Str data type
String is a collection of characters; these characters can be alphabets, digits or special characters. String is non numeric data type.

How to represent string?
In python string values are represented in 3 ways

1. Within single quotes
2. Within double quotes
3. Within triple single quotes or double quotes

The string which consists of alphabets is alphabetic string.
The string which consists of alphabets and digits is called alphanumeric string.

Within single quotes, programmer can represent one line string.

```
>>> s1='python'
>>> s2='python language'
>>> s3='python language is
SyntaxError: incomplete input
>>> s4='nit123'
>>> type(s1)
<class 'str'>
>>> type(s2)
<class 'str'>
>>> type(s4)
<class 'str'>
>>> s1
'python'
>>> s2
'python language'
>>> s4
'nit123'
>>> s5='45'
>>> s5
'45'
>>> type(s5)
<class 'str'>
>>> s6='1.6'
>>> s6
'1.6'
>>> name='नरेश'
>>> name
'नरेश'
>>> name='నరేష్'
>>> name
'నరేష్'
```

Within single quotes, programmer can embed/insert double quotes.

```
>>> s1='Python"s'
>>> print(s1)
Python"s
>>> s2='python's'
SyntaxError: unterminated string literal (detected at line 1)
```

Double quotes
Within double quotes programmer can represent single line string.

```
>>> s3="python"
>>> s4="65"
>>> s6="p101"
>>> print(s3,s4,s6)
python 65 p101
>>> s5="python programming
SyntaxError: incomplete input
>>> s7="Monty Python's Flying Circus"
>>> print(s7)
Monty Python's Flying Circus
```

Triple Quotes
Within triple quotes programmer can represent multiline string.

```
>>> s8='''Python is a
... programming
... language'''
>>> print(s8)
```

```
Python is a
programming
language
>>> s9="""Python is a
... programming
... language"""
>>> print(s9)
Python is a
programming
language
```

Escape Sequences

Escape sequences are special characters.

\n	New line
\t	Tab space
\v	Vertical tab space
\\	\
\”	“
\’	‘

```
>>> s10='Monty python\'s flying circus'
>>> print(s10)
Monty python's flying circus
>>> s11="Monty Python\"s flying circus"
>>> print(s11)
Monty Python"s flying circus
>>> s12="Python\nProgramming\nLanguage"
>>> print(s12)
Python
Programming
Language
>>> s13='Python\nProgramming\nLanguage'
>>> print(s13)
Python
Programming
Language
>>> s14="\"
SyntaxError: incomplete input
>>> s14="\\"
>>> print(s14)
\
>>> s15="Rollno\tName\tCourse\tFees"
>>> print(s15)
Rollno	Name	Course	Fees
```

Int,float,complex,NoneType,bool → scalar data types
str → collection

Every program required 3 components

1. Input
2. Process
3. Output

Input is nothing but data or information given to program. This input is given from various sources (keyboard, file, database, …).

Output is nothing but data or information given by program.

Process is nothing but performing operations on input data.

print()

print() is standard output function. This function is used to print or display information/data on console/monitor.
It is a predefined function and exists in __builtins__ library and it is default library imported by any python program.

Syntax-1: print(data/info/values)
Syntax-2: print(values,sep=’ ‘)
Syntex-3: print(values,sep=’ ‘,end=’\n‘)

Example:
# Example of print function

```
print(10)
print(1.5)
print(1+2j)
print(True)
print(None)
print("Python")
```

Output:

```
10
1.5
(1+2j)
True
None
Python
```

Example:

```
# Example of print function

a=10
b=1.5
c=1+2j
d=True
e=None
f="Python"

print(a)
print(b)
print(c)
print(d)
print(e)
print(f)
```

Output:

```
10
1.5
(1+2j)
True
None
Python
```

Syntax-2: print(values,sep=’ ‘)
Print function print/display more than one value using default separator space.

Example:

```
# Example of print function to print multiple values


print(10,20,30,40,50)
print(1,1.5,1+2j,False,"Python")

print(10,20,30,40,50,sep=";")
print(10,20,30,40,50,sep=",")
print(10,20,30,sep="*")
print(10,20,30,sep="\n")
print(10,sep="%")
print(10,20,sep=':')
```

Output:

```
10 20 30 40 50
1 1.5 (1+2j) False Python
10;20;30;40;50
10,20,30,40,50
10*20*30
10
20
30
10
10:20
```

Example:

```
# Example of print function

a=10
b=1.5
c=1+2j
d=True
e=None
f="Python"
```

```
print(a,b,c,d,e,f)

print(a,b,c,d,e,f,sep=',')
print(a,b,c,d,e,f,sep=';')
```

Output:

```
10 1.5 (1+2j) True None Python
10,1.5,(1+2j),True,None,Python
10;1.5;(1+2j);True;None;Python
```