
PYTHON
OS LIBRARY
Comic Story!
Next:
shutil
pathlib


Python is cool... but how does it talk to my computer?
Woah! Hi!
Meet the os module! It connects Python to your operating system!
It lets Python work with files, folders, and system info!
import os
It lets Python work with files, folders, and system info!
Exactly!
The os module helps you talk to folders, paths, and even environment variables.
/path/to.file.txt
Every beginner MUST know this module.
Yeah!
DIRECTORIES
Next, let's
Yeja, let's


Hmm, where am I right now?
Woah! Hi!
Your Python program is always in a folder called the current working directory!
It's where your code is running from!
CURRENT WORKING DIRECTORY
Type this to find it:
> print(os.getcwd())
print(os.getcwd())—got it!
Your current working directory is shown!
print(os.getcwd())
Lost? Let's go home with
> /Users/sriju/projects
Magic!
But, be careful! Moving to non-existent folders causes errors!
> /Users/sriju/homework
Nonexistent
Phew! Ready to list files?
Totally!

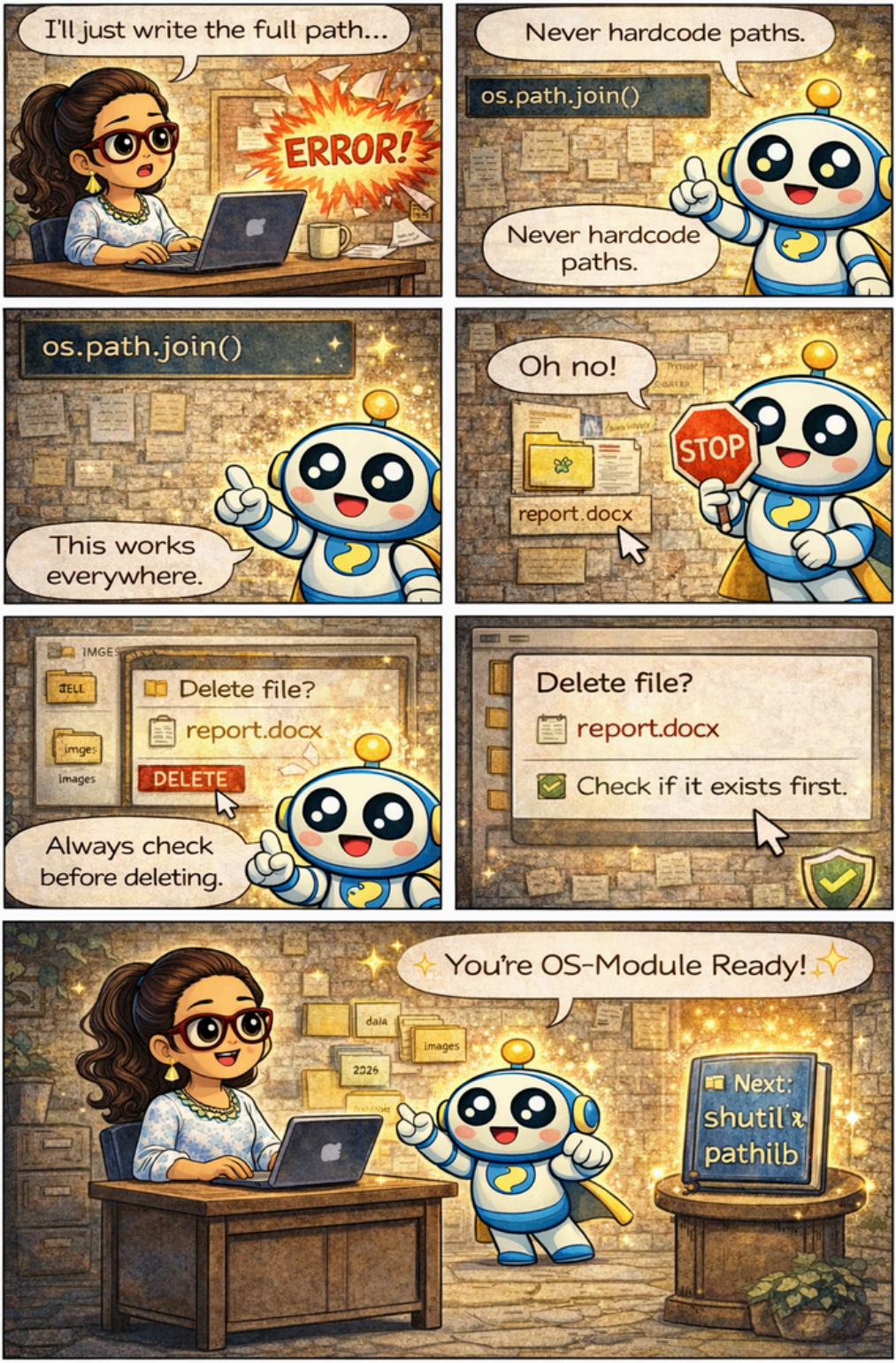

Hmm, where am I right new?
Let's take a look!
Type this to see everything in the current folder:
> os.listdir( )
It returns a list of everything inside!
['report.pdf'
[script.py'
[notes.txt'
[data]
To peek inside another folder, use os.listdir("data")
os.listdir("data")
Ohhh, it includes files AND folders!
Exactly!
To peek inside another folder, use
> os.listdir("data")
os.listdir("data")
DATA
> ['data.csv," data2.csv,info.txt]
Just be careful!
> OSEreor: [Errno 2] No such file or directory: "missing_folder"
CREATE FOLDERS
Let's move on and make new folders!
Let's do it!


I have files... but I need folders to organize them.
Python can create folders for you using the os module.
import os
First, we import the os module.
First, we import the os module.
import os
os.mkdir("images")
This creates one new folder called images.
images
Nice! My folder is created.
Nice! My folder is created.
images
This creates multiple folders at once.
os.makedorks("2026/january/data")
2026
january
data
What if I want to delete a folder?
