

Trabalhando com Dados em Python Cheat Sheet

Lendo e escrevendo arquivos

Pacote/Método	Descrição	Exemplo de Sintaxe e Código
Modos de abertura de arquivos	Diferentes modos para abrir arquivos para operações específicas.	Sintaxe: r (leitura) w (escrita) a (adição) + (atualização: ler/escrever) b (binário, caso contrário texto) Exemplos: with open("data.txt", "r") as file: content = file.read() print(content) with open("output.
Métodos de leitura de arquivos	Diferentes métodos para ler o conteúdo do arquivo de várias maneiras.	Sintaxe: file.readlines() # lê todas as linhas como uma lista readline() # lê a próxima linha como uma string file.read() # lê todo o conteúdo do arquivo como uma string Exemplo: with open("data.txt", "r") as file: lines = file.readlines() next_line = file.readline() content = file.read()
Métodos de escrita de arquivos	Diferentes métodos de escrita para gravar conteúdo em um arquivo.	Sintaxe: file.write(content) # escreve uma string no arquivo file.writelines(lines) # escreve uma lista de strings no arquivo Exemplo: lines = ["Olá\n", "Mundo\n"] with open("output.txt", "w") as file: file.writelines(lines)
Iterando sobre linhas	Itera por cada linha no arquivo usando um `loop`.	Sintaxe: for line in file: # Código para processar cada linha Exemplo: with open("data.txt", "r") as file:

		<pre>for line in file: print(line)</pre>
open() e close()	Abre um arquivo, realiza operações e fecha explicitamente o arquivo usando o método close().	<p>Sintaxe:</p> <pre>file = open(filename, mode) # Código que usa o arquivo file.close()</pre> <p>Exemplo:</p> <pre>file = open("data.txt", "r") content = file.read() file.close()</pre>
with open()	Abre um arquivo usando um bloco with, garantindo o fechamento automático do arquivo após o uso.	<p>Sintaxe:</p> <pre>with open(filename, mode) as file: # Código que usa o arquivo</pre> <p>Exemplo:</p> <pre>with open("data.txt", "r") as file: content = file.read()</pre>

Pandas

Pacote/Método	Descrição	Exemplo de Sintaxe e Código
.read_csv()	Lê dados de um arquivo `.CSV` e cria um DataFrame.	Sintaxe: dataframe_name = pd.read_csv("filename.csv") Exemplo: df = pd.read_csv("data.csv")
.read_excel()	Lê dados de um arquivo do Excel e cria um DataFrame.	<p>Sintaxe:</p> <pre>dataframe_name = pd.read_excel("filename.xlsx")</pre> <p>Exemplo:</p> <pre>df = pd.read_excel("data.xlsx")</pre>

.to_csv()	Escreve o DataFrame em um arquivo CSV.	<p>Sintaxe:</p> <pre>dataframe_name.to_csv("output.csv", index=False)</pre> <p>Exemplo:</p> <pre>df.to_csv("output.csv", index=False)</pre>
Acessar Colunas	Acessa uma coluna específica usando [] no DataFrame.	<p>Sintaxe:</p> <pre>dataframe_name["column_name"] # Acessa uma única coluna dataframe_name[["column1", "column2"]] # Acessa várias colunas</pre> <p>Exemplo:</p> <pre>df["idade"] df[["nome", "idade"]]</pre>
describe()	Gera um resumo estatístico das colunas numéricas no DataFrame.	<p>Sintaxe:</p> <pre>dataframe_name.describe()</pre> <p>Exemplo:</p> <pre>df.describe()</pre>
drop()	Remove linhas ou colunas especificadas do DataFrame. axis=1 indica colunas. axis=0 indica linhas.	<p>Sintaxe:</p> <pre>dataframe_name.drop(["column1", "column2"], axis=1, inplace=True) dataframe_name.drop(index=[row1, row2], axis=0, inplace=True)</pre>

