

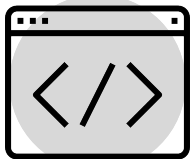
Profissão: Analista de dados



DE OLHO NO CÓDIGO



Coleta de dados I



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

- **Manuseie arquivos CSV**
- **Manipule arquivos TXT**
- **Manuseie arquivos de Excel**



Manuseie arquivos CSV

Os arquivos CSV (*Comma Separated Values*) são amplamente utilizados para armazenar dados em um formato de texto simples e estruturado. Aqui estão algumas boas práticas para o uso de arquivos CSV.

● Usar uma única vírgula como separador

Embora o nome sugira que o separador é uma vírgula, alguns programas podem usar outro separador por padrão, como ponto e vírgula ou tabulação. É importante garantir que um único separador seja usado consistentemente em todo o arquivo para evitar problemas ao ler os dados.

Para usar um separador definido nos arquivos CSV, é possível especificá-lo no momento em que o arquivo é lido usando uma biblioteca de leitura de arquivos CSV. A maioria das bibliotecas de processamento de dados em Python, como Pandas, Numpy e csv, permitem que você especifique um separador personalizado.



Manuseie arquivos CSV

Os arquivos CSV (*Comma Separated Values*) são amplamente utilizados para armazenar dados em um formato de texto simples e estruturado. Aqui estão algumas boas práticas para o uso de arquivos CSV.

Por exemplo, usando a biblioteca Pandas, você pode ler um arquivo CSV com um separador personalizado da seguinte forma:

```
import pandas as pd
df = pd.read_csv('arquivo.csv', sep=';')
```

Neste exemplo, o separador do arquivo CSV é definido como ;. Se o arquivo CSV usar um separador diferente, basta substituir ; pelo separador correto. A biblioteca Pandas detectará o novo separador e lerá corretamente o arquivo CSV.



Manuseie arquivos CSV

Os arquivos CSV (*Comma Separated Values*) são amplamente utilizados para armazenar dados em um formato de texto simples e estruturado. Aqui estão algumas boas práticas para o uso de arquivos CSV.

Em Python, por exemplo, as listas são criadas usando colchetes [] e podem conter diferentes tipos de dados, incluindo números, strings e outros objetos. Por exemplo, a seguinte linha de código cria uma lista de três elementos:

```
minha_lista = [1, "hello", True]
```

