Python String: Operators and Functions

Anoop S Babu
Faculty Associate
Dept. of Computer Science & Engineering
bsanoop@am.amrita.edu

String Operators

- Concatenation - +
- Joining two or more strings into one string

```
>>> s = "Python"
>>> t = "programming"
>>> c = s + t
>>> print(c)
Pythonprogramming
```

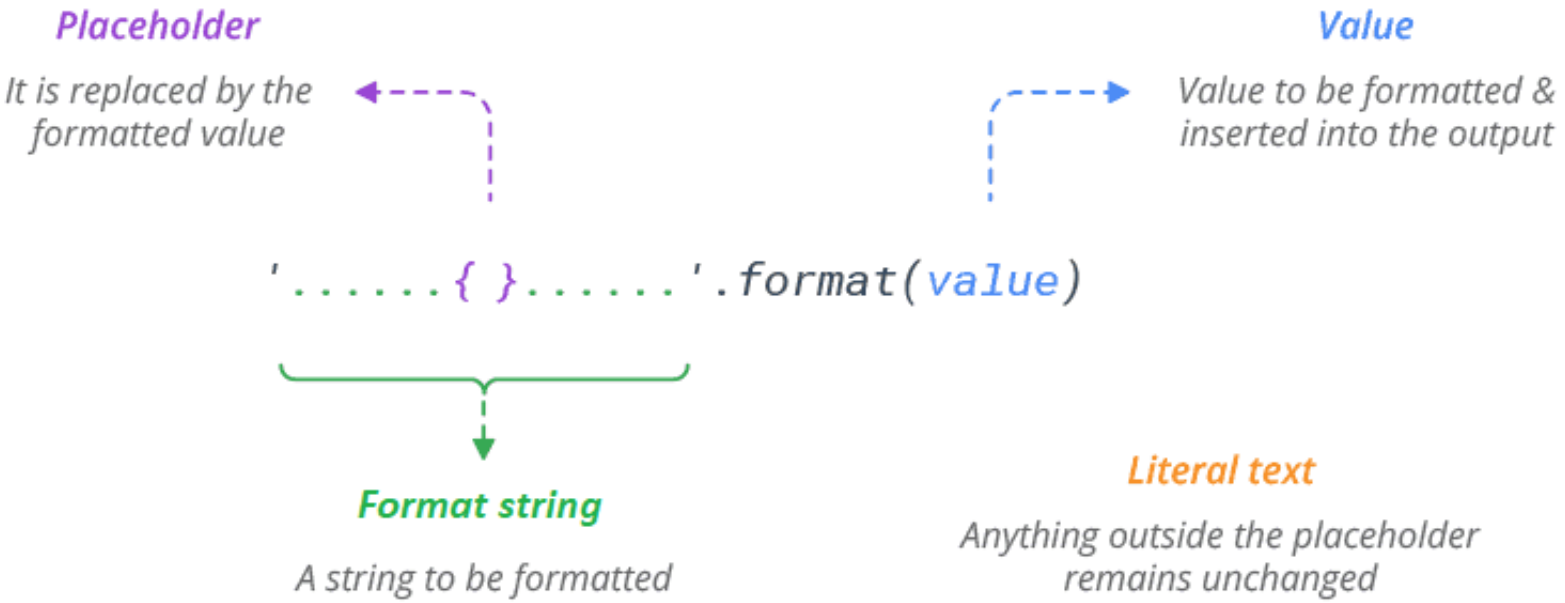
String Operators

- Repetition - *
- Repeat the string for a given number of times

```
>>> s = "Python"
>>> print(s * 3)
PythonPythonPython
```

String Formatting

- format() method
- The format string contains curly braces which act as place holders for the values.



String Formatting Orders

- Implicit
- Positional
- Keyword

Implicit

```
>>> s1 = "Python"
>>> s2 = "String"
>>> s3 = "Formatting"

>>> print("{} {} in {}".format(s3,s2,s1))
Formatting String in Python
```

Positional

```
>>> s1 = "Python"
>>> s2 = "String"
>>> s3 = "Formatting"

>>> print("{1} {0} in {2}".format(s3,s2,s1))
String Formatting in Python

>>> print("{2} {0} in {1}".format(s3,s2,s1))
Python Formatting in String
```

Keyword

```
>>> s1 = "Python"
>>> s2 = "String"
>>> s3 = "Formatting"


>>> print("{s} {f} in {p}".format(p = s1,s = s2,f = s3))
String Formatting in Python
```

String Function

- **len()** – returns the length of the string

```
>>> s= "Python Programming"
>>> len(s)
18
```

String Methods – find

- **find(*char*)** – returns the first occurrence of the character

```
>>> s= "Python Programming"
>>> s.find("g")
10
```

String methods – replace

- **replace(***char1*,*char2*,*[no:]***)** – replace all or specified number of occurrences

```
>>> s= "Python Programming"
>>> s.replace("P","C",1)
'Cython Programming’

>>> s.replace('o','*')
'Pyth*n Pr*gramming'
```

String Method - join

- **join(*seq*)** – join all iterables into a string separated by the given separator

```
>>> words = ("Python", "Programming")
>>> space = " "
>>> sentence = space.join(words)
>>> sentence
'Python Programming’
>>> sentence = '@'.join(words)
>>> print(sentence)
Python@Programming
```

String Method - split

- **split()** – splits the string into substrings w.r.t the separator

```
>>> sentence = 'Python Programming'
>>> sentence.split()
['Python', 'Programming’]

>>> s = "String,methods"
>>> s.split(',')
['String', 'methods']
```

String Traversal

```
>>> s = "Python"
>>> for i in s:
        print(i)
```

```
P
y
t
h
o
n
```