

Import Necessary Module

import os

- os module provides functions for interacting with the operating system, such as creating directories and files.

Define Variables

- *Define the desktop path*

```
desktop_path = os.path.join(os.path.expanduser('~'), 'Desktop')
```

Folder name

```
main_folder_name = "My Organized Folder"
```

Number of subfolders

```
num_subfolders = 10
```

- desktop_path: Stores the path to the user's desktop.
- main_folder_name: Specifies the name of the main folder to be created.
- num_subfolders: Determines the number of subfolders to be created within the main folder.

Create Main Folder

Create the main folder if it doesn't exist

```
main_folder_path = os.path.join(desktop_path, main_folder_name)
```

try:

```
os.makedirs(main_folder_path)
```

except FileExistsError:

```
print(f'Folder '{main_folder_name}' already exists on your desktop.")
```

- `os.makedirs(main_folder_path)`: Creates the main folder at the specified path.
- `try-except` block: Handles the case where the folder already exists, preventing errors.

Create Subfolders and Files

`for i in range(1, num_subfolders + 1):`

`subfolder_name = f"Subfolder {i}"`

`subfolder_path = os.path.join(main_folder_path, subfolder_name)`

`os.makedirs(subfolder_path, exist_ok=True)`

`file_path = os.path.join(subfolder_path, "shridevi.py")`

`with open(file_path, 'w') as f:`

`f.write("# This is an empty shridevi.py file.\n")`

- `for` loop: Iterates `num_subfolders` times to create the specified number of subfolders.
- `subfolder_name`: Generates the name for each subfolder.
- `subfolder_path`: Creates the path for each subfolder.
- `os.makedirs(subfolder_path, exist_ok=True)`: Creates the subfolder, handling the case where it already exists.
- `file_path`: Creates the path for the `shridevi.py` file within each subfolder.
- `with open(file_path, 'w') as f`: Opens the file in write mode and writes content to it.

Print Success Message

`print(f'Successfully created folder '{main_folder_name}' with {num_subfolders} subfolders and shridevi.py files on your desktop.')`

- Prints a message indicating the successful creation of the folders and files.

Key Points for task

1. Clarity and Conciseness: Explain each step clearly and avoid unnecessary jargon.
2. Visual Aids: Use diagrams or screenshots to illustrate the folder structure and code execution.
3. Code Demonstration: If possible, demonstrate the code execution live to show the results.
4. Customization: Explain how the code can be customized by changing variable values or adding more features.
5. Error Handling: Emphasize the importance of error handling and how the try-except block prevents issues.
6. Best Practices: Discuss coding best practices, such as using descriptive variable names and following Python style guidelines.

By following these guidelines and providing a clear and engaging presentation, you can effectively explain the code to your audience.