

TD ÉVALUÉ

CONSIGNES GÉNÉRALES

Dans le cadre de ce TD évalué, vous avez 2 exercices différents à réaliser dans les 4h imparties. Réalisez les exercices dans l'ordre de priorité suivant: Exercice 1 -> Exercice 2 -> Les consignes "POUR ALLER PLUS LOIN"

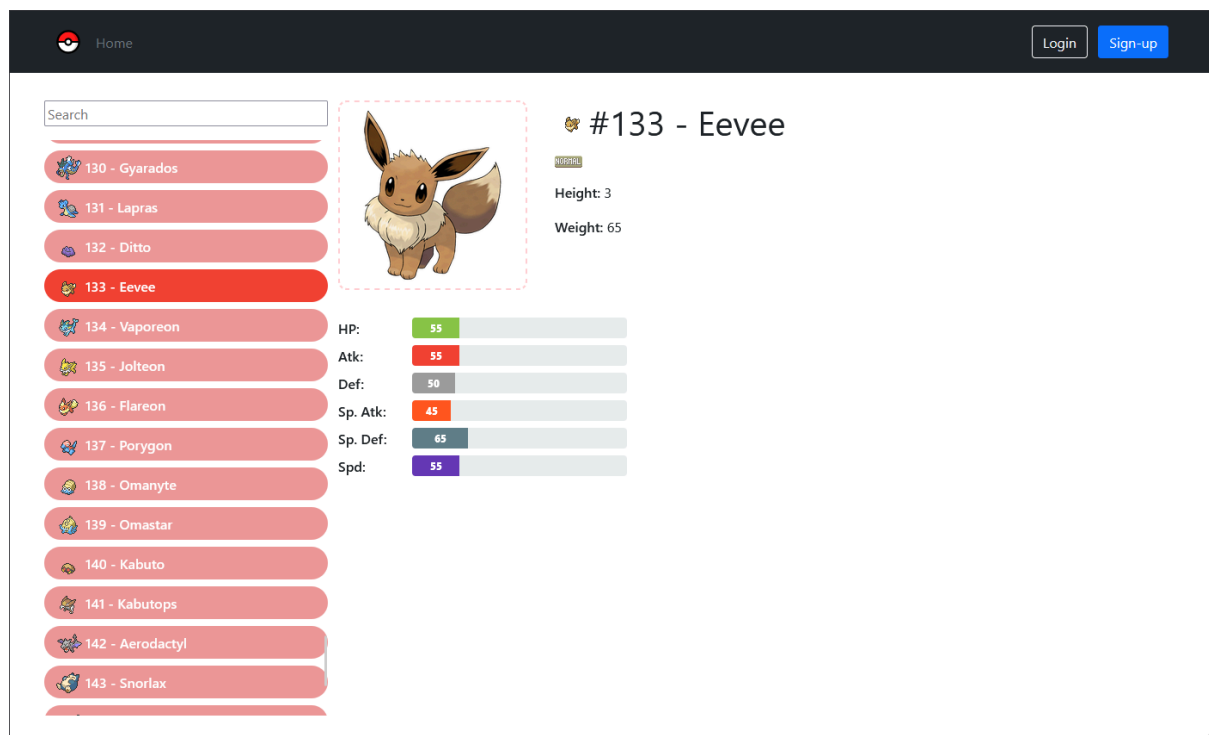
Vous partez comme d'habitude d'une base en HTML/CSS et vous devez construire à partir de celle-ci les fonctionnalités décrites dans ce document. Vous n'avez théoriquement pas besoin de toucher au HTML/CSS (mais ce n'est pas interdit).

EXERCICE 1 - POKÉDEX

Le but de ce premier exercice est de réaliser une application Pokédex basée sur une API fournie gratuitement par <https://pokeapi.co/>. Pour ceux qui ne seraient éventuellement pas familier avec le concept, il s'agit essentiellement d'une encyclopédie de créatures fictives. Pour faire simple, nous allons gérer uniquement l'affichage des 151 premiers Pokémon.

Ce sujet ne va pas rentrer dans trop de détails en ce qui concerne ces créatures et devrait être abordable même si vous n'y connaissez rien. Mais en cas de soucis, n'hésitez vraiment pas à poser vos questions.

Vous aurez à réaliser une page de vue générale présentant votre Pokédex, ainsi qu'un petit formulaire d'inscription. Voici à quoi ressemble l'application :



I/ LA SOURCE DE DONNÉES

A) L'API À UTILISER

Comme spécifié dans l'en-tête. Vous utiliserez l'API fournie par pokeapi. Vous pouvez accéder à la documentation de cette API ici : <https://pokeapi.co/docs/v2>. Elle est détaillée et permet d'accéder à beaucoup d'informations au besoin.

