



How to Search for Specific Files Only in Subdirectories in Python?

by [Shubham Sayon](#)

5/5 - (1 vote)

Table of Contents

☰

Method 1: Using os.walk + endswith + join

```
import os

root_dir = r"C:\Users\SHUBHAM SAYON\Desktop\Parent"
for folder, subfolders, files in os.walk(root_dir):
    if folder != root_dir:
        for f in files:
            if f.endswith(".csv"):
                print("File Name: ", f)
                print(f"Path: ", os.path.join(folder, f))
```

A glimpse at the solutions to follow

Problem Formulation: Let’s say we have a directory containing other subdirectories which further contain files. How do we search for a specific file in the subdirectories in our Python script?

Scenario: We have a parent folder (`Parent`) with child folders (`child_1` , `child_2` , and `child_3`). There are files in the parent directory/folder as well as the subdirectories. We need to find only the `.csv` files that are present only within the subfolders, i.e., `sample.csv` , `heart-disease.csv` , and `car-sales.csv` and ignore the files present in the parent folder and any other file with a different extension. How should we approach this scenario?

Let’s have a quick look at the directory structure that we have to deal with.

```
Parent --> (C:\Users\SHUBHAM SAYON\Desktop\Parent)
| countries.csv
| demo.py
| Diabetes.xls
| hello world.py
| tree.txt
|
+---child_1
|   read me.txt
|   sample.csv
|
+---child_2
|   heart-disease.csv
|   read me.txt
|
+---child_3
|   car-sales.csv
|   read me.txt
```

Raw Copy Extern

Why Fintxter?

"Give me a lever long enough [...] and I shall move the world." 🌐 - Archimedes

Fintxter aims to be your lever! Our single purpose is to increase humanity's **collective intelligence** via programming tutorials so you can leverage infinite computational intelligence to your success! 🧠

Learning Resources

Feel free to [join our free email academy](#) with 1000+ tutorials in Python, freelancing, data science and machine learning, and Blockchain technology!


Also, feel free to check out our [Fintxter books](#) and the [world's #1 freelancer course](#) to create your thriving coding business online. ★★★★★

Freelance Coder

If you're not quite ready to go all in, read over [our blog article on earning your first \\$3,000 as a freelance coder](#).

ALL SIDEBAR LINKS OPEN IN A NEW TAB!

The problem might look daunting initially, but it can be solved with ease since Python provides us with numerous libraries and modules to deal with directories, subdirectories, and files contained within them. So, without further delay, let us dive into the solutions to our mission-critical question.

-  **Important Note:** Each solution takes care of a couple of key points:
- i. How to select only sub-directories files and eliminate the parent directory files?
 - ii. How to select only specific files (that is, `.csv` files in this case) and eliminate other files in the subdirectories?

Video Walkthrough



Method 1: Using `os.walk` + `endswith` + `join`

A Quick Recap of the Prerequisites

- `os.walk` is a function of the `os` module in Python that basically returns a list of three things
 1. The name of the **root directory**.
 2. A list of the **names of the sub-directories**.
 3. A list of the **file names in the current directory**.
- `endswith()` is a built-in method in Python that returns `True` or `False` depending on whether the string ends with a specified value or not.
- The `join()` function allows us to concatenate the elements in a given iterable.

Approach:

- The idea is to use the `os.walk` method and fetch the sub-directories and files within the subdirectories with respect to the parent folder.
- If the folder extracted is not the root/parent folder itself, then we iterate over all the files within the subdirectory. Simultaneously, we check if the file ends with the `.csv` extension with the help of the `endswith` method.
- If `True`, then we simply return the filename. To get the path of the file, `join` the path of the subdirectory and the file name.

