

Trabajando con Datos en Python Cheat Sheet

Lectura y escritura de archivos

Paquete/Método	Descripción	Sintaxis y Ejemplo de Código
Modos de apertura de archivos	Diferentes modos para abrir archivos para operaciones específicas.	Sintaxis: r (lectura) w (escritura) a (agregar) + (actualización: lectura/escritura) b (binario, de lo contrario texto) Ejemplos: with open("data.txt", "r") as file: content = file.read() print(content) with open("output.txt", "w") as file: file.write("¡Hola, mundo!") with open("log.txt", "a") as file: file.write("Entrada")
Métodos de lectura de archivos	Diferentes métodos para leer el contenido de un archivo de varias maneras.	Sintaxis: file.readlines() # lee todas las líneas como una lista readline() # lee la siguiente línea como una cadena file.read() # lee todo el contenido del archivo como una cadena Ejemplo: with open("data.txt", "r") as file: lines = file.readlines() next_line = file.readline() content = file.read()
Métodos de escritura de archivos	Diferentes métodos de escritura para escribir contenido en un archivo.	Sintaxis: file.write(content) # escribe una cadena en el archivo file.writelines(lines) # escribe una lista de cadenas en el archivo Ejemplo: lines = ["Hola\n", "Mundo\n"] with open("output.txt", "w") as file: file.writelines(lines)
Iterando sobre líneas	Itera a través de cada línea en el archivo usando un `bucle`.	Sintaxis: for line in file: # Código para procesar cada línea Ejemplo: with open("data.txt", "r") as file: for line in file: print(line)
Open() y close()	Abrir un archivo, realizar operaciones y cerrar explícitamente el archivo usando el método close().	Sintaxis: file = open(filename, mode) # Código que usa el archivo file.close() Ejemplo: file = open("data.txt", "r") content = file.read() file.close()

with open()	Abrir un archivo usando un bloque with, asegurando el cierre automático del archivo después de su uso.	<p>Sintaxis:</p> <pre>with open(filename, mode) as file: # Código que usa el archivo</pre> <p>Ejemplo:</p> <pre>with open("data.txt", "r") as file: content = file.read()</pre>
-------------	--	---

Pandas

Paquete/Método	Descripción	Sintaxis y Ejemplo de Código
.read_csv()	Lee datos de un archivo '.CSV' y crea un DataFrame.	Sintaxis: dataframe_name = pd.read_csv("filename.csv") Ejemplo: df = pd.read_csv("data.csv")
.read_excel()	Lee datos de un archivo de Excel y crea un DataFrame.	Sintaxis: <pre>dataframe_name = pd.read_excel("filename.xlsx")</pre> Ejemplo: <pre>df = pd.read_excel("data.xlsx")</pre>
.to_csv()	Escribe el DataFrame en un archivo CSV.	Sintaxis: <pre>dataframe_name.to_csv("output.csv", index=False)</pre> Ejemplo: <pre>df.to_csv("output.csv", index=False)</pre>
Acceder a Columnas	Accede a una columna específica usando [] en el DataFrame.	Sintaxis: <pre>dataframe_name["column_name"] # Accede a una sola columna dataframe_name[["column1", "column2"]] # Accede a múltiples columnas</pre> Ejemplo: <pre>df["edad"] df[["nombre", "edad"]]</pre>
describe()	Genera un resumen estadístico de las columnas numéricas en el DataFrame.	Sintaxis: <pre>dataframe_name.describe()</pre> Ejemplo: <pre>df.describe()</pre>
drop()	Elimina filas o columnas específicas del DataFrame. axis=1 indica columnas. axis=0 indica filas.	Sintaxis: <pre>dataframe_name.drop(["column1", "column2"], axis=1, inplace=True) dataframe_name.drop(index=[row1, row2], axis=0, inplace=True)</pre> Ejemplo: <pre>df.drop(["edad", "salario"], axis=1, inplace=True) # Eliminará columnas df.drop(index=[5, 10], axis=0, inplace=True) # Eliminará filas</pre>
dropna()	Elimina filas con valores NaN faltantes del DataFrame. axis=0 indica filas.	Sintaxis: <pre>dataframe_name.dropna(axis=0, inplace=True)</pre>

