dnc

data expert.

Data Science & Machine Learning

STRINGS E PRINTS

MATERIAL COMPLEMENTAR

COMO USAR O PRINT

MODO 1 - Usando virgula (**,**)

Para exibir **variáveis** no seu comando **print**, você pode usar uma vírgula para separar o texto da variável que você vai usar, a `,` acrescenta um **espaço automático** antes e depois da variável no seu texto quando usada.

```
fruta = 'maça'
nome = 'Priscila'
print('Eu gosto de' , fruta , 'demais‘)
# Eu gosto de maça demais
print('Meu nome é:' , nome, 'e eu gosto de: ‘, fruta )
# Meu nome é Priscila e eu gosto de maça
```

MODO 2 - Usando virgula (**+**)

Você também pode usar `+` para exibir as variáveis, porém diferente do uso da `,` , não se aplica espaços automáticamente.

```
fruta = ‘maça’
nome = ‘Priscila’
print(‘Meu nome é: ’  + nome +  ' e eu gosto de: ‘ + fruta )
# Meu nome é Priscila e eu gosto de maça
```

dnc

MODO 3 - Usando format.()

O comando **format** ajuda você a organizar melhor seu print, passando todas as variáveis no **final do comando**:

```
nome= 'Vitor'
idade = 24
altura = 1.84
print('Nome: {} - idade: {} - altura: {} '.format( nome, idade, altura))
# Nome: Vitor - idade: 24 - altura: 1.84
```

MODO 4 - Concatenando usando o comando f

A partir da **versão 3.6** do Python, podemos usar esse comando que deixa bem mais **enxuto** o seu comando, nesse comando você começa o print com f, seguido pelo texto, nesse texto você colocará as variáveis dentro de {} .

```
nome= 'Vitor'
idade = 24
altura = 1.84
print(f'Nome: {nome} - idade: {idade} - altura: {altura}' )
# Nome: Vitor - idade: 24 - altura: 1.84
```

MODO 5 - Brincando com formatações.

Com o format, também conseguimos configurar **caracteres de preenchimento**, **alinhamento**, **larguras** mínima e máxima, e também **casas decimais.**
Para isso vamos preencher dentro do {} com parametros, explicando:
{(posicao): (preenchimento) (alinhamento) (quantidade de caracteres) (precisao)}
Exemplo: {**0:.>10f**}

- **0** = posição da variável na lista de variáveis.
- **.** = preenche os espaços vazios com `.`.
- **>** = alinhamento do texto na direita espaço. ex: `>` (direita) `<` (esquerda) `^` (centro)
- **10** = quantidade de caracteres máximos na exibição dessa variável.
- f = precisão da variável, ex: `d` (inteiro), `f` (float).

O uso desses campos são opcionais, podemos usar somente alguns, como por exemplo:

```
texto = 'texto exemplo'
print('texto antes | {:.>20} | texto depois'.format(texto))
# texto antes |.......texto exemplo| texto depois
print('texto antes | {:.<20} | texto depois''.format(texto))
# texto antes |texto exemplo.......| texto depois
print('texto antes | {:.^20} | texto depois''.format(texto))
# texto antes |...texto exemplo....| texto depois

# brincando com casas decimais:
numero = 45.9567
print('texto antes | {:.2f} | texto depois'.format(numero ))
# texto antes | 45.96 | texto depois
```

dnc