**🗓️ WEEK 1 — Practical Data Handling with Python**

**Goal:** Build comfort reading, writing, and organizing data files in different formats — and logging everything.

**DAY 1 – Project Setup & File Exploration**

**Focus:** Environment, directories, and basic file inspection.  
**Learn / Do**

* Create a folder structure:
* data\_engineering/
* data/
* logs/
* scripts/
* Explore Python modules: os, pathlib, glob
* List and filter files by extension or size:
* from pathlib import Path
* [p for p in Path("data").rglob("\*.csv")]

**Mini-Task:**  
List all CSV files, sort by modified date, print names and sizes.

**DAY 2 – Reading and Writing CSV Files**

**Focus:** Efficient CSV handling.  
**Learn / Do**

* Read large CSV with pandas.read\_csv(chunksize=50000)
* Measure read time using time module
* Write cleaned subset back to disk.  
  **Mini-Task:**  
  Load a large CSV (Kaggle dataset, for example), extract specific columns, and save as sample\_10k.csv.

**DAY 3 – JSON and YAML Handling**

**Focus:** Non-tabular formats.  
**Learn / Do**

* Parse JSON:
* import json
* data = json.load(open('sample.json'))
* Convert nested JSON → flat DataFrame (pandas.json\_normalize)
* Read/write YAML with pyyaml  
  **Mini-Task:**  
  Convert a JSON file to YAML and back again.

**DAY 4 – Parquet and Feather Formats**

**Focus:** Columnar storage for big data.  
**Learn / Do**

* df.to\_parquet("data.parquet")
* pd.read\_parquet("data.parquet")
* Compare file sizes between CSV and Parquet.  
  **Mini-Task:**  
  Write a script that takes any CSV and converts it to Parquet automatically.

**DAY 5 – Logging and Error Handling**

**Focus:** Production hygiene.  
**Learn / Do**

* Configure Python’s logging module.
* Write logs to logs/app.log.
* Use try/except blocks with meaningful messages.  
  **Mini-Task:**  
  Wrap your CSV→Parquet converter in a try/except and log start/end timestamps and any errors.

**DAY 6 – Modularizing Your Code**

**Focus:** Organize and reuse code.  
**Learn / Do**

* Create a utils.py file with functions like read\_csv\_safely(path) and convert\_to\_parquet(df, path).
* Import those functions in your main script.  
  **Mini-Task:**  
  Refactor yesterday’s script into two modules: converter.py and utils.py.

**DAY 7 – Mini Project Wrap-Up**

**Project:** **File Format Converter**

* Input folder: /data/input/
* Output folder: /data/output/
* Converts CSV → Parquet and JSON → CSV
* Logs every action and handles bad files gracefully.

**Outcome:**  
✅ You now know file systems, formats, and safe I/O with logs — the daily bread of data engineers.

**🗓️ WEEK 2 — Data Cleaning & Transformation**

**Goal:** Clean, reshape, and standardize data with pandas like a professional ETL developer.

**DAY 8 – Inspecting and Profiling Data**

**Focus:** Understanding raw data.  
**Learn / Do**

* df.info(), df.describe(), df.isna().sum()
* Quick EDA summary script.  
  **Mini-Task:**  
  Write a “data profiler” function that prints:
  + shape
  + column names/types
  + null counts

**DAY 9 – Handling Missing and Duplicate Values**

**Focus:** Data integrity.  
**Learn / Do**

* df.dropna(), df.fillna(value), df.duplicated()
* Create a clean\_missing(df) function.  
  **Mini-Task:**  
  Clean missing values using median for numeric and mode for categorical columns.

**DAY 10 – Type Conversions and Date Parsing**

**Focus:** Correct data types.  
**Learn / Do**

* pd.to\_datetime(), .astype()
* Detect wrong dtypes and fix them.  
  **Mini-Task:**  
  Convert all date columns to datetime and all IDs to string.

**DAY 11 – Outlier Detection and Normalization**

**Focus:** Numeric sanity checks.  
**Learn / Do**

* Z-score method using scipy.stats.zscore
* Capping or removing outliers.  
  **Mini-Task:**  
  Remove values outside ±3 z-scores for numeric columns.

**DAY 12 – Aggregation and Reshaping**

**Focus:** Summarization and structure.  
**Learn / Do**

* groupby, pivot\_table, melt
* Combine multiple CSVs.  
  **Mini-Task:**  
  Aggregate sales by region and month from multiple CSVs into one summary file.

**DAY 13 – Reusable Cleaning Pipeline**

**Focus:** Automation.  
**Learn / Do**

* Create a function clean\_dataframe(df) applying all steps:
  + drop duplicates
  + fix types
  + fill nulls
  + remove outliers
* Use it in a loop for multiple files.  
  **Mini-Task:**  
  Apply clean\_dataframe to every file in /data/input/ and save to /data/processed/.

**DAY 14 – Mini Project Wrap-Up**

**Project:** **Data Cleaning Pipeline**

* Reads all raw CSVs
* Cleans with your reusable pipeline
* Generates a summary CSV with dataset stats
* Logs all steps

**Outcome:**  
✅ You can ingest messy real-world data, clean it systematically, and output standardized datasets ready for ETL or SQL load.

**🎯 After Week 2**

You’ll be fluent in:

* File I/O (CSV, JSON, Parquet)
* Logging & error handling
* pandas-based data cleaning
* Writing modular, reusable ETL scripts