- **created by**: Sudip Ghimire
- **URL**: https://www.sudipghimire.com.np
- **GitHub**: https://github.com/ghimiresdp

go to course contents

Chapter 2.4: String Formatting

Table of Contents

Python has various methods for formatting strings. It has the capability to format strings, numbers, their precision, and many more. some of the methods to format the strings are as follows:

Formatting Strings with %s, %d and %f

We can format strings with the following formatting strings

1. %s: Used for strings
2. %<a>d: Used for integers
3. %<a>.<b>f: Used for floating point numbers

Whenever we format strings with these notation, they accept the sequence of characters after quotation marks of the string.

- a defines the minimum number of digits allocated to represent the number
- b defines the maximum number of digits expected after decimal point.

```
name = 'John'
age = 10
print("My name is %s"%name)
print("I'm %s and I'm %d years old"%(name, age))
print("the value is:%10.3f" % 44.2345345)  # output: The value is:    44.235
```

Format method format()

Format method can be used as the string method which accepts the pair of curley brackets {} inside the string. We have to make sure that the number of parameters should match the number of bracket pairs {} to work it correctly.

```
name = 'John'
print("Hi {}, how are you doing? Are you {} now?".format(name, age))
```

`format()` method with named arguments

We can also pass named arguments in the format string which is independent of order of parameters. in the example below, the parameters `n` and `pi` are named arguments.

```
print("Hi {n}, is the value {pi:10.2f}?".format(pi=3.1415, n=name))
```

F-strings

Source: https://peps.python.org/pep-0498/

F-strings were introduced in python 3.6. With the help of f-strings, we can format our string in even easier way. F-strings were created since all formattings are not supported using `%`-formatting methods.

F-strings provide a concise, readable way to include the value of Python expressions inside strings. F-strings are also called as templated strings where we use variable inside a string to generate dynamic content.

F-strings are always written inside quotation marks starting with f (i.e.`f''` or `f""`).

Basic structure of f-strings

```
# adding only value
f'content {statement}  content'

# adding formatted value
f'content {statement: format} content'
```

example:

```
PI = 3.1415926535

print(f"The value of pi is {PI}.")

print(f"The value of pi is {PI: 5.3f}.")    # formatting the floating point
characters
```

Unlike `%` formatting, we can even pass complex statements inside the formatted string.

Example:

```
word = 'Hello World'

print(f'The word "{word}" has {word.__len__()} characters.')
print(f'The word "{word}" in upper case is {word.upper()}.')

# even complex one
print(f'The word {name}  is {"long" if name.__len__() > 10 else "short"}')
```