

**Course
&
Test Series**

Working with Directories in Python

 **CBSE**

 **ICSE**

 **NTSE**

 **Banking &
Insurance**

 **Central Govt.
Service**

 **State Govt.
Services**

 **LAW
Entrance**

 **MBA
Entrance**

 **Railways & Metro
Services**

...many more

abhyasonline.in

Working with Directories in Python

A **directory** (also called a folder) is used to **organize files** on your computer. For example:
Projects → contains subfolders like Python, Images, Documents, etc.
Python provides built-in tools to **create, navigate, modify, and delete directories** using the **os** and **shutil** modules.

Importing Required Modules

Before working with directories, import the **os** module (and **shutil** if needed):

```
import os
import shutil
```

os (Operating System) Module

The **os** module allows you to **interact with the operating system** – such as creating folders, deleting files, checking paths, or listing directory contents.

Common uses of os module:

Task	Command	Example
Get current working directory	os.getcwd()	Returns current folder path
Change directory	os.chdir("path")	Move to another directory
Create folder	os.mkdir("NewFolder")	Makes a new folder

Solved Example 1 - Using os module

```
import os

# Create a new folder
os.mkdir("TestFolder")

# Check the current working directory
print("Current Directory:", os.getcwd())

# List all folders/files
print("Contents:", os.listdir())
```

Output:

Current Directory: C:\Users\ABC\Documents
Contents: ['TestFolder']

**Course
&
Test Series**



Working with Directories in Python

shutil (Shell Utilities) Module

The shutil module provides high-level file operations, like copying, moving, or deleting entire folders.

It's more powerful than os for bulk operations.

Common uses of shutil module:

Task	Command	Example
Copy a file	shutil.copy("a.txt", "b.txt")	Copies contents of one file to another
Move file/folder	shutil.move("a.txt", "NewFolder/")	Moves file to another location
Delete a folder (even if not empty)	shutil.rmtree("FolderName")	Removes folder and all contents
Copy entire directory	shutil.copytree("Source", "Destination")	Copies all files and subfolders

Solved Example - Using shutil module

```
import shutil
```

```
# Delete a folder (and everything inside it)
shutil.rmtree("TestFolder")
```

```
print("Folder deleted successfully!")
```

Output:

Folder deleted successfully!

In short:

- os → For creating, navigating, and managing directories and paths.
- shutil → For copying, moving, or deleting entire files and folders.

1. Getting the Current Working Directory

When Python runs, it works inside a default folder known as the **current working directory (CWD)**.

To find it:

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

Output example:

C:\Users\ABC\Documents

You can think of this as “where Python is currently working.”



Course
&
Test Series

Working with Directories in Python

2. Changing the Current Directory

You can change Python's current working directory using:

```
os.chdir("C:\\Users\\ABC\\Desktop")  
print("Now in:", os.getcwd())
```

Tip:

Use **double backslashes (\)** or raw strings (r"path") for Windows paths.

Example:

```
os.chdir(r"C:\Users\ABC\Desktop")
```

3. Listing All Files and Folders

To see what's inside a directory:

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

Output example:

```
['project1.py', 'data.txt', 'Images', 'Videos']
```

You can also specify a path:

```
print(os.listdir("C:\\Users\\ABC\\Desktop"))
```

4. Creating a New Folder

You can make a new directory using **os.mkdir()**:

```
os.mkdir("NewFolder")  
print("Folder created successfully!")
```

This creates a folder named NewFolder inside the current working directory.

To create nested folders (one inside another), use **os.makedirs()**:

```
os.makedirs("MainFolder/SubFolder/Images")
```

5. Renaming a Directory

To rename an existing folder:

```
os.rename("OldFolder", "UpdatedFolder")
```

Result:

Folder OldFolder is renamed to UpdatedFolder.

6. Removing a Directory

If you no longer need a folder, you can remove it.

