

RESUMO PYTHON (1)



Interpretador online: <https://replit.com/languages/python3>;

Recomendado: Thonny (<https://thonny.org/>)

EXPRESSÕES ARITMÉTICAS:

- Soma: +

```
1 print (1+1)
```

```
2
```

- Subtração: -

- Divisão: / (sempre retorna ponto flutuante, ou seja, na forma decimal)

- Multiplicação: *

- Potência: **

- Resto da divisão: %

```
1 print (10 % 8)
```

```
2
```

- Parte inteira da divisão: //

```
1 print (9//8)
```

```
1
```

Sequência de resolução: 1- Potência, 2- Multiplicação e divisão, 3- Soma e subtração (mas a ordem pode ser alterada por parênteses!).

VARIÁVEIS:

Faz uma ponte entre o código e a memória do computador.

n1 (nome); = (atribuição); 10 (valor)

```
1 n1 = 10
2 print(n1)
```

```
10
```

O valor da variável pode ser mudado durante o código:

```
1 #queremos o resultado da multiplicação de n1 e n2 com +1
2 n1 = 10
3 n2 = 20
4 resultado = n1 * n2
5 resultado = resultado + 1
6 print (resultado)
```

```
201
```

OBS: com o # podemos comentar o código, isso significa que o interpretador não lerá essa linha. Isso é usado principalmente para explicar o que você está pensando ao desenvolver o código de tal forma, caso outra pessoa tenha acesso a ele.

➔ Alguns tipos de variáveis

n1 = 'abc' (String [usamos '' ou ""])

```
n1 = true (Lógico [bool])
n1 = 1.5 (Ponto flutuante [float])
n1 = 10 (Inteiro [int])
```

!!!!OBS: em outras linguagens, como java e C/C++ é um pouquinho diferente, precisamos declarar o tipo de variável

```
int pin_LDR = A0;
int vLDR = 0;
```

Int (tipo); pin_LDR (nome); = (atribuição); A0 (valor)

ENTRADA E SAÍDA

Entrada é tudo o que entra e saída o que sai :) Por exemplo, o teclado é uma entrada, pois estamos inserindo informações, enquanto a tela é uma saída, ela exibe essas informações. Para entrada usaremos o `input()` e para saída o `print()`.

```
1 nome = input ('digite seu nome: ')
2 idade = input ('digite sua idade: ')
3 cidade = input ('digite sua cidade: ')
4 print (f'Então seu nome é {nome}, você tem {idade} anos,
e você está em {cidade}. Legal!')
```

```
digite seu nome: Duda
digite sua idade: 18
digite sua cidade: Blumenau
Então seu nome é Duda, você tem 18 anos, e você está em Blumenau.
Legal!
```

Porém, todas as vezes que usamos o input, a variável fica armazenada como string. Logo, nome, idade e cidade são textos, o que torna impossível fazer contas com eles. Isso está indicado no exemplo abaixo. A parte em vermelho indica após *TypeError* que não podemos fazer a soma da idade com o número 1, porque seria a soma de um texto com um número inteiro.

```
1 nome = input ('digite seu nome: ')
2 idade = input ('digite sua idade: ')
3 cidade = input ('digite sua cidade: ')
4 print (f'Então seu nome é {nome}, você vai fazer {idade +
1 } anos, e você está em {cidade}. Legal!')
```

```
digite seu nome: Duda
digite sua idade: 18
digite sua cidade: Blumenau
Traceback (most recent call last):
  File "main.py", line 4, in <module>
    print (f'Então seu nome é {nome}, você vai fazer {idade + 1 }
anos, e você está em {cidade}. Legal!')
TypeError: can only concatenate str (not "int") to str
```

Então precisamos converter esse texto em número, e para isso usamos `int()` ou `float()`

```
1 nome = input ('digite seu nome: ')
2 idade = input ('digite sua idade: ')
3 cidade = input ('digite sua cidade: ')
4
5 idade = int (idade)
6
7 print (f'Então seu nome é {nome}, você vai fazer {idade +
1 } anos, e você está em {cidade}. Legal!')
```

```
digite seu nome: Duda
digite sua idade: 18
digite sua cidade: Blumenau
Então seu nome é Duda, você vai fazer 19 anos, e você está em Blum
enau. Legal!
```

OBS: sobre operações com strings recomendo procurar no livro *Pense em Python* (tem no drive). Não colocarei aqui para não ficar muito extenso.

OPERADORES RELACIONAIS

Sempre resultam em um valor lógico (True ou False)

`==` (igualdade); `!=` (diferença)

```
1 n1 = 1==1
2 print(n1)
```

True

```
1 n1 = 1!=1
2 print(n1)
```

False

`<` (menor que); `>` (maior que); `<=` (menor ou igual que); `>=` (maior ou igual que)

```
1 media_final = 5
2 media_necessaria = 5.75
3 resultado = media_final >= media_necessaria
4 print(resultado)
```

False

```
1 media_final = 6
2 media_necessaria = 5.75
3 resultado = media_final >= media_necessaria
4 print(resultado)
```

True

OPERADORES LÓGICOS

Não = `not`, Ou = `or`, E = `and`

```
1 print(not True)
```

False

```
1 media_final = 6
2 media_necessaria = 5.75
3 frequencia_final = 80
4 frequencia_necessaria = 75
5 resultado_media = media_final >= media_necessaria
6 resultado_freq = frequencia_final >= frequencia_necessaria
7 aprovado = resultado_media and resultado_freq
8 print(aprovado)
```

True

```
1 media_final = 5
2 media_necessaria = 5.75
3 frequencia_final = 80
4 frequencia_necessaria = 75
5 resultado_media = media_final >= media_necessaria
6 resultado_freq = frequencia_final >= frequencia_necessaria
7 aprovado = resultado_media and resultado_freq
8 print(aprovado)
```

False

OBS: prestar atenção nas diferenças do `or` e do `and`, segue a tabela de valores obtidos e do valor final:

Valores inseridos	and	or
True True	True	True
True False	False	True
False True	False	True
False False	False	False