Uma lista pode conter outras listas como elementos. Por exemplo:

```
lista_aninhada = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```



Manuseie arquivos CSV

Os arquivos CSV (*Comma Separated Values*) são amplamente utilizados para armazenar dados em um formato de texto simples e estruturado. Aqui estão algumas boas práticas para o uso de arquivos CSV.



Saiba Mais

A escolha do separador mais adequado para arquivos CSV depende do contexto em que o arquivo é usado e das práticas comuns no país em que é compartilhado. Em alguns países, como Brasil e Portugal, a vírgula é usada como separador decimal, o que pode tornar confusa a leitura de arquivos CSV que usam a vírgula como separador de campo. Nesses casos, é recomendável usar um separador diferente, como o ponto-e-vírgula, para evitar ambiguidades. Por outro lado, em países que usam o ponto como separador decimal, a vírgula pode ser uma opção mais natural como separador de campo em arquivos CSV.

Manuseie arquivos CSV

Os arquivos CSV (*Comma Separated Values*) são amplamente utilizados para armazenar dados em um formato de texto simples e estruturado. Aqui estão algumas boas práticas para o uso de arquivos CSV.



- Incluir cabeçalhos de coluna**
 Incluir um cabeçalho de coluna descritivo no topo do arquivo pode ajudar a entender melhor os dados e evitar confusão. Os cabeçalhos também podem ser úteis ao importar os dados em um software de planilha ou banco de dados.

- Evitar caracteres especiais**
 É melhor evitar o uso de caracteres especiais, como aspas duplas, vírgulas e quebras de linha nos dados. Se eles precisarem ser usados, é importante garantir que eles sejam adequadamente citados para que o software possa entender corretamente o arquivo.

Manuseie arquivos CSV

Os arquivos CSV (*Comma Separated Values*) são amplamente utilizados para armazenar dados em um formato de texto simples e estruturado. Aqui estão algumas boas práticas para o uso de arquivos CSV.



Validar os dados

É importante garantir que os dados sejam válidos e estejam no formato correto antes de salvar em um arquivo CSV. Isso pode ajudar a evitar erros ao importar os dados para outros programas ou ao usá-los em análises.




Especificar a codificação

Para garantir que os caracteres sejam exibidos corretamente em diferentes sistemas operacionais e softwares, é importante especificar a codificação do arquivo. UTF-8 é amplamente suportado e recomendado.



Manuseie arquivos CSV

Os arquivos CSV (*Comma Separated Values*) são amplamente utilizados para armazenar dados em um formato de texto simples e estruturado. Aqui estão algumas boas práticas para o uso de arquivos CSV.

- 
Fornecer documentação
 Fornecer documentação sobre o formato e a estrutura dos dados pode ajudar outros usuários a entender e trabalhar com os dados com mais facilidade. Isso pode incluir informações sobre os tipos de dados, unidades de medida e outras considerações relevantes.



Manipule arquivos TXT

O regex (expressão regular) é uma ferramenta poderosa para análise de dados que permite pesquisar padrões em strings e textos.

Acompanhe algumas boas práticas para o uso do regex.

• Compreender a sintaxe

O regex tem sua própria sintaxe e regras para a definição de padrões. É importante ter uma compreensão sólida dessas regras antes de começar a usá-lo para evitar erros e otimizar o desempenho.

• Testar as expressões regulares

Testar as expressões regulares com antecedência para garantir que elas estejam captando corretamente os padrões desejados. Existem muitas ferramentas online disponíveis para testar expressões regulares.



Manipule arquivos TXT

O regex (expressão regular) é uma ferramenta poderosa para análise de dados que permite pesquisar padrões em strings e textos.

Acompanhe algumas boas práticas para o uso do regex.



● Escolher o melhor mecanismo regex

Existem vários mecanismos regex disponíveis e cada um tem seus próprios recursos e limitações. É importante escolher o melhor mecanismo para a tarefa em questão, considerando a linguagem de programação e o ambiente em que o regex será usado.

● Manter as expressões regulares simples

Embora seja possível criar expressões complexas, é importante manter as expressões regulares simples sempre que possível. Isso pode melhorar a legibilidade e o desempenho.

Manipule arquivos TXT

O regex (expressão regular) é uma ferramenta poderosa para análise de dados que permite pesquisar padrões em strings e textos.

Acompanhe algumas boas práticas para o uso do regex.

'\S+@\S+' é uma expressão regular que corresponde a uma sequência de caracteres que tenha o formato de um endereço de e-mail. Mais especificamente, a expressão regular começa com \S+, que corresponde a um ou mais caracteres que não sejam espaços em branco. Em seguida, é esperado um sinal de arroba (@) literal, seguido novamente por \S+, que corresponde a um ou mais caracteres que não sejam espaços em branco. Essa expressão regular é comumente usada para encontrar endereços de e-mail em textos ou em conjunto de dados, como por exemplo em arquivos CSV.

Para saber mais sobre expressões regulares, acesse o artigo [7 Best Regular Expression Courses for Developers to Learn in 2023](#).
Para testar seu Regex, acesse [Regexr: Learn, Build, & Test Regex](#).



Manipule arquivos TXT

O regex (expressão regular) é uma ferramenta poderosa para análise de dados que permite pesquisar padrões em strings e textos.

Acompanhe algumas boas práticas para o uso do regex.

- Considerar a segurança**
 Como as expressões regulares podem ser usadas para injetar código malicioso, é importante considerar a segurança ao usar o regex. Validar cuidadosamente os dados de entrada e limitar o escopo da expressão regular sempre que possível.

- Usar grupos de captura**
 Usar grupos de captura para extrair informações específicas de uma string ou texto. Isso pode ajudar a simplificar o processamento e a manipulação dos dados.



Manuseie arquivos de Excel

Openpyxl é uma biblioteca Python que permite aos usuários ler, escrever e manipular arquivos do Microsoft Excel (.xlsx e .xlsm). Pode ser usado para automatizar tarefas repetitivas, gerar relatórios, extrair dados de arquivos do Excel etc. Confira as dicas a seguir.

● Usar a função `with open()` para abrir arquivos Excel

Essa prática garante que o arquivo seja fechado automaticamente após a execução do código, evitando possíveis problemas de corrupção de dados.

● Especificar corretamente as planilhas e células

Ao referenciar planilhas e células, é importante especificá-las corretamente para evitar erros de referência. O openpyxl usa um sistema de indexação baseado em 1, ou seja, a primeira célula em uma planilha é representada como A1.



Manuseie arquivos de Excel

Openpyxl é uma biblioteca Python que permite aos usuários ler, escrever e manipular arquivos do Microsoft Excel (.xlsx e .xlsm). Pode ser usado para automatizar tarefas repetitivas, gerar relatórios, extrair dados de arquivos do Excel etc. Confira as dicas a seguir.



● Evitar operações em larga escala em grandes conjuntos de dados

Ao trabalhar com grandes conjuntos de dados, é importante evitar operações em larga escala que possam consumir muita memória ou processamento. Por exemplo, ler todas as células de uma planilha em uma única operação pode levar a problemas de desempenho.

● Usar estilos e formatação de células

O openpyxl permite a definição de estilos e formatação de células para melhorar a aparência dos dados em uma planilha. Isso pode incluir a definição de cores de fundo, bordas, alinhamento de texto e formatos numéricos.

Manuseie arquivos de Excel

Openpyxl é uma biblioteca Python que permite aos usuários ler, escrever e manipular arquivos do Microsoft Excel (.xlsx e .xlsm). Pode ser usado para automatizar tarefas repetitivas, gerar relatórios, extrair dados de arquivos do Excel etc. Confira as dicas a seguir.

● Testar o código com diferentes arquivos Excel

Essa ação garante que ele funcione corretamente em diferentes situações e formatos de arquivo. É recomendado criar diferentes planilhas de teste com diferentes formatos e estruturas de dados.

● Documentar o código e as funcionalidades

Essa ação pode ajudar outros desenvolvedores a entender o código e as funcionalidades, bem como a depurar possíveis problemas. O uso de comentários e documentação pode tornar o código mais legível e mais fácil de manter.



Manuseie arquivos de Excel

Openpyxl é uma biblioteca Python que permite aos usuários ler, escrever e manipular arquivos do Microsoft Excel (.xlsx e .xlsm). Pode ser usado para automatizar tarefas repetitivas, gerar relatórios, extrair dados de arquivos do Excel etc. Confira as dicas a seguir.

Saiba Mais

Para baixar o pacote Python e interagir com as planilhas de Excel, acesse [openpyxl - A Python library to read/write Excel 2010 xlsx/xlsm files - openpyxl 3.1.1 documentation](#)



Bons estudos!

