

Working with Data in Python Cheat Sheet

Reading and writing files

Package/Method	Description	Syntax and Code Example
File opening modes	Different modes to open files for specific operations.	Syntax: r (reading) w (writing) a (appending) + (updating: read/write) b (binary, otherwise text) 1 Examples: with open("data.txt", "r") as file: content = file.read() print(content) with open("output.txt", "w") as file: file.write("Hello, world!") with open("log.txt", "a") as file: file.write("Log entr
File reading methods	Different methods to read file content in various ways.	Syntax: 1 file.readlines() # reads all lines as a list 2 readline() # reads the next line as a string 3 file.read() # reads the entire file content as a string Example: 1 with open("data.txt", "r") as file: 2 lines = file.readlines() 3 next_line = file.readline() 4 content = file.read()
File writing methods	Different write methods to write content to a file.	Syntax: 1 file.write(content) # writes a string to the file 2 file.writelines(lines) # writes a list of strings to the file Example: 1 lines = ["Hello\n", "World\n"] 2 with open("output.txt", "w") as file: 3 file.writelines(lines)
Iterating over lines	Iterates through each line in the file using a 'loop'.	Syntax: 1 for line in file: # Code to process each line Example: 1 with open("data.txt", "r") as file: 2 for line in file: print(line)
Open() and close()	Opens a file, performs operations, and explicitly closes the file using the close() method.	Syntax: 1 file = open(filename, mode) # Code that uses the file 2 file.close() Example: 1 file = open("data.txt", "r") 2 content = file.read() 3 file.close()
with open()	Opens a file using a with block, ensuring automatic file closure after usage.	Syntax: 1 with open(filename, mode) as file: # Code that uses the file Example: 1 with open("data.txt", "r") as file: 2 content = file.read()

Pandas

Package/Method	Description	Syntax and Code Example
.read_csv()	Reads data from a ".CSV" file and creates a DataFrame.	Syntax: dataframe_name = pd.read_csv("filename.csv") Example: df = pd.read_csv("data.csv")
.read_excel()	Reads data from an Excel file and creates a DataFrame.	Syntax: 1 dataframe_name = pd.read_excel("filename.xlsx") Example: 1 df = pd.read_excel("data.xlsx")
.to_csv()	Writes DataFrame to a CSV file.	Syntax: 1 dataframe_name.to_csv("output.csv", index=False) Example: 1 df.to_csv("output.csv", index=False)
Access Columns	Accesses a specific column using [] in the DataFrame.	Syntax: 1 dataframe_name["column_name"] # Accesses single column 2 dataframe_name[["column1", "column2"]] # Accesses multiple columns Example: 1 df["age"] 2 df[["name", "age"]]
describe()	Generates statistics summary of numeric columns in the DataFrame.	Syntax: 1 dataframe_name.describe() Example: 1 df.describe()

