

# #\_ the Python + Excel Automation [ Cheat Sheet ]

## 1. 📦 Automating Basic Excel Operations

- **Packages:** `openpyxl`, `xlwings`
- **Description:** Create, open, save, and close Excel files through Python scripts.
- **Commands:**
  - `wb = openpyxl.Workbook()`
  - `wb.save('file.xlsx')`
  - `xlwings.Book('file.xlsx').save()`
  - `wb.close()`

## 2. 📦 Managing Sheets and Ranges

- **Packages:** `openpyxl`, `xlwings`
- **Description:** Create, delete, or move sheets and manipulate cell ranges.
- **Commands:**
  - `wb.create_sheet('Sheetname')`
  - `ws['A1':'B10']`
  - `xlwings.sheets.add('Sheetname')`
  - `xlwings.Range('A1').value`

## 3. 📦 Data Cleaning and Preprocessing Automation

- **Packages:** `pandas`, `openpyxl`
- **Description:** Automate the cleaning and preprocessing of Excel data.
- **Commands:**
  - `df.dropna(inplace=True)`
  - `ws['A1'] = df['column'].mean()`
  - `df['column'] = df['column'].apply(func)`

## 4. 📦 Excel Formulas and Calculations Automation

- **Packages:** `openpyxl`
- **Description:** Automate the application of Excel formulas.
- **Commands:**

- `ws['A1'] = "=SUM(A2:A10)"`
- `ws['B1'] = "=AVERAGE(B2:B10)"`
- `ws['C1'] = "=VLOOKUP(C2, Sheet2!A1:B10, 2, FALSE)"`

## 5. 📦 Chart Automation for Visual Insights

- **Packages:** `openpyxl`
- **Description:** Create and modify Excel charts programmatically.
- **Commands:**
  - `chart = openpyxl.chart.LineChart()`
  - `chart.add_data(data)`
  - `ws.add_chart(chart, 'E5')`

## 6. 📦 Automating Pivot Tables for Data Summary

- **Packages:** `pandas`, `openpyxl`
- **Description:** Create Pivot Tables programmatically for summarizing data.
- **Commands:**
  - `pivot = pd.pivot_table(df, index, columns, values)`
  - `pivot.to_excel('pivot.xlsx', engine='openpyxl')`

## 7. 📦 Automation of Custom User-Defined Functions

- **Packages:** `xlwings`
- **Description:** Implement and call custom functions in Excel via Python.
- **Commands:**
  - `@xlwings.udf`
  - `xlwings.quickstart my_udf`
  - `=my_function()` in Excel

## 8. 📦 Real-Time Data Retrieval and Automation

- **Packages:** `pandas`, `yfinance`
- **Description:** Automatically retrieve and analyze real-time data such as stocks.
- **Commands:**
  - `stock_data = yfinance.Ticker('AAPL').history(period='1d')`
  - `df.to_excel('realtime_data.xlsx')`

## 9. 📦 Automation of Security Measures in Excel

- **Packages:** `openpyxl`
- **Description:** Ensure data security in Excel files through automated techniques.
- **Commands:**
  - `ws.protection.set_password('password')`
  - `ws.sheet_properties.tabColor = "1072BA"`

## 10. 📦 Performance Optimization Automation for Large Files

- **Packages:** `pandas`, `dask`
- **Description:** Automate performance optimization when dealing with large Excel files.
- **Commands:**
  - `dask_df = dask.dataframe.read_csv('large_file.csv')`
  - `dask_df.compute()`
  - `pandas.read_excel('file.xlsx', usecols=['A', 'B'])`

## 11. 📦 Excel Add-ins and Extensions Automation

- **Packages:** `xlwings`, `pyxl`
- **Description:** Develop and deploy Excel add-ins and extensions through automation.
- **Commands:**
  - `xlwings addin install`
  - `@xlwings.func` for custom functions

## 12. 📦 Scheduling Automated Excel Tasks

- **Packages:** `schedule`, `pandas`, `openpyxl`
- **Description:** Schedule Excel-related tasks to run at specific intervals.
- **Commands:**
  - `schedule.every(10).minutes.do(task)`
  - `schedule.every().day.at('10:30').do(task)`
  - `while True: schedule.run_pending()`

### 13. 📦 Web Scraping to Excel Automation

- **Packages:** `beautifulsoup4, requests, pandas`
- **Description:** Automate the process of scraping web data and storing it in Excel.
- **Commands:**
  - `soup = BeautifulSoup(page.content, 'html.parser')`
  - `df.to_excel('web_data.xlsx')`

### 14. 📦 Email Automation with Excel Attachments

- **Packages:** `smtplib, email`
- **Description:** Automate sending emails with Excel files as attachments.
- **Commands:**
  - `server = smtplib.SMTP('smtp.gmail.com', 587)`
  - `server.login(username, password)`
  - `email.attach(attachment, 'application/excel', 'file.xlsx')`

### 15. 📦 Database Integration with Excel Automation

- **Packages:** `pandas, sqlalchemy`
- **Description:** Automate data extraction from databases to Excel and vice versa.
- **Commands:**
  - `engine = create_engine('database_url')`
  - `df.to_sql('table_name', engine)`
  - `df = pd.read_sql('table_name', engine)`

### 16. 📦 Error Handling in Excel Automation

- **Packages:** `openpyxl`
- **Description:** Implement error handling in your Excel automation to ensure smooth operation.
- **Commands:**
  - `try: ws['A1'] = 'value'`
  - `except Exception as e: print(e)`

## 17. 📦 Excel Data Validation Automation

- **Packages:** `openpyxl`
- **Description:** Automate validation rules in Excel to ensure data integrity.
- **Commands:**
  - `validation = openpyxl.worksheet.datavalidation.DataValidation()`
  - `ws.add_data_validation(validation)`

## 18. 📦 Batch Processing Automation with Excel

- **Packages:** `pandas`, `openpyxl`
- **Description:** Automate batch processing of multiple Excel files at once.
- **Commands:**
  - `for file in files: pd.read_excel(file).process()`

## 19. 📦 Creating Interactive Dashboards in Excel

- **Packages:** `openpyxl`, `pandas`
- **Description:** Automate the creation of interactive dashboards in Excel.
- **Commands:**
  - `ws['A1'] = "=PIVOT(Sheet2!A1:C10)"`
  - `ws['A2'] = "=CHART(Sheet2!D1:E10)"`

## 20. 📦 Automating Multi-Language Support in Excel

- **Packages:** `openpyxl`, `translate`
- **Description:** Facilitate internationalization by automating translations in Excel.
- **Commands:**
  - `translator = translate.Translator(to_lang='es')`
  - `ws['A1'] = translator.translate(ws['A1'].value)`