

## Day 28



### os Module in Python:

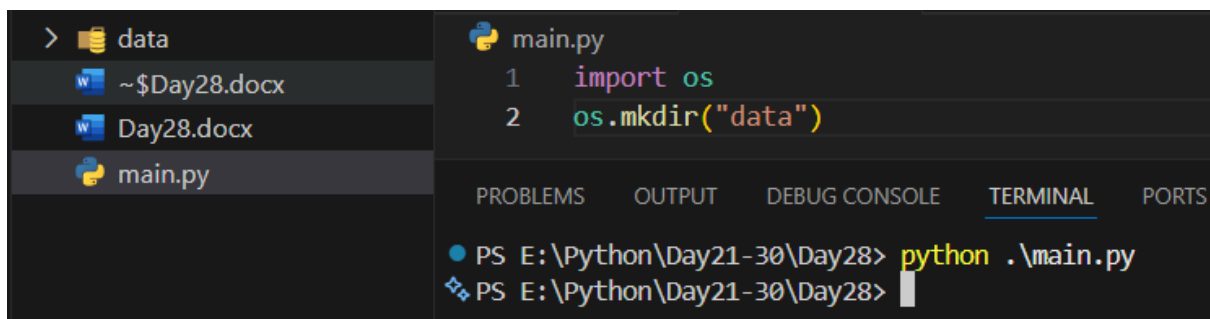
#### What Is the os Module?

- os is a built-in Python module for interacting with the operating system
- Lets you perform tasks like:
  - File and directory management
  - Environment variables
  - Process management

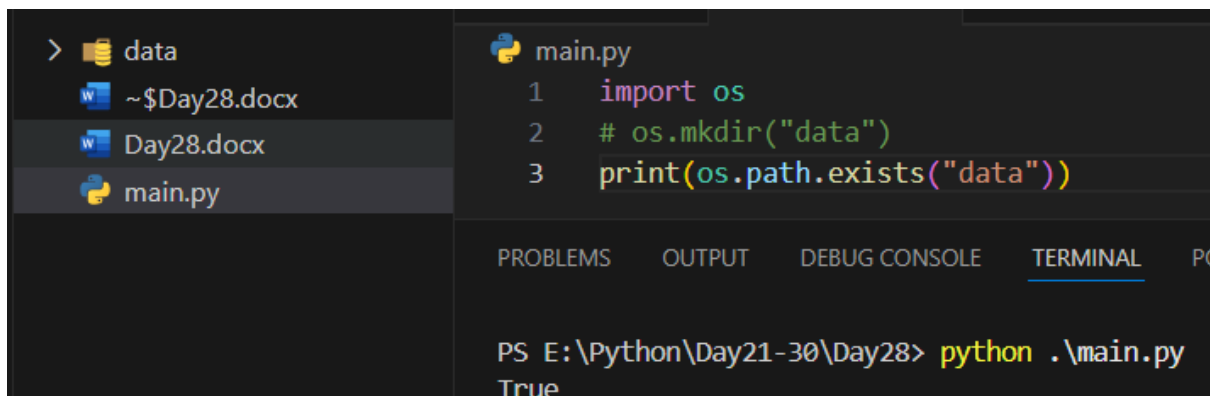
Importing the module:

```
main.py
1 import os
```

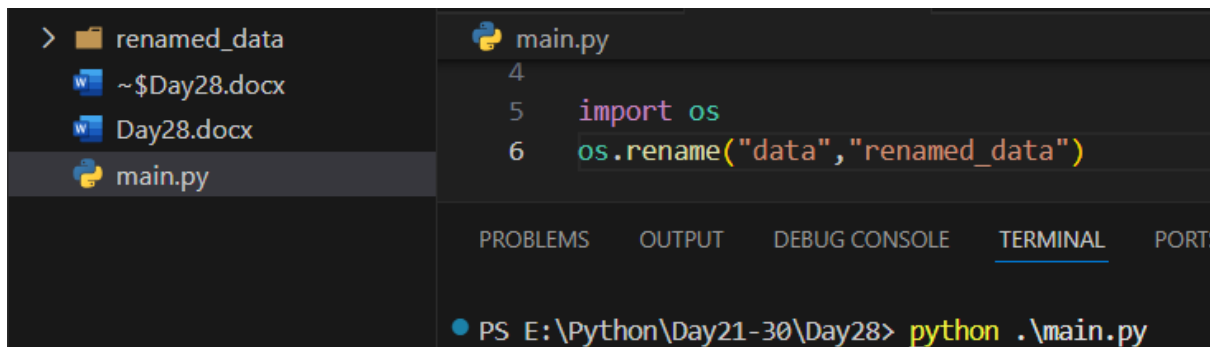
Example: Creating a folder using the os module.

