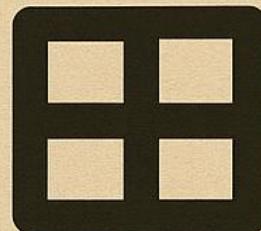
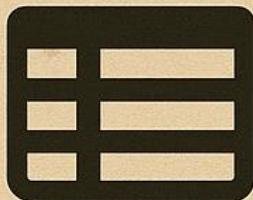


READING & WRITING
Excel Files

using openpyxl

INTRODUCTION TO
DICTIONARIES





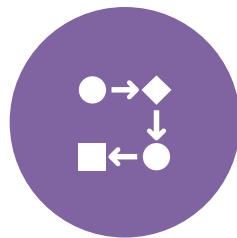
Why Excel Automation Matters



- EXCEL
EVERYWHERE



- REAL-WORLD
AUTOMATION



- GREAT PRACTICE
WITH LOOPS



- PREPARES FOR
PANDAS



Goals for This Video



- VIRTUAL ENVIRONMENT



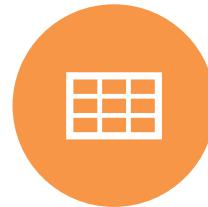
- INSTALL OPENPYXL



- READ EXCEL



- DICTIONARIES



- WRITE EXCEL



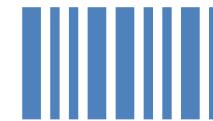
Prepare the Project Folder



- Create folder



- Add sample
Excel



- Open in VS Code



Definitions

- Workbook – entire Excel .xlsx file
- Worksheet – a tab inside the workbook
- Cell – smallest data container (e.g., A1)
- Row – horizontal group of cells
- Column – vertical group of cells
- Header Row – labels like Name, Email, Grade
- Cell Value – actual data or formula in a cell
- openpyxl – library for reading/writing .xlsx
- Virtual Environment – isolated Python setup
- Dictionary – key/value data structure



Creating a Virtual Environment



`python -m venv venv`



Activate with
yourprojectroot/venv/scripts/activate
.ps1



`pip list`



<https://realpython.com/python-virtual-environments-a-primer/> -
Friendly Intermediate



<https://docs.python.org/3/library/venv.html> - Official

Installing openpyxl



Need for Package Indexer



<https://pypi.org/> - 99% of packages



Python Package Index find



- pip install openpyxl



- pip show



Exploring Sample Excel File

- Columns

- Header row

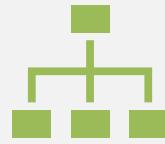
- Table structure



Loading a Workbook



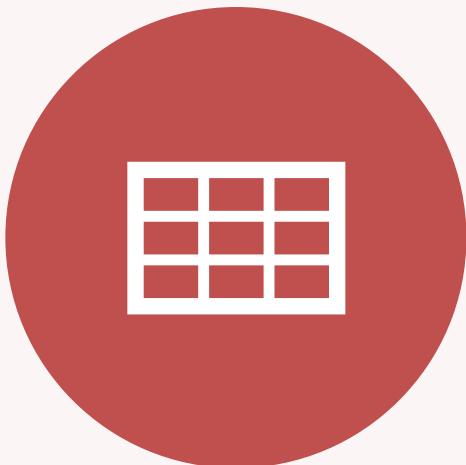
- `load_workbook()`



- `wb.active`



Iterating Through Rows



- `ITER_ROWS(VALUES_ONLY=TRUE)`
- `SKIP HEADER`



Introducing Dictionaries



- Keys = column names



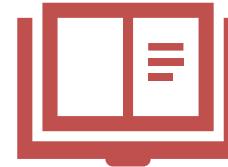
- Easier than tuples



Converting Rows to Dicts



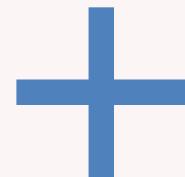
- student['name']



- Clean, readable



Using Dictionaries



- Filter

- Cleaning

- Add fields

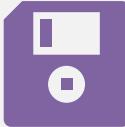
Creating a New Excel File



- Workbook()