Code:

```
import os
```



```
root_dir = r"C:\Users\SHUBHAM SAYON\Desktop\Parent"
for folder, subfolders, files in os.walk(root_dir):
    if folder != root_dir:
        for f in files:
            if f.endswith(".csv"):
                print("File Name: ", f)
                print(f"Path: ", os.path.join(folder, f))
```

Output:

```
File Name:  sample.csv
Path:  C:\Users\SHUBHAM SAYON\Desktop\Parent\child_1\sample.csv
File Name:  heart-disease.csv
Path:  C:\Users\SHUBHAM SAYON\Desktop\Parent\child_2\heart-disease.csv
File Name:  car-sales.csv
Path:  C:\Users\SHUBHAM SAYON\Desktop\Parent\child_3\car-sales.csv
```

Method 2: Using os.listdir + os.path.isdir + endswith

Prerequisites: We already learned about the `endswith` and `join` methods in the previous solution. Let's have a quick look at some other methods that will help us in this approach:

- `os.listdir` is a method of the `os` module that lists all the files and subdirectories present within a specified directory.
- `os.path.isdir()` is another method of the `os` module that is used to check if a specified path is an existing directory or not.
- `os.path.isfile()` is similar to the `os.path.isdir` method, with the only difference being that it checks if the given path is an existing regular file or not.

Approach:

- Iterate over all the subdirectories and files present within the parent folder with the help of the `listdir` function.
- Check if a component within the parent directory is a subdirectory or not. If yes, iterate across all the subdirectories and further check if the content within the subdirectory is a file or not.
- If it is a file, also check if the file ends with a `.csv` extension and then display the filename along with its path.

```
import os

root_dir = r"C:\Users\SHUBHAM SAYON\Desktop\Parent"
for name in os.listdir(root_dir):
    if os.path.isdir(os.path.join(root_dir, name)):
        for file in os.listdir(os.path.join(root_dir, name)):
            if os.path.isfile(os.path.join(root_dir, name, file)) and file.endswith('.csv'):
                print("File Name: ", file)
                print("Path: ", os.path.join(root_dir, name, file))
```

Output:

```
File Name:  sample.csv
Path:  C:\Users\SHUBHAM SAYON\Desktop\Parent\child_1\sample.csv
File Name:  heart-disease.csv
Path:  C:\Users\SHUBHAM SAYON\Desktop\Parent\child_2\heart-disease.csv
File Name:  car-sales.csv
Path:  C:\Users\SHUBHAM SAYON\Desktop\Parent\child_3\car-sales.csv
```

Method 3: Using os.scandir + os.listdir + endswith()

Note: The `os.scandir()` method was introduced in Python 3.5 and is one of the latest methods in Python that allows us to list all the files in a directory. This method does not return a list; instead, it returns an iterator.

Approach:

- List all the contents (files and folders) within the parent directory with the help of the `os.scandir` method.
- Check whether the content is a subdirectory or not. If it is a directory, find the list of all the files present within the subdirectory.
- Check if a file ends with `.csv` extension or not. If yes, display the name of the file and the path of the file.

```
import os
root_dir = r"C:\Users\SHUBHAM SAYON\Desktop\Parent"
for i in os.scandir(root_dir):
    if i.is_dir():
        for file in os.listdir(i):
            if file.endswith(".csv"):
                print(f"Path:{i.path}")
                print("File Name: ", file)
```

Output:

```
Path:C:\Users\SHUBHAM SAYON\Desktop\Parent\child_1
File Name:  sample.csv
Path:C:\Users\SHUBHAM SAYON\Desktop\Parent\child_2
File Name:  heart-disease.csv
Path:C:\Users\SHUBHAM SAYON\Desktop\Parent\child_3
File Name:  car-sales.csv
```

Method 4: Using Pathlib

Approach:

- The idea here is to utilize Python’s `pathlib` module to iterate over the existing contents within the parent directory: `for path in`
`pathlib.Path(root_dir).iterdir()`
- Check if the content is a directory or not. If it is a directory, then use the `pathlib` modules’ `glob` method to check if the subdirectory further has files that end with a `.csv` extension.
- Finally, display the filename along with its path as shown below.

```
import pathlib
root_dir = r"C:\Users\SHUBHAM SAYON\Desktop\Parent"
for path in pathlib.Path(root_dir).iterdir():
    if path.is_dir():
        for file in pathlib.Path(path).glob('*.csv'):
            print("File Name: ", file.name)
            print("Path: ", file)
```

