

Programação em Python

Comando *print()*

Prof. Daniel Di Domenico

https://github.com/danidomenico/gex003_algprog

Comando **print()**

- Utilizado para saídas de dados
- Impressão de vários tipos de dados na tela (numéricos, texto, etc)
- Separador de dados: vírgula

```
print(9)
9
print("9")
9
i = 1
print("Valor de i =", i)
Valor de i = 1
```

Parâmetros do comando **print()**

- **sep**
 - Indica o separador entre os dados
 - Padrão: " " (espaço em branco)

```
print("a", "b")
```

a b

```
print("a", "b", sep="")
```

ab

```
print("a", "b", sep=":-)")
```

a:-)b

```
print(192, 168, 178, 42, sep=".")
```

192.168.178.42

Parâmetros do comando **print()**

- **end**
 - Indica o que será impresso ao final do comando
 - Padrão: `\n` (nova linha)

```
print("a", "b", end=" - Aqui é o final da linha")
```

a b - Aqui é o final da linha

```
for i in range(4):  
    print(i, end=" ")
```

0 1 2 3

```
for i in range(4):  
    print(i, end=" :-) ")
```

0 :-) 1 :-) 2 :-) 3

Parâmetros do comando **print()**

- **file**
 - Arquivo onde o conteúdo do comando será impresso
 - Padrão: `sys.stdout` (saída do terminal)

```
#A saída do comando print será impressa no arquivo saida.txt  
arquivo = open("saida.txt", "w")  
print("Teste de impressão de comando em arquivo", file=arquivo)  
arquivo.close()
```

Formatação da saída

- Utiliza parâmetros para inserir valores dentro de uma string
- Deve-se chamar o método **format** a partir da string:

```
texto = "1º parâmetro: {0}, 2º parâmetro: {1}".format(47, 11)  
print(texto)
```

1º parâmetro: 47, 2º parâmetro: 11

```
texto = "2º parâmetro: {1}, 1º parâmetro: {0}".format(47, 11)  
print(texto)
```

2º parâmetro: 11, 1º parâmetro: 47

```
texto = "1º parâmetro: {}, 2º parâmetro: {}".format(47, 11)  
print(texto)
```

1º parâmetro: 47, 2º parâmetro: 11

Formatação da saída

- Caso numerados, os parâmetros podem ser utilizados mais de uma vez:

```
texto = "Imprime uma vez: {0}, e depois outra: {0}".format(47)
print(texto)
```

Imprime uma vez: 47, e depois outra: 47

- Os parâmetros também podem ser nomeados:

```
texto = "A: {a}, B: {b}, D: {d}, C: {c}".format(a=1, b="BB", c=5, d=8.1)
print(texto)
```

A: 1, B: BB, D: 8.1, C: 5

Formatação da saída

- Números inteiros são formatados (precisão) utilizando o caracter **d**
 - Colocar a precisão após a identificação do parâmetro (**:5d**)
 - Define o mínimo de casas que serão deixadas à esquerda

```
#Cinco casas à esquerda
print("Número inteiro: {0:5d}".format(12))
print("Número inteiro: {0:5d}".format(12254))
print("Número inteiro: {0:5d}".format(487515))
Número inteiro:    12
Número inteiro: 12254
Número inteiro: 487515
```


Formatação da saída

- Números reais são formatados (precisão) utilizando o caracter **f**
 - Colocar a precisão após a identificação do parâmetro (**:8.2f**)
 - Define o mínimo de casas totais e após o separador decimal

```
#Oito casas ao todo, duas decimais
print("Número real: {0:8.2f}".format(12))
print("Número real: {0:8.2f}".format(12.65478))
print("Número real: {0:8.2f}".format(45664.123456))
print("Número real: {0:8.2f}".format(445545.123456)) #Aqui 9 casas
Número real:      12.00
Número real:      12.65
Número real: 45664.12
Número real: 445545.12

#Quatro decimais
print("Número real: {0:.4f}".format(45664.123456))
Número real: 45664.1235
```

Formatação da saída

- Mais sobre formatação de strings:

https://www.python-course.eu/python3_formatted_output.php