

Nom de la SAE	Ecriture et lecture de fichiers de données		semestre / Période	Semestre 1
volume horaire consacré par l'étudiant	avec enseignant	6h	en autonomie	12h
coéquipiers :	Magui Azmirly			

Sujet spécifique	Transformation et structuration de données de films depuis un format non classique vers un CSV
Objectifs	Créer un script Python qui lit et convertit un fichier de données textuel spécifique en un fichier CSV structuré, utilisable dans Excel et que les données soient correctement encodées et que les colonnes soient ordonnées et formatées selon les exigences définies
Livrables	Le script python Le fichier CSV généré avec les données de films correctement structurée

Bilan de la SAE

(reproduire le tableau autant de fois que de compétences mobilisées dans la SAE)

Compétence	Traiter des données à des fins décisionnelles
Apprentissages critiques sollicités	Correctement interpréter et prendre en compte le besoin du commanditaire ou du client
	Prendre conscience de l'intérêt de la programmation
	Comprendre les structures algorithmiques de base et leur contexte d'usage
Composantes essentielles à respecter	En écrivant un programme correctement structuré et documenté, respectant les bonnes pratiques
	En intervenant à toutes les étapes du cycle de vie de la donnée (insertion, modification, extraction, suppression)
	En traduisant correctement les demandes métier en programmes, avec le respect du cahier des charges s'il existe

Ma démarche

Savoirs / connaissances	Savoir-faire	Savoir-être
Utilisation du langage Python pour à écrire des scripts pour le traitement et la manipulation des données textuelles.	<ul style="list-style-type: none">-Utiliser des méthodes de chaînes pour formater les textes et les nombres-Transformer un format de données complexe en un format CSV structuré- Maîtriser des structures de données en Python, comme les listes et les chaînes de caractères	<ul style="list-style-type: none">-Porter un attention rigoureuse aux détails lors de la manipulation des données-Communication en équipe

Evaluation du résultat

- Ce que je trouve bien réalisé, pourquoi ?

Je suis satisfaite de la manière dont j'ai réussi à gérer le traitement des données textuelles en Python pour générer un fichier CSV clair et qui respecte les consignes grâce au fait d'avoir rigoureusement suivi les directives pour la disposition des données dans le CSV, comme par exemple l'utilisation de la méthode « .title() » pour les titres et l'ajustement des types de données pour les colonnes. Cela m'a aidée à comprendre tout le processus de la conversion des types et du nettoyage des données.

- Ce que je n'ai pas bien compris ; ce qui serait à améliorer pour une prochaine fois : pourquoi ? comment ?

J'ai eu un peu du mal à simplifier le code ; il y avait une grande boucle qui aurait pu être simplifiée dans le code. La prochaine fois, je pourrais rendre le code plus simple en utilisant des fonctions ou en écrivant des morceaux de code qui peuvent être réutilisés. Cela aiderait à rendre le script plus clair et plus facile à suivre.

Eléments de preuve, ce que je peux montrer

Preuve pour l'apprentissage critique « **Correctement interpréter et prendre en compte le besoin du commanditaire ou du client** » :

```
6 f = open("data_groupe_11-12_B.txt", 'r', encoding='utf-8')
7 texte = f.read()
8 f.close()
9
10 lignes = texte.split('&&') #séparer les films
11
12 data=[]
13 data.append(['Titre','Année','Durée (min)','Revenu (M$)','Nombre de votes','Score',
14             'Metascore','Restriction','Description']) #création de la 1ere ligne
15
```

J'ai commencé par lire les données à partir d'un fichier avec un encodage spécifié (utf-8) et sépare ensuite chaque film à l'aide d'un délimiteur personnalisé (&&). Ainsi, en adaptant le script pour séparer les films et pour insérer les données nettoyées dans la liste « data » j'ai montré que j'ai bien compris et interprété correctement les besoins spécifiques du projet en matière de format et de structure des données, et que j'ai bien compris comment les données doivent être structurées pour répondre aux attentes du commanditaire.

Preuve pour l'apprentissage critique « **Prendre conscience de l'intérêt de la programmation** » et « **Comprendre les structures algorithmiques de base et leur contexte d'usage** » :

```
16 for l in lignes[0:]:
17     if len(l)>0:
18         li = []
19         for c in l.split(', ', maxsplit=8): #séparation de la ligne
20             n=c.split('=')[0] #donnée : nom
21             v=c.split("=")[1] #donnée : valeur
22             if n == "title" or n == "&title":
23                 li.insert(0, v.strip().title())
24             elif n == "year":
25                 v=int(float(v))
26                 li.insert(1, v)
27             elif n == "duration":
28                 if len(v)>0:
29                     v=int(float(v))
30                     li.insert(2, v)
31             elif n == "revenue":
32                 if len(v)>0:
33                     v=float(v)
34                     li.insert(3, v)
35             elif n == "nbvotes":
36                 if len(v)>0:
37                     v=int(float(v))
38                     li.insert(4, v)
39             elif n == "score":
40                 if len(v)>0:
41                     v=float(v)
42                     li.insert(5, v)
43             elif n == "metascore":
44                 if len(v)>0:
45                     v=int(float(v))
46                     li.insert(6, v)
47             elif n == "restriction":
48                 li.insert(7, v.strip().title())
49             elif n == "desc":
50                 li.insert(8, v.strip().replace(";", ", "))
51         data.append(li)
```

En construisant une boucle « for » qui parcourt les éléments de la liste « lignes » et gère différentes conditions pour nettoyer et organiser les données montrent que j'ai eu une utilisation pratique et ciblée des boucles et des structures de contrôle en Python. Cela met en évidence que j'ai bien compris l'utilité de la programmation pour organiser les données efficacement et une compréhension des structures algorithmiques de base adaptées au contexte du projet.