- To delete an empty folder:

```
os.rmdir("UpdatedFolder")
```
- To delete a folder with contents:

```
shutil.rmtree("MainFolder")
```

Be careful: **shutil.rmtree()** deletes everything inside the folder permanently.



CBSE



ICSE



NTSE



Banking &
Insurance



Central Govt.
Service



State Govt.
Services



LAW
Entrance



MBA
Entrance



Railways & Metro
Services

...many more

abhyasonline.in



Course
&
Test Series

Working with Directories in Python

7. Checking if a File or Folder Exists

Before performing any action, it's a good habit to check if a file/folder exists.

```
if os.path.exists("NewFolder"):  
    print("Folder exists!")  
else:  
    print("Folder not found.")
```

Output example:

Folder exists!

8. Working Example: Creating and Listing Folders

```
import os  
  
# Create folders  
os.mkdir("PythonProjects")  
os.mkdir("PythonProjects/Assignments")
```

```
# Change current directory  
os.chdir("PythonProjects")
```

```
# Print current path  
print("Working inside:", os.getcwd())
```

```
# List all contents  
print("Contents:", os.listdir())
```

Output:

Working inside: C:\Users\Mehul\Desktop\PythonProjects
Contents: ['Assignments']

9. Combining File and Directory Work

You can also use directory operations with file handling:

```
import os  
  
folder = "Reports"  
file_path = os.path.join(folder, "summary.txt")  
  
if not os.path.exists(folder):  
    os.mkdir(folder)  
  
with open(file_path, "w") as f:  
    f.write("Monthly report generated successfully!")  
  
print("File created inside Reports folder.")
```

 CBSE

 ICSE

 NTSE

 Banking & Insurance

 Central Govt. Service

 State Govt. Services

 LAW Entrance

 MBA Entrance

 Railways & Metro Services

...many more

abhyasonline.in

Course
&
Test Series

Working with Directories in Python

This program:

1. Checks if the folder "Reports" exists.
2. Creates it if it doesn't.
3. Writes a file named "summary.txt" inside it.

Why Directory Handling Is Important

1. Helps organize files and projects.
2. Automates creating folders for reports, backups, or logs.
3. Allows working with many files at once (useful in data analysis, web scraping, etc.).
4. Prevents file clutter and accidental overwriting.

Assignment

Ques 1: Create and List a Folder

Question:

Write a Python program to:

1. Create a new folder named "PythonPractice".
2. Print the current working directory.
3. List all the files and folders inside it.

Expected Output Example:

Current Working Directory: C:\Users\Mehul\Documents
Folder 'PythonPractice' created successfully!
Contents: ['PythonPractice']

Ques 2: Check and Rename a Folder

Question:

Write a Python program to:

1. Check if a folder named "OldProjects" exists.
2. If it does, rename it to "ArchivedProjects".
3. Otherwise, print a message saying "Folder not found."

Expected Output Example:

Folder found. Renamed to 'ArchivedProjects'

 CBSE

 ICSE

 NTSE

 Banking & Insurance

 Central Govt. Service

 State Govt. Services

 LAW Entrance

 MBA Entrance

 Railways & Metro Services

...many more

abhyasonline.in

Course
&
Test Series

Working with Directories in Python

Summary

Task	Function	Description
Get Current Directory	os.getcwd()	Shows current working directory
Change Directory	os.chdir(path)	Moves to a new folder
List Items	os.listdir(path)	Lists all files and folders
Create Folder	os.mkdir() / os.makedirs()	Creates directories
Rename Folder	os.rename(old, new)	Renames directory
Delete Folder	os.rmdir() / shutil.rmtree()	Deletes folder
Check Existence	os.path.exists()	Checks if folder/file exists

 CBSE

 ICSE

 NTSE

 Banking & Insurance

 Central Govt. Service

 State Govt. Services

 LAW Entrance

 MBA Entrance

 Railways & Metro Services

...many more

abhyasonline.in