

Report

File System - Directory Traversal

The objective of the provided Python code is to traverse a specified directory, generate a report containing information about folders and files within the directory, and save the report to a user-specified output file. Additionally, the code locates its own file, retrieves its content, and creates a new file to copy the code content programmatically.

Code Structure:

- **Traversing Directory (Function: `traverse_directory`):**
 - The code uses the `os.walk` method to traverse through the specified directory.
 - Information about folders and files, including their names, extensions, sizes, and modification timestamps, is collected and written to the specified output file.
 - During this process, the code also prints the details of each file to the console for real-time progress monitoring.
- **File Content Extraction (Function: `get_current_file_content`):**
 - Utilizes the `inspect` module to determine the path of the currently executing Python file.
 - Reads the content of the current file and returns it as a string.
- **File Content Copying (Function: `copy_content_to_new_file`):**
 - Accepts content and a new file path as parameters.
 - Creates a new file at the specified path and writes the provided content into it.
- **Main Execution (Block: `__main__`):**
 - Takes user input for the folder path to traverse and the output file name.
 - Initiates the directory traversal and writes the report to the output file.
 - Prints progress to the console.
 - Retrieves the content of the current Python file.
 - Copies the content to a new file at a predefined location.

In conclusion, the provided Python script demonstrates a practical application of directory traversal, file I/O, and introspection using the `'inspect'` module. The code successfully fulfills its primary objectives of creating a detailed report on the contents of a specified directory and programmatically copying its own content to a new file. The inclusion of real-time progress updates on the console enhances user awareness during the directory traversal process.

While the script accomplishes its goals effectively, there is room for improvement. The absence of comprehensive error handling might make the script vulnerable to unexpected situations, and incorporating try-except blocks would bolster its reliability. Additionally, providing clearer user guidance, especially in terms of required inputs, could enhance the script's usability for individuals unfamiliar with its functionality.

A notable positive aspect is the utilization of modularity in the form of distinct functions for different tasks. However, further refinement in this regard could be beneficial for code readability and maintainability. For instance, breaking down complex functionalities into smaller, focused functions would facilitate easier comprehension and potential reuse in other contexts.

Furthermore, the script's use of an OS-specific path separator in the `'new_file_path'` assignment might lead to issues on different operating systems. Adopting `'os.path.join'` would ensure cross-platform compatibility, addressing this potential concern.

In summary, the provided Python script is a functional and practical tool for directory analysis and file manipulation. With the suggested improvements, it can be made more robust, user-friendly, and adaptable to diverse environments, aligning with best practices in Python scripting and development.

Screenshot

The screenshot shows the VS Code editor interface with the file `DirectoryTraversal.py` open. The script is a Python program that traverses a directory and its subdirectories, recording the path, file name, extension, size, and modification date of each file. The script is executed in a terminal window, and the output shows the results of the traversal for the folder `/Users/harshsiddhapura/Harsh/Education/MS_IT/Sem-1/IFT510 - Architecture /Python Labs/Module-8/Lab-1/InputTestingFolder`.

```
1  ## Student Name: Harsh Siddhapura
2  ## Student ID: 1230169813
3  ## Date: 11/23/2023
4
5  import os
6  from datetime import datetime
7
8  def traverse_directory(folder_path, output_file):
9      with open(output_file, "w") as file:
10         for root, dirs, files in os.walk(folder_path):
11             file.write(f"Folder: {root}\n")
12             for directory in dirs:
13                 directory_path = os.path.join(root, directory)
14                 folder_name = os.path.basename(directory_path)
15                 file.write(f" - {folder_name}\n")
16             for file_name in files:
17                 file_path = os.path.join(root, file_name)
18                 file_name = os.path.basename(file_path)
19                 file_extension = os.path.splitext(file_name)[1]
20                 file_stats = os.stat(file_path)
21                 file_size = file_stats.st_size
22                 file_modified = file_stats.st_mtime
23                 file_modified_date = datetime.fromtimestamp(file_modified).strftime("%Y-%m-%d %H:%M:%S")
24                 file.write(f" - {file_name} ({file_extension}) - Size: {file_size} bytes, Modified: {file_modified_date}\n")
25             file.write("\n")
26
27 if __name__ == "__main__":
28     folder_path = input("Enter the path of the folder to traverse: ")
29     output_file = input("Enter the name of the output file: ")
30     traverse_directory(folder_path, output_file)
31     print("Traversal completed. The results are saved in", output_file)
32
```

