

String :-

String in python is surrounded by either

either

or

or

→ String is stored as arrays

- Each characters in String are having index value starting from 0
- And from end, the index value starts from -1

→ String Operators

The + Operator is used to concatenate strings

Ex:-

S = 'foo'
t = 'bar'
u = 'baz'

S+t

'foobar'

S+t+u

'foobarbaz'

Print('Go team') + ('!!!')

Go team !!!

x =

Any ~~any~~ by

My Plot no. is 363.

'+' operator only concatenate string characters
so when we need to concatenate string with integers or integers with integers
than we have to convert integers into strings

Ex: `str(integer value)`

The '*' operator create multiple copies of the string

Ex:-

`s = 'foo'`

`s * 4`

O/P

`foo . foo . foo . foo`

'in' operators

The 'in' operator return true if the first operand is contain in either second or return false.

Ex:-

`s = 'foo'`

`s in "that food for thought,"`

Then it return 'True'

`s in "that good for health,"`

Then it return 'False'

String Functions:

- `char()` → Convert an integer into character
- `ord()` → Convert a character to an integer

• `len()` → Return the length of a string

• `str()` → Return a string representation of an object.

→ This gives ASCII value of the character given

$S = \text{Gain a string - }$
len(S) O/P: 13

~~NDP~~

0	1	2	3	4	5	6	7
N	I	H	A	R	I	K	A
-8	-7	-6	-5	-4	-3	-2	-1

numbers in forward
reading order

→ String

Have some predefined functions

• `upper()`

• `lower()`

• `capitalize()`

• `title()`

• `replace()`

`Replace()`

`S = "We are here to learn Python";`

`Print("Printing the string:");`

`Print(S)`

`Print("Replacing Python to String")`

`S = S.replace('Python', 'String')`

→ which is → new changed string

`Count():`— Counting the occurrence of the given characters

`Print("Counting the characters:");`

`Print(S.Count('a'))`

`endswith()` → It is used to return 'True' if we pass the character is same as in the string and 'False' if passed character is not available at the end of the given string.

`startswith()` → It is used to return 'True' if the passed character is present in the starting of the string and 'False' if the passed character is not present in the starting of the string.

Ex :- `S = "We are here to learn Python from GPCS"`

`Print("Printing the string:");`

`Print(S)`

Print ("Ending with :")

print (S. ends with ('S'))

Print ("Starts with :")

print (S. starts with ('W'))

O/P:

Printing the string:

We are here to learn Python from IPCS

Ending with

True

Starts with

True

Boolean String functions (C. m. true) (E. f. false)

Functions that

return the O/P in the form of
'True' and 'False'

1. isalpha ()

2. isalnum ()

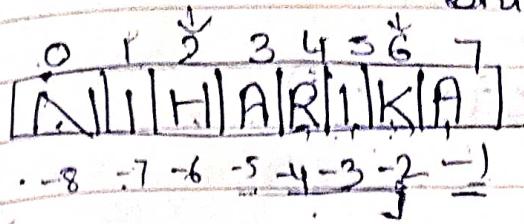
3. isspace ()

4. isdigit ()

5. isupper ()

6. islower ()

String Slicing :- It is used to slice the string that is given.



S = "NIHARIKA"

Print ("Pointing the String :")

Point (S2)

Print ("Print first character :")

Point (S2[0])

O/P :

N (Note: output printing is trained)

Print ("Print characters from 0 to 3")

Print (S2[0:3])

O/P :

NIH

Print ("Print characters from 3 to 6")

Print (S2[3:6])

O/P : ART

Print ("Print last character")

Print (S2[-1])

O/P :

Print ("Print last character to -6 :")

Print (S2[-1:-6:-1])

from last

destination value + 1

increment of index value

O/P:

A K I R A

Print ("Printing from -3 to -6")

Point ($S_2[-3:-6:-1]$)

$\downarrow \quad \downarrow$ increment of index (i.e.)

Starting value Ending index Right to left (-1)

index Value index Value

O/P:-

A R T

($S_2[0:7:2]$)

Print (" Skipping two characters from last")

Print ($S_2[-1:-6:-2]$)

O/P

K R

Print (" Skipping two characters ")

Print ($S_2[2:6:2]$)

④

(\rightarrow value is +ve

because increment

from left to right)

Print (" Printing Reversed")

Print ($S_2[: -1]$)

O/P

\rightarrow by default value is ~~1~~ 1st index value

By default value is -1 1st index value

KIRAHIN

N H R K - R

Point C (Point from standard to 10³)

Print(*, S[1:11])

↳ By default Blant indexes noise

Point C "Print from 10 to last"

Print C S [10 :]

↓ By default the destination range is taken as the last index value).

If string is method: → For Big using this
type conversion is not needed
there is integer value of
String.

Ex:-

Var = 123

Var = 123
Point (f ' a dog says ' var y ')
S 2 1.

~~form~~ ~~function~~

©/P:

a dog says 123

↓
Here 123 is
integer value
but we using
Placeholder

as characters without their type conversion.

My PC's name is Doge

Escape characters: — To insert the characters whose use is illegal in string, to maintain their use we have to use escape character.

Ex: ~~we are the so-called "Vikings"~~
txt = "We are the so-called "Vikings"
from the north."

Escape character

O/P: ~~We are the so-called "Vikings" from the north.~~
We are the so-called "Vikings" from the north.

Escape character	Result
'	Single Quote
\	Backslash
\n	Newline
\r	Carriage Return
\t	Tab
\b	Backspace
\f	Form Feed
\ooo	Octal Value
\xhh	Hex Value

Ex: ~~msg~~

msg

obstruction

Alt + backspace

W+Q + backspace