		<p>Exemplo:</p> <pre>df.drop(["idade", "salário"], axis=1, inplace=True) # Vai remover colunas df.drop(index=[5, 10], axis=0, inplace=True) # Vai remover linhas</pre>
dropna()	Remove linhas com valores NaN ausentes do DataFrame. axis=0 indica linhas.	<p>Sintaxe:</p> <pre>dataframe_name.dropna(axis=0, inplace=True)</pre> <p>Exemplo:</p> <pre>df.dropna(axis=0, inplace=True)</pre>
duplicated()	Valores ou registros duplicados ou repetitivos dentro de um conjunto de dados.	<p>Sintaxe:</p> <pre>dataframe_name.duplicated()</pre> <p>Exemplo:</p> <pre>duplicate_rows = df[df.duplicated()]</pre>
Filtrar Linhas	Criam um novo DataFrame com linhas que atendem a condições especificadas.	<p>Sintaxe:</p> <pre>filtered_df = dataframe_name[(Conditional_statements)]</pre> <p>Exemplo:</p> <pre>filtered_df = df[(df["idade"] > 30) & (df["salário"] < 50000)]</pre>

groupby()	Divide um DataFrame em grupos com base em critérios especificados, permitindo agregação, transformação ou análise subsequente dentro de cada grupo.	<p>Sintaxe:</p> <pre>grouped = dataframe_name.groupby(by, axis=0, level=None, as_index=True, sort=True, group_keys=True, squeeze=False, observed=False, dropna=True)</pre> <p>Exemplo:</p> <pre>grouped = df.groupby(["categoria", "região"]).agg({"vendas": "soma"})</pre>
head()	Exibe as primeiras n linhas do DataFrame.	<p>Sintaxe:</p> <pre>dataframe_name.head(n)</pre> <p>Exemplo:</p> <pre>df.head(5)</pre>
Importar pandas	Importa a biblioteca Pandas com o alias pd.	<p>Sintaxe:</p> <pre>import pandas as pd</pre> <p>Exemplo:</p> <pre>import pandas as pd</pre>
info()	Fornece informações sobre o DataFrame, incluindo tipos de dados e uso de memória.	<p>Sintaxe:</p> <pre>dataframe_name.info()</pre> <p>Exemplo:</p> <pre>df.info()</pre>

merge()	Mescla dois DataFrames com base em várias colunas comuns.	<p>Sintaxe:</p> <pre>merged_df = pd.merge(df1, df2, on=["column1", "column2"])</pre> <p>Exemplo:</p> <pre>merged_df = pd.merge(vendas, produtos, on=["id_produto", "id_categoria"])</pre>
imprimir DataFrame	Exibe o conteúdo do DataFrame.	<p>Sintaxe:</p> <pre>print(df) # ou apenas digite df</pre> <p>Exemplo:</p> <pre>print(df) df</pre>
replace()	Substitui valores específicos em uma coluna por novos valores.	<p>Sintaxe:</p> <pre>dataframe_name["column_name"].replace(old_value, new_value, inplace=True)</pre> <p>Exemplo:</p> <pre>df["status"].replace("Em Andamento", "Ativo", inplace=True)</pre>
tail()	Exibe as últimas n linhas do DataFrame.	<p>Sintaxe:</p> <pre>dataframe_name.tail(n)</pre>

		<p>Exemplo:</p> <pre>df.tail(5)</pre>
--	--	---------------------------------------

Numpy

Pacote/Método	Descrição	Exemplo de Sintaxe e Código
Importando NumPy	Importa a biblioteca NumPy.	<p>Sintaxe:</p> <pre>import numpy as np</pre> <p>Exemplo:</p> <pre>import numpy as np</pre>
np.array()	Cria um array unidimensional ou multidimensional.	<p>Sintaxe:</p> <pre>array_1d = np.array([valores da lista1]) # Array 1D array_2d = np.array([valores da lista1], [valores da lista2]) # Array 2D</pre> <p>Exemplo:</p> <pre>array_1d = np.array([1, 2, 3]) # Array 1D array_2d = np.array([1, 2], [3, 4]) # Array 2D</pre>
Atributos do Array Numpy	<ul style="list-style-type: none">- Calcula a média dos elementos do array- Calcula a soma dos elementos do array- Encontra o valor mínimo no array- Encontra o valor máximo no array- Computa o produto escalar de dois arrays	<p>Exemplo:</p> <pre>np.mean(array) np.sum(array) np.min(array) np.max(array) np.dot(array_1, array_2)</pre>



Skills Network

© IBM Corporation. Todos os direitos reservados.