A screenshot of the Visual Studio Code editor. On the left, the Explorer sidebar shows a file tree with a folder named 'data', a Word document '~\$Day28.docx', another Word document 'Day28.docx', and a Python file 'main.py'. The main editor area shows the 'main.py' file with two lines of code: '1 import os' and '2 os.mkdir("data")'. Below the editor, the TERMINAL panel is open, showing the command 'python .\main.py' being executed in a PowerShell prompt at 'E:\Python\Day21-30\Day28', with the output being an empty line.

Example: code to check if a folder exists in our folder or not.

A screenshot of the Visual Studio Code editor. The Explorer sidebar is the same as in the previous screenshot. The main editor area shows the 'main.py' file with three lines of code: '1 import os', '2 # os.mkdir("data")', and '3 print(os.path.exists("data"))'. The TERMINAL panel shows the command 'python .\main.py' being executed, with the output 'True' displayed.

Example: to rename

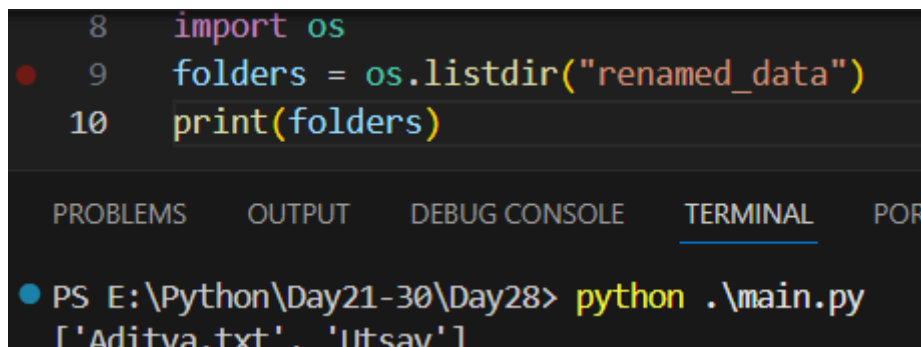


```
4
5 import os
6 os.rename("data","renamed_data")

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS E:\Python\Day21-30\Day28> python .\main.py
```

Example: to print the list of the folders in a specific folder.

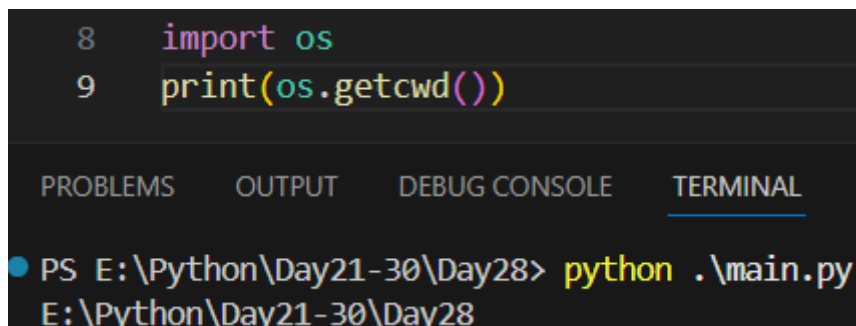


```
8 import os
9 folders = os.listdir("renamed_data")
10 print(folders)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS E:\Python\Day21-30\Day28> python .\main.py
['Aditya.txt', 'Utsav']
```

Example: Current Working Directory

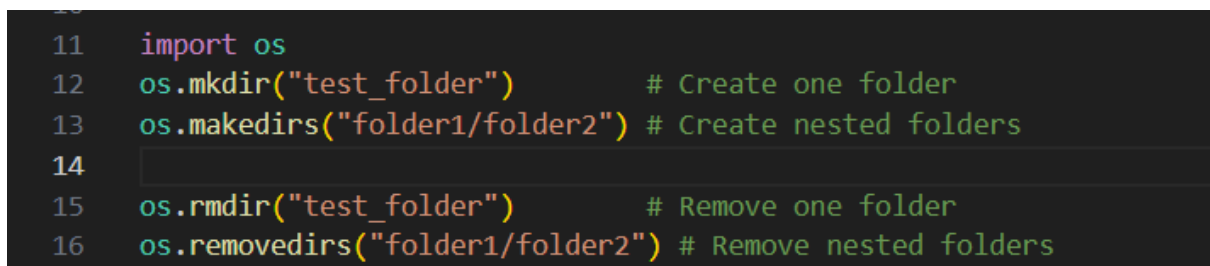


```
8 import os
9 print(os.getcwd())

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

● PS E:\Python\Day21-30\Day28> python .\main.py
E:\Python\Day21-30\Day28
```

Example: Creating and Removing Directories



```
11 import os
12 os.mkdir("test_folder") # Create one folder
13 os.makedirs("folder1/folder2") # Create nested folders
14
15 os.rmdir("test_folder") # Remove one folder
16 os.removedirs("folder1/folder2") # Remove nested folders
```

--The End--