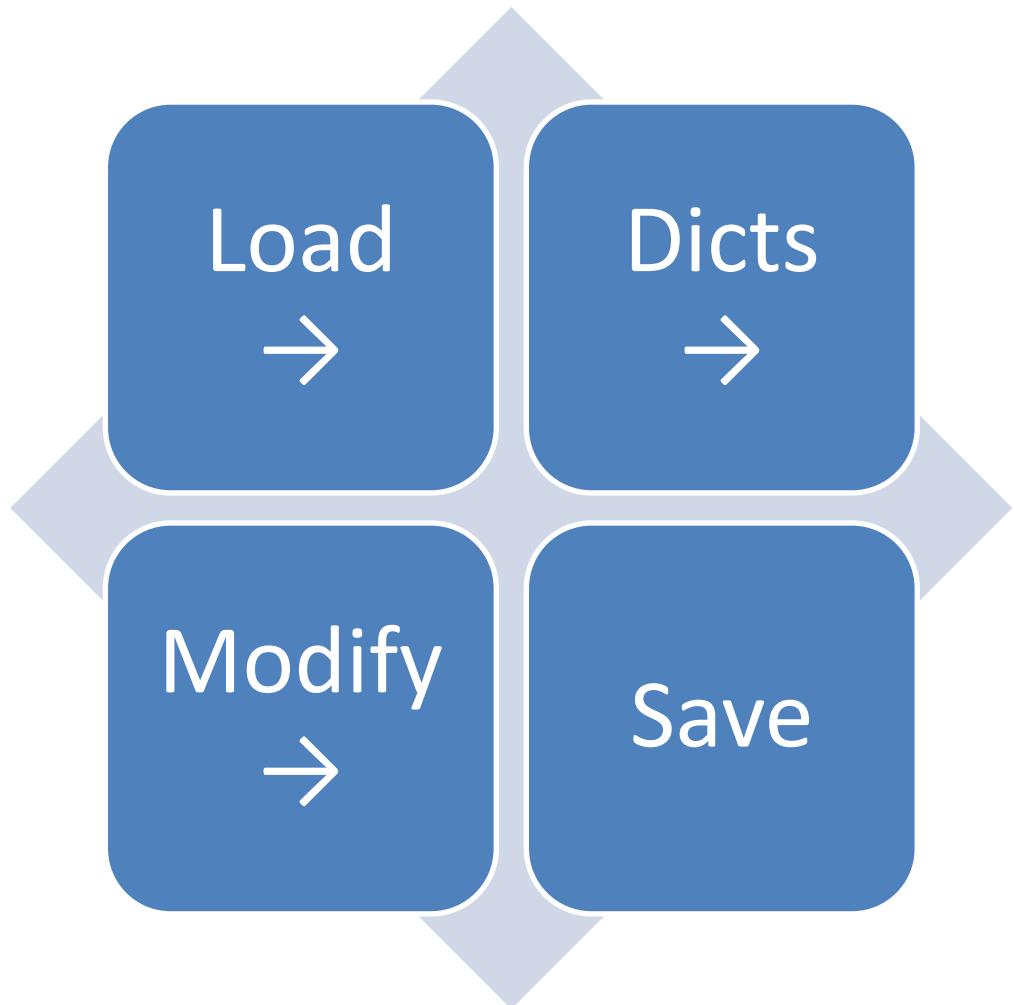
- append()



- save()



