

# TP9 : Manipulation de données en csv

## PRÉLÉMINAIRE

Sur le site internet, télécharger l'archive *TP9.zip*.

La désarchiver pour obtenir les fichiers suivants :

- *TP9.py*, le fichier source que l'on va modifier au cours de la séance.
- *series.csv*, le fichier comportant les données sur les séries que l'on va manipuler.
- *genres.csv*, le fichier comportant les données des genres possibles.
- *lien\_genres\_series.csv*, le fichier comportant les numéros liant les séries et les genres enregistrés précédemment.
- *commentaires.csv*, le fichier comportant les données des commentaires sur les différentes séries.
- *dicoFrancais.py*, le fichier qui met en place l'ordre alphabétique français. Ce fichier ne sera pas modifié seulement utilisé.

## EXERCICE 1 : COMPRÉHENSION D'UN CSV

1. A l'aide d'un éditeur de texte (comme **Emacs** par exemple), ouvrir le fichier *series.csv*.
2. A l'aide de **libre office**, ouvrir le fichier *series.csv*. On sélectionnera « séparé par » puis « virgule ». Cette deuxième ouverture permet une lecture plus facile des informations.
  - a. Identifier les différents critères enregistré sur chacune des séries.
  - b. En ouvrant de la même manière *genres.csv* et *lien\_genres\_series.csv*, identifier les genres de la série « Sherlock ».
  - c. En ouvrant de la même manière *commentaires.csv*, déterminer le nombre de commentaires de la série « Game of Thrones ».

## EXERCICE 2 : AUTOMATISATION DE LA LECTURE D'UN CSV

1. Ouvrir le fichier *TP6.py* avec **Idle**, puis exécuter le script :

---

```
print(dict_series())
```

---

Déterminer comment sont enregistrées les données des séries.

2. Maintenant, exécuter le script :

---

```
print(retourner_le_titre_des_series())
```

---

En comprendre les différentes lignes de la fonction utiliser.

3. Sur le même principe, écrire la fonction « retourner\_le\_titre\_des\_series\_et\_date() » qui affiche le message :

---

```
'le titre de la serie est "???", elle existe depuis ???'
```

---

Les points d'interrogations seront remplacés par les informations du fichier *series.csv*.

4. Sur le même principe, écrire la fonction « retourner\_le\_titre\_des\_series\_et\_date\_en\_production() » qui affiche le même message mais seulement avec les séries encore en cours de production.

## EXERCICE 3 : MÉLANGER PLUSIEURS CSV

1. Exécuter le script :

---

```
print(retourner_le_numero_genre_d_une_serie('7'))
```

---

Qu'affiche-t-il ?

2. Modifier la fonction pour obtenir uniquement les numéros de la série « Sherlock ».
3. Ecrire une fonction « dict\_genres() » qui retourne la lecture du fichier *genres.csv* en liste de dictionnaires.
4. Ecrire une fonction « retourner\_le\_genre\_d\_une\_serie(IdSerie) » qui retourne le message :

---

```
'les genres sont ???, ???, ...'
```

---

Les points d'interrogations seront remplacés par les genres dont le numéro aura été saisi.

#### EXERCICE 4 : L'ÉCRITURE D'UN CSV

1. Nous allons utiliser la date et l'heure de l'ordinateur. Pour ceci rajouter le module suivant :

---

```
import datetime
```

---

2. Exécuter le script :

---

```
ecrire_commentaire('IdSerie', 'Note', 'Pseudo', 'Message')
```

---

Comprendre cette fonction, on pourra changer les valeurs saisies.

3. Ecrire une fonction « nouveau\_commentaire() » qui pose les différentes questions sur les données à saisir et exécute la fonction précédente avec les valeurs saisies.
4. Exécuter le script :

---

```
print(retourner_commentaires_d_une_serie('2'))
```

---

Modifier la fonction pour qu'elle affiche également l'année du commentaire.

#### EXERCICE 5 : POUR ALLER PLUS LOIN

1. Un peu de tri, pour ceci on va ajouter les fonctions suivantes :

---

```
from operator import itemgetter  
from dicoFrancais import Compfr
```

---

Puis exécuter le script suivant :

---

```
print(retourner_les_genres())
```

---

Déterminer l'erreur produite. Pour la corriger, on va modifier le mode de triage par :

---

```
Genres.sort(cmp=Compfr(), key=itemgetter('genre'))
```

---

2. Ecrire une fonction qui calcule la moyenne des notes d'une série donnée ou qui retourne « Aucune note » s'il n'existe aucune note pour cette série.
3. Ecrire une fonction qui retourne le titre de cinq séries prises aléatoirement parmi l'ensemble des séries (il faut que les titres soit tous distincts).  
On utilisera la fonction suivante :

---

```
from random import randint
```

---