Python Auto File Sorter

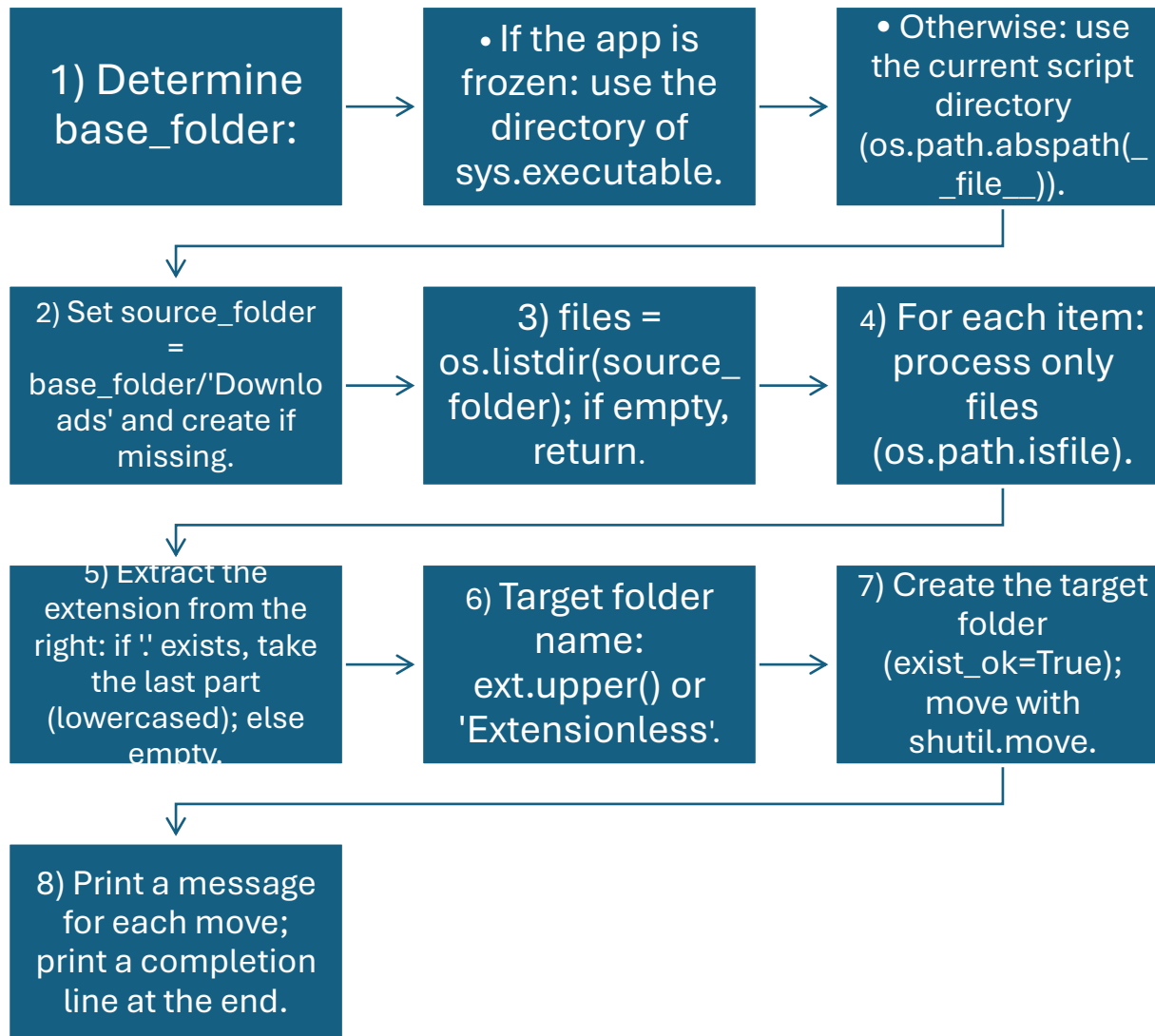Moves files from the 'downloads' folder into extension-based subfolders

Aim & Scope

Aim: Organize the 'Downloads' folder in the same directory as the script/executable.

Scope: Move files into top-level folders according to their extensions.

Result: Files with extensions go into UPPERCASE folders; extensionless files go into 'Extensionless'.

How It Works (Step by Step)

1) Determine base_folder: → • If the app is frozen: use the directory of sys.executable. → • Otherwise: use the current script directory (os.path.abspath(__file__)).

2) Set source_folder = base_folder/'Downloads' and create if missing. → 3) files = os.listdir(source_folder); if empty, return. → 4) For each item: process only files (os.path.isfile).

5) Extract the extension from the right: if '.' exists, take the last part (lowercased); else empty. → 6) Target folder name: ext.upper() or 'Extensionless'. → 7) Create the target folder (exist_ok=True); move with shutil.move.

8) Print a message for each move; print a completion line at the end.

What We Used (From Code)

os: os.path.join, os.path.isfile, os.path.dirname, os.path.abspath, os.listdir, os.makedirs.

shutil: move files via shutil.move.

sys: getattr(sys, 'frozen', False) check and sys.executable.

Conditionals & loops: extension parsing and target folder selection flow.

Folder Layout (Before / After)

```
project/
    ├── auto_sort.py (example)
    └── Downloads/   ← mixed files here
        ├── report.pdf
        ├── photo.jpg
        └── note
```

After running:

```
project/
    ├── auto_sort.py
    ├── PDF/     ← report.pdf
    ├── JPG/     ← photo.jpg
    └── Extensionless/  ← note
```

Code Snippet — Setup

```python
import os, shutil, sys
def auto_sort_files():
    if getattr(sys, 'frozen', False):
        base_folder = os.path.dirname(sys.executable)
    else:
        base_folder =
os.path.dirname(os.path.abspath(__file__))
    source_folder = os.path.join(base_folder, 'Downloads')
    os.makedirs(source_folder, exist_ok=True)
```

Code Snippet — Loop & Move

```python
    files = os.listdir(source_folder)
    if not files:
        return
    for file_name in files:
        source_path = os.path.join(source_folder, file_name)
        if os.path.isfile(source_path):
            if '.' in file_name:
                ext = file_name.split('.')[-1].lower()
            else:
                ext = ''
            folder_name = 'Extensionless' if ext == '' else
ext.upper()
            destination_folder = os.path.join(base_folder,
folder_name)
            os.makedirs(destination_folder, exist_ok=True)
            shutil.move(source_path,
os.path.join(destination_folder, file_name))
            print(f"{file_name} moved to {folder_name}.")
    print('\nAll files are moved according to their extensions.')
```

How to Run?

Place the script in the project folder.

Keep a 'Downloads' folder in the same directory (the code creates it if missing).

Put files into 'Downloads'.

Run the script: auto_sort_files() is called automatically at the end.

Constraints (Current Code)

No name conflict handling (possible overwrite when files share the same name).

No try/except error handling; locked/inaccessible files will raise errors.

Processes only files; does not process subfolders.

Conclusion

auto_sort_files() moves files inside 'Downloads' into extension-based top-level folders.

Extensionless files are collected under 'Extensionless'

For each move, the console prints "<file> moved to <folder>."

If 'Downloads' is empty, the function exits without changes.

It is designed to operate on 'Downloads' next to the script/executable.