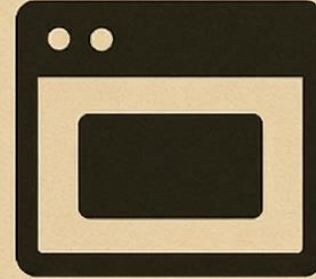


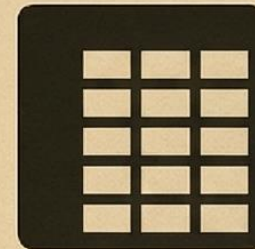
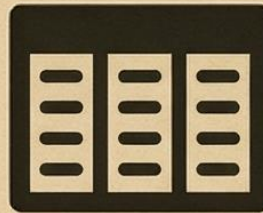
pandas



**READING, TRANSFORMING,  
WRITING Excel, CSV  
with Pandas data  
analysis library**



Introduction to Sets





## Why pandas?

- Fast, powerful data analysis toolkit
- Works great with CSV & Excel
- Cleaner & faster than pure Python loops
- Industry-standard for data tasks
- References:
  - <https://pandas.pydata.org/about/index.html>
  - [https://pandas.pydata.org/docs/user\\_guide/10min.html](https://pandas.pydata.org/docs/user_guide/10min.html)
  - <https://numpy.org/doc/stable/user/whatisnumpy.html>



## Virtual Environment Reminder

- Keeps project dependencies isolated
- Prevents conflicts with system Python
- Packages stored per project
- References:
  - <https://docs.python.org/3/library/venv.html>
  - <https://realpython.com/python-virtual-environments-a-primer/>
  - <https://packaging.python.org/en/latest/guides/installing-using-pip-and-virtual-environments/>



# Activating Your venv

- `python -m venv venv`
- Activate venv
- Install pandas & Jupyter
- Verify with `pip list`
- References:
  - <https://docs.python.org/3/library/venv.html#creating-virtual-environments>
  - <https://code.visualstudio.com/docs/python/environments>
  - <https://pip.pypa.io/en/stable/installation/>



## Installing pandas & Jupyter

```
pip install pandas
```

```
pip install jupyter
```

Confirm installation in venv

### References:

- [https://pandas.pydata.org/docs/getting\\_started/install.html](https://pandas.pydata.org/docs/getting_started/install.html)
- <https://jupyter.org/install>
- [https://pip.pypa.io/en/stable/user\\_guide/](https://pip.pypa.io/en/stable/user_guide/)



# Jupyter Notebooks in VS Code

- Install Python + Jupyter extensions
- Create .ipynb file
- Select venv kernel
- Great for data exploration
- References:
  - <https://code.visualstudio.com/docs/datascience/jupyter-notebooks>
  - <https://code.visualstudio.com/docs/python/python-tutorial>
  - <https://marketplace.visualstudio.com/items?itemName=ms-toolsai.jupyter>



# Notebook Basics

- Run code in cells
- Outputs appear below code
- Use Markdown cells for notes
- Restart kernel clears variables
- References:
  - <https://jupyter-notebook.readthedocs.io/en/stable/notebook.html>
  - <https://docs.jupyter.org/en/latest/>
  - <https://ipython.readthedocs.io/en/stable/interactive/htmlnotebook.html>



# Introducing pandas


- import pandas as pd
- Series and DataFrame
- DataFrame = table-like structure
- References:
  - [https://pandas.pydata.org/docs/getting\\_started/intro\\_tutorials/01\\_table\\_oriented.html](https://pandas.pydata.org/docs/getting_started/intro_tutorials/01_table_oriented.html)
  - [https://pandas.pydata.org/docs/user\\_guide/dsintro.html](https://pandas.pydata.org/docs/user_guide/dsintro.html)
  - <https://realpython.com/pandas-python-explore-dataset/>





