing

i) Concatenation → This term means joining. (+ operator is used to concatenate two strings)

- "Hello" + "World"
  = HelloWorld

» "Hello" + " " + "World"
= Hello World

» 2 + "World"
= Error

» "2" + "3"
= 23

ii) Repetation → (*) operator is used to create multiple copies of the same string.

» 3 * "Hello"
= HelloHelloHello

» 2 * "2"
= 22

» "3" * "3"
= Error.

iii) Membership Operator → Python offers two membership operator (in, not in) to check whether the given character is present in the string or not.
It returns the value in true or false.

» 'H' in "Hello"
= True.

» 'K' not in "Kit"
= False.

iv) Comparison Operator → We can use relational operator (<, >, >=, <=, ==, !=) to compare two strings. String compares using ASCII Code.

Character	=	ASCII Code
'0' - '9'	=	48 to 57
'A' - 'Z'	=	65 to 90
'a' - 'z'	=	97 to 122

11/10/22

Q) Give the output of the following:-

» 'Ram' == 'Raja'
= False

» "free" != "Freedom"
= True

» "arrow" >> "aron"
= True

» "right" >= "left"
= True

» "Mary" < "Mac"
= False

» "abc" > ""
= True

v) String Slicing → Slicing is used to retreat a subset value.

A Slice of a String is nothing but a sub-string. This extracted ~~substring~~ sub-string is termed as slice.

Syntax :-

String_name [start : end : step]

e.g. → ① S1 = "I love Python"
S = S1 [2 : 10 : 1]
print (S)

0	1	2	3	4	5	6	7	8	9	10	11	12
I		l	o	v	e		P	y	t	h	o	n

O/P → LOVE Pyt

② S = S1 [2 : 10 : 2]
print (S)

O/P ⇒ lv y

Q) Consider a given string that is A = "SAVE MONEY". Find the output for the same.

a) A[1:3]
= AVE

b) A[:3]
= SAV

c) A[2:]
= VE MONEY

d) A[:]
= SAVE MONEY

e) A[-2:]
= ~~ENOM EVAS~~ EY

f) A[:-2]
= ~~EY~~ SAVE MON

g) A[::2]
= ~~EOVS A AEMNY~~
= SV OE

* Strings are immutable, means that the content of the string cannot be changed after it is created.

e.g → S = "Python"
S[0] = 'A'
⇒ Error

String Functions

Python provide several build in function associated with string. It allows us to modify and manipulate the strings.

1) len() → It is used to count total number of character in a string.

Syntax → len(str)

```
>>> Word = "Good Boy"
>>> len(Word)
>>> 8.
```

2) Capitalize() - This returns the exact copy of the string with first character in Uppercase.

Syntax :- str. Capitalize()

```
>>> S1 = "'welcome"
>>> S1. Capitalize()
>>> Welcome
```

3) Replace() - The function replaces all the occurance of the old string to the new string.

Syntax :-

Str. replace (old, new)

ex :-
```
>>> S1 = " This is a string example"
>>> S1. replace ("is", 'was')
>>> Thwas was a string example.
```

4) Upper() - This function converts the lower case letter into upper case.

Syntax :-

str.upper()

ex:- >>> S1 = "Welcome"
>>> S1.upper()
>>> WELCOME

5) lower() → This function converts all upper case letters to lower case.

Syntax :- str.lower()

ex:- >>> S = "WELCOME"
>>> S.lower()
>>> welcome

6) title() → It returns the string with first letter of every word in uppercase and rest in lowercase.

Syntax :- str.title()

ex:- >>> s = "hello, its all python"
>>> s.title()
>>> Hello, Its All Python

7) Swapcase() → It converts all uppercase letter into lower case and vice versa.

Syntax :- str.swapcase()

ex:- >>> s1 = "PYthON"
>>> s1.swapcase()
>>> pyTHon

8) isalpha() → This function checks for alphabet in a input string, it returns true if the string contains only letter, otherwise, it returns false.

