# Python: formatação avançada com `str.format()` e `format()`

A partir do Python 2.6 existe uma nova notação para formatação de strings, números e outros objetos, substituindo a formatação com o operador `%`. Esta notação é usada principalmente em dois contextos:

| contextos de uso | exemplos de uso |
|---|---|
| **`str.format(*args, **kwargs)`**<br>Método aplicado sobre uma string marcada com **`{❶:❷}`** onde:<br>❶ é uma especificação de valor e<br>❷ é uma especificação de formato. | `>>> fmt = 'valor de {0} com 4 casas: {0:.4f}'`<br>`>>> fmt.format(math.pi)`<br>`'valor de 3.14159265359 com 4 casas: 3.1416'`<br>`>>> fmt2 = 'valor de {0} com {n:02} casas: {0:.{n}f}'`<br>`>>> fmt2.format(math.pi, n=5)`<br>`'valor de 3.14159265359 com 05 casas: 3.14159'` |
| **`format(valor, ❷)`** função *built-in* onde<br>❷ é uma especificação de formato. | `>>> format(math.pi, '6.3f')`<br>`' 3.142'` |

# ❶ Especificação de valor

# Exemplos

| | expressão | resultado | equivalente usando % |
|---|---|---|---|
| 1 | `format('Diretor','.<12')` | `'Diretor.....'` | |
| 2 | `format(math.pi, '_>+8.3f')` | `'__+3.142'` | |
| 3 | `format(123, '0= 6x')` | `' 0007b'` | |
| | `format(123, '#06x')` | `'0x007b'` | |
| | `format(12345678.9876,'18.10n')` | `'   12.345.678,9876'` | |

# ❷ Especificação de formato

**«alinhamento»«sinal»#«largura»,«.precisao»«tipo»**

## *Elementos do formato*

**«alinhamento»** Um dos sinais **<**, **^**, **>** ou **=**, indicando:
- **<** alinhamento à esquerda
- **^** centralizado
- **>** à direita
- **=** à direita com preenchimento após o sinal

O sinal pode ser precedido de um caractere qualquer (exceto **{** ou **}**) a ser usado em vez do espaço para preencher o campo se **«largura»** for definida. Ver ex. 1, 2 e 3.

**«sinal»** Os caracteres **+**, **-** ou **␣** (um espaço em branco). Ex. 2 e 3.
- **+** exibir sinal **+** ou **–** à esquerda do número
- **-** exibir apenas sinal **–** à esquerda de números negativos
- **␣** (espaço em branco) exibir sinal **–** à esquerda de números negativos e branco à esquerda dos positivos.

**#** Use para exibir **0b**, **0o** ou **0x** à esquerda do número nas apresentações de **«tipo»** binário, octal ou hexadecimal.

**«largura»** Número de caracteres da largura total mínima do campo. O conteúdo não é truncado se exceder essa largura. Se o conteúdo for menor, haverá preenchimento conforme o **«alinhamento»** definido. Se a largura começar com um **0** (zero), o campo será preenchido com zeros à esquerda (igual a **«alinhamento» 0=**)

**,** Exibir **,** (vírgula) como separador de milhares. Para obter outros separadores de milhares, use o **«tipo» n**.

**«.precisao»** Um **.** (ponto) seguido de um inteiro cuja função depende do **«tipo»** especificado.
No **«tipo» s**, precisão é o máximo de caracteres a exibir
No **«tipo» f**, é o número de casas decimais após o ponto
No **«tipo» g** ou **n**, é o total de algarismos significativos

**«tipo»** Um dos caracteres abaixo. Se omitido, vale o default assinalado com * para cada tipo (ex. **d** para **int**):
- **s** **str/unicode** *
- **b** **int** como número binário
- **c** **int** como caractere Unicode correspondente
- **d** **int** como número decimal *
- **o** **int** como número octal
- **x X** **int** como número hexadecimal em caixa baixa ou alta
- **e E** **float** em notação exponencial (indicador do expoente em caixa baixa ou alta)
- **f F** **float** sem usar notação exponencial
- **g G** **float** como nos tipos **e E** ou **f F**, dependendo da precisão e do expoente *
- **n** **float** como no tipo **g**, mas usando separadores decimal e de milhares conforme o locale ativo
- **%** **float** como porcentagem, usando formato do tipo **f**, com o valor ×100, seguido do sinal **%**

```
1>> format('Diretor','.<12')
'Diretor.....'
2>> format(3.14159265, '_>+8.3f')
'__+3.142'
3>> format(123, '0= 6x')
' 0007b'
4>> format(123, '#06x')
'0x007b'
5>> format(12345678.9876,'18.10n')
'       12345678.99'
>>> import locale
>>>
locale.setlocale(locale.LC_NUMERIC,
'de_DE.UTF-8')
'de_DE.UTF-8'
6>> format(12345678.9876,'18.10n')
'     12.345.678,99'
>>>
```