

TP Pkmn

- 1) Créer une simple application web

```
/pkmntp  
  index.html  
  /css  
    base.css  
  /js  
    app.js  
  /assets
```

index.html point d'entrée de l'application

base.css fichier css global de l'application

app.js point d'entrée javascript de l'application

/assets dossier fourre tout pour les ressources de l'application

- 2) Intégrer Bulma

```
<link rel="stylesheet"  
href="https://cdn.jsdelivr.net/npm/bulma@0.9.4/css/bulma.min.css">
```

Cet outils css permet de gérer facilement la disposition de la page web :

<https://bulma.io/documentation/overview/start/>

- 3) A l'aide de bulma créer un header avec :

a gauche le nom de l'application, a droite le nom du joueur ainsi qu'une image de profil

- 4) Créer à l'aide de la balise template une carte d'affichage (image du pkm + nom de celui-ci) pour un pokémon

exemple : (vous avez le droit de faire des trucs plus beau ou mettre un peu plus d'infos)



- 5) A l'aide de pokeapi, récupérer la liste de tous les pokémon :

<https://pokeapi.co/api/v2/pokemon?limit=100000>

Pour chaque pokémon reçu avoir la possibilité de cliquer dessus et d'afficher les détails de celui-ci au travers d'une modale. le détail d'un pokémon est accessible à l'adresse suivante : <https://pokeapi.co/api/v2/pokemon/muk> (ici muk = grotadmorv, c'est un paramètre qui changera en fonction du nom du pokémon sur lequel on a cliqué).

exemple de code pour une modale simple :

https://www.w3schools.com/howto/howto_css_modals.asp

Le code html de la modale doit être dans une balise template afin d'être récupéré et manipulé plus facilement.

6) Modifier le code de la page pour mettre en place une pagination. Charger l'ensemble des pokémons peut être long (plus de 1000). N'en charger que 50, le afficher, puis arriver au bout du scroll (bas de la page) charger les 50 suivants, et ainsi de suite.

la pagination côté serveur (pokeapi) est gérée à partir du paramètre offset :

<https://pokeapi.co/api/v2/pokemon?limit=50&offset=0> va renvoyer les 50 premiers pokémon (paramètre limit) en partant du premier (paramètre offset = 0)

<https://pokeapi.co/api/v2/pokemon?limit=50&offset=50> va renvoyer 50 pokémon (paramètre limit) en partant du 50eme (paramètre offset = 50)

7) Créer un joueur pour la suite du TP

a) créer un bouton 'pokedex' et un bouton 'joueur'

appuyer sur le bouton joueur affiche soit la fiche du joueur remplie soit un formulaire permettant de créer un nouveau joueur (nom, pseudo, âge, lieux de départ, image de profil, ...)

appuyer sur le bouton pokedex affiche la page des pokémon précédemment créée

Le formulaire de saisie du joueur doit permettre d'enregistrer les infos saisies dans le localStorage afin d'être sauvegardées et d'être utilisées lors du rechargement de la page.

b) Si un joueur existe dans le local storage, remplacer dans le header, le nom et l'image de profil en fonction des données sauvegardées.

8) Choisir un starter

Une fois le joueur créé, ajouter la possibilité de choisir un pokémon starter (récupérer les informations et ids des différents pokémon sur pokeapi), et enregistrer l'information.

Modifier la fiche du joueur pour ajouter l'information de son pokémon starter choisi.

A partir de là on va partir sur la création d'un jeu Pokémon lite. Avec du déplacement de dresseur, de l'apparition de pokémon aléatoire, du combat, de la capture,