Syntax :- str.isalpha()

ex:- >>> str = "Good"
>>> print(str.isalpha())
>>> True.

ex:- >>> str = "Working....in Python"
>>> print(str.isalpha())
>>> False.

9) isdigit() → This function returns true if the string contains only digits otherwise false.

Syntax:- str.isdigit()

ex:- >>> s = "12343"
>>> print(s.isdigit())
>>> True.

ex:- >>> s = "Ram123"
>>> print(s.isdigit())
>>> False.

10) isalnum() → The isalnum() method returns if all the characters are alphanumeric i.e. alphabets and numbers except all the special characters.

Syntax → string.isalnum()

ex:- >>> ~~string.isalnum()~~ s = "python 3.8"
>>> print(s.isalnum)
>>> False.
# because space and dot is present.

ex:- >>> s = "Python38"
>>> print (s.isalnum)
>>> True.

11/10/22

H.W- WAP to input a string and count
Q1) total no. of uppercase and lowercase character.

```
s = input ("Enter a string")
c = 0
d = 0
i = 0
while i < len(s):
for i in range (0, len(s)):
    if (s[i].is lower()):
        c = c + 1
    elif (s[i].is upper()):
        d = d + 1
print ("Total no. of lowercase letters = ", c)
print ("Total no. of uppercase letters = ", d)
```

12/10/22

11) ord() → This function returns the ASCII value of the character.

```
>> ch = 'A'
>> ord (ch)
>>  65
```

12) chr() → This function returns the character represented by the inputted unique code.

```
>> chr (65)
>> 'A'
```

13) is lower() → This function returns true if all the characters are in lower case otherwise it will display false.

Syntax :- str. is lower()

```
>>> S = "python"
>>> S.islower()
>>> True
```

14) isupper() → This function returns true if all the characters are in upper case otherwise it will display false.

Syntax :- str.isupper()

```
>>> s= "HELLO"
>>> s.isupper()
>>> True
```

15) endswith() → This function returns true if the given string ends with the specified substring else returns false.

Syntax :- str.endswith(substring)

```
>>> s= "Hello World"
>>> s.endswith("ld")
>>> True
>>> s.endswith("wo")
>>> False
```

16) startswith() → The function returns true if the given string starts with specified substring else returns false.

Syntax:- str.startswith(substring)

```
>>> s = "Hello World"
>>> s.startswith("He")
>>> True
```

17) join(sequence) → This function returns a string in which the string element have been joined by a string separator.

Syntax:- str.join(sequence)

```
>>> s = "12345"
>>> s1 = '-'
>>> s1.join(s)
>>> 1-2-3-4-5
```

17) istitle() → This function does not take any argument., It returns true if the string is titled case else returns false if the string is not a titled case.

Syntax :- str.istitle()

```
>>> s = "Hello World"
>>> print (s.istitle())
>>> True

>>> s1 = "Hello world"
>>> print (s1.istitle())
>>> False
```

18) count() → This function returns the no. of times the given substring occurs. If we do not give start and stop index, the start index will be '0' and the end index will be 'length of the string.

Syntax :- str.count (substring, start, stop)

Ex:- >>> s = "Hello World ! Hello Hello"
>>> s.count('Hello') → 3
>>> s.count("Hello", 12, 25) → 2
>>> s.count("e", 5, 15) → 1

19) strip() → This function returns the string by removing the spaces both on the left and the right side.

Syntax :- str.strip()

>>> s = "HelloWorld "
>>> s.strip()
>>> 'HelloWorld'

20) lstrip() → This function returns the string after removing the spaces from the left hand side.

Syntax :-

str.lstrip()
or
str.lstrip(chars)

Ex:- >>> s = "Green world"
>>> s.lstrip()
>>> Green world

>>> s.lstrip('Gr')
>>> een world
>>>
>>> s = "Green world"
>>> s.lstrip('rG')
>>> een world

21) r.strip() → This function removes the white space from the right side of the string.

