Day 34: Introduction to File System Automation in Python

📘 Introduction to File System Automation in Python

Objective

By the end of this session, you will:

- Understand what *file system automation* means.
- Know the role of `pathlib`, `os`, and `shutil` libraries.
- Be able to perform basic file and folder operations.

1️⃣ Warm-up & Context

- **Question**: How do you usually manage files/folders on your computer? (create, rename, delete, move)
- **Definition**: *File System Automation* = using Python to perform these tasks automatically.
- File system automation in Python refers to the use of Python scripts to automatically perform tasks related to files and directories on a computer's file system. This automation aims to eliminate manual intervention in repetitive or time-consuming file management operations, thereby increasing efficiency and reducing the potential for human error.
- **Real-world use cases**:
  - Organizing "Downloads" folder.
  - Creating backups.
  - Bulk renaming files.
  - Cleaning up unused data.

2️⃣ Introduction to Libraries

🔷 Pathlib (modern, recommended)

- Object-oriented way of working with file paths.
- Works across operating systems.

```
from pathlib import Path

print(Path.cwd())   # Current directory

path = Path("Documents/example.txt")
print(path.name)    # example.txt
print(path.parent)  # Documents
print(path.suffix)  # .txt
```

Use case: Preferred for clean and intuitive code.

🔷 os (older, procedural)

- Provides functions to interact with the OS.
- Good for low-level operations.

```
import os

print(os.getcwd())        # Current directory
print(os.listdir("."))    # List all files

os.mkdir("NewFolder")     # Create a directory
```

Use case: Still widely used in many projects.

🔷 shutil (high-level operations)

- Used for copying, moving, deleting files/folders.

```
import shutil

shutil.copy("example.txt", "backup.txt")
shutil.move("backup.txt", "Documents/")
shutil.rmtree("OldFolder")   # Delete folder with contents
```

Use case: When working with entire files/folders.

3️⃣ Comparative Overview

Task	pathlib	os	shutil
Get current dir	Path.cwd()	os.getcwd()	❌
List files	.iterdir()	os.listdir()	❌
Create folder	.mkdir()	os.mkdir()	❌
Delete file	.unlink()	os.remove()	❌
Delete folder	.rmdir()	os.rmdir()	shutil.rmtree()
Copy file/folder	❌	❌	shutil.copy() , shutil.copytree()
Move file/folder	.replace()	❌	shutil.move()

4️⃣ Hands-On Demos

1. Check working directory

```
from pathlib import Path
print("Pathlib:", Path.cwd())

import os
print("os:", os.getcwd())
```

2. Create and list folders

```
folder = Path("TestFolder")
folder.mkdir(exist_ok=True)
for item in folder.iterdir():
    print(item)
```

3. Create, rename, and delete a file

```
file = Path("sample.txt")
file.touch()                # create
file.rename("renamed.txt")  # rename
Path("renamed.txt").unlink() # delete
```

4. Copy & Move (shutil)

```
import shutil

Path("copyme.txt").write_text("Hello")
shutil.copy("copyme.txt", "copyme_backup.txt")
shutil.move("copyme_backup.txt", "TestFolder/")
```

5️⃣ Mini-Exercise

Task for Student:

1. Create a folder named `Practice`.
2. Inside it, create two files (`a.txt`, `b.txt`).
3. Rename `a.txt` to `first.txt`.
4. Copy `first.txt` into a new folder `Backup`.
5. Delete `b.txt`.
