TP VILLES – PAYS v2.0

Créer une fonction importation :

#fonction importation d'un csv dans un dictionnaire de dictionnaires dont les clefs se situent dans la première ligne du CSV (les descripteurs)

# entrée le nom du fichier CSV

# sortie le dictionnaire de dictionnaires

def importation(fichier):

    file = open(fichier, "r", encoding = "utf8")

    table = {}

    descripteurs = ???

#remplissage de table …

**… # à compléter**

file.close()

return table

countries = importation("countries.csv")

cities = importation("cities.csv")

**Dans la fonction imporation()**

1. Lire les descripteurs les mettre dans une liste **descripteurs**
2. Lire les données puis pour chaque ligne créer un dictionnaire en utilisant les données de la liste descripteurs comme clef et les données comme valeur
3. Attention à mettre les données entières ou réelles sous leurs types respectifs.

**Dans le reste du programme python**

* 1. Créer une fonction afficher(datas) permettant d’avoir un affichage pour une ville ou un pays (donné en paramètre, **datas** est un dictionnaire avec les infos d'une ville ou d'un pays)

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

* 1. Créer une fonction afficherTout(datas) permettant d’avoir un affichage de toutes les villes ou de tous les pays (donné en paramètre, **datas** est un dictionnaire de villes ou de pays)

1. Ecrire une fonction **recherche()** qui permet de retourner le pays ou la ville (le dictionnaire) par rapport à son nom
2. Ecrire une fonction **nbPopulationFR()** permettant de retourner le nombre d’habitants de la France.
3. Ecrire une fonction **nbPopulation(pays)** permettant de retourner le nombre d’habitants du pays donné en paramètre. Le paramètre **pays** est le nom du pays.
4. Écrire une fonction **codesDollar()** permettant de retourner la liste des **codes de toutes les monnaies qui s’appellent ‘Dollar’**.
5. Écrire la fonction **top20paysPopulation()** permettant de retourner la liste des **noms des 20 pays les plus peuplés** sous la forme (Pays, Population).
6. Ecrire une fonction **moyenneHabitantVille()** permettant de calculer et retourner la moyenne du nombre d’habitant par villes de chaque pays. Cette fonction doit retourner un dictionnaire avec en clé le nom du pays et en valeur la moyenne du nombre d’habitant par ville
7. Ecrire une fonction **ajouterVille(newCity)**, qui permet d’ajouter une ville dans le dictionnaire et dans le fichier cities.csv. newCity est un dictionnaire

# Tris

Pour trier les pays et villes par rapport à une clé spécifique, on peut écrire la fonction suivante :

def tri(table, cle\_tri):

    # Création d'une liste de tuples à partir du dictionnaire, où chaque tuple est (clé, sous-dictionnaire)

    items\_liste = list(table.items())

    # Tri de la liste de tuples en fonction de la valeur de la clé spécifiée dans chaque sous-dictionnaire

    items\_tries = sorted(items\_liste, key=lambda item: item[1][cle\_tri])

    return items\_tries

# Exemple d'utilisation pour trier les pays par leur nom

pays\_tries\_par\_nom = tri(countries, "Name")

Cette fonction **trier\_dictionnaire\_par\_cle** fonctionne en créant d'abord une liste de tuples à partir du dictionnaire donné, où chaque tuple est formé de la clé principale du dictionnaire (par exemple, le code ISO pour un pays) et du sous-dictionnaire correspondant. Ensuite, elle trie cette liste de tuples en utilisant la fonction **sorted** avec une fonction lambda qui spécifie de trier les tuples en fonction de la valeur associée à la clé de tri dans le sous-dictionnaire (par exemple, le "Name"). La liste triée est ensuite retournée.

1. Créer une nouvelle fonction tri qui se nomme **tri\_par\_nom**() qui prend en paramètre table (le dictionnaire de pays ou de ville) et qui retourne la liste de tuples triés par rapport au nom de la ville ou du pays. (Il faut reprendre la fonction tri et changer un tout petit truc)

# Carte (Bonus)

Les *séries* peuvent faire l’objet de graphiques.

On peut réaliser toutes sortes de **graphiques** en exploitant les bibliothèques numpy et matplotlib  :

**import** numpy

**import** matplotlib**.**pyplot **as** plt

Par exemple, on peut réaliser une carte des villes utilisant la [projection de Mercator](https://fr.wikipedia.org/wiki/Projection_de_Mercator) en effectuant :

def afficher\_monde():

    x=[]

    y=[]

    for citie in cities.values():

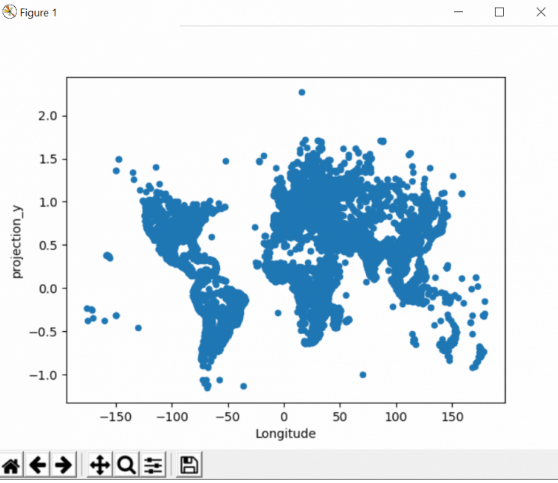
        x.append(float(citie['Longitude']))

        y.append(numpy.arcsinh(numpy.tan(float(citie['Latitude']) \* numpy.pi / 180)))

    plt.scatter(x, y)

    plt.show()

1. Afficher toutes les villes du monde (juste reprendre le code au-dessus et appeler la fonction)



1. Créer une fonction afficher\_USA() qui affiche toutes les villes des USA.
2. Créer une fonction afficher\_amerique() qui affiche toutes les villes du continent américain.
3. Créer une fonction afficher\_carte(pays) qui affiche toutes les villes du pays donné en paramètre