

# String Formatting

String formatting allows us to create dynamic strings by combining (plugging in) values and variables in them.

It is also used in print statement where we pass strings.

# Format String Method

1. Formatting with % operator.
2. Formatting with format() method.
3. f-string

# String Formatting using % formatter

s1: 'mayank'

Print ("my name is %.s" %(s1))



print ("Hi %.s , I am %.s" %(s1, 'mayank'))



%.d → integer

%.s → string

%.f → float

## \* Precision Handling with % operator

Floating point number use the format

% w.p  $\Rightarrow$  precision

$\downarrow$   
whole

print ( 'The number is % 2.3f' % (3.1415) )

$\downarrow$   
rounded off to 3.142

# String Formatting using format()

⇒ Introduced in Python 3.

Syntax: 'Hi, { } welcome to { }'.format(s1, s2)

It can also be index / keyword based.

Let us see some examples.

# Python f-string

It makes string formatting very easy and efficient than previous method.

Syntax : f' Hello {name} , welcome to {&13}'

Let us see some examples.

It also support expression & precision

```
print (f' The final amount is ${price + tax}')
```

```
print (f' The value of pi is: {pi:.5f}')
```

$\{ \text{value} : \{ \text{width} \} . \{ \text{precision} \} \}$

