

TP: Application Agence de Voyage

Objectif

Dans ce TP, vous allez apprendre à manipuler le DOM en JavaScript pour afficher dynamiquement des destinations sous forme de cartes, gérer les erreurs de saisie et implémenter un système de recherche.

Guide pour la Réalisation de l'Application Agence de Voyage

Vous allez créer une application web permettant d'ajouter des destinations de voyage et de les rechercher. L'interface se base sur le fichier `voyage.html` fourni, qui comporte un formulaire pour ajouter une destination et une section pour afficher les destinations existantes.

1. Comprendre le Fichier HTML

Le formulaire d'ajout de destination :

Vous trouverez plusieurs champs :

- **Pays** : `<input>` avec `<p id="pays-error">` pour afficher un message d'erreur si nécessaire.
- **Ville** : `<input>` avec `<p id="ville-error">`.
- **Prix (€)** : `<input type="number">` avec `<p id="prix-error">`.
- **URL de l'image** : `<input>` avec `<p id="image-error">`.

La section de recherche :

Un champ `#filtre` permet de rechercher les destinations par pays.

La zone d'affichage des destinations :

Une `<div id="destinations">` où seront ajoutées les cartes de destination créées.

2. Ajouter une Destination

a. Intercepter le formulaire

- Écoutez l'événement `submit` sur le formulaire identifié par `id="formulaire"`.
- Utilisez `e.preventDefault()` pour éviter que la page ne se recharge lors de la soumission.

b. Récupérer et Valider les Données

Pour chaque champ, récupérez la valeur saisie et effectuez une validation simple :

- **Pays & Ville** : Vérifiez que le champ n'est pas vide, sinon affichez un message dans `<p id="pays-error">` ou `<p id="ville-error">`.
- **Prix** : Vérifiez que le prix est un nombre positif. Si le prix est négatif, affichez un message dans `<p id="prix-error">`.
- **Image** : Vérifiez que l'URL de l'image est renseignée. Si vide, affichez un message dans `<p id="image-error">`.

Assurez-vous de vider le message d'erreur si l'utilisateur corrige sa saisie.

c. Créer et Afficher la Carte

Si toutes les validations sont correctes :

- Créez dynamiquement une carte (par exemple, un `<div>` avec des classes Bootstrap) contenant :
 - **Le pays et la ville** (`<h5 class="card-title">`).
 - **Le prix** (`<p class="card-text">`).
 - **L'image** (``).
 - **Un bouton de suppression** (`<button class="btn btn-danger">`).
- **Ajoutez la carte dans `<div id="destinations">`.**

3. Recherche de Destination

Ajoutez un champ de recherche permettant de filtrer les destinations affichées.

- L'utilisateur tape un **nom de pays**, seules les cartes correspondantes doivent rester visibles.
- Utilisez `toLowerCase()` pour rendre la recherche insensible à la casse.
- Masquez les cartes qui ne correspondent pas et affichez celles qui conviennent.

4. (Bonus) Supprimer une Destination

- Pour chaque carte créée, ajoutez un bouton « **Supprimer** ».
- Lorsque l'utilisateur clique sur ce bouton, retirez la carte correspondante du DOM.

5. (Bonus) Tri par Prix

- Implémentez une fonctionnalité pour trier les destinations selon leur prix (du moins cher au plus cher, ou l'inverse).
- Vous pouvez ajouter un bouton ou un menu déroulant pour activer cette fonctionnalité.

Points Importants

- Utilisez `document.createElement()` pour créer chaque élément.
- Utilisez `appendChild()` pour ajouter les éléments dans la bonne structure.
- Pensez à écouter l'événement `click` sur le bouton pour supprimer la carte du DOM.
- Ajoutez un `eventListener` sur le champ de recherche pour filtrer dynamiquement les cartes.

Conseils

- **N'oubliez pas `trim()` !** Il permet de supprimer les espaces inutiles lorsque vous récupérez les valeurs du formulaire.
- **Testez par étapes.** Commencez par créer une seule carte avant de généraliser.
- **Vérifiez la console (`console.log()`)** pour déboguer vos erreurs.
- **Commentez votre code** pour expliquer chaque étape.

À vous de jouer ! 🚀