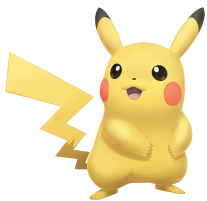


Pokémon

Un jour, je serai le meilleur Développeur...



Développer votre version du jeu de Pokémon. Soyez le meilleur dresseur en remportant des combats et en collectionnant un maximum de Pokémon.



Chaque Pokémon a un **nom**, **des points de vie** (par défaut 100), un **niveau**, une **puissance d'attaque** et une **défense** (par défaut 0). La défense permet de se protéger des dégâts subis.

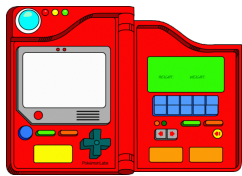
Dans l'univers de ces petites créatures, il existe 18 types de Pokémon. Chaque Pokémon possède au minimum **un type** qui conditionne les caractéristiques du Pokémon, le nombre de points de vie, la puissance d'attaque, sa défense...



Un système de combat entre deux Pokémon doit être développé.

Lors des combats, la vie des Pokémon est vérifiée. Si la vie d'un Pokémon est à zéro, un message est affiché en console avec le nom du Pokémon gagnant.

Bien sûr, il peut arriver qu'un Pokémon loupe son attaque.



Comme vous êtes un dresseur de Pokémon avertie, vous avez toujours sur vous votre Pokédex. Le Pokédex est un outil indispensable. Il sert à enregistrer les informations des Pokémon combattus.(nom, type, défense, puissance d'attaque et point de vie).

Les Pokémon seront sauvegardés dans un fichier nommé **pokédex.json**, une vérification doit être effectuée afin d'éviter les doublons. Cet outil permet d'afficher l'ensemble des Pokémon rencontrés ainsi que leur nombre.

En prenant en compte les informations ci-dessus, écrire la classe "**Pokemon**", dont certains de ses attributs doivent être **privés** (nom et point de vie).

Créer au minimum **quatre types** de Pokémon : **Normal**, **Feu**, **Eau** et **Terre**.

Les classes concernant les types **héritent** de la classe "**Pokemon**". Chacune de ces catégories modifie le nombre de points de vie, la puissance d'attaque et la défense. Une méthode doit être implémentée permettant d'afficher en console les informations générales du Pokémon (nom, vies, défense, attaque).

Ajouter une classe "**Combat**". Cette classe permet la gestion complète du jeu.

Cette classe doit avoir au minimum :

- Une méthode qui **vérifie** si l'un des deux Pokémon n'est plus en vie.
- Une méthode qui **renvoie** le nom du vainqueur.
- Une méthode qui **choisit aléatoirement** 1 ou 0. Si 1 est retourné, alors le Pokémon peut attaquer et infliger des dégâts à son adversaire, sinon il loupe son attaque.
- Une méthode qui **recupère** le type de l'adversaire et **recupère** sa puissance d'attaque et le multiplie en suivant le tableau suivant :

Attaque\Défense	L'eau	Feu	Terre	Normal
L'eau	1	2	0,5	1
Feu	0,5	1	2	1
Terre	2	0,5	1	1
Normal	0,75	0,75	0,75	1

Par exemple, si l'attaquant est de type eau avec 10 en attaque, qu'il attaque un Pokémon de type terre, les dégâts subis seront de 5 PV. ($10 * 0.5$).

- Une méthode qui **enlève** des points de vie en fonction de la défense
- Une méthode qui **renvoie** de nom du Pokémon perdant
- Une méthode qui **enregistre** le Pokémon rencontré dans votre Pokédex

Un menu en début de partie est proposé au joueur permettant les actions suivantes :

- Lancer une partie
- Ajouter un Pokémon (dans un fichier texte nommé "**pokemon.json**")
- Accéder à son Pokedex.

Lors du démarrage de la partie, le joueur doit renseigner avec quel Pokémon il veut jouer, selon la liste des Pokémon. L'adversaire est quant à lui choisi de façon aléatoire dans le fichier "**pokemon.json**".



Faites combattre deux Pokémon.

Chaque classe doit être dans un fichier séparé.

Pour aller plus loin

Ajoutez plus de types de Pokémon et faites en sorte qu'un Pokémon ait deux types. De plus, un Pokémon a la possibilité d'évoluer. Ajoutez tout ce dont vous avez besoin afin de les faire évoluer.

Afin que votre jeu soit plus agréable, ajouter une interface graphique (celle de votre choix)

Pour aller beaucoup plus loin

Au lancement du programme, l'utilisateur peut choisir de lancer une nouvelle partie ou de reprendre une sauvegarde existante. Si le joueur relance une nouvelle partie, son pokédex est remis à zéros.

Rendu

Dans votre répertoire GitHub public "**pokemon**".

N'oubliez pas d'envoyer vos modifications dès qu'une étape est avancée ou terminée et utilisez des commentaires explicites.

Compétences visées

- Maîtriser l'architecture POO en Python
-

Base de connaissances

- [Les classes - Documentation officielle](#)
- [Apprenez la programmation orientée objet avec Python](#)
- [Tutoriel Classe python](#)
- [Classe python](#)
- [L'héritage](#)
- [Passage de référence](#)
- [Liste des Pokémon](#)
- [JSON - Lire et écrire dans un fichier JSON](#)