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`

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.	Sintaxis:
		`with open(filename, mode) as file: # Código que usa el archivo`
		
		Ejemplo:
		`with open("data.txt", "r") as file:`
		`content = file.read()`

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:
		`dataframe_name = pd.read_excel("filename.xlsx")`
		Ejemplo:
		`df = pd.read_excel("data.xlsx")`

.to_csv()	Escribe el DataFrame en un archivo CSV.	Sintaxis:
		`dataframe_name.to_csv("output.csv", index=False)`
		Ejemplo:
		`df.to_csv("output.csv", index=False)`

Acceder a Columnas	Accede a una columna específica usando [] en el DataFrame.	Sintaxis:
		`dataframe_name["column_name"] # Accede a una sola columna`
		`dataframe_name[["column1", "column2"]] # Accede a múltiples columnas`
		Ejemplo:
		`df["edad"]`
		`df[["nombre", "edad"]]`

describe()	Genera un resumen estadístico de las columnas numéricas en el DataFrame.	Sintaxis:
		`dataframe_name.describe()`
		Ejemplo:
		`df.describe()`

drop()	Elimina filas o columnas específicas del DataFrame. axis=1 indica columnas. axis=0 indica filas.	Sintaxis:
		`dataframe_name.drop(["column1", "column2"], axis=1, inplace=True)`
		`dataframe_name.drop(index=[row1, row2], axis=0, inplace=True)`
		Ejemplo:
		`df.drop(["edad", "salario"], axis=1, inplace=True) # Eliminará columnas`
		`df.drop(index=[5, 10], axis=0, inplace=True) # Eliminará filas`

dropna()	Elimina filas con valores NaN faltantes del DataFrame. axis=0 indica filas.	Sintaxis:
		`dataframe_name.dropna(axis=0, inplace=True)`
		Ejemplo:
		`df.dropna(axis=0, inplace=True)`

duplicated()	Valores o registros duplicados o repetitivos dentro de un conjunto de datos.	Sintaxis:
		`dataframe_name.duplicated()`
		Ejemplo:
		`duplicate_rows = df[df.duplicated()]`

Filtrar Filas	Crea un nuevo DataFrame con filas que cumplen condiciones específicas.	Sintaxis:
		`filtered_df = dataframe_name[(Condiciones)]`
		Ejemplo:
		`filtered_df = df[(df["edad"] > 30) & (df["salario"] < 50000)`

groupby()	Divide un DataFrame en grupos según criterios específicos, permitiendo la agregación, transformación o análisis posterior dentro de cada grupo.	Sintaxis:
		`grouped = dataframe_name.groupby(by, axis=0, level=None, as_index=True, sort=True, group_keys=True, squeeze=False, observed=False, dropna=True)`
		Ejemplo:
		`grouped = df.groupby(["categoría", "región"]).agg({"ventas": "suma"})`

head()	Muestra las primeras n filas del DataFrame.	Sintaxis: `dataframe_name.head(n)`

		Ejemplo: `df.head(5)`
Importar pandas	Importa la biblioteca Pandas con el alias pd.	Sintaxis: `import pandas as pd` Ejemplo: `import pandas as pd`

info()	Proporciona información sobre el DataFrame, incluidos los tipos de datos y el uso de memoria.	Sintaxis:
		`dataframe_name.info()`
		Ejemplo:
		`df.info()`

merge()	Combina dos DataFrames basándose en múltiples columnas comunes.	Sintaxis: `merged_df = pd.merge(df1, df2, on=["column1", "column2"])`

		Ejemplo: `merged_df = pd.merge(ventas, productos, on=["product_id", "category_id"])`

imprimir DataFrame	Muestra el contenido del DataFrame.	Sintaxis:
		`print(df) # o simplemente escribe df`
		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	- 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)`

Skills Network