		<p>Ejemplo:</p> <pre>df.dropna(axis=0, inplace=True)</pre>
<code> duplicated()</code>	Valores o registros duplicados o repetitivos dentro de un conjunto de datos.	<p>Sintaxis:</p> <pre>dataframe_name.duplicated()</pre> <p>Ejemplo:</p> <pre>duplicate_rows = df[df.duplicated()]</pre>
Filtrar Filas	Crea un nuevo DataFrame con filas que cumplen condiciones específicas.	<p>Sintaxis:</p> <pre>filtered_df = dataframe_name[(Condiciones)]</pre> <p>Ejemplo:</p> <pre>filtered_df = df[(df["edad"] > 30) & (df["salario"] < 50000)]</pre>
<code>groupby()</code>	Divide un DataFrame en grupos según criterios específicos, permitiendo la agregación, transformación o análisis posterior dentro de cada grupo.	<p>Sintaxis:</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>Ejemplo:</p> <pre>grouped = df.groupby(["categoria", "region"]).agg({"ventas": "suma"})</pre>
<code>head()</code>	Muestra las primeras n filas del DataFrame.	<p>Sintaxis:</p> <pre>dataframe_name.head(n)</pre> <p>Ejemplo:</p> <pre>df.head(5)</pre>
Importar pandas	Importa la biblioteca Pandas con el alias pd.	<p>Sintaxis:</p> <pre>import pandas as pd</pre> <p>Ejemplo:</p> <pre>import pandas as pd</pre>
<code>info()</code>	Proporciona información sobre el DataFrame, incluidos los tipos de datos y el uso de memoria.	<p>Sintaxis:</p> <pre>dataframe_name.info()</pre> <p>Ejemplo:</p> <pre>df.info()</pre>
<code>merge()</code>	Combina dos DataFrames basándose en múltiples columnas comunes.	<p>Sintaxis:</p> <pre>merged_df = pd.merge(df1, df2, on=["column1", "column2"])</pre> <p>Ejemplo:</p> <pre>merged_df = pd.merge(ventas, productos, on=["product_id", "category_id"])</pre>
imprimir DataFrame	Muestra el contenido del DataFrame.	<p>Sintaxis:</p> <pre>print(df) # o simplemente escribe df</pre>

		Ejemplo: print(df) df
replace()	Reemplaza valores específicos en una columna por nuevos valores.	Sintaxis: dataframe_name["column_name"].replace(old_value, new_value, inplace=True) Ejemplo: df["estado"].replace("En Progreso", "Activo", inplace=True)
tail()	Muestra las últimas n filas del DataFrame.	Sintaxis: dataframe_name.tail(n) Ejemplo: df.tail(5)

Numpy

Paquete/Método	Descripción	Sintaxis y Ejemplo de Código
Importar NumPy	Importa la biblioteca NumPy.	Sintaxis: import numpy as np Ejemplo: import numpy as np
np.array()	Crea un array unidimensional o multidimensional,	Sintaxis: array_1d = np.array([valores de lista1]) # Array 1D array_2d = np.array([[valores de lista1], [valores de lista2]]) # Array 2D Ejemplo: array_1d = np.array([1, 2, 3]) # Array 1D array_2d = np.array([[1, 2], [3, 4]]) # Array 2D
Atributos de Array de Numpy	<ul style="list-style-type: none">- Calcula la media de los elementos del array- Calcula la suma de los elementos del array- Encuentra el valor mínimo en el array- Encuentra el valor máximo en el array- Calcula el producto punto de dos arrays	Ejemplo: np.mean(array) np.sum(array) np.min(array) np.max(array) np.dot(array_1, array_2)

