

# その他

## restructure\_folders

---

```
#!/usr/bin/env python3
```

```
"""
```

---

Script to restructure raw\_firm folder: - Convert position folders to .txt files - Each company folder will contain position files instead of folders

```
"""
```

---

```
import os
```

---

```
import shutil
```

---

```
from pathlib import Path
```

---

```
def restructure_folders(base_path):
```

```
    """
```

---

Restructure the folder hierarchy by converting position folders to text files.

**Args:**

---

base\_path: Path to the raw\_firm directory

```
    """
```

---

```
base_path = Path(base_path)
```

---

```
# Counter for statistics
```

```
companies_processed = 0
```

---

```
positions_converted = 0
```

---

```
# Iterate through each company folder for company_dir in base_path.iterdir(): # Skip files and special directories
```

**if not company\_dir.is\_dir():**

---

**continue**

---

# Skip special folders like 'jobs', 'processed', etc. if company\_dir.name in ['jobs', 'processed', '.git']:

**continue**

---

print(f"Processing company: {company\_dir.name}")

**companies\_processed += 1**

---

# Iterate through position folders within each company for position\_dir in company\_dir.iterdir():

**if position\_dir.is\_dir():**

---

# Create the new text file name position\_file\_name = f"{position\_dir.name}.txt" position\_file\_path = company\_dir / position\_file\_name

# Create an empty text file for the position with open(position\_file\_path, 'w', encoding='utf-8') as f: # Write a placeholder text that can be edited later f.write(f"# {position\_dir.name}\n\n") f.write("求人情報をここに記入してください。 \n") f.write("(Job information to be added here)\n")

# Remove the empty position folder

**try:**

---

**shutil.rmtree(position\_dir)**

---

**positions\_converted += 1**

---

print(f" ✓ Converted: {position\_dir.name}")

**except Exception as e:**

---

print(f" ✗ Error removing folder {position\_dir.name}: {e}")

print(f"\n=== Restructuring Complete ===") print(f"Companies processed: {companies\_processed}") print(f"Positions converted: {positions\_converted}")

return companies\_processed, positions\_converted

**if \_\_name\_\_ == "\_\_main\_\_":**

---

# Get the base path base\_path = "/Users/bloom0035/Desktop/raw\_firm"

print(f"Starting restructuring of: {base\_path}") print("This will convert all position folders to .txt files")

```
print("-" * 50)
```

---

```
try:
```

---

```
companies, positions = restructure_folders(base_path)
```

```
if positions > 0:
```

---

```
print(f"\n✅ Successfully restructured {positions} positions in {companies} companies!")
```

```
else:
```

---

```
print(f"\n⚠️ No positions were converted. The folders might already be restructured.")
```

```
except Exception as e:
```

---

```
print(f"\n❌ Error during restructuring: {e}")
```

```
import traceback
```

---

```
traceback.print_exc()
```

---