os.rmdir("2026/january/data")
2026
january
data


How does Python know whether something is a file or a folder?
notes
How does Python know whether
Python can check that using the os.path module.
import os
os.path.isfile(("notes.txt")
This checks whether the path points to a file.
import os
os.path.isfile("notes.txt")
This checks whether the path points to a file.
data
So it returns True if it really is a file.
os.path.isdvr("data")
notes
So it returns True if it really is a file.
Always check before opening, deleting, or modifying paths.
data
Next, let's learn about environment variables.
images
I'm ready!
images
ENVIRONMENT VARIABLES


Where does my computer store its hidden settings?
SECRET SETTINGS
These are environment variables.
os.environ
PATH = /usr/bin
HOME = /home/user
USER = alice
os.environ
These are environment variables.
That's a long list of system paths!
/usr/local/bin
/opt/bin:/usr/bin/usr/bin/usr/sbin
/ust/bin/usr/bin/ust/sibin/sbin
os.environ.get("PATH")
Used for configs & security.
os.name
This tells Python which OS it's running on.
Now my code is cross-platform!
MISTAKES
Linux
Linux


Why do file paths look so confusing?
Don't worry, Sriju. Python can handle paths safely for every system.
import os
path = os.path.join("data", "file.txt")
Windows: data\file.txt
Linux/Mac: data/file.txt
First, we import the os module.
path = os.path.join("data", "file.txt")
data
file.txt
This joins folder and file names correctly on any operating system.
/path/to/file.txt
So I don't have to worry about slashes anymore!
/14" Users/sriju/path (to/file.txt"
This gives the full absolute path of the file.
absolute_path = os.path.absath("file.txt")
file.txt
MORE OS.PATH!
No more hardcoding paths!
Exactly. Let's dive deeper next!


I'll just write the full path...
ERROR!
Never hardcode paths.
os.path.join()
Never hardcode paths.
os.path.join()
This works everywhere.
Oh no!
STOP
report.docx
Delete file?
report.docx
DELETE
Always check before deleting.
Delete file?
report.docx
Check if it exists first.
You're OS-Module Ready!
data
images
2026
Next: shutil & pathlib