

Material Complementar: Formataores especiais

A Especificação de Formatação (do inglês “*Format Specification*” - acesse a [documentação aqui](#)) oferece modificadores que podem ser utilizados em conjunto com strings para alterar a forma que ela será mostrada em tela. A especificação é bem extensa e contém diversos componentes, portanto sugiro dar uma olhadinha lá.

Sua forma é a seguinte:

```
{[nome][!conversão][:modificadores]}
```

- **Nome:** é o nome da variável que será utilizada.
- **Conversão:** notação utilizada para se converter a representação da variável passada.
- **Modificadores:** a parte de modificadores é bem complexa e possui os seguintes campos abaixo:

```
:[preenchimento]alinhamento[sinal][#][0][comprimento][grupo][.precisão][tipo]
```

Cada campo acima é opcional e é utilizado para alterar características da string que será mostrada em tela. Alguns exemplos:

- **Preenchimento:** Caracter utilizado para preencher espaços vazios da string.
- **Alinhamento:** Diz qual será o alinhamento da string resultante. "<" para alinhamento à esquerda, ">" para alinhamento à direita e "^" para centralizado.
- **Sinal:** Funciona apenas para strings contendo números. Possíveis valores: "+", "-", e espaço.
- **Comprimento:** Determina o comprimento mínimo da string resultante.

Vamos compreender através de exemplos:

1) Suponha o seguinte código:

```
valor = 5.5 / 40.0

print(
    f'Resultado original: {valor}\n'
    f'Resultado formatado: {valor:.1%}'
)
```

A saída será a seguinte;

```
Resultado original: 0.1375
```

Resultado formatado: 13.8%

Explicando os formatadores utilizados:

- O **.1** diz ao Python que a string resultante deve ter apenas uma casa decimal;
- O **%** multiplica o valor por 100 e inclui o símbolo de porcentagem **%** ao final.

2) Agora outro exemplo:

```
valor = 255
```

```
print(f'{{valor:-^10x}}')
```

O resultado será:

```
'----ff----
```

Agora vamos entender os formatadores utilizados:

- **-** é o **[preenchimento]**: é esse caracter que vai preencher os espaços vazios;
- **^** é o **[alinhamento]**: diz como a string deve ser alinhada. No caso, **^** diz que a string deve ser centralizada.
- **10** é o **[comprimento]**: diz que a string resultante deve ter 10 caracteres.
- **x** é o **[tipo]**: diz que a string deve ser convertida em hexadecimal (255 em Hexadecimal é ff, por isso o "ff" apareceu no resultado).

Agora te convido a abrir a documentação, brincar com os formatadores e testar as diversas combinações possíveis! 😊