

# Évaluation – Hackathon de Mise en Pratique pour le Module Scripting Python

**Responsable module :** M. Naceur – [m.naceur@eductive.fr](mailto:m.naceur@eductive.fr)

**Date :** 02/07/2025

**Durée :** 4h30

**Travail en binôme ou trinôme**

**Rendu attendu :** Dossier de projet Python structuré + scripts + README.md + démonstration

**À remettre via MyGes :** .zip + lien Git

## Objectif général

Mettre en œuvre l'ensemble des compétences vues dans le module à travers un **projet**

**DevOps intégrateur** mêlant :

- Automatisation de tâches
- Structuration logicielle
- Interaction avec des services réseau (API météo)
- Utilisation de fichiers et formats variés (CSV, JSON, .env, logs...)

## Contexte du projet

Vous travaillez pour le **Campus Eductive** qui souhaite améliorer sa logistique quotidienne.

**Votre mission :** développer une solution légère permettant de :

- Gérer automatiquement ses listes d'étudiants
- Générer des statistiques simples
- **Décider automatiquement du mode de cours (présentiel ou visioconférence)** selon la météo dans chaque ville d'étudiant
- Sauvegarder les données importantes

💡 En cas de météo défavorable (pluie, neige...), le cours passe automatiquement en **visioconférence**. Sinon, il se tient **en présentiel**.

# Spécifications techniques

## Partie 1 – Gestion des utilisateurs (~1h)

À partir du fichier `etudiants.csv` (nom, âge, email, ville) :

- Demander à l'utilisateur une **ville** (`input()`).
- Filtrer les étudiants de cette ville.
- Trier les étudiants par âge croissant.
- Sauvegarder les résultats dans `exports/<ville>.json`.

## Partie 2 – Organisation des cours selon la météo (~45 min)

Pour chaque ville présente dans le fichier :

- Appeler l'API météo **OpenWeatherMap**
- Décider du mode de cours :
  - **Présentiel** si météo = "Clear" ou "Clouds"
  - **Visioconférence** si météo = "Rain", "Snow", "Thunderstorm", "Drizzle"
- Afficher un message pour chaque étudiant :
  - "Cours en présentiel pour Alice Dupont (Paris) – Météo : nuageux"
  - "Cours en visio pour Bob Martin (Lyon) – Météo : pluie"
- Générer un fichier `decisions_<date>.json` récapitulatif

## Partie 3 – Structure logicielle (*en continu*)

Votre projet doit suivre une structure modulaire :

- `gestion.py` – fonctions liées aux étudiants
- `meteo.py` – appel API et logique météo
- `main.py` – exécution principale
- Paramètres dans un fichier `.env` :
  - `API_KEY`
  - `CSV_PATH`
- Ajouter des logs dans un fichier (logs/) :
  - Lancements, erreurs, résultats météo...

## Partie Bonus (30 à 45 min)

- Interface CLI avec **argparse** :
  - Choix de la ville
  - Option --full pour tout lancer automatiquement
- Génération d'un report.txt avec :
  - Nombre de villes traitées
  - Nombre d'étudiants en visio
  - Nombre d'étudiants en présentiel

## Livrables attendus

- Projet Python bien structuré
- Scripts fonctionnels
- Fichier README.md avec :
  - Instructions d'installation (pip install -r requirements.txt)
  - Instructions d'exécution
- .zip du projet + lien Git (GitHub) à déposer sur MyGes

## Grille d'évaluation (sur 20 points)

CRITERE	POINTS
FONCTIONNALITES OBLIGATOIRES RESPECTEES	/8
QUALITE DU CODE (MODULARITE, CLARTE, LOGS, ERREURS)	/5
STRUCTURE DU PROJET + QUALITE DU README.MD	/3
BONUS (CLI, RAPPORT, DEMONSTRATION)	/2
TOTAL	/20