



escola  
britânica de  
artes criativas  
& tecnologia

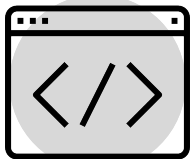
# Python para Análise de Dados



# DE OLHO NO CÓDIGO



# Fluxo condicional e repetição



Confira boas práticas da linguagem Python por assunto relacionado às aulas.

- **Conheça as estruturas condicionais if / else / elif**
- **Aprenda as estruturas condicionais try / except / finally**
- **Saiba sobre a estrutura de repetição for / in**
- **Referência Bibliográfica**



# Conheça as estruturas condicionais if / else / elif

De olho no código

- Estruturas condicionais são o princípio básico de lógica. Utilizamos essas estruturas para comparar dois valores e tomar uma decisão. Sendo assim, podemos comparar por exemplo quais das variáveis é maior  $x=1$  e  $y=2$ .

```
if (x > y):
    print("X é maior que Y")
```

```
if (x < y):
    print("Y é menor que X")
```



# Conheça as estruturas condicionais if / else / elif

De olho no código

- Mas, para facilitar as nossas decisões, temos algumas outras estruturas que podemos usar:

**elif <2ª comparação lógica> == True:**

<execute este código se a segunda condição for verdade>

//utilizamos esse quando temos dois pontos a serem validados.

**else:**

<senão execute este código>

//e caso nenhum dos dois seja verdadeiro, utilizamos esse.



# Aprenda as estruturas condicionais try / except / finally

De olho no código

- Exceções são criadas para prevenção de erros, por isso as utilizamos para tratar algum cenário que poderia não ser previsto. No exemplo a seguir, identificamos uma operação que não é possível obter o resultado e retornar ao usuário a informação.

```

try:
    valor_por_pessoa = preco / pessoas
    print(valor_por_pessoa)
except ZeroDivisionError:
    print('Número de pessoas inválido. Espera-se um
    valor maior que 0 e obteve-se um valor igual a ' +
    str(pessoas))
  
```

Para mais informações,  
 acesse [Try e Except no Python](#).



# Saiba sobre a estrutura de repetição `for / in`

De olho no código

- Repetir tarefas é o que mais fazemos no nosso dia a dia. Ao esperar uma encomenda, olhamos o celular inúmeras vezes para saber se a encomenda já saiu para entrega. Isso se trata de um **laço de repetição**: treinamos o nosso cérebro para que, enquanto a encomenda não sair para entrega, confirmaremos seu status no celular, e caso, ela saia para entrega, encerramos a validação.

Na programação não é diferente, criamos rotinas através do `for / in`.

Para aprofundar seus conhecimentos sobre laços, acesse [estruturas de repetição](#)



# Referências Bibliográficas

- <https://www.devmedia.com.br/for-python-estrutura-de-repeticao-for/38513>
- <https://www.hashtagtreinamentos.com/try-e-except-no-python>
- <https://blog.somostera.com/desenvolvimento-web/dicionario-python>





# Bons estudos!