Regardez le détail si vous voulez. Mais essentiellement, vous aurez uniquement besoin d'utiliser les deux routes suivantes :

<https://pokeapi.co/api/v2/pokemon?limit=151&offset=0>

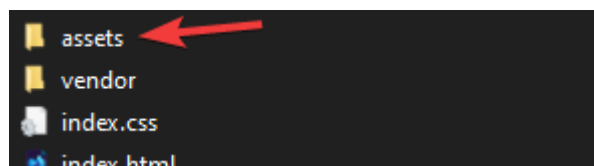
Cette première route va retourner les 151 premiers Pokémon présents dans la DB de l'API. Elle servira à peupler votre liste à gauche sur l'écran principal.

[https://pokeapi.co/api/v2/pokemon/\\${idPokemon}/](https://pokeapi.co/api/v2/pokemon/${idPokemon}/)

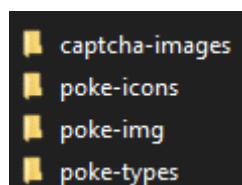
Cette deuxième route permet de récupérer les informations sur un Pokémon en particulier. Il faut juste remplacer dans l'URL son identifiant.

B) LES FICHIERS FOURNIS

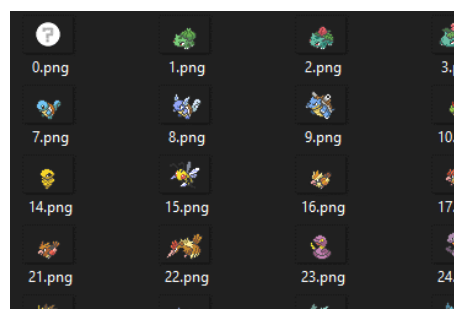
Vous aurez dans les fichiers initiaux du répo un certain nombre de fichiers à disposition. Pas de fichiers JS cette fois-ci, juste des fichiers images que vous devrez utiliser dans votre projet :



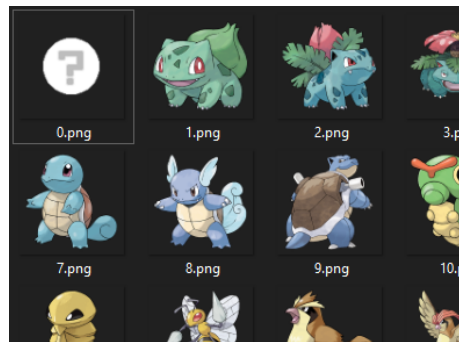
Tout se trouve dans le dossier « assets ». A l'intérieur, vous trouverez les dossiers suivants :



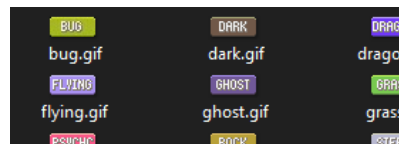
- Le dossier « captcha-images » est utilisé dans le fichier « signup.html » et contient les images à utiliser dans l'élaboration d'un captcha pour votre formulaire d'inscription
- Le dossier « poke-icons » contient les images des Pokémon en version « icône ». Le nom de fichier correspond à l'identifiant :



- Le dossier « poke-img » contient aussi des images de Pokémon mais en version « grand format ». Comme pour les icônes, le nom de fichier correspond à l'identifiant :



- Enfin, le dossier « poke-types » contient des images par rapport aux types de Pokémon. Leur nom de fichier match ce que vous avez en retour dans l'API :

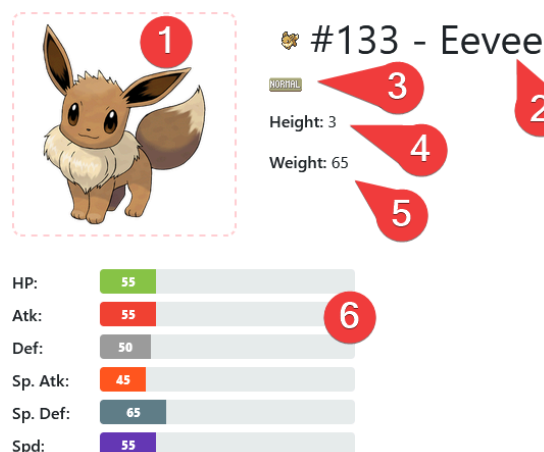


Avec tous ces fichiers, vous devriez avoir ce qui est nécessaire niveau images pour réaliser ce TD

II/ TÂCHES À RÉALISER

A) LA PAGE "POKÉDEX"