# Why DataFrames Are Powerful

- Labeled rows & columns
- Handles multiple data types
- Vectorized operations
- Easy filtering/joining/grouping
- References:
  - [https://pandas.pydata.org/docs/user\\_guide/dsintro.html#dataframe](https://pandas.pydata.org/docs/user_guide/dsintro.html#dataframe)
  - [https://pandas.pydata.org/docs/user\\_guide/basics.html](https://pandas.pydata.org/docs/user_guide/basics.html)
  - [https://pandas.pydata.org/docs/user\\_guide/10min.html](https://pandas.pydata.org/docs/user_guide/10min.html)



# Pandas Data Model & NumPy Foundation

- Columns are NumPy ndarrays
- Each column has strict dtype
- NumPy provides speed
- pandas adds labels + ops
- References:
  - [https://pandas.pydata.org/docs/user\\_guide/basics.html#dtypes](https://pandas.pydata.org/docs/user_guide/basics.html#dtypes)
  - <https://numpy.org/doc/stable/reference/arrays.ndarray.html>
  - [https://pandas.pydata.org/docs/user\\_guide/basics.html#numpy-interop](https://pandas.pydata.org/docs/user_guide/basics.html#numpy-interop)



# Reading CSV Files

- `pd.read_csv`
- Preview with `head()`
- Inspect with `info()`
- Type detection
- References:
  - [https://pandas.pydata.org/docs/reference/api/pandas.read\\_csv.html](https://pandas.pydata.org/docs/reference/api/pandas.read_csv.html)
  - [https://pandas.pydata.org/docs/user\\_guide/io.html#csv-text-files](https://pandas.pydata.org/docs/user_guide/io.html#csv-text-files)
  - <https://realpython.com/python-csv/>



# Reading Excel Files

- `pd.read_excel`
- Supports multiple sheets
- Handles missing values
- Preview first rows
- References:
  - [https://pandas.pydata.org/docs/reference/api/pandas.read\\_excel.html](https://pandas.pydata.org/docs/reference/api/pandas.read_excel.html)
  - [https://pandas.pydata.org/docs/user\\_guide/io.html#excel-files](https://pandas.pydata.org/docs/user_guide/io.html#excel-files)
  - <https://openpyxl.readthedocs.io/en/sheet/>



# Viewing Data Efficiently

- `head()`, `tail()`
- `describe()` for numerics
- `shape` for dims
- columns to inspect names
- References:
  - [https://pandas.pydata.org/docs/user\\_guide/basics.html](https://pandas.pydata.org/docs/user_guide/basics.html)
  - <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.info.html>
  - <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.describe.html>



# Selecting Data (Rows & Columns)

- `df['col']`
- `df[['a','b']]`
- Boolean filtering
- `loc` and `iloc` basics
- References:
  - [https://pandas.pydata.org/docs/user\\_guide/indexing.html](https://pandas.pydata.org/docs/user_guide/indexing.html)
  - <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.loc.html>
  - <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.iloc.html>



# Cleaning & Transforming Data

- Rename columns
- Create new columns
- Convert types
- Handle missing data
- References:
  - [https://pandas.pydata.org/docs/user\\_guide/basics.html](https://pandas.pydata.org/docs/user_guide/basics.html)
  - [https://pandas.pydata.org/docs/user\\_guide/reshaping.html](https://pandas.pydata.org/docs/user_guide/reshaping.html)
  - <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.fillna.html>



# Sorting & Filtering

- `sort_values`
- Logical conditions
- Combine filters with `&` and `|`
- Filter examples
- References:
  - [https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.sort\\_values.html](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.sort_values.html)
  - [https://pandas.pydata.org/docs/user\\_guide/indexing.html#boolean-indexing](https://pandas.pydata.org/docs/user_guide/indexing.html#boolean-indexing)
  - [https://pandas.pydata.org/docs/user\\_guide/basics.html#basics-filtering](https://pandas.pydata.org/docs/user_guide/basics.html#basics-filtering)





