

# La pokéAPI

## 1. Création d'une Page Web Simple

**Question :** Comment structurer une page HTML pour afficher une liste de Pokémon et un formulaire de recherche ?

**Suggestions :**

- Utilisez `<div>` ou `<section>` pour créer deux sections principales : une pour la liste de Pokémon et une autre pour le formulaire de recherche.
- Pour le formulaire, envisagez un champ de saisie `<input>` pour le numéro du Pokémon et un bouton `<button>` pour soumettre la recherche.
- Prévoyez un espace pour afficher les résultats de la recherche, comme une autre section.

## 2. Affichage de Plusieurs Pokémon Issus d'une Liste

**Question :** Comment récupérer et afficher une liste initiale de Pokémon avec l'API PokeAPI ?

**Suggestions :**

- Utilisez l'endpoint `/pokemon` pour obtenir une liste de Pokémon. Par exemple, <https://pokeapi.co/api/v2/pokemon?limit=10>.
- Utilisez `fetch` pour faire une requête à l'API. Traitez la réponse avec `.json()` pour la convertir en format JSON.
- Parcourez le JSON pour extraire les informations nécessaires (nom du Pokémon, peut-être l'image) et affichez-les dans votre HTML.

## 3. Présentation de l'API PokeAPI

**Question :** Comment explorer l'API PokeAPI pour trouver les informations nécessaires ?

**Suggestions :**

- Visitez la documentation de l'API à <https://pokeapi.co/docs/v2> pour comprendre la structure de l'API et les endpoints disponibles.
- Identifiez les endpoints qui fournissent les informations dont vous avez besoin pour votre application, comme les détails d'un Pokémon spécifique.

## 4. Recherche de Pokémon Grâce à un Formulaire

**Question :** Comment implémenter une fonctionnalité de recherche pour trouver des Pokémon par leur numéro ?

**Suggestions :**

- Ajoutez un écouteur d'événements `submit` à votre formulaire pour intercepter la soumission et effectuer la recherche.
- Utilisez l'endpoint `/pokemon/{id or name}/` pour obtenir les détails d'un Pokémon spécifique. Par exemple, <https://pokeapi.co/api/v2/pokemon/1> pour Bulbasaur.

- Assurez-vous de lire la valeur du champ de saisie pour utiliser le numéro du Pokémon comme paramètre dans votre requête à l'API.

## 5. Gestion d'Erreur

**Question :** Comment gérer les erreurs, comme une recherche qui ne retourne aucun résultat ?

**Suggestions :**

- Utilisez un bloc `try/catch` autour de votre appel `fetch` pour gérer les erreurs de réseau.
- Vérifiez le statut de la réponse. Si `response.ok` est `false`, lancez une erreur ou affichez un message d'erreur à l'utilisateur.

## 6. Manipulation d'Objet et de Tableau en JS

**Question :** Comment manipuler les objets et tableaux retournés par l'API pour afficher les informations des Pokémon ?

**Suggestions :**

- Utilisez des méthodes comme `.map()`, `.filter()`, et `.forEach()` pour traiter les tableaux de données.
- Accédez aux propriétés des objets pour extraire les informations nécessaires, comme le nom du Pokémon, ses stats, ou son image.

## 7. Utilisation de Fetch et Résolution de Promesses

**Question :** Comment utiliser `fetch` pour faire des requêtes asynchrones et traiter les données retournées ?

**Suggestions :**

- Rappelez-vous que `fetch` retourne une promesse. Utilisez `.then()` pour traiter la réponse une fois que la promesse est résolue.
- Pour un code plus propre, utilisez `async/await` pour attendre la réponse de l'API avant de continuer l'exécution du code. (Optionnel)

## 8. Usage du Format JSON

**Question :** Comment travailler avec le format JSON pour extraire les données retournées par l'API ?

**Suggestions :**

- Après avoir reçu la réponse de `fetch`, utilisez la méthode `.json()` pour convertir la réponse en un objet JavaScript.
- Explorez la structure de l'objet JSON dans la console pour comprendre comment accéder aux différentes valeurs dont vous avez besoin.

## Évaluation

Ce TP sera noté sur les critères suivants :

- Fonctionnalité : La page web fonctionne comme demandé, sans erreurs.

- Clarté du code : Le code est bien organisé, commenté, et facile à lire.
- Utilisation correcte des technologies : HTML, CSS, JavaScript, et Fetch sont utilisés de manière appropriée pour répondre aux besoins du TP.
- Créativité et design : L'interface utilisateur est agréable et intuitive.

**L'objectif est de créer une page web qui affiche une liste de Pokémon et permet de rechercher des Pokémon spécifiques par leur numéro. Les détails de chaque Pokémon (nom, image, stats, etc.) doivent être affichés de manière claire et lisible.**