

Move to a Temporary Location Instead of Deletion:

- Instead of deleting the file with `os.remove`, you could move it to a temporary directory (like `/tmp` on Unix systems) using `shutil.move`. Then, after a set period or when space is needed, you can clear out the temporary folder.

```
import shutil
import os

file_path = "your_file.txt"
temp_dir = "/path/to/temp/dir/"
shutil.move(file_path, os.path.join(temp_dir, os.path.basename(file_path)))
```

Rename Instead of Delete:

- Rename the file with a timestamp or other indicator of its deletion status. Later, you can decide whether to delete or restore these files.

```
import os
import time

file_path = "your_file.txt"
renamed_path = f"{file_path}_{int(time.time())}.bak"
os.rename(file_path, renamed_path)
```

Use a File Deletion Library with Trash Support:

- Libraries like `send2trash` move files to the system's trash or recycle bin, allowing you to recover files later if needed.

```
from send2trash import send2trash

file_path = "your_file.txt"
send2trash(file_path)
```

Create Backups Before Deletion:

- Copy or back up the file to another location before using `os.remove`. This provides a form of rolling storage.