Terminal Output:

```
(.venv) harshsiddhapura@Harshs-MacBook-Air Lab-1 % python3 DirectoryTraversal.py
Enter the path of the folder to traverse: /Users/harshsiddhapura/Harsh/Education/MS_IT/Sem-1/IFT510 - Architecture /Python Labs/Module-8/Lab-1/InputTestingFolder
Enter the name of the output file: Output.txt
Traversal completed. The results are saved in Output.txt
(.venv) harshsiddhapura@Harshs-MacBook-Air Lab-1 % date
Tue Nov 21 23:40:28 MST 2023
(.venv) harshsiddhapura@Harshs-MacBook-Air Lab-1 %
```

The screenshot shows the VS Code editor interface with the file `Output.txt` open. The file contains the results of the directory traversal performed by the `DirectoryTraversal.py` script. The output shows the folder path, the files and subdirectories, and their respective sizes and modification dates.

```
1  Folder: /Users/harshsiddhapura/Harsh/Education/MS_IT/Sem-1/IFT510 - Architecture /Python Labs/Module-8/Lab-1/InputTestingFolder
2  - testing-1.py (.py) - Size: 23 bytes, Modified: 2023-11-21 23:22:29
3  - testing-2.txt (.txt) - Size: 23 bytes, Modified: 2023-11-21 23:22:43
4  - testing-3.json (.json) - Size: 66 bytes, Modified: 2023-11-21 23:23:30
5
6
```

Terminal Output:

```
(.venv) harshsiddhapura@Harshs-MacBook-Air Lab-1 % python3 DirectoryTraversal.py
Enter the path of the folder to traverse: /Users/harshsiddhapura/Harsh/Education/MS_IT/Sem-1/IFT510 - Architecture /Python Labs/Module-8/Lab-1/InputTestingFolder
Enter the name of the output file: Output.txt
Traversal completed. The results are saved in Output.txt
(.venv) harshsiddhapura@Harshs-MacBook-Air Lab-1 % date
Tue Nov 21 23:40:28 MST 2023
(.venv) harshsiddhapura@Harshs-MacBook-Air Lab-1 %
```

```

1  ## Student Name: Harsh Siddhapura
2  ## Student ID: 1230169813
3  ## Date: 11/23/2023
4
5  import os
6  from datetime import datetime
7  import inspect
8
9  def traverse_directory(folder_path, output_file):
10     with open(output_file, "w") as file:
11         for root, dirs, files in os.walk(folder_path):
12             file.write(f"Folder: {root}\n")
13             for directory in dirs:
14                 directory_path = os.path.join(root, directory)
15                 folder_name = os.path.basename(directory_path)
16                 file.write(f" - {folder_name}\n")
17             for file_name in files:
18                 file_path = os.path.join(root, file_name)
19                 file_name = os.path.basename(file_path)
20                 file_extension = os.path.splitext(file_name)[1]
21                 file_stats = os.stat(file_path)
22                 file_size = file_stats.st_size
23                 file_modified = file_stats.st_mtime
24                 file_modified_date = datetime.fromtimestamp(file_modified).strftime("%Y-%m-%d %H:%M:%S")
25                 file.write(f" - {file_name} ({file_extension}) - Size: {file_size} bytes, Modified: {file_modified_date}\n")
26                 print(f" - {file_name} ({file_extension}) - Size: {file_size} bytes, Modified: {file_modified_date}\n")
27             file.write("\n")
28
29  def get_current_file_content():
30     current_file_path = inspect.getfile(inspect.currentframe())
31     with open(current_file_path, "r") as current_file:
32         return current_file.read()
33
34  def copy_content_to_new_file(content, new_file_path):

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH TERMINAL OUTPUT GITLENS COMMENTS

zsh - Lab-1 + v ... ^ x

- (.venv) harshsiddhapura@Harshs-MacBook-Air Lab-1 % python3 DirectoryTraversalUpdated.py
 - Enter the path of the folder to traverse: /Users/harshsiddhapura/Harsh/Education/MS_IT/Sem-1/IFT510 - Architecture /Python Labs/Module-8/Lab-1/InputTestingFolder
 - Enter the name of the output file: Output.txt
 - Traversing the directory and writing to the output file...
 - testing-1.py (.py) - Size: 23 bytes, Modified: 2023-11-21 23:22:29
 - testing-2.txt (.txt) - Size: 23 bytes, Modified: 2023-11-21 23:22:43
 - testing-3.json (.json) - Size: 66 bytes, Modified: 2023-11-21 23:23:30
- Traversal completed. The results are saved in Output.txt
- Content copied to the new file: /Users/harshsiddhapura/Harsh/Education/MS_IT/Sem-1/IFT510 - Architecture /Python Labs/Module-8/Lab-1/CopiedNewFolder/CopiedNewFile.txt
- (.venv) harshsiddhapura@Harshs-MacBook-Air Lab-1 %

Ln 11, Col 55 Spaces: 4 UTF-8 LF Python 3.9.6 (.venv: venv) Blackbox tabnine starter Prettier

