

Forum:[Hack Forums](#)

Thread: [Delete All Files & Folders | Recursive Functions \(FREE, Python & PHP\)](#)

Hackers - January 7, 2024, 8:09 am

Delete All Files & Folders

So, you want to bulk delete files & folders on your server or computer right? Easy as pie! In this tutorial you will learn a bit more about PHP and Python. You can use either or for this, and there are also many other ways you could do this (C++, C#, C, Bash, Powershell, etc, etc). I chose PHP and Python because so much of the web is ran by PHP and I know many people use shared hosting which enables this by default. For Python, I chose it because it is a fast scripting language and is commonly used to perform bulk functions like recursive delete, read, analyze, etc.

This is a great way to clear out files and folders on your server, but keep in mind that it won't be able to remove files if it is not run with the right permissions, if there is a protection on the target folder/file, etc. Please be careful: you may not be able to recover deleted files! In the code I am using the path `"/rm"` so that you are forced to test it with a folder `/rm/` first, or change the code yourself. Use at your own risk!

Python - Recursive Deletion - Source Code

Tested as of 1/7/2024 with Python 3.12

Spoiler ([Click to View](#))

Code import os

import shutil

```
def delete_directory(dir_path):
    if not os.path.isdir(dir_path):
        raise ValueError(f"{dir_path} must be a directory")

    for root, dirs, files in os.walk(dir_path, topdown=False):
        for name in files:
            os.remove(os.path.join(root, name))
            # print("os remove" + os.path.join(root, name))
        for name in dirs:
            shutil.rmtree(os.path.join(root, name))
            # print("shutil" + os.path.join(root, name))
    os.rmdir(dir_path)
    #print("os rmdir" + dir_path)
```

Usage

`delete_directory('/rm')`

In our code, we have 1 function: `delete_directory`.

This will take a starting path (ex: `./rm`)

It will then check to make sure it is a directory.

Then, it uses `os.walk`. `os` is the package that allows Python to manipulate and read the operating system for these types of operations.

We use the `os.path` function with the root dir to create the full filepath for the found file/folder.

It will recursively loop through until it targets all subfolders and files from the starting path, for unlimited depth.

It deletes all using a mix of the `shutil` package to remove a whole tree of folders, `os.rmdir` for a specific single folder, and `os.remove` for a single file.

To run our script, it's as easy as:

Code `python run.py` and `run.py` is our main file, in the same root directory as our `./rm` folder for testing.

PHP - Recursive Deletion - Source Code

Tested as of 1/7/2024 with PHP 7.4x.

I don't use PHP much and don't have 8 installed on this specific laptop, but it should be fine for PHP 8x too.

Spoiler ([Click to View](#))

Code `<?php`

```
function deleteDirectory($dirPath) {
    if (!is_dir($dirPath)) {
        throw new InvalidArgumentException("$dirPath must be a directory");
    }
    if (substr($dirPath, strlen($dirPath) - 1, 1) != '/') {
        $dirPath .= '/';
    }
    $files = glob($dirPath . '*', GLOB_MARK);
    foreach ($files as $file) {
        if (is_dir($file)) {
            deleteDirectory($file);
        } else {
            unlink($file);
            // echo " removing " . $file;
        }
    }
    rmdir($dirPath);
    // echo " removing " . $dirPath;
}
deleteDirectory("./rm");
?>
```

We have 1 function 'deleteDirectory' that takes the path. We use `glob` for getting all of the contents inside the folder.

You will see how this is recursive, because if we find another folder inside the starting folder, we re-use `deleteDirectory` func to now delete all files in the found subfolder.

You can see that we use `unlink()` func for single file deletions, and we use `rmdir()` func for removing a directory path. It's important to delete all files in the found folder before deleting the main folder, and this is also why we re-trigger the recursive `deleteDirectory` on the subfolders before deleting the

main parent as well. This is so system doesn't error out that we can't delete a folder that still has files inside.

To run our script, it's as easy as:

Code `php run.php`

or if hooked into the web, access like this:

Code `yoursite.com/path/to/run.php`

and `run.php` is our main file, in the same root directory as our `./rm` folder for testing.

I hope this has helped you! Now you can save these and use them as-needed for bulk deletions.