TP Pokémons v2.0

# Préparation du travail

1. Créer un nouveau dossier : **tp\_pokemon\_v2.0** (dans votre clé ou espace perso)
2. Dans VSCODE cliquer sur **Open Folder** et sélectionner le dossier tp\_pokemon\_v2.0
3. Créer un fichier python (.py) nommé : **pokemon\_v2.0\_nom-prenom.py**
4. Après avoir récupéré le fichier **pokemon.csv**, le déposer dans le dossier tp\_pokemon\_v2.0
5. Une fois le TP terminé, remettre dans l’affectation TEAM le fichier .py et le .csv

# Le travail à effectuer

Dans ce TP nous allons proposer une structure de données pour représenter des Pokémons et sauvegarder les Pokémons dans un fichier csv

Ci-dessous un extrait du fichier **pokemon.csv**

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

Description générée automatiquement

**Note importante** : Il est à noter que l’ID ici **#**, ne permet d’identifier de façon unique les pokémons car certains pokémons ont le même ID. Il faudra donc choisir comme clé primaire le **Name** qui lui est bien unique. Aucun Pokémon peut avoir deux fois le même nom.

# La base

1. Créer la fonction **importation(fichier)** qui permet de retourner un dictionnaire de dictionnaires. Chaque dictionnaire est associé à une clé : le nom du pokémon et toutes les informations de chaque pokémon se trouvent dans le sous-dictionnaire (Reprendre la v1.0 de pokémon)

#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 la liste créée

def importation(fichier):

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

    table = {}

    file.close()

    descripteurs = ???

#remplissage de table …

return table

pokemons = importation("pokemon.csv")

1. Reprendre les différentes fonctions de la v1.0 et modifier en conséquence les fonctions de recherche, d'affichage et d'ajout
   * afficherPokemon()
   * afficherPokemons()
   * ajouterPokemon()

Le Pokémon ajouté doit être ajouté au fichier CSV !

# Filtres

1. Écrire une fonction **pokemonEau()** permettant de retourner la liste des **noms** des **Pokémons de type Eau (Water dans le CSV)**
2. Écrire une fonction **pokemonMore50HP()** permettant de retourner la liste les **noms des Pokémons de plus de 50HP**
3. Ecrire une fonction **filtre(critere, val)** qui prend en paramètre **critere** (un descripteur existant dans le dictionnaire) et **val** la valeur du critère et qui retourne la liste des noms de pokémons qui correspondent au critère.

# Tris

1. Créer une fonction de tri par nom de Pokémon. La fonction retourne une liste de nom de Pokémons trié.

# Pour terminer

1. Écrire une fonction doublons permettant de rechercher les doublons ID (#) de Pokémon dans la table pokemon.csv, ici l'algorithme retourne une liste de tuples : les doublons : l’ID (#) et le nom du pokémon

Extrait du résultat attendu :

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

Description générée automatiquement

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

Description générée automatiquement

**Objectif** :

 🡺 Une image contenant noir et blanc, croquis, portrait, texte

Description générée automatiquement 🡺 🡺Une image contenant dessin humoristique, Animation

Description générée automatiquement