- Au chargement de la page, vous devez récupérer la liste des 151 Pokémon via l'API et les charger dans votre liste latérale
- Au clic sur un des éléments de la liste, vous devez charger les informations sur le Pokémon que vous avez cliqué et les afficher dans le volet de droite. Toutes les informations sont disponibles dans le retour de l'API. Prenez soin d'update les endroits indiqués ci-dessous :



- Au chargement de la liste des Pokémon, sélectionnez par défaut le premier Pokémon de la liste
- POUR ALLER PLUS LOIN :** Votre application pourra accepter un paramètre GET (dans l'URL) pour inclure une ID de Pokémon dedans. C'est cette ID qui sera chargée au lancement de l'application. Pensez aussi à changer l'URL à la sélection d'un Pokémon. Le but est de pouvoir partager un lien vers un Pokémon en particulier. Par exemple, ouvrir `http://127.0.0.1/index.html?pokeId=28` devrait ouvrir l'application avec le Pokémon numéro 28 chargé initialement.
- POUR ALLER PLUS LOIN :** Il y a un input texte au-dessus de la liste des Pokémon. A la saisie, vous devez uniquement afficher les Pokémon dans la liste qui correspondent à la recherche.

- *POUR ALLER PLUS LOIN : Rajoutez des indicateurs de chargement lorsque vous attendez la réponse de l'API dans votre application. Rajoutez aussi un message d'erreur sous la forme de votre choix si jamais une erreur survient au moment de récupérer les données de l'API.*

B) LA MODALE DE LOGIN

Dans la page HTML, il y a un bout de code correspondant à une modale de login, cachée par défaut via son attribut style :

```
<main>
<div
  class="custom-modal"
  id="login-modal"
  tabindex="-1"
  style="display: none"
>
  <div class="modal-dialog">
    <div class="modal-content">
      <form>
        <div class="modal-header">
          <h5 class="modal-title">Login</h5>
          <button type="button" class="btn-close"></button>
        </div>
        <div class="modal-body">
          <div class="mb-3">
            <label for="email" class="form-label">
              Email address
            <input type="text" class="form-control" id="email" />
          </div>
          <div class="mb-3">
            <label for="password" class="form-label">
              Password
            <input type="password" class="form-control" id="password" />
          </div>
        </div>
        <div class="modal-footer">
          <button type="button" class="btn btn-secondary">Close</button>
          <button type="button" class="btn btn-primary">Login</button>
        </div>
      </form>
    </div>
  </div>
</div>
<script src="index.js"></script>
```

Lorsque cette modale est affichée dans la page, vous devrez voir ceci :

- Lorsque l'utilisateur clique sur le bouton « Login » en haut à droite de la page principale, affichez cette modale.
- La modale doit se fermer lorsque l'utilisateur clique sur le bouton « Close », la croix en haut à droite de la modale, ou bien sur le backdrop derrière la modale.
- Vous n'avez **PAS** besoin de gérer ce formulaire. Rien ne doit se passer lorsque l'on clique sur le bouton « Login »
- *POUR ALLER PLUS LOIN : Un appui sur le bouton « Show Password » devrait changer l'affichage du champ « mot de passe » pour afficher les caractères saisis par l'utilisateur à la place des « • » qui s'affichent dans le champ en temps normal.*

C) LA PAGE D'INSCRIPTION

Vous avez également une page d'inscription à faire avec un petit formulaire d'inscription avec quelques vérifications. La page se trouve sur « `signup.html` » :

Sign-up form

Email address

Password

Password strength:


weak

Password Confirm

☐ You agree to our [terms and conditions](#)

Captcha

Please select all the kitten pictures below:



Submit

Voici les tâches à faire sur cette page :

- Vérifiez si l'e-mail saisi est une adresse mail valide
- Le mot de passe doit faire au moins 8 caractères
- La confirmation de mot de passe doit correspondre au mot de passe
- La case termes et conditions doit être cochée
- Chaque champ erroné doit être indiqué clairement à côté du champ (il existe une balise HTML préexistante dans la page pour contenir les messages d'erreur)
- Empêchez l'envoi du formulaire si un champ n'est pas bon
- **POUR ALLER PLUS LOIN : Implémentez le captcha dans votre formulaire:**
 - Bloquez l'envoi du formulaire si le Captcha n'est pas rempli correctement.
 - Faites en sorte que toutes les images du captcha soient mélangées aléatoirement à chaque chargement de page
- **POUR ALLER PLUS LOIN : Implémentez un fonctionnement pour la barre « password strength » juste en dessous du champ mot de passe. Suggestion d'implémentation : Une échelle allant jusqu'à 3 points, où il est possible de « gagner » des points lorsqu'on utilise des variétés de caractères différents. Par exemple, au moins une lettre minuscule donne 1 point, au moins une lettre majuscule donne 1 point, au moins 1 chiffre donne 1 point, et au moins 1 caractère spécial donne 1 point. Pensez à adapter le style avec la couleur de la barre, et le texte à afficher.**

EXERCICE 2 - PANIER E-COMMERCE

Le but de cet exercice va être de créer les fonctionnalités de base d'un panier (shopping cart).

