24/9/25 - Python shutil Module

The **shutil module** provides functions for **file operations and directory management**.
It is more powerful than the os module for copying and removing folders.

6.1 copy() – Copy a File

```
import shutil
shutil.copy("file1.txt", "copy.txt")
```

Output (effect):

```
['file1.txt', 'copy.txt']
```

🔎 **Theory**

- shutil.copy(src, dest) copies the contents of **src file** to a new file **dest**.
- If dest exists, it may be overwritten.

6.2 copy2() – Copy with Metadata

```
import shutil
shutil.copy2("file1.txt", "backup.txt")
```

Output (effect):

```
['file1.txt', 'backup.txt']
```

🔎 **Theory**

- Similar to copy(), but also copies **file metadata** (creation time, modification time, permissions).
- Useful for making exact backups.

6.3 copytree() – Copy Entire Folder

```
import shutil
shutil.copytree("data", "data_backup")
```

Output (effect):

```
['data', 'data_backup']
```

🔎 **Theory**

- Copies the entire folder (with all files/subfolders) to a new location.
- ❌ Raises error if data_backup already exists.

6.4 move() – Move/Rename File or Folder

```
import shutil
shutil.move("copy.txt", "renamed.txt")
```

Output (effect):

```
['file1.txt', 'renamed.txt']
```

🔎 **Theory**

- Moves a file/folder to another location.
- If used in the same directory, it works as a **rename**.

6.5 rmtree() – Delete Entire Folder

```
import shutil
# ⚠️ Be careful! This will delete folder and all contents permanently
shutil.rmtree("data_backup")
```

Output (effect):

```
['file1.txt', 'renamed.txt']
```

🔎 **Theory**

- Removes the entire directory along with its files and subdirectories.
- **Dangerous**: Once deleted, data cannot be recovered.

6.6 disk_usage() – Disk Space Information

```
import shutil
usage = shutil.disk_usage("/")
print("Total:", usage.total)
print("Used:", usage.used)
print("Free:", usage.free)
```

Output (example):

```
Total: 500107862016
Used: 320000000000
Free: 180107862016
```

🔎 **Theory**

- Returns disk usage statistics for the given path.
- Values are in **bytes** (convert to GB by dividing by 1024**3).