🐍Python `os` Module – Quick Notes

🌍What is `os`?

- A **built-in Python module** for interacting with the **operating system**.
- Helps with **files, directories, environment variables, and processes**.

🕳️Directory Handling

```
import os

print(os.getcwd())        # 📌 Get Current Working Directory
os.chdir("C:/Users")      # 🕐 Change Directory
print(os.listdir())       # 📃 List files & folders
```

🦁Creating & Removing Directories

```
os.mkdir("new_folder")       # 💻 Create single directory
os.makedirs("a/b/c")         # 📣 Create nested directories
os.rmdir("new_folder")       # 🔩 Remove empty directory
os.removedirs("a/b/c")       # 🔩 Remove nested empty directories
```

🕊️File Handling

```
os.remove("file.txt")            # 🗑️ Delete file
os.rename("old.txt", "new.txt")  # ✏️ Rename file
```

🦜Path Handling (`os.path`)

```
os.path.exists("file.txt")     # 🔗 File exists?
os.path.isfile("file.txt")     # 📄 Is it a file?
os.path.isdir("folder")        # 📁 Is it a folder?
os.path.join("folder", "a.txt")# 🔗 Join paths
os.path.abspath("file.txt")    # 📌 Absolute path
```

🌱Environment Variables

```
print(os.environ)              # 🕐 All environment variables
print(os.environ.get("PATH"))  # 📌 Get specific variable
```

System & Process Info

```
os.name         # 💻 OS name → 'nt' (Windows), 'posix' (Linux/Mac)
os.getlogin()   #    Current user
os.getpid()     # 🕐 Process ID
```

🪻 Example: Loop through files

```
import os

for file in os.listdir():
    if os.path.isfile(file):
        print("📄 File:", file)
    else:
        print("📁 Folder:", file)
```

🔔**In short**: `os` = your **toolbox** for OS-level tasks (files, folders, processes, environment).