Output:

```
File Name:  sample.csv
Path:  C:\Users\SHUBHAM SAYON\Desktop\Parent\child_1\sample.csv
File Name:  heart-disease.csv
Path:  C:\Users\SHUBHAM SAYON\Desktop\Parent\child_2\heart-disease.csv
File Name:  car-sales.csv
Path:  C:\Users\SHUBHAM SAYON\Desktop\Parent\child_3\car-sales.csv
```


Method 5: Using Glob

The **glob module** in Python is a very effective module that has certain built-in functions that facilitate us with the ability to list *specific* files in a directory. **glob.glob()** is one such function that provides wildcards like “*”, “?”, **[ranges]** that make the process of retrieving a path easy.

Approach:

- Use `glob.glob(path, recursive=True)` to allow Python to recursively search existing subdirectories.
- `/**/*.extension` ensures that all subdirectories are matched, and `.extension` is used to specify the type of file being searched.
- `glob` simply returns the path of the file. To get the filename, you can split the entire path string into a list and grab the last element from the list, which will contain the file name.

```
import glob
root_dir = r"C:\Users\SHUBHAM SAYON\Desktop\Parent"
for path in glob.glob(f'{root_dir}/**/*.csv', recursive=True):
    print("File Name: ", path.split('/')[-1])
    print("Path: ", path)
```

Output:

```
File Name:  sample.csv
Path:  C:\Users\SHUBHAM SAYON\Desktop\Parent\child_1\sample.csv
File Name:  heart-disease.csv
Path:  C:\Users\SHUBHAM SAYON\Desktop\Parent\child_2\heart-disease.csv
File Name:  car-sales.csv
Path:  C:\Users\SHUBHAM SAYON\Desktop\Parent\child_3\car-sales.csv
```

Conclusion

Well! We have discussed as many as five methods to solve the given problem. However, here’s a list of highly recommended articles if you wish to dive deeper into problems like this

- [Find all files in a directory with extension .txt in Python](#)
- [How Do I List All Files of a Directory in Python?](#)
- [How To Get The Filename Without The Extension From A Path In Python?](#)
- [The Most Pythonic Way to Check if a File Exists in Python](#)

Please stay tuned for more interesting articles and discussions. Happy learning!



Shubham Sayon

I am a professional Python Blogger and Content creator. I have published numerous articles and created courses over a period of time. Presently I am working as a full-time freelancer and I have experience in domains like Python, AWS, DevOps, and Networking.

You can contact me @:

[UpWork](#)
[LinkedIn](#)

Related Tutorials

- [The Ultimate Guide to Python Lists](#)
- [100 Code Puzzles to Train Your Rapid Python Understanding](#)
- [Python Regex Superpower \[Full Tutorial\]](#)
- [The Complete Guide to Freelance Developing](#)
- [56 Python One-Liners to Impress Your Friends](#)
- [Freelance Developers, Specialize! 44 Profitable Niches for...](#)

📁 [Operating System](#), [Python](#), [Windows](#)

New Finxter Tutorials:

- [Computer Science Researcher – Income & Opportunity](#)
- [Pandas DataFrame Methods \[Cheat Sheet\]](#)
- [Pandas DataFrame to_coo\(\) Method](#)
- [Pandas DataFrame to_gbq\(\) Method](#)
- [Pandas DataFrame to_string\(\) Method](#)
- [Pandas DataFrame to_feather\(\) Method](#)
- [Pandas DataFrame to_parquet\(\) Method](#)
- [Pandas DataFrame to_xml\(\) Method](#)
- [Pandas DataFrame to_latex\(\) Method](#)
- [Pandas DataFrame to_hdf\(\) Method](#)

Finxter Categories:

Select Category 

[About](#)

[Impressum](#)

[Privacy](#)

[Terms](#)

[Earnings Disclaimer](#)

[Puzzles](#)

[Academy](#)

[Books & Courses](#)

[Freelancer Course](#) 🌴

[Books](#)

[Puzzles](#)

[User Stories](#) ❤️

[Finxter Feedback from ~1000 Python](#)

[Developers](#)

[About](#)

[Course Review from Cristian](#)

[Course Review from Adam](#)