# Saving CSV & Excel

- to\_csv
- to\_excel
- Multi-sheet support
- Avoid path mistakes
- References:
  - [https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to\\_csv.html](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_csv.html)
  - [https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to\\_excel.html](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_excel.html)
  - [https://pandas.pydata.org/docs/user\\_guide/io.html#excel-writer](https://pandas.pydata.org/docs/user_guide/io.html#excel-writer)



# Introducing Python Sets

- Unordered unique collection
- Fast membership test
- Use {} or set()
- References:
  - <https://docs.python.org/3/library/stdtypes.html#set>
  - <https://realpython.com/python-sets/>
  - <https://docs.python.org/3/tutorial/datastructures.html#sets>



# Set Operations

- union
- intersection
- difference
- Find new/missing values
- References:
  - <https://docs.python.org/3/library/stdtypes.html#set-types-set-frozenset>
  - <https://realpython.com/python-sets/#set-operations>
  - <https://docs.python.org/3/howto/functional.html>



## Using Sets with pandas

- `set(df['col'])`
- Compare across files
- Detect mismatches
- Check consistency
- References:
  - [https://pandas.pydata.org/docs/user\\_guide/basics.html#basics-duplicate-values](https://pandas.pydata.org/docs/user_guide/basics.html#basics-duplicate-values)
  - <https://pandas.pydata.org/docs/reference/api/pandas.Series.unique.html>
  - <https://pandas.pydata.org/docs/reference/api/pandas.Index.isin.html>



# Automation Workflow — Overview



**Combine many  
CSVs**



**Clean structure**



**Summaries &  
aggregates**



**Save outputs**



**References:**

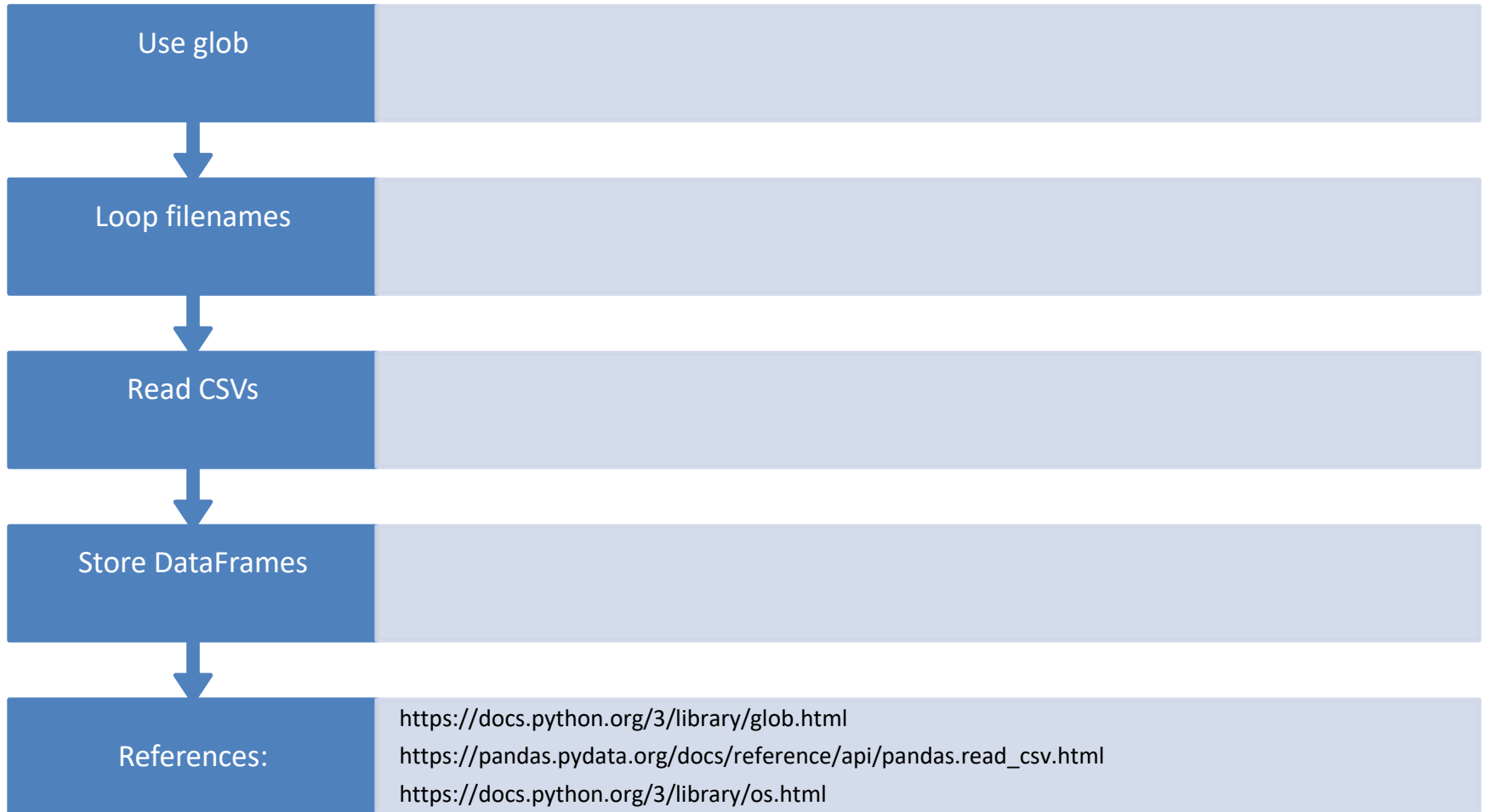
[https://pandas.pydata.org/docs/getting\\_started/intro\\_tutorials/](https://pandas.pydata.org/docs/getting_started/intro_tutorials/)

