Un projet qu'une autre personne peut ouvrir

Où vivent les données brutes, le code et les sorties, pourquoi un chemin en dur garantit que le script ne tourne que sur une machine, et la règle qui rend une analyse rejouable — ne jamais écrire dans le dossier que l'on lit.

Pierre Robentz Cassion

Leçon 2 sur 8

Python pour les données de programme — Un environnement qui tiendra

Ce que couvre cette leçon

- La forme d'un projet
- data/raw est en lecture seule
- Des chemins qui fonctionnent sur la machine d'un autre
- Les sorties sont jetables, et c'est le test
- Ce qui entre dans le contrôle de version
- Notebooks et scripts font des métiers différents
- Un README réellement lu
- Ce qui vient ensuite

La forme d'un projet — Exemple

```
muac-analysis/
  data/
    raw/            l'export tel qu'il est arrive, jamais modifie
    interim/        fichiers intermediaires, supprimables sans risque
  outputs/
    tables/
    figures/
  scripts/
    read_register.py
    indicator_table.py
  notebooks/
    exploration.ipynb
  pyproject.toml
  uv.lock
  README.md
  .gitignore
```

data/raw est en lecture seule — Exemple

```
data/raw/muac-screening-artibonite-2024.v1.csv
data/raw/muac-screening-artibonite-2024.v2.csv
```

data/raw est en lecture seule — Exemple

```
data/raw/muac_final.csv
data/raw/muac_final_v2.csv
data/raw/muac_FINAL_corrige(1).csv
```

Des chemins qui fonctionnent sur la machine d'un autre — En Python

```
# S'execute sur exactement un ordinateur.
muac = pd.read_csv("C:/Users/marie/Bureau/muac-screening-artibonite-2024.v1.csv")
```

Des chemins qui fonctionnent sur la machine d'un autre — En Python

```python
from pathlib import Path
import pandas as pd

# __file__ est ce script ; son parent est scripts/, dont le parent est le projet.
PROJECT = Path(__file__).resolve().parent.parent
RAW = PROJECT / "data" / "raw"
OUTPUTS = PROJECT / "outputs"

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

Des chemins qui fonctionnent sur la machine d'un autre

- **Il s'exécute depuis n'importe où.** `python scripts/indicator_table.py` et ‘python. . .
- **Il fonctionne sous Windows.** `Path` assemble avec `/` dans votre source et produit `\` sous Windows. Ne construisez. . .
- **Il est repérable.** Chaque fichier touché par le script se trouve sous `RAW` ou `OUTPUTS` : un lecteur voit les. . .

Des chemins qui fonctionnent sur la machine d'un autre — En Python

```python
from pathlib import Path

PROJECT = Path.cwd().parent      # le notebook vit dans notebooks/
assert (PROJECT / "data" / "raw").exists(), f"pas la racine du projet : {PROJECT}"
```

Les sorties sont jetables, et c'est le test — Shell

```
rm -rf outputs/
uv run python scripts/indicator_table.py
git status                        # ne doit rien afficher d'inattendu
```

Les sorties sont jetables, et c'est le test — En Python

```
OUTPUTS = PROJECT / "outputs" / "tables"
OUTPUTS.mkdir(parents=True, exist_ok=True)

indicator_table.to_csv(OUTPUTS / "gam_by_commune.csv", index=False)
```

Ce qui entre dans le contrôle de version

À versionner	À ne pas versionner
`scripts/`, `notebooks/`	`data/raw/` — voir ci-dessous
`pyproject.toml`, `uv.lock`	`data/interim/`
`README.md`	`outputs/`
`.gitignore`	`.venv/`, `__pycache__/`

Ce qui entre dans le contrôle de version — Exemple

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

Ce qui entre dans le contrôle de version

- Ne versionnez jamais de données brutes de bénéficiaires — identifiées ou pseudonymisées

Ce qui entre dans le contrôle de version — En Python

```python
import os

# Lu depuis l'environnement ; la valeur n'entre jamais dans le depot.
token = os.environ["DHIS2_TOKEN"]
```

Notebooks et scripts font des métiers différents — En Python

```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 et scripts font des métiers différents — En Python

```python
# Dans le 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")
```

Un README réellement lu — Exemple

```
# Tableaux d'indicateurs du depistage PB

Produit les taux de MAG et de MAS par commune a partir du registre de
depistage de l'Artibonite.

## Executer

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

## Donnees

data/raw n'est pas versionne. Telechargez le registre depuis le partage
du programme et placez-le la sous son nom de fichier versionne.
```

Ce qui vient ensuite

- Le projet a une forme et les chemins survivent à un changement de machine.

Lire la leçon complète, avec le code exécutable

https://data-analysis.cassion.dev/fr/cours/python-for-programme-data/python-02-project-layout