

# OS Python Library

Interact with your OS

It's a built-in module, so just 'import os' to use it



**os.rename**  
("initial.txt", "final.txt")

Renames file from initial.txt to final.txt. Similar to mv initial.txt final.txt in shell.

**os.getcwd()**

Get the path of your current working directory. Similar to pwd in shell.

**os.listdir('path')**

Get a list of directories and files in the specified path. Similar to ls path in shell.



**os.mkdir('test')**

Creates a directory with the name test. Similar to mkdir test in shell.



**os.remove**  
('location/filename')

Removes file "filename" from path "location". Similar to rm location/filename in shell.



**os.rmdir**  
('location/dirname')

Removes directory "dirname" from path "location". Similar to rm -r location/dirname in shell.

Can't we just use a shell like bash for this?

But the os library lets you interact with the OS from within a Python program. Which means you can combine other Python features with these.

