

TP 3 sur les données en table

Nous allons, durant ce TP, utiliser deux tables :

- La première, stockée dans le fichier « cities.csv », regroupe des informations sur des villes dont monde entier.
- La deuxième, stockée dans le fichier « countries.csv », nous fournira des informations sur les pays.
- **Attention :** Vérifiez bien le séparateur utilisé dans votre fichier csv pour le définir correctement pendant l'appel du fichier.
- **Attention,** pour la fonction `chargement_table`, il faut bien préciser "encoding = "UTF-8" lors de l'ouverture du fichier csv car beaucoup de villes ont des noms avec des accents.

1. IMPORTATION DES DONNÉES

1 EXERCICE 1 : CONSTRUCTION DU MODULE TRAITEMENT_CSV.PY

1. Écrire une fonction `lecture_table` qui charge dans une liste de dictionnaire le contenu d'un fichier csv. Elle prendra deux paramètres :
 - le nom du fichier contenant la table de données à charger
 - le caractère utilisé comme séparateur entre les données d'une ligne

Aide : Inspirez-vous de l'exercice 2 du TP1 sur le traitement de données en table.

2. Compléter le code ci-dessous pour :
 - utiliser la fonction `lecture_table` pour charger le contenu du fichier `cities.csv`
 - afficher les informations de la première ville puis le nombre de villes
 - utiliser la fonction `lecture_table` pour charger le contenu du fichier `countries.csv`
 - Afficher les informations du premier pays puis le nombre de pays.

```
if __name__ == '__main__':  
    ### code lu uniquement si ce programme est celui exécuté  
    # Faire la question 2 ici
```

2. RÉVISIONS SUR LES FILTRES ET LES PROJECTIONS

1 EXERCICE 2

1. Importer le module `traitement_csv.py` et utiliser la fonction `lecture_table` pour charger le contenu du fichier `cities.csv` dans une variable que vous appellerez `villes`
2. Filtrer les villes françaises (reconnaissables car la valeur associée au descripteur 'country_code' vaut 'FR') et stocker le résultat dans une variable que vous appellerez `villes_france`
Aide : Inspirez-vous de la question 3 de l'exercice 3 du TP1 sur le traitement de données en table.
3. Afficher, ligne par ligne, les éléments contenus dans la liste `villes_france`
4. Afficher le nombre de villes françaises

2 EXERCICE 3

1. Importer le module `traitement_csv.py` et utiliser la fonction `lecture_table` pour charger le

contenu du fichier countries.csv dans une variable que vous appellerez `pays`

2. Filtrer les pays dont la superficie (valeur associée au descripteur 'area') est supérieure à 100 000 km². Effectuer une projection pour ne conserver que le nom et la superficie de ces pays. Stocker le résultat dans une variable que vous appellerez `grands_pays`

Aide : Inspirez-vous de la question 3 de l'exercice 3 du TP1 sur le traitement de données en table.

3. Afficher, ligne par ligne, les éléments contenus dans la liste `grands_pays`
4. Afficher le nombre de pays dont la superficie est supérieure à 100 000 km².

3. RÉVISION SUR LA FUSION DE TABLES ET LES TRIS

1 EXERCICE 4

1. Importer le module `traitement_csv.py` et utiliser la fonction `lecture_table` pour charger le contenu du fichier `cities.csv` dans une variable que vous appellerez `villes` puis charger le contenu du fichier `countries.csv` dans une variable que vous appellerez `pays`
2. Croiser ces deux tables pour en construire une nouvelle contenant les capitales des pays. Pour chacune, on ne conservera comme information que le nom du pays, sa densité (nombre d'habitants au km², à calculer en arrondissant au centième !), le nom de la capitale, la latitude et la longitude de la capitale. Stocker cette nouvelle table dans une variable que vous appellerez `r_pays`.
3. Trier `r_pays` par ordre alphabétique des noms des capitales
4. Afficher, ligne par ligne, les éléments contenus dans la liste `r_pays`

----- Exploitation des données de géo-localisation -----

Nous souhaitons créer une carte où seront marquées les capitales des pays avec en popup le nom du pays, sa densité et sa capitale. Pour cela, recopier et compléter le code suivant.

```
ma_carte=folium.Map(location=[0,0],zoom_start="3")
for p in r_pays :
    folium.Marker(.....,.....).add_to(ma_carte)
ma_carte.save("CartePays.html")
```

Coordonnées GPS correspondant à la
marque à placer (sous forme de liste)

Informations que devra contenir le
pop-up (sous forme de chaîne de caractère)

2 EXERCICE 5

1. Importer le module `traitement_csv.py` et utiliser la fonction `lecture_table` pour charger le contenu du fichier `cities.csv` dans une variable que vous appellerez `villes` puis charger le contenu du fichier `countries.csv` dans une variable que vous appellerez `pays`
2. Croiser ces deux tables pour en construire une nouvelle contenant les villes de plus d'un million d'habitants. Pour chacune, on conservera comme information le nom de la ville, son nombre d'habitants, sa latitude et sa longitude, le nom du pays et le continent. Stocker cette nouvelle table dans une variable que vous appellerez `grandes_villes`
3. Trier `grandes_villes` par ordre décroissant de population
4. Afficher, ligne par ligne, les éléments contenus dans la liste `grandes_villes`

----- Exploitation des données de géo-localisation -----

Nous souhaitons créer une carte où seront marquées les villes de plus d'un million d'habitants avec en popup le nom de la ville, son nombre d'habitants, le nom du pays et le continent. Inspirez-vous du travail fait en fin de l'exercice précédent.