```

1  import os
2  from datetime import datetime
3  import inspect
4  import shutil
5
6  def traverse_directory(folder_path, output_file):
7     with open(output_file, "w") as file:
8         for root, dirs, files in os.walk(folder_path):
9             file.write(f"Folder: {root}\n")
10            for directory in dirs:
11                directory_path = os.path.join(root, directory)
12                folder_name = os.path.basename(directory_path)
13                file.write(f" - {folder_name}\n")
14            for file_name in files:
15                file_path = os.path.join(root, file_name)
16                file_name = os.path.basename(file_path)
17                file_extension = os.path.splitext(file_name)[1]
18                file_stats = os.stat(file_path)
19                file_size = file_stats.st_size
20                file_modified = file_stats.st_mtime
21                file_modified_date = datetime.fromtimestamp(file_modified).strftime("%Y-%m-%d %H:%M:%S")
22                file.write(f" - {file_name} ({file_extension}) - Size: {file_size} bytes, Modified: {file_modified_date}\n")
23                print(f" - {file_name} ({file_extension}) - Size: {file_size} bytes, Modified: {file_modified_date}\n")
24            file.write("\n")
25
26  def get_current_file_content():
27     current_file_path = inspect.getfile(inspect.currentframe())
28     with open(current_file_path, "r") as current_file:
29         return current_file.read()
30
31  def copy_content_to_new_file(content, new_file_path):
32     with open(new_file_path, "w") as new_file:
33         new_file.write(content)
34
35  if __name__ == "__main__":
36     folder_path = input("Enter the path of the folder to traverse: ")
37     output_file = input("Enter the name of the output file: ")

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH TERMINAL OUTPUT GITLENS COMMENTS

zsh - Lab-1 + v ... ^ x

- (.venv) harshsiddhapura@Harshs-MacBook-Air Lab-1 % python3 DirectoryTraversalUpdated.py
 - Enter the path of the folder to traverse: /Users/harshsiddhapura/Harsh/Education/MS_IT/Sem-1/IFT510 - Architecture /Python Labs/Module-8/Lab-1/InputTestingFolder
 - Enter the name of the output file: Output.txt
 - Traversing the directory and writing to the output file...
 - testing-1.py (.py) - Size: 23 bytes, Modified: 2023-11-21 23:22:29
 - testing-2.txt (.txt) - Size: 23 bytes, Modified: 2023-11-21 23:22:43
 - testing-3.json (.json) - Size: 66 bytes, Modified: 2023-11-21 23:23:30
- Traversal completed. The results are saved in Output.txt
- Content copied to the new file: /Users/harshsiddhapura/Harsh/Education/MS_IT/Sem-1/IFT510 - Architecture /Python Labs/Module-8/Lab-1/CopiedNewFolder/CopiedNewFile.txt
- (.venv) harshsiddhapura@Harshs-MacBook-Air Lab-1 %

