

Ce dernier TP vous permet de faire de la manipulation de fichiers.

1 Ouverture, lecture, écriture

Mise en place

Créez un dossier TP3 sur votre ordinateur, téléchargez l'archive "ressources.zip" depuis mon site internet et décompressez là dans ce dossier. Voici une liste (non exhaustive) des commandes dont vous risquez d'avoir besoin.

- **import os** : permet d'importer le module os, qui vous permettra de vous déplacer dans l'ordinateur, créer des dossiers, etc
- **os.getcwd()** : renvoie le dossier de travail de python (attention, ce n'est pas toujours le dossier où se trouve votre fichier .py)
- **os.chdir(<chemin>)** : permet de modifier le dossier de travail pour se rendre au chemin indiqué en paramètre.
- **os.makedirs(<nom de dossier>)** : permet de créer un dossier.
- **fichier = open(<nom de fichier>, mode, encoding)** : créer un objet contenant un fichier sur votre ordinateur. on utilisera généralement "encoding = 'UTF-8'" et "mode = 'r'" (pour lire) ou "mode = 'w'" (pour écrire). Si vous souhaitez écrire dans un fichier qui n'existe pas, cela le crée.
- **for ligne in fichier** : permet de parcourir les différentes lignes du fichier.
- **fichier.close()** : permet de fermer le fichier afin de ne pas surcharger la mémoire.
- **fichier.read()** renvoie l'ensemble des caractères dans le fichier. Les sauts de lignes sont représentés par le caractère `\n`

Question 0

Ouvrez le fichier et affichez ses différentes lignes.

Essayer d'afficher 2 fois le contenu du fichier.

Question 1

Créez une classe *pokemon* ayant pour attributs : *num*, *nom*, *pdv*, *attaque*, *defense*, *vitesse* et *special*. Tous ces attributs seront des entiers sauf *nom* qui sera une chaîne de caractère.

2 Manipulation des données

Question 2

Créez une liste *pokedex* et stockez-y les 151 Pokémon du fichier (on créera donc des objets de type *pokemon* pour chacun d'eux). on pourra utiliser la méthode *split('^t')* du type str permettant de séparer une chaîne de caractère à chaque occurrence du caractère `'\t'`, correspondant à une tabulation.

Question 3

Ajouter à la classe *pokemon* un attribut *type* valant (*None*, *None*) par défaut. Charger le fichier *types.txt* et mettez à jour votre liste *pokedex* pour que chaque *pokemon* qu'elle contient ait bien ses types enregistrés. Si un *pokemon* a deux types, on les enregistrera sous la forme (*type1*, *type2*), s'il n'en a qu'un, on l'enregistrera sous la forme (*type1*, *None*).

3 Création de données

Question 4

À l'aide d'un *try*, *except*, Créer un dossier *pokedex* dans le répertoire de travail. On rattrapera l'erreur si le dossier existe déjà pour juste le signaler à l'utilisateur.

Question 5

Exécutez le code suivant :

```
1 from PIL import Image
2 image = Image.open('sprites/abo.png')
3 image.show()
```

Ajoutez une méthode `__repr__` à la classe *pokemon* pour que lorsque l'on *print* un Pokémon, cela affiche une image du *pokemon* et renvoie son nom.

Question 6

Importer la fonction `code_HTML` du fichier *generateur_HTML.py* à l'aide de la commande suivante :

```
1 from generateur_HTML import code_HTML
```

Cette fonction prend en entrée un titre, un *pokemon* et un chemin vers une photo et génère le texte de la page html qui contient la page pokedex du Pokémon.

Écrire un script qui génère dans le dossier pokedex les 151 pages HTML des 151 Pokémon.