[https://pandas.pydata.org/docs/user\\_guide/io.html](https://pandas.pydata.org/docs/user_guide/io.html)

<https://realpython.com/pandas-python-explore-dataset/>



# Automation Step 1 — Load Many Files





# Automation

## Step 2 — Merge All Data

- `pd.concat`
- Verify row counts
- Clean column names
- Drop duplicates
- References:
  - <https://pandas.pydata.org/docs/reference/api/pandas.concat.html>
  - [https://pandas.pydata.org/docs/user\\_guide/merging.html](https://pandas.pydata.org/docs/user_guide/merging.html)
  - [https://pandas.pydata.org/docs/user\\_guide/basics.html#basics-reindexing-and-alignment](https://pandas.pydata.org/docs/user_guide/basics.html#basics-reindexing-and-alignment)



# Automation

## Step 3 — Aggregate Data

- groupby for summaries
- Average per course
- Counts per group
- Use sets for unique IDs
- References:
  - <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.groupby.html>
  - [https://pandas.pydata.org/docs/user\\_guide/groupby.html](https://pandas.pydata.org/docs/user_guide/groupby.html)
  - <https://pandas.pydata.org/docs/reference/api/pandas.core.groupby.DataFrameGroupBy.agg.html>





# Automation Step 4 — Export Final Output

- Save merged data
- Save summaries
- Multi-sheet Excel
- Check paths
- References:
  - [https://pandas.pydata.org/docs/user\\_guide/io.html#excel-files](https://pandas.pydata.org/docs/user_guide/io.html#excel-files)
  - <https://pandas.pydata.org/docs/reference/api/pandas.ExcelWriter.html>
  - [https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to\\_excel.html](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.to_excel.html)



# Wrap-Up — What You Learned

- pandas basics
- Jupyter workflows
- Sets for comparisons
- Automation pipeline
- References:
  - [https://pandas.pydata.org/docs/getting\\_started/intro\\_tutorials/](https://pandas.pydata.org/docs/getting_started/intro_tutorials/)
  - <https://jupyter.org/documentation>
  - <https://docs.python.org/3/library/stdtypes.html#set>