Syntax :- str.rstrip()
or
str.rstrip(optional)

>>> s = "Green world"
>>> s.rstrip()
>>> Green world

>>> s.rstrip("orld")
>>> Green we

22) find() → This function is used to search the first occurance of the substring. It returns the lowest index of the substring, if it is found otherwise it returns -1.

Syntax :- str. find (sub, start, end)

```
>>> word = " Green word"
>>> print ( word.find ("Green")
>>> 0

>>> print (word.find ('0', 8, 9)
>>> -1
```

23) index() → This function is similar to as of find() function. It searches the first occurance and returns the lowest index of the substring if it is found, but raises an exception if substring is not found.

Syntax :- str.index (sub, start, end)

```
>>> str = "Hello World of Python"
>>> str.index("Hello")
>>> 0

>>> str.index ("Hi")
>>> error.
```

Note!- Both (22) and (23) have same working.

24) split() → The split function break the string at a specified separator and returns a list of string.

Syntax!- str.split ((separator), (maxsplit))

```
>>> x = "blue; red; green"
>>> x.split (";")
>>> ["blue", "red", "green"]
```

→ The separator is a delimiter.
→ Maxsplit (optional) → It defines the maximum number of split which is by default -1, which means no limit on a number of splits.

```
>>> x.split()
>>> ['blue', 'red', 'green']

>>> x.split(";", 2)
>>> ['blue', 'red', 'green']

>>> x.split(";", 0)
>>> ["blue; red; green"]
```

14/10/22

(Q2) WAP to input a string and check it is a pallendrome or not.

```
s = input("Enter a string")
k = s[::-1]
if s == k:
    print("Pallendrome string")
else:
    print("Not a pallendrome string")
```

(Q3) WAP to input a string and count total no. of vowels as consonant in a given string.

⇒
```
s = input ("Enter string")
v=0
c=0
for i in s:
    if i in "aeiouAEIOU":
        v= v+1
    else:
        c= c+1
print ("no. of vowel is", v)
print ("no. of consonant is", c)
```

(Q4) What will be the output of the given ~~snippet~~ snippet.

(1)
```
str1 = "My name is raja"
str2 = str1[3:7]
s= len (str2)
print (s)
```

= O/P ⇒ 4.

② print ("Computer Science". split ("er", 2))

= O/P = ['Comput', 'Science']

③ How many times the loop will execute:-

```
s= 'python rocks'
for ch in s[3:8]:
    print ("Hello")
```

O/P = 5 times.

Q5) Find the output of the given code:-

```
word = " green vegetables"
print (word.find ('g', 2))
print (word.find ('veg', 2))
print (word.find ('tab', 4, 15))
print (word.find ('eg', 6, 8))
```

⇒ O/P 8
6
10
-1

Q6) Comider a given string :—
str = "Green Revolution" and write the statements for the following :—

i) To display the last four characters.
= str [-4 :]

ii) Repeat the string three times.
= str * 3

iii) Check the string contains ('vol') or not.
= 'vol' in str.

iv) Display the starting index of sub string ('vo').
= str. find ('vo')

(Q7) What will be the output of the given string :—

```
x = "Amazing"
print (x [3:], "and", x [ :2])
print (x [-7:], "and", x [-4:-2])
print (x [2:7], "and", x [-4:-1])
```

a) O/P = zing and Am
Amazing and zi
azing and zin

Q8) What will be the output of the following code :-

```
str = "My python programming"
str [-5 : -1]
str [1 : 5]
str [-4 : ]
str [ : -4]
str [0 : ]
str [ : 13-4]
str [ : : -1]
```

= O/P

```
mmin
y pyt
hing
progra
My Python programming
My Python
```

17/10/22

Q1) Output Based Questions :-

i) str = "aNDarIel"
n = " "
for i in range (len(str)):
    if str[i].is upper()
        n = n + str[i].lower()
    else:
        n = n + str[i].upper()

