Strings :

Data type used to represent textual data. They are sequences of characters and are enclosed in either.

How to create String :

You can create strings using single, double, or triple quotes. Triple quotes are used for multiline strings or to include special characters like line breaks.

String Indexing :

Strings are ordered sequences, and you can access individual characters using indexing. Python uses zero-based indexing, where the first character has an index of 0.

Str = " PYTHON "

Positive - 0 1 2 3 4 5

Negative - -6 -5 -4 -3 -2 -1

Code :-

```
Str = " PYTHON "
Print (Str[0])
Print (Str[2])
Print (Str[-1])
```

String slicing :- Division of string into several parts.

Name = " Sreenath Bathala "

String_Variable [start : stop : step]

Starting Index — End Index — Increment

Name [0] = S

Name [0:4] = Sree

Name [0:4:2] = Se

Ex :- s = hello World

0 1 2 3 4 5 6 7 8 9 10

-11 -1

Print (s[1]) = e

Print (s[-1]) = d

Print (s[1:3]) = el

Print (s[1:-1]) = ello world

Print (s[::2]) = hlo wrd

Print (s[1::2]) = el-ol

Print (s[::-1]) = dlrow olleh

Print (s[2:]) = llo world

Print (s[:-1]) = hello worl

String concatenation :-

You can concatenates strings using the + operator.

```
first_name = "John"
last_name = "Doe"
full_name = first_name + " " + last_name
Print (full_name)
```

O/P :- "John Doe"

String Length :- (Built in function) ⟶

You can find the length of a string using the len() function.

Ex:-

```
String 1 = "Coding is fun!"
Print (len (string1))
```

O/P :- 14

String Methods :-

Python provides numerous built-in methods for manipulating strings, such as converting cases, removing whitespaces, replacing characters, splitting, joining and more.

Methods :- S = "Hello, World!"

Print (s.upper()) O/P :- "HELLO, WORLD!"

Print (s.lower()) O/P :- 'hello, world!'

print (s.strip()) O/P :- 'Hello, world!'

Print (s.replace ('o', 'x')) o/p : 'Hellx, wxrld!'

Print ('Abracadabra'.count ('a')) o/p : 4

String formatting :-

Python supports multiple ways of formatting strings, including old-style % formatting, str.format(), and f-strings

```
name = "Alice"
age = 30
```

old - Print ("My name is %s and I am %d years old." % (name, age))

format() - Print ("My name is {} and I am {} years old.". format (name, age))

formatted string literals → Print (f"My name is {name} and I am {age} years