



# Séance 4 – Projet final

- **Objectifs et compétences :**
    - Concevoir un projet complet en mobilisant les acquis.
    - Collaborer en groupe (3–4 personnes).
  - **Modalités d'animation :**
    - Mise en place des groupes.
    - Définition du cahier des charges (outil de parsing multi-format CSV/JSON/XML avec logs + packaging minimal).
    - Travail autonome avec encadrement.
  - **Évaluation :** état d'avancement du projet.
  - **Matériel :** Python, GitHub/GitLab, VSCode/PyCharm.
- 

## 1 Déroulement

- Constitution de **groupes de 3 à 4 étudiants**.
- Chaque groupe conçoit un **outil de parsing multi-format** capable de :
  - Lire des fichiers CSV, JSON et XML.
  - Transformer les données dans une structure standard Python (list/dict).
  - Ajouter des **logs** pour tracer l'exécution.
  - Être organisé en **package minimal** (modules, `__init__.py`, `main.py`).

## Compétences mobilisées

- Gestion de projet et travail en équipe.
- Structuration et packaging Python.
- Utilisation des logs.

- Parsing de données hétérogènes.
- Documentation et style (PEP8, docstrings).

## 2 Cahier des charges (à définir en groupe)

Chaque groupe doit formaliser un **cahier des charges** minimal comprenant :

- Les formats pris en charge (CSV, JSON, XML).
- Les fonctionnalités prévues (fusion, conversion, affichage, export).
- L'organisation du code (modules, packages).
- La gestion des erreurs (fichier manquant, format incorrect, encodage).
- L'utilisation de logs (niveau INFO, WARNING, ERROR).

👉 Exemple attendu :

```
project_parser/ |— __init__.py |— main.py # Point d'entrée |— parsers/  
| |— __init__.py | |— csv_parser.py | |— json_parser.py | |— xml_parse  
r.py |— utils/ | |— logger.py
```

## 3 Collaboration avec GitHub/GitLab

- Création d'un dépôt Git partagé.
- Règles de contribution (branches, commits clairs, pull requests).
- Suivi des tâches avec Issues/Projects.

### Bonnes pratiques

- Un commit = une idée (pas de "big commit"). <https://www.conventionalcommits.org/en/v1.0.0/>
- Convention de nommage des branches ( `feature/` , `fix/` ).
- Respect du PEP8 ( `flake8` , `black` ).

## 5 Évaluation

- État d'avancement du projet (commit history, organisation, clarté du code).
- Respect du cahier des charges (parsers + logs + packaging).

- **Qualité technique :**
  - Documentation ( `docstrings` , `README.md` ).
  - Logs présents et utiles.
- **Travail collaboratif** (participation équilibrée sur Git).