drop()	Removes specified rows or columns from the DataFrame. axis=1 indicates columns. axis=0 indicates rows.	<div>Syntax:</div> <div><div>1</div><div>dataframe_name.drop(["column1", "column2"], axis=1, inplace=True)</div><div>2</div><div>dataframe_name.drop(index=[row1, row2], axis=0, inplace=True)</div></div> <div>Example:</div> <div><div>1</div><div>df.drop(["age", "salary"], axis=1, inplace=True) # Will drop columns</div><div>2</div><div>df.drop(index=[5, 10], axis=0, inplace=True) # Will drop rows</div></div>
dropna()	Removes rows with missing NaN values from the DataFrame. axis=0 indicates rows.	<div>Syntax:</div> <div><div>1</div><div>dataframe_name.dropna(axis=0, inplace=True)</div></div> <div>Example:</div> <div><div>1</div><div>df.dropna(axis=0, inplace=True)</div></div>
uplicated()	Duplicate or repetitive values or records within a data set.	<div>Syntax:</div> <div><div>1</div><div>dataframe_name.duplicated()</div></div> <div>Example:</div> <div><div>1</div><div>duplicate_rows = df[df.duplicated()]</div></div>
Filter Rows	Creates a new DataFrame with rows that meet specified conditions.	<div>Syntax:</div> <div><div>1</div><div>filtered_df = dataframe_name[(Conditional_statements)]</div></div> <div>Example:</div> <div><div>1</div><div>filtered_df = df[(df["age"] > 30) & (df["salary"] < 50000)]</div></div>
groupby()	Splits a DataFrame into groups based on specified criteria, enabling subsequent aggregation, transformation, or analysis within each group.	<div>Syntax:</div> <div><div>1</div><div>grouped = dataframe_name.groupby(by, axis=0, level=None, as_index=True,</div><div>2</div><div>sort=True, group_keys=True, squeeze=False, observed=False, dropna=True)</div></div> <div>Example:</div> <div><div>1</div><div>grouped = df.groupby(["category", "region"]).agg({"sales": "sum"})</div></div>
head()	Displays the first n rows of the DataFrame.	<div>Syntax:</div> <div><div>1</div><div>dataframe_name.head(n)</div></div> <div>Example:</div> <div><div>1</div><div>df.head(5)</div></div>
Import pandas	Imports the Pandas library with the alias pd.	<div>Syntax:</div> <div><div>1</div><div>import pandas as pd</div></div> <div>Example:</div> <div><div>1</div><div>import pandas as pd</div></div>
info()	Provides information about the DataFrame, including data types and memory usage.	<div>Syntax:</div> <div><div>1</div><div>dataframe_name.info()</div></div> <div>Example:</div> <div><div>1</div><div>df.info()</div></div>
merge()	Merges two DataFrames based on multiple common columns.	<div>Syntax:</div> <div><div>1</div><div>merged_df = pd.merge(df1, df2, on=["column1", "column2"])</div></div> <div>Example:</div> <div><div>1</div><div>merged_df = pd.merge(sales, products, on=["product_id", "category_id"])</div></div>
print DataFrame	Displays the content of the DataFrame.	<div>Syntax:</div> <div><div>1</div><div>print(df) # or just type df</div></div> <div>Example:</div> <div><div>1</div><div>print(df)</div><div>2</div><div>df</div></div>
replace()	Replaces specific values in a column with new values.	<div>Syntax:</div> <div><div>1</div><div>dataframe_name["column_name"].replace(old_value, new_value, inplace=True)</div></div> <div>Example:</div> <div><div>1</div><div>df["status"].replace("In Progress", "Active", inplace=True)</div></div>
tail()	Displays the last n rows of the DataFrame.	<div>Syntax:</div> <div><div>1</div><div>dataframe_name.tail(n)</div></div> <div>Example:</div> <div><div>1</div><div>df.tail(5)</div></div>

Numpy

Package/Method	Description	Syntax and Code Example
Importing NumPy	Imports the NumPy library.	<div>Syntax:</div> <div><div>1</div><div>import numpy as np</div></div> <div>Example:</div> <div><div>1</div><div>import numpy as np</div></div>
		<div>Syntax:</div>

np.array()	Creates a one or multi-dimensional array.	<div><div>Copy icon</div><pre>1 array_1d = np.array([list1 values]) # 1D Array 2 array_2d = np.array([[list1 values], [list2 values]]) # 2D Array</pre></div> <div>Example:<div><div>Copy icon</div><pre>1 array_1d = np.array([1, 2, 3]) # 1D Array 2 array_2d = np.array([[1, 2], [3, 4]]) # 2D Array</pre></div></div>
Numpy Array Attributes	<ul style="list-style-type: none">- Calculates the mean of array elements- Calculates the sum of array elements- Finds the minimum value in the array- Finds the maximum value in the array- Computes dot product of two arrays	<div>Example:<div><div>Copy icon</div><pre>1 np.mean(array) 2 np.sum(array) 3 np.min(array) 4 np.max(array) 5 np.dot(array_1, array_2)</pre></div></div>