



- Saída de Dados
- Operadores
- Strings
- Condicionais (if)

Python

Lógica de Programação

Principais conceitos da Linguagem Python

01 Saída de Dados

- Usamos a função print para apresentação dos dados
- No exemplo abaixo:
 - Na linha 1, a variável saldo recebe um valor com 5 casas decimais
 - Na linha 2, exibimos o saldo com as 5 casas decimais
 - Na linha 3, imprimimos uma linha em vazio
 - Por fim, na linha 4, exibimos o mesmo saldo (da linha 2), mas com apenas 2 casas decimais
 - Verifique que, na parte `%.2f`, o `f` se refere a *float*, enquanto que o número 2, se refere a quantidade de casas decimais
 - Se fosse `int`, seria `d` (ao invés de `f`)

```
1 saldo = 1000.50123
2 print(' o saldo é ',saldo)
3 print()
4 print(" o saldo é %.2f " %saldo)
```

```
1 nome = 'Daniel '
2 sobrenome = 'Abella'
3 nomeCompleto = nome + sobrenome
4 print(' o meu nome completo é', nomeCompleto)
```

- No exemplo acima, criamos duas variáveis (nome e sobrenome) e, uma terceira variável que é a junção das duas anteriores. A esta operação de junção, comumente chamamos de concatenação.
 - A concatenação deve sempre ser entre *str* (Strings) e inteiros, sem poder misturar. Ou seja, o exemplo a seguir (misturando *str* e *int*) não funciona em Python.

```
1 nome = 'Daniel '
2 idade = 26
3 nomeComIdade = nome + idade
4 print(' nome com idade é ', nomeCompleto)
```



- No exemplo a seguir, apresentamos as principais operações representadas na tabela anterior.

```
1 print(3+2)      5
2 print(3-2)      1
3 print(3*2)      6
4 print(3/2)      1.5
5 print(3**2)     9
6 print(3%2)      1
7 print(22//8)    2
```

Saída

02 Operadores Matemáticos

- Na tabela a seguir, apresentamos os principais operadores matemáticos disponíveis no Python.

Operador	Operação	Exemplo
**	Potência	2 ** 3 = 8
%	Módulo	5 % 2 = 1
//	Divisão inteira	22 // 8 = 2
*	Multiplicação	3 * 8 = 24
-	Subtração	8 - 2 = 6
+	Soma	2 + 2 = 4
/	Divisão	33 / 4 = 8.25

- O operador `+`, que se refere obviamente a soma, pode ser utilizado em números (*int* e *float*), bem como em strings (*str*).

- Como sempre, as mesmas operações feitas acima com literais, podem ser feitas com uso de variáveis (lidas ou não pelo usuário)

```
1 membro1 = 3
2 membro2 = 2
3
4 print(membro1+membro2)      5
5 print(membro1-membro2)      1
6 print(membro1*membro2)      6
7 print(membro1/membro2)      1.5
8 print(membro1**2)           9
9 print(membro1%membro2)      1
10 print(membro1//membro2)     1
```

Saída

03 Trabalhando com Strings (str)

- Para armazenar caracteres (que ainda assim, podem ser números), usamos o tipo str (String). No exemplo a seguir, apresentamos um exemplo do uso de Strings.
 - Para declarar Strings, basta atribuir um valor entre aspas simples ou duplas a uma variável
- Nas linhas 1 a 3, declaramos 3 variáveis. Verifique que, a variável idade, tende a ser int, mas ainda assim, podemos declarar como string, pois a colocamos entre aspas
- Nas linhas seguintes, apenas apresentamos os textos concatenados

```
1 nome = 'Daniel '  
2 sobrenome = 'Abella'  
3 idade = '36'  
4  
5 print(nome + sobrenome)  
6 print('Daniel' + 'Abella')  
7 print('possui a idade ', idade)
```

- Para finalizar, no exemplo a seguir apresentamos o uso do método `len`, que se refere a `length`, que é o tamanho da String. Note que, imprimimos o tamanho de duas maneiras o tamanho da String 'Daniel', que possui 6 caracteres.
 - Na primeira (linha 2), chamamos diretamente o método `len` durante o `print`
 - A segunda abordagem (linhas 4 e 5), atribui o tamanho da String a uma variável antes de entrar no `print`.

```
1 nome = 'Daniel'  
2 print('o tamanho do nome tem', len(nome), 'caracteres')  
3  
4 tamanhoTexto = len(nome)  
5 print('o tamanho do nome tem', tamanhoTexto, 'caracteres')
```

- O uso de operadores matemáticos com Strings é funcional
 - O operador soma, funciona para juntar 2 Strings (concatenar)
 - Os operadores subtração e divisão não possuem funcionamento com Strings
 - Como vou fazer 'Daniel'/? 'Daniel'-2?
 - O operador multiplicação possui uso no Python
 - 'Daniel'*3 exibe DanielDanielDaniel
 - O exemplo a seguir demonstra o funcionamento do que acabamos de discutir. As linhas 1,2,3 e 6 funcionam, enquanto todas as outras apresentam erro.

```
1 nome = 'Daniel '  
2 sobrenome = 'Abella'  
3 print(nome + sobrenome)  
4 print(nome + 3)  
5 print(nome - 3)  
6 print(nome * 3)  
7 print(nome / 3)
```

- Dentre as operações mais comuns com Strings, são a de tomar uma string em maiúscula, minúscula ou substituir um dado caractere por outro. Um exemplo das 3 operações é apresentado a seguir:
 - Na linha 3, colocamos 'Daniel' em maiúsculo
 - Na linha 6, em minúsculo
 - Na linha 9, substituímos o caractere 'd' de Daniel por 'r', tomando-o 'Raniel'

```
1 nome = 'Daniel '  
2  
3 nome = nome.upper()  
4 print(nome)  
5  
6 nome = nome.lower()  
7 print(nome)  
8  
9 nome = nome.replace('d','r')  
10 print(nome)
```

04 Condicionais (If)

- Outra coisa muito importante em programação é o uso de condicionais e aqui vamos aprender o uso do IF (Se, em português). No exemplo a seguir, apresentamos o texto 'é maior de idade' caso a variável idade tenha valor `>=` (maior ou igual) a 18.

```
1 idade = 18  
2  
3 if idade >= 18 :  
4 |   print('é maior de idade')  
5   print('fim do programa')
```

- Ainda com relação ao exemplo anterior, verifique que, o `print` da linha 4, possui um recuo para indicar que, esse `print` só será executado se a condição do `if` (maior de 18) for verdadeira.
 - E, neste exemplo, é verdadeiro 😊
- A linha 5, independente do `if` ser executado ou não, será impressa, pois não possui um recuo, de modo que, entende-se que não pertence ao `if` e será executado sempre.
- Para melhor entendimento do exemplo, apresentamos a seguir o funcionamento do `if`.
 - Note que, a linha em roxo representa o recuo necessário, indicando que linhas de código 1, 2 e 3 pertencem ao `if` (em azul)
 - E, a linha "outra linha qualquer" será apresentada sempre.

