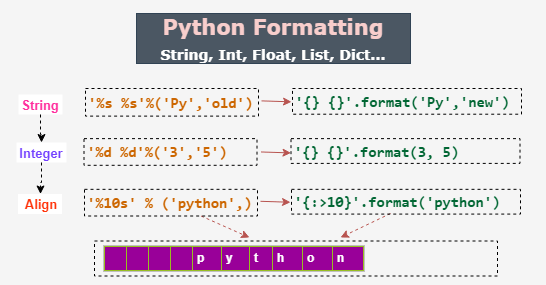
**Formatting**

How to format string and other data types in Python?

**Using format function**



Format syntax : '{}'.format(param)

\* Description:

\*\* '{}': It is the format target, i.e., the placeholder.

\*\* param: It can be a string, integer, float, or any of the collection types.

\*\* Return: It returns a formatted output with the input argument substituting the placeholder.

Format syntax : '{} {}'.format(arg1, arg2)

\* Description:

\*\* '{} {}': We can have multiple placeholders.

\*\* arg1, arg2: They can be a string, integer, float, or any of the collection types.

\*\* Return: It returns a formatted output with the input argument substituting the placeholder.

test = "Hello"

print('{}'.format(test))

op🡺Hello

**Padding and Align strings**

print(format("Hello", ">10s"))

Hello

print(format("Hello", "#>10s"))

#####Hello

print(format("Hello", "#<10s"))

Hello#####

print(format("Hello", "#^15s"))

#####Hello#####

**Justify a Variable String Expansion**

>>> print('<-- {0:30} -->'.format('Python Programming'))

<-- Python Programming -->

>>> print('<-- {0:>30} -->'.format('Python Programming'))

<-- Python Programming -->

>>> print('<-- {0:<30} -->'.format('Python Programming'))

<-- Python Programming -->

>>> print('<-- {0:^30} -->'.format('Python Programming'))

<-- Python Programming -->

**Integer’s formats**

print("I've <{}> years of experience and my salary is <{}> USD per annum.".format(10, 75000))

I've <10> years of experience and my salary is <75000> USD per annum.

Using separator number formatting

>>> print("I've <{}> years of experience and my salary is <{:,}> USD per annum.".format(10, 75000))

I've <10> years of experience and my salary is <75,000> USD per annum.

**Padding for numbers**

print("I've <{:#>8}> years of experience and my salary is <{:z>20,}> USD per annum.".format(10, 75000))

I've <######10> years of experience and my salary is <zzzzzzzzzzzzzz75,000> USD per annum.

**Binary**

print('{0:b}'.format(10))

1010

**Octal**

print('{0:o}'.format(10))

12

**Hex decimal**

>>print('{0:x}'.format(10))

a

>>> print('{0:X}'.format(10))

A

**Floating number**

print("{0:f}".format(1.123456))

1.123456

print("{0:.3f}".format(1.123456))

1.123

>>> print('Fixed-point example: <{0:f}>'.format(2.2183))

Fixed-point example: <2.218300>

>>> print('Fixed-point with right alignment example: <{0:25f}>'.format(2.2183))

Fixed-point with right alignment example: < 2.218300>

>>> print('Fixed-point with precision and right alignment example: <{0:<25.10f}>'.format(2.2183))

Fixed-point with precision and right alignment example: <2.2183000000 >

>>> print('General format example: <{0:g}>'.format(2.2183))

General format example: <2.2183>

>>> print('General format with right alignment example: <{0:25g}>'.format(2.2183))

General format with right alignment example: < 2.2183>

>>> print('General format with precision and center alignment example: <{0:^25.10g}>'.format(2.2183))

General format with precision and center alignment example: < 2.2183 >

>>print('Scientific format example: <{0:e}>'.format(2.2183))

Scientific format example: <2.218300e+00>

>>> print('Scientific format with left alignment example: <{0:<25e}>'.format(2.2183))

Scientific format with left alignment example: <2.218300e+00 >

>>> print('General format with precision and right alignment example: <{0:>25.5e}>'.format(2.2183))

General format with precision and right alignment example: < 2.21830e+00>

**Formatting a List**

langs = ['C', 'C++', 'CSharp']

>>> print('Skillset: {}'.format(langs))

Skillset: ['C', 'C++', 'CSharp']

print('Skillset: {0[1]}'.format(langs))

Skillset: C++

print('Skillset: {}'.format(\*langs))

Skillset: C

**Formatting Dict**

print(" Jake's salary is {0[jake]} \n Anand's salary is {0[anand]}".format({'jake': '$100K', 'anand': '$120K'}))

Jake's salary is $100K

Anand's salary is $120K

print(" Jake's salary is {sal[jake]} \n Anand's salary is {sal[anand]}".format(sal={'jake': '$100K', 'anand': '$120K'}))

Jake's salary is $100K

Anand's salary is $120K