



escola
britânica de
artes criativas
& tecnologia

Python para Análise de Dados



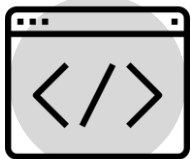
BCA

escola
britânica de
artes criativas
& tecnologia

DE OLHO NO CÓDIGO



Python para Análise de Dados



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

- **Desvende as variáveis**
- **Descubra os números**
- **Conheça as strings**
- **Aprenda booleanos**
- **Referência Bibliográfica**



Desvende as variáveis

De olho no código

- Cada nome de variáveis é uma letra ou o símbolo '_' (underscore) seguida de letras, números e '_'. Atenção, maiúscula é diferente de minúscula. Assim `maior` e `Maior` são duas variáveis diferentes.
- O caractere para sublinhado (`_`) pode aparecer em um nome. Ele é muito utilizado em nomes com múltiplas palavras, tal como em `meu_nome` ou `preco_do_cha_na_china`.



Desvende as variáveis

De olho no código

- Palavras reservadas definem as regras e a estrutura da linguagem e não podem ser usadas como nomes de variáveis. Python tem 33 palavras reservadas:

<code>and</code>	<code>def</code>	<code>for</code>	<code>is</code>	<code>raise</code>	<code>False</code>
<code>as</code>	<code>del</code>	<code>from</code>	<code>lambda</code>	<code>return</code>	<code>None</code>
<code>assert</code>	<code>elif</code>	<code>global</code>	<code>nonlocal</code>	<code>try</code>	<code>True</code>
<code>break</code>	<code>else</code>	<code>if</code>	<code>not</code>	<code>while</code>	
<code>class</code>	<code>except</code>	<code>import</code>	<code>or</code>	<code>with</code>	
<code>continue</code>	<code>finally</code>	<code>in</code>	<code>pass</code>	<code>yield</code>	

Para aprofundar seus conhecimentos em variáveis, acesse o capítulo 2, Variáveis, expressões e comandos, do livro [Aprenda computação com Python](#).



Descubra os números

De olho no código

- A função `type()` retorna o tipo do valor passado como parâmetro.

No exemplo:

`print(type(37))`, a função `type()` retorna `<class 'int'>` indicando que o valor é de fato um número inteiro.

- Em operações envolvendo números decimais `float`, há uma limitação de dígitos imposta por essa notação. Por isso, ao realizarmos operações, os resultados de cálculos realizados pelos computadores são aproximações dos valores reais.

Para aprofundar seus conhecimentos sobre funções, acesse o site [Pense Python](#).



Descubra os números

De olho no código

- Quando se realiza diversos cálculos em operações de adição e subtração de números de decimais de magnitudes muito diferentes são envolvidos, os dígitos menos significativos são perdidos. Essa perda pode gerar consequências insignificantes ou um grande erro no resultado.
- As operações de divisão e multiplicação de números decimais podem ser executadas em qualquer ordem;

Para saber mais sobre números decimais `float` e boas práticas para minimizar seus possíveis erros, acesse o site [The Floating-point Guide](#).



Conheça as strings

De olho no código



- Python aceita aspas simples ou duplas para delimitar strings.
- Perceba o exemplo: `"17"` e `"3.2"`. Eles parecem números, mas estão entre aspas, então são strings.
- Para usar strings formatadas, comece uma string com `f` ou `F`, antes de abrir as aspas. Dentro dessa string, pode-se escrever uma expressão Python entre caracteres `{e}`, que podem se referir a variáveis, ou valores literais.
- Para acessar um elemento de uma lista, usamos o operador de indexação `[]`. A expressão dentro dos colchetes especifica o índice. O índice do primeiro elemento é 0.

Conheça as strings

De olho no código

- Em operações de fatiamento de *strings*, dizemos que uma lista é uma estrutura sequencial indexada pois os seus elementos podem ser acessados sequencialmente utilizando índices. O primeiro elemento da lista tem índice **0**, o segundo tem índice **1**, e assim por diante.

Para aprofundar seus conhecimentos sobre métodos de strings, acesse o conteúdo Python String Methods do site [W3 School](#).



Aprenda booleanos

De olho no código

- Em operadores de comparação de strings, um erro comum é usar um sinal de igual sozinho (=) em vez de um duplo (==). Lembre-se de que = é um operador de atribuição e == é um operador de comparação.



Para aprofundar seus conhecimentos sobre métodos de strings, acesse o conteúdo Python String Methods do site [W3 School](https://www.w3schools.com/python/python_strings_methods.asp).



Referências Bibliográficas

- https://www.w3schools.com/python/python_ref_string.asp
- <https://floating-point-gui.de/errors/propagation/>
- <https://panda.ime.usp.br/pensepy/static/pensepy/index.html>
- https://mange.ifrn.edu.br/python/aprenda-com-py3/capitulo_02.html



Bons estudos!