IUT Cart

Product Name	Unit Price	Quantity	Subtotal	Action
IUT Rubber Duck	\$25.00	<input type="text" value="0"/>	\$0	<button>Remove</button>
<input type="text" value="Product Name"/>	<input type="text" value="0"/>			Create Product

Calculate Prices

Total: \$0

Vous ferez la majeure partie de votre travail dans le fichier `js/index.js`. Nous avons ajouté le balisage initial dans `index.html` et un peu de style de base. Pendant le développement, assurez-vous d'utiliser les mêmes noms de classe que ceux déjà utilisés (et disponibles dans le fichier CSS) pour rendre notre panier d'achat agréable et propre.

I/ UPDATESUBTOTAL

Ouvrez le fichier `index.html` dans votre navigateur. Comme vous pouvez le voir, il n'y a qu'une seule ligne dans le tableau qui représente un produit. Dans cette première itération, vous vous concentrerez uniquement sur ce produit, et plus tard, vous implémenterez les fonctionnalités restantes pour plusieurs produits.

Le seul produit que nous avons actuellement dans notre `#cart` est placé dans le `tr` avec la classe `produit` (qui se trouve à l'intérieur du `tbody`) :

```
<tr class="product">
  <td class="name">
    <span>Ironhack Rubber
    Duck</span>
  </td>
  <td
    class="price">$<span>25.00</span>
  </td>
  <td class="quantity">
    <input type="number" value="0"
    min="0" placeholder="Quantity" />
  </td>
  <td
    class="subtotal">$<span>0</span>
  </td>
  <td class="action">
    <button class="btn btn-
    remove">Remove</button>
  </td>
</tr>
```

Le produit a un prix et une quantité (où la quantité représente combien d'articles d'un produit spécifique un utilisateur a ajouté dans le panier). Dans le code fourni, nous voyons qu'il y a aussi un prix subtotal. Le prix subtotal sera le résultat de la multiplication de ces valeurs.

Votre objectif est de calculer le prix subtotal et de l'afficher au bon endroit lorsque le champ "Quantity" est modifié.

II/ CALCULATE ALL

Notre code actuel est optimisé pour ne gérer qu'un seul produit, mais il est prévu d'accueillir plusieurs produits dans notre panier. Par conséquent, votre tâche consiste à découvrir comment calculer la somme des "subtotals" dès que l'on appuie sur le bouton "calculate prices" et à afficher cette somme à l'emplacement actuel de "total: \$0".

Product Name	Unit Price	Quantity	Subtotal	Action
IUT Rubber Duck	\$25.00	<input type="text" value="2"/>	\$50	<button>Remove</button>
Item 2	\$7	<input type="text" value="3"/>	\$21	<button>Remove</button>
<input type="text" value="Item 2"/>	<input type="text" value="7"/>			Create Product

Calculate Prices

Total: \$71

III/ AJOUTER UN PRODUIT

Les deux champs de saisie situés dans le tfoot représentent respectivement le nom du nouveau produit et son prix unitaire. Le bouton "Créer le produit" devrait ajouter un nouveau produit au panier lorsqu'il est déclenché.

Ajoutez un événement de clic au bouton "Créer le produit" qui devra cibler les input de nom et de prix unitaire, extraire leurs valeurs, ajouter une nouvelle ligne au tableau avec le nom du produit et le prix unitaire, ainsi que l'entrée de quantité et le bouton "Supprimer", et vous assurer que toute la fonctionnalité fonctionne comme prévu.

Rappelez-vous, le nouveau produit devrait paraître indifférencié et se comporter comme n'importe lequel des produits précédemment inclus dans le panier. Ainsi, on devrait pouvoir calculer son subtotal lorsque le bouton "Calculer tout" est cliqué, et retirer le produit (exercice suivant).

Lorsque la création du produit est finalisée, veuillez, nettoyer les champs de saisie dans le formulaire de création.

<input type="text" value="Item 2"/>	<input type="text" value="7"/>			Create Product
-------------------------------------	--------------------------------	--	--	----------------

IV/ SUPPRIMER UN ITEM

Les utilisateurs doivent avoir la possibilité de supprimer des produits de leur panier. Pour cela, un bouton "Supprimer" est présent à l'extrémité de chaque ligne de produit dans notre tableau. Implementer cette fonctionnalité manquante.

RENDU

Veillez rendre votre projet en fin de TD sur le Google Forms suivant au format .zip : <https://forms.gle/SkrXkY7HQKVs2Ge28>.