

## TP 13 - Python

---

Comme d'habitude, vous devez rendre votre TP dans le délai indiqué. Lorsqu'un fichier contenant les en-têtes des fonctions et des doctests est fourni, vous devez le compléter. Sinon, vous devez écrire un programme par exercice, avec comme nom **exoX.py** (X est le numéro de l'exercice) et mettre en commentaires les tests que vous avez faits pour vérifier que votre programme fonctionne.

### Exercice 1. Box-office

Pour cet exercice, vous aurez besoin d'utiliser les fichiers :

- **box-office.txt** qui contient les films ayant rapporté plus de 200 millions de dollars au box-office. Chaque ligne du fichier indique, séparés par le caractère # :
  - le titre du film,
  - son revenu (en millions de dollars),
  - son année de sortie.
- **movies-info.txt** qui contient les titres des mêmes films. Après chaque ligne de titre, apparaissent 4 lignes contenant :
  - le genre du film,
  - son studio de production,
  - son ou ses réalisateurs (séparés par des #),
  - son ou ses acteurs principaux (séparés par des #).

#### Indications pour cet exercice (enfin, pas seulement) :

- Assurez-vous d'ouvrir les fichiers avec le bon encodage ('utf-8')
- Pensez à faire des dictionnaires (vous avez le droit d'en faire des variables globales).
- Faites de petites fonctions réutilisables.
- Un fichier **exemple.txt** est fourni, pour voir le genre de sortie attendue.
- Un fichier **exo1.py** est fourni, avec le code du **main** utilisé pour générer l'exemple.

Écrire un programme qui ...

1. ... affiche le réalisateur de votre film préféré (bien entendu, vous devez pouvoir aisément changer de film préféré).
2. ... affiche toutes les informations possibles sur votre film préféré.
3. ... affiche tous les films de votre réalisateur préféré (qui doit pouvoir changer aussi).
4. ... affiche le meilleur (ayant le plus gros revenu) film au box-office, dans votre genre préféré.
5. ... affiche les 10 meilleurs films au box-office, dans votre genre préféré.
6. ... affiche le meilleur film sorti durant l'année de votre naissance.
7. ... affiche le nombre de films sortis cette année-là.
8. ... affiche la liste des films sortis cette année-là.
9. ... affiche le revenu total des films de votre réalisateur préféré.
10. ... affiche le réalisateur réalisant le plus gros revenu au box-office.
11. ... affiche les 10 réalisateurs réalisant les plus gros revenu au box-office.

12. ... affiche l'acteur ayant joué dans le plus de films au box-office.
13. ... affiche les 10 acteurs ayant joué dans le plus de films au box-office.
14. ... affiche les acteurs ayant joué dans exactement 15 films au box-office.
15. ... affiche le genre, le studio et l'acteur réalisant, chacun, le plus gros revenu au box-office.
16. ... affiche les genres des films auxquels a participé votre acteur préféré.
17. ... affiche la liste des films de votre acteur préféré, avec leur genre.
18. ... affiche le studio avec le plus de variété dans les genres produits.
19. ... affiche l'ensemble des acteurs qui ont joué dans les mêmes films que votre acteur préféré.
20. ... affiche l'acteur qui a joué avec le plus d'acteurs différents.

## Exercice 2. ★ Génération du box-office (optionnel)

Les 2 fichiers de données utilisés dans l'exercice précédents ont été générés par un programme Python à partir des informations trouvées là <sup>1</sup> : [https://www.boxofficemojo.com/chart/top\\_lifetime\\_gross/?area=XWW](https://www.boxofficemojo.com/chart/top_lifetime_gross/?area=XWW).

Pour cela, on utilise le module `requests` de Python qui permet de récupérer le source d'une page `html`. Et on extraie les informations qui nous intéressent à grand renfort d'expressions régulières (module `re` introduit au TP 11).

Écrire le programme permettant de générer les fichiers `box-office.txt` et `movies-info.txt`.

---

1. Le format de ces pages ont changé depuis que les fichiers du TP ont été générés, on ne trouve plus que les 200 premiers films au box-office.