

Week 1 Project: File Organizer CLI using `argparse` and `os`

Varun Kumar

Objective

Build a **command-line tool** in Python that automatically organizes files in a directory **based on file extensions**.

Problem Statement

Your **Downloads/** or any working folder often becomes cluttered with mixed file types like `.pdf`, `.jpg`, `.py`, `.zip`, etc.

This tool will **automatically move files into categorized folders** based on their file extensions—like a personal assistant for your messy folders.

Requirements

1. **Command-line Tool** — Run it like this:

```
python file_organizer.py --path /home/ubuntu/Downloads
```

2. The script should:

- Accept a directory path using **argparse**
- Go through every file (not folders) in that directory
- Move each file into a new folder based on its type:
 - `.jpg`, `.png` → **Images/**
 - `.pdf`, `.docx` → **Documents/**
 - `.py`, `.js`, `.c` → **Code/**
 - `.zip`, `.tar`, `.rar` → **Archives/**
 - `.mp3`, `.wav` → **Audio/**

- Others → **Others/**
- Create the folders if they don't exist
- Skip folders and hidden files

Optional Features (Bonus)

- **--dry-run** flag: Show what would be moved, but don't actually move files
- **--recursive**: Traverse subdirectories too
- **--preview**: Print a summary table of how many files per type
- **--log**: Write moved file logs to a **.log** file

Key Python Modules

Module	Purpose
os	Directory listing, file checking
shutil	Move files
argparse	CLI argument parsing
pathlib	(Optional) Modern path handling

Example Before and After

Before Running

```
Downloads/
|- image1.jpg
|- resume.pdf
|- song.mp3
|- program.py
```

After Running

```
Downloads/
|- Images/
|   |- image1.jpg
```

```
|– Documents/  
|   |– resume.pdf  
|– Audio/  
|   |– song.mp3  
|– Code/  
|   |– program.py
```

Test Cases to Verify

1. Run on folder with:
 - No files → should do nothing
 - Mixed file types → should organize properly
2. Run with **--dry-run** → No files should move
3. Run twice → Second time should not crash or duplicate
4. Check if **.log** file correctly logs moved files (if implemented)

Final Deliverables

You should have:

- **file_organizer.py** with clean CLI logic
- A **README.md** file explaining usage
- A screenshot or video showing it working on a messy folder