Ln 10, Col 35 Spaces: 4 UTF-8 LF Plain Text Blackbox tabnine starter Prettier

The image shows a VS Code editor interface with a Python script named `DirectoryTraversalUpdated.py` open in the editor. The script is located in the `Module-8 > Lab-1` directory. The script defines a `traverse_directory` function that recursively traverses a directory and writes file information to an output file. It also includes a `get_current_file_content` function and a `copy_content_to_new_file` function. The main function prompts the user for a folder path and an output file name, then executes the traversal and copying tasks.

```

21     file_stats = os.stat(file_path)
22     file_size = file_stats.st_size
23     file_modified = file_stats.st_mtime
24     file_modified_date = datetime.fromtimestamp(file_modified).strftime("%Y-%m-%d %H:%M:%S")
25     file.write(f"{file_name} ({file_extension}) - Size: {file_size} bytes, Modified: {file_modified_date}\n")
26     print(f"{file_name} ({file_extension}) - Size: {file_size} bytes, Modified: {file_modified_date}\n")
27     file.write("\n")
28
29 # Task 1: Print progress on Terminal/Console
30 print("Traversing the directory and writing to the output file...")
31
32 traverse_directory(folder_path, output_file)
33 print("Traversal completed. The results are saved in", output_file)
34
35 # Task 2: Locate the current file and get its content
36 current_file_content = get_current_file_content()
37
38 # Task 3 and 4: Create a new file and copy the content
39 new_file_path = "/Users/harshidhappura/Harsh/Education/MS_IT/Sem-1/IFT510 - Architecture /Python Labs/Module-8/Lab-1/CopiedNewFolder/CopiedNewFile.txt"
40 copy_content_to_new_file(current_file_content, new_file_path)
41 print(f"Content copied to the new file: {new_file_path}")
42
43 if __name__ == "__main__":
44     folder_path = input("Enter the path of the folder to traverse: ")
45     output_file = input("Enter the name of the output file: ")
46
47     # Task 1: Print progress on Terminal/Console
48     print("Traversing the directory and writing to the output file...")
49
50     traverse_directory(folder_path, output_file)
51     print("Traversal completed. The results are saved in", output_file)
52
53     # Task 2: Locate the current file and get its content
54     current_file_content = get_current_file_content()
55
56     # Task 3 and 4: Create a new file and copy the content
57     new_file_path = "/Users/harshidhappura/Harsh/Education/MS_IT/Sem-1/IFT510 - Architecture /Python Labs/Module-8/Lab-1/CopiedNewFolder/CopiedNewFile.txt"
58     copy_content_to_new_file(current_file_content, new_file_path)
59     print(f"Content copied to the new file: {new_file_path}")

```

The terminal window shows the execution of the script. It prompts for the folder path and output file name, then displays the traversal results and the content copied to the new file.

```

(.venv) harshidhappura@Harshs-MacBook-Air Lab-1 % python3 DirectoryTraversalUpdated.py
Enter the path of the folder to traverse: /Users/harshidhappura/Harsh/Education/MS_IT/Sem-1/IFT510 - Architecture /Python Labs/Module-8/Lab-1/InputTestingFolder
Enter the name of the output file: Output.txt
Traversing the directory and writing to the output file...
- testing-1.py (.py) - Size: 23 bytes, Modified: 2023-11-21 23:22:29
- testing-2.txt (.txt) - Size: 23 bytes, Modified: 2023-11-21 23:22:43
- testing-3.json (.json) - Size: 66 bytes, Modified: 2023-11-21 23:23:30
Traversal completed. The results are saved in Output.txt
Content copied to the new file: /Users/harshidhappura/Harsh/Education/MS_IT/Sem-1/IFT510 - Architecture /Python Labs/Module-8/Lab-1/CopiedNewFolder/CopiedNewFile.txt
(.venv) harshidhappura@Harshs-MacBook-Air Lab-1 %

```