

## RESUMO PYTHON (2)



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

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

### IF / ELSE / ELSE IF

Caso haja o cumprimento de certa condição, algumas ações serão realizadas.

[inicialização]

if [condição] :

[ações a serem realizadas]

```
1 n1 = 4
2
3 if (n1 > 1):
4     print ('a condição foi satisfeita!')
```

a condição foi satisfeita!

OBS: o afastamento da margem esquerda do texto que informa as ações a serem realizadas. Caso esteja alinhado com o if, não é considerado dentro do bloco do if

```
1 n1 = 4
2
3 if (n1 < 1):
4     print ('a condição foi satisfeita!')
5 print ('esse print está fora do bloco do if, então mesmo
se a condição não for satisfeita, esse texto será
exibido')
```

esse print está fora do bloco do if, então mesmo se a condição for satisfeita, esse texto será exibido

O Else é usado para realizar ações caso a condição não seja satisfeita:

```
1 n1 = 4
2
3 if (n1 < 1):
4     print ('a condição foi satisfeita!')
5 else:
6     print ('a condição NÃO foi satisfeita!')
```

a condição NÃO foi satisfeita!

Porém, vemos que o else é muito geral. E se quisermos testar outras condições? Usamos o else if :

```
1 n1 = 4
2
3 if (n1 == 1):
4     print ('n1 é igual a 1')
5 else:
6     if (n1 == 2):
7         print ('n1 é igual a 2')
8     else:
9         if (n1 == 3):
10            print ('n1 é igual a 3')
11        else:
12            if (n1 == 4):
13                print ('n1 é igual a 4')
```

n1 é igual a 4

Contudo, podemos “juntar” o else if e escrever só elif, buscando deixar o código mais organizado

```
1 n1 = 3
2
3 if (n1 == 1):
4     print ('n1 é igual a 1')
5 elif (n1 == 2):
6     print ('n1 é igual a 2')
7 elif (n1 == 3):
8     print ('n1 é igual a 3')
9 elif (n1 == 4):
10    print ('n1 é igual a 4')
```

n1 é igual a 3

OBS: em C/C++ é um pouco diferente. Ao invés de usar os dois pontos e a indentação, usamos as chaves {} para indicar o que está dentro do bloco do if.

```
if (dswitch==1)
{
    digitalWrite (pinmotor,HIGH);
    Serial.println ("1");
}
else
{
    digitalWrite (pinmotor,LOW);
}
```

## WHILE

Enquanto a condição for satisfeita, o bloco de código será executado.

[inicialização]

while [condição]:

Ação a ser executada

[atualização]

```
1 #queremos contar até 10
2 n1 = 1
3
4 while (n1 <= 10):
5     print (n1)
6     n1 = n1 + 1
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10

OBS: a questão da indentação continua.

## BREAK E CONTINUE

BREAK: sai imediatamente do laço.

Nesse exemplo estaremos somando os números inseridos. Porém, se for inserido um número negativo a soma será encerrada, e será exibido o valor da soma até o momento.

```
1 #soma dos números positivos
2 lidos = 0
3 soma = 0
4
5 while (lidos < 5):
6     n = int (input('digite um número: '))
7     if (n >= 0):
8         soma = soma + n
9         lidos = lidos + 1
10    else:
11        print ('Não pode ser negativo. Desisto!')
12        break
13
14 print (soma)
```

```
digite um número: 3
digite um número: 3
digite um número: -3
Não pode ser negativo. Desisto!
6
>
```

MAS é opcional, pois podemos simplesmente “sair” da condição do while

```
1 #soma dos números positivos
2 lidos = 0
3 soma = 0
4
5 while (lidos < 5):
6     n = int (input('digite um número: '))
7     if (n >= 0):
8         soma = soma + n
9         lidos = lidos + 1
10    else:
11        print ('Não pode ser negativo. Desisto!')
12        lidos = 5
13
14 print (soma)
```

```
digite um número: 3
digite um número: 3
digite um número: -3
Não pode ser negativo. Desisto!
6
>
```

CONTINUE: Para de executar o bloco e volta para a condição.

No próximo exemplo também estaremos fazendo a soma dos números inseridos, porém quando o código receber um número negativo esse será desconsiderado.

```
1 #soma dos números positivos
2 lidos = 0
3 soma = 0
4
5 while (lidos < 5):
6     n = int (input('digite um número: '))
7     if (n < 0):
8         print('não pode ser negativo!')
9         continue
10    soma = soma + n
11    lidos = lidos + 1
12
13
14 print (soma)
```

```
digite um número: 1
digite um número: 1
digite um número: -1
não pode ser negativo!
digite um número: 1
digite um número: 1
digite um número: 1
5
>
```

MAS também é descartado simplesmente adicionando um else, ou reorganizando o código.

```
1 #soma dos números positivos
2 lidos = 0
3 soma = 0
4
5 while (lidos < 5):
6     n = int (input('digite um número: '))
7     if (n < 0):
8         print('não pode ser negativo!')
9     else:
10        soma = soma + 1
11        lidos = lidos + 1
12
13
14 print (soma)
```

```
digite um número: 1
digite um número: 1
digite um número: -1
não pode ser negativo!
digite um número: 1
digite um número: 1
digite um número: 1
5
>
```