
Formateo cadenas

Python cuenta con diferentes métodos para ejecutar el formateo de una cadena de texto las cuales son:

Formateo utilizando `%`

También es conocido como **'operador formateo de cadena'**.

Ejemplo:

```
>>> x = 'con'
>>> print("Camilo %s futbol %s sus amigos"%('juega',x))
Camilo juega futbol con sus amigos
```

Para tener en cuenta: - `%s`: es utilizado para valores de cadena - `%d`: es utilizado para valores enteros - `%f`: es utilizado para valores flotantes.

Ejemplo:

```
>>> print('%s tiene %d perros' %('Carlos', 4))
Carlos tiene 4 perros
```

```
>>> print('El valor de PI es de %f' %(3.141593))
El valor de PI es de 3.141593
```

Formateo utilizando el metodo `format()`

Este método funciona colocando uno o mas campos de remplazo, y estos deben estar definidos por los caracteres '{}', y posteriormente llamar al método `.format()`

Ejemplo:

```
>>> print('{} tiene {} perros'.format('Carlos', 4))
Carlos tiene 4 perros
```

Este método también otras ventajas como:

Inserción de datos por posición:

```
>>> print('{1} tiene {0} perros'.format(4, 'Carlos'))
Carlos tiene 4 perros
```

Inserción de datos por llaves (permite también el reuso):

```
>>> print('{nombre} tiene {valor} perros'.format(nombre='Carlos', valor=4))
Carlos tiene 4 perros
```

Formateo utilizando los `F-strings`

Conocido de esta forma debido a que se precede el caracter `f` al inicio de la cadena. Este método es similar a `format()`.

Ejemplo:

```
>>> nombre='CARLOS'
>>> valor=4
>>> print(f'{nombre.title()} tiene {valor} perros')
Carlos tiene 4 perros
```

Formateo de números

Los valores numéricos en python también pueden ser impresos haciendo uso del método `format()` o `F-strings`. Con ayuda de la siguiente tabla, podemos revisar los diferentes formatos que pueden ser utilizados:

Número	Formato	Salida	Descripción
3.1415926	{:.2f}	3.14	Formato flotante con 2 decimales
3.1415926	{:+.2f}	+3.14	Formato flotante con 2 decimales con signo
-1	{:+.2f}	-1.00	Formato flotante con 2 decimales con signo
2.71828	{:.0f}	3	Formato flotante con sin decimales
5	{:0>2d}	05	Completar hasta tener 2 caracteres con ceros a la izquierda
5	{:x<4d}	5xxx	Completar hasta tener 4 caracteres con 'x' a la derecha
1000000	{:,}	1,000,000	Formato numérico con coma como separador
0.25	{:.2%}	25.00%	Formato porcentaje
1000000000	{:.2e}	1.00e+09	Notación exponencial
13	{:10d}	13	Alineado a la derecha hasta completar 10 caracteres
13	{:<10d}	13	Alineado a la izquierda hasta completar 10 caracteres
13	{:^10d}	13	Alineado al centro hasta completar 10 caracteres

Ejemplo:

```
>>> figura = 'circulo'
>>> area = '2 x pi x r^2'
>>> print(f'El area del {figura} es igual a {area}')
El area del circulo es igual a 2 x pi x r^2
>>> radio = 10
>>> from math import pi
>>> print(f'El area del {figura} de radio = {radio} es de {2*pi*(radio**2):.2f}')
El area del circulo de radio = 10 es de 628.32
```

Del ejemplo anterior, se puede ver como dentro de un `F-string` pueden ser llevadas a cabo operaciones de python