→ O/P AndARiEL

ii) Write the output for the following snippet given below :-

str = "CBSE Digital India"
for i in range ("len(str)-1, 0, -1):
    if str[i].is upper():
        print (str[i].lower(), end " ")
    elif i % 2 == 0:
        if str[i].islower():
            print (str[i].upper(), end" ")
    else:
        print ("@", end" ")

→ O/P @I @N @ @L@ @N@
IN i@AIId@esb

2) Write a program to input a line and count total no. of words present in that particular line.

```
S = input("Enter a line")
x = S.split()
c = 0
for i in x:
    c = c + 1
print("Total no. of words = ", c)
```

3) WAP to read a line and count how many times the substring 'is' appears in the given line.

```
S = input("Enter a line")
x = S.split()
c = 0
str = "is"
for i in x:
    if str == i:
        c = c + 1
print("Total no. of times 'is' occurs", c)
```

4) WAP to input a string and display the occurance of those words which starts with a vowel and also display those words.

```
s= input("Enter a line")
x= s.split()
c=0
d= [aeiouAEIOU]
for i in x:
    if i.startswith(d)
        c=c+1
        print(i)
print("Total no. of vowels ", c)
```

OR

```
x= s.split()
c=0
for i in x:
    if i[0] in "aeiou" or i[0] in "AEIOU"
        c= c+1
        print(i)
print("total no. of vowels ", c)
```

5) Consider the following code & give the output of the following :-

```
str = input ("Enter a string")
while (len (str) <= 4):
    if str [-1] == 'z':
        str = str [0:3] + 'c'
    elif 'a' in str:
        str = str [0] + 'bb'
    elif not int (str[0]):
        str = 'I' + str[1:] + "2"
    else:
        str = str + '*'
print (str)
```

~~i) O/P~~ str = 'Grazing'

i) 1bzz

ii) abcd

1bzc

i) O/P 1bzc

ii) O/P abb

18/10/22

Q) What will be the output of the following code:-

```
Text = "Mind@work!"
ln = len(Text)
n = " "
for i in range (0, ln):
    if Text[i].isupper():
        n = n + Text[i].lower()
    elif Text[i].isalpha():
        n = n + Text[i].upper()
    else:
        n = n + 'A'
print (n)
```

⇒ O/P ⇒ mINDAWORKA

Q7) WAP to input a string & remove all the vowels from the given string.

```
s = input("Enter a string")
b = len(s)
c = ""
for i in range(0,b):
    if s[i] in 'AEIOUaeiou':
        continue
    else:
        c = c + s[i]
print("Original string =",s)
print("New string =", c)
```

Q8) WAP to input a string / word and make a new word from adding the 1st two characters & last two characters.

⇒
```
s = input ("Enter word")
ln = len(s)
b = ''
for i in range (0, 2):
    if s[i]
while s[i] < 2:
n = s[0:2] + s[l-2; l]
print (n).
```

Q9) WAP to input a string and convert all the vowels in uppercase.

```
S= input ('Enter string')
b= ""
ln = len(S)
for i in range (0, ln):
    if S[i] in 'aeiouAEIOU'
        b= b + S[i].upper()
    else:
        b = b + S[i]
print ("Original string=", S)
print ("New string =", b)
```

Q10) WAP to input a string & display the first character of every word & make a substring of it.

⇒
```
s = input("Enter string")
b = " "
p = s.split()
for i in p:
    if p[i] < 1:
        b =
s1 = " "
s = " " + s
l = len(s)
for i in range(0, l):
    if s[i] == " ":
        s1 = s1 + s[i+1]
print("New string", s1)
```

2nd method

```
s = input("Enter a string")
l = len(s)
b = "   "
for i in range (s.split())
    S1 = S1 + S[i]
print ("New string of first letters=",S
```