

A project someone else can open

Where raw data, code and outputs live, why a hardcoded path guarantees the script runs on one machine, and the one rule that makes an analysis rerunnable — never write into the folder you read from.

Pierre Robentz Cassion

Lesson 2 of 8

Python for Programme Data — An environment that will still run

What this lesson covers

- The shape of a project
- data/raw is read-only
- Paths that work on someone else's machine
- Outputs are disposable, and that is the test
- What goes in version control
- Notebooks and scripts do different jobs
- A README that is actually read
- What comes next

The shape of a project — Example

```
muac-analysis/  
  data/  
    raw/           the export exactly as it arrived, never edited  
    interim/       intermediate files, safe to delete  
  outputs/  
    tables/  
    figures/  
  scripts/  
    read_register.py  
    indicator_table.py  
  notebooks/  
    exploration.ipynb  
pyproject.toml  
uv.lock  
README.md  
.gitignore
```

data/raw is read-only — Example

`data/raw/muac-screening-artibonite-2024.v1.csv`

`data/raw/muac-screening-artibonite-2024.v2.csv`

data/raw is read-only — Example

```
data/raw/muac_final.csv
```

```
data/raw/muac_final_v2.csv
```

```
data/raw/muac_FINAL_corrected(1).csv
```

Paths that work on someone else's machine — In Python

Runs on exactly one computer.

```
muac = pd.read_csv("C:/Users/marie/Desktop/muac-screening-artibonite-2024.v1.csv")
```

Paths that work on someone else's machine — In Python

```
from pathlib import Path
import pandas as pd

# __file__ is this script; its parent is scripts/, whose parent is the project.
PROJECT = Path(__file__).resolve().parent.parent
RAW = PROJECT / "data" / "raw"
OUTPUTS = PROJECT / "outputs"

muac = pd.read_csv(RAW / "muac-screening-artibonite-2024.v1.csv")
```

Paths that work on someone else's machine

- **It runs from anywhere.** `python scripts/indicator_table.py` and `'python...`
- **It works on Windows.** `Path` joins with `/` in your source and produces `\` on Windows. Never build a path by string...
- **It is greppable.** Every file the script touches is under `RAW` or `OUTPUTS`, so a reader can see the inputs and...

Paths that work on someone else's machine — In Python

```
from pathlib import Path

PROJECT = Path.cwd().parent      # notebook lives in notebooks/
assert (PROJECT / "data" / "raw").exists(), f"not the project root: {PROJECT}"
```

Outputs are disposable, and that is the test — Shell

```
rm -rf outputs/  
uv run python scripts/indicator_table.py  
git status                # should show nothing unexpected
```

Outputs are disposable, and that is the test — In Python

```
OUTPUTS = PROJECT / "outputs" / "tables"  
OUTPUTS.mkdir(parents=True, exist_ok=True)  
  
indicator_table.to_csv(OUTPUTS / "gam_by_commune.csv", index=False)
```

What goes in version control

Commit	Do not commit
scripts/, notebooks/ pyproject.toml, uv.lock README.md .gitignore	data/raw/ — see below data/interim/ outputs/ .venv/, __pycache__/_

What goes in version control — Example

```
.venv/  
__pycache__/  
*.pyc  
data/raw/  
data/interim/  
outputs/  
.ipynb_checkpoints/
```

What goes in version control

- Never commit raw beneficiary data — identified or pseudonymised

What goes in version control — In Python

```
import os
```

```
# Read from the environment; the value never enters the repository.
```

```
token = os.environ["DHIS2_TOKEN"]
```

Notebooks and scripts do different jobs — In Python

```
# scripts/read_register.py
from pathlib import Path
import pandas as pd

def read_register(path: Path) -> pd.DataFrame:
    return pd.read_csv(
        path,
        dtype={"child_id": "string", "commune": "string"},
        na_values={"muac_mm": ["-99"]},
    )
```

Notebooks and scripts do different jobs — In Python

```
# In the notebook  
import sys  
sys.path.append(str(PROJECT / "scripts"))  
  
from read_register import read_register  
  
muac = read_register(RAW / "muac-screening-artibonite-2024.v1.csv")
```

A README that is actually read — Example

```
# MUAC screening indicator tables
```

Produces GAM and SAM rates by commune from the Artibonite screening register.

```
## Run
```

```
uv sync
```

```
uv run python scripts/indicator_table.py \  
    --input data/raw/muac-screening-artibonite-2024.v1.csv \  
    --output outputs/tables
```

```
## Data
```

data/raw is not committed. Download the register from the programme share and place it there under its versioned filename.

What comes next

- The project has a shape and the paths survive a change of machine.

Read the full lesson, with runnable code

[https://data-analysis.cassion.dev/courses/python-for-programme-data/
python-02-project-layout](https://data-analysis.cassion.dev/courses/python-for-programme-data/python-02-project-layout)