

23/9/25 - Python **os** Module

The **os** module in Python provides functions to interact with the operating system. It is mainly used for **working with files, directories, and environment variables**.

5.1 **getcwd()** – Get Current Working Directory

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

Output (example):

Current directory: /home/user/project

Theory

- Returns the **absolute path** of the current working directory (CWD).
- The CWD is the folder where your Python script is running.

5.2 **listdir()** – List Files and Folders

```
import os
print(os.listdir())
```

Output (example):

['file1.py', 'data.txt', 'images', 'notes.docx']

Theory

- Lists all files and folders in the current directory.
- Helps to see directory contents.

5.3 **mkdir()** and **rmdir()** – Create and Remove Directory


```
import os
os.mkdir("new_folder")
print("After creation:", os.listdir())
```

```
os.rmdir("new_folder")
print("After deletion:", os.listdir())
```

Output (example):

```
After creation: ['file1.py', 'data.txt', 'new_folder']
After deletion: ['file1.py', 'data.txt']
```

Theory

- `os.mkdir(name)` → Creates a new empty folder.
- `os.rmdir(name)` → Deletes an empty folder.
-  If the folder is not empty, `rmdir()` will give an error.

5.4 `rename()` – Rename File or Folder

```
import os
os.rename("data.txt", "info.txt")
print(os.listdir())
```

Output (example):

```
['file1.py', 'info.txt', 'images']
```

Theory

- Renames a file or folder.
- If the target name already exists, it may overwrite (OS-dependent).

5.5 `remove()` – Delete File

```
import os
os.remove("old_file.txt")
```

Theory

- Deletes the specified file permanently.
 - ⚠ Use carefully: once deleted, it cannot be undone.
-

5.6 path Functions

```
import os
print(os.path.exists("file1.py"))  # Check if file exists
print(os.path.isfile("file1.py"))  # Check if it is a file
print(os.path.isdir("images"))     # Check if it is a directory
```

Output (example):

```
True
True
True
```

Theory

- `os.path.exists(path)` → Checks whether a file/folder exists.
- `os.path.isfile(path)` → True if path is a file.
- `os.path.isdir(path)` → True if path is a directory.

5.7 getlogin() and environ

```
import os
print("Logged in as:", os.getlogin())
print("Path variable:", os.environ.get("PATH"))
```

Output (example):

```
Logged in as: student
Path variable: /usr/local/bin:/usr/bin:/bin
```

Theory

- `os.getlogin()` → Returns the username of the logged-in user.
- `os.environ` → Access system environment variables (like `PATH`, `HOME`).