Full Simple Workflow





Advanced Workflows



- Multi-sheet



- Formulas



- Templates

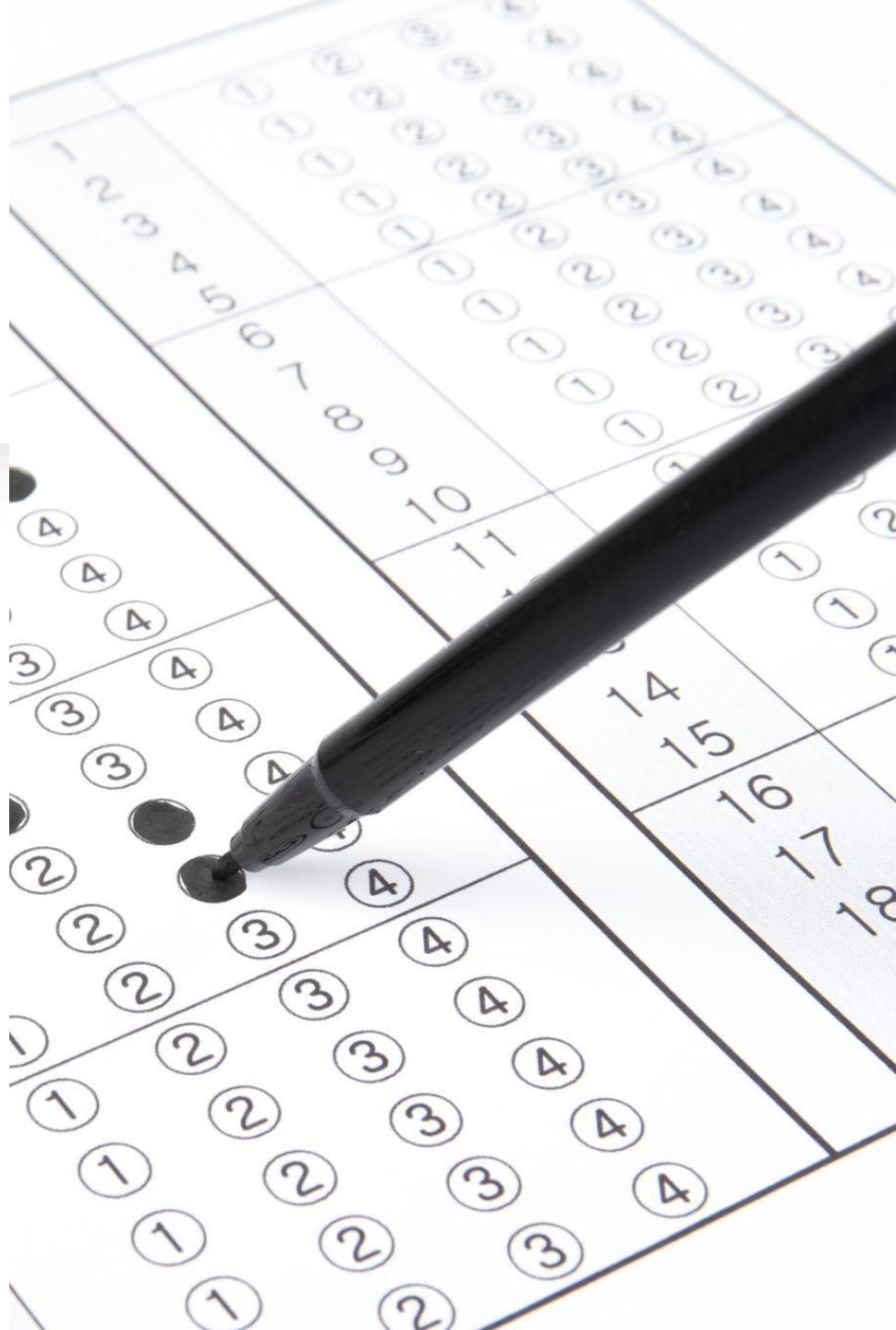


- Styling



Advanced Case Study – Attendance

- Student attendance Tracked
- Many worksheets
- Grades
- Present/absent logic





Color Formatting

- PatternFill
- Red/yellow/green



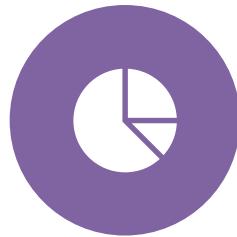
openpyxl Strengths



- FORMATTING



- TEMPLATES



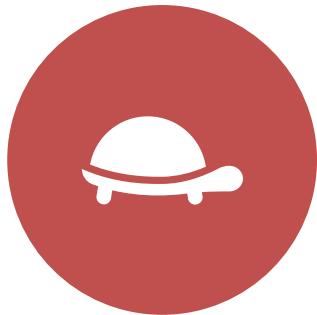
- CHARTS



- MULTI-SHEET



openpyxl Limitations



- SLOW FOR LARGE DATA



- NO FORMULA CALC(CAN CREATE)



- MANUAL STEPS



Skills Learned

- How to set up and use a **Python virtual environment**
- Installing and using **openpyxl** to read and write Excel files
- Iterating through Excel worksheets and converting rows into **Python dictionaries**
- Processing, transforming, and generating new structured data from spreadsheets
- Creating new Excel files and saving results using Workbook() and append()
- Working with **multiple worksheets** and merging them into a single summary
- Applying basic **cell formatting** (colors) to communicate data meaning

References - URLs

- Official -
<https://pypi.org/project/openpyxl/>
- Intermediate -
<https://realpython.com/openpyxl-excel-spreadsheets-python/>
- Beginner -
<https://automatetheboringstuff.com/3e/chapter14.html>



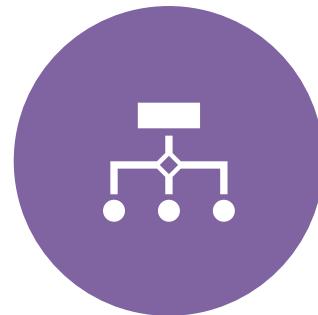
→ Next Up: Pandas



- DATAFRAMES



- FAST OPERATIONS



- MERGE AND RESHAPE