

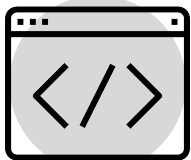
Profissão: Cientista de Dados



BOAS PRÁTICAS



Elementos básicos de Python



- **Conheça o Python**
- **Compreenda os tipos de dados**
- **Analise as estruturas**
- **Compreenda a sintaxe**



Conheça o Python

- **Entenda a ordem das operações:** a ordem das operações em Python é importante para garantir que seus códigos sejam executados corretamente. Use parênteses para alterar a ordem padrão das operações, se necessário.
- **Siga uma ordem lógica:** ao escrever código em Python, especialmente no Jupyter Notebook, é importante seguir uma ordem lógica, pois o código é executado de cima para baixo.



Compreenda os tipos de dados

- Sempre verifique o tipo de dados que você está trabalhando. Isso é especialmente importante ao lidar com números e strings, pois um número entre aspas é considerado uma string, não um número. Use a função `type()` para isso.
- Ao trabalhar com strings, lembre-se de que a função `len()` pode ser usada para determinar o tamanho da string.



Compreenda os tipos de dados

- Lembre-se de que a **divisão de inteiros** em Python resulta em um **float**, mesmo que os números envolvidos sejam inteiros. Isso pode ser importante ao lidar com cálculos numéricos.
- Ao trabalhar com dados **booleanos**, lembre-se de que eles só podem ter dois valores: **True ou False**. Eles são usados para representar o resultado de uma condição lógica.



Analise as estruturas

- Use **listas** para coleções de **dados ordenadas e mutáveis**. As listas são úteis quando você precisa de uma coleção de itens que pode ser alterada e que mantém uma ordem específica.
- Use **tuplas** para **coleções de dados imutáveis**. As tuplas são úteis quando você precisa de uma coleção de itens que não deve ser alterada.



Analise as estruturas

- Use **conjuntos** para **coleções de dados únicos**. Conjuntos são úteis quando você precisa de uma coleção de itens únicos, e não se importa com a ordem.
- Use **dicionários** para **pares de chave-valor**. Dicionários são úteis quando você precisa associar valores a chaves específicas.



Analise as estruturas

- Aproveite as operações de conjunto. Se você está trabalhando com conjuntos, pode usar operações de conjunto, como união e interseção, para manipular seus dados.
- Lembre-se de que **as tuplas e os conjuntos** requerem **sintaxe especial**. As tuplas são definidas por valores entre parênteses, enquanto os conjuntos são definidos por valores entre chaves.



Analise as estruturas

- Use o método '**append**' para **adicionar elementos a uma lista**. Este é um método fácil e eficiente para adicionar novos itens a uma lista existente.
- Acesse os valores em um dicionário usando a chave correspondente. Esta é uma maneira eficiente de recuperar um valor específico de um dicionário.



Conheça a sintaxe

- Use comentários para explicar o que seu código faz. Isso é especialmente útil para pessoas que não estão familiarizadas com seu projeto ou para você mesmo quando revisar o código no futuro.
- Aproveite as estruturas de decisão condicional, como "if", para executar código com base em condições booleanas. Isso pode ajudar a tornar seu código mais eficiente e fácil de entender.



Conheça a sintaxe

- Crie módulos e pacotes para trechos de código que você usa com frequência. Isso pode ajudar a organizar seu código e torná-lo mais reutilizável.
- Pratique o que você aprende através de exercícios e revisão do material da aula. A prática é a chave para se tornar proficiente em qualquer habilidade, incluindo a programação em Python.



Bons estudos!

