

# TP : Application de gestion de tâches avec JavaScript et Bootstrap

## Objectif

Développer une application web de gestion de tâches (To-Do List) qui permet à l'utilisateur d'ajouter, de supprimer et de marquer des tâches comme terminées.

## Compétences à mobiliser

- **DOM** : Manipulation du Document Object Model pour ajouter, modifier et supprimer des éléments HTML.
- **BOM** : Utilisation du Browser Object Model pour interagir avec le navigateur (par exemple, `localStorage`).
- **Objets ES6** : Création et manipulation d'objets à l'aide de la syntaxe ES6.
- **Tableaux** : Gestion et manipulation de listes de données.
- `this` : Compréhension du contexte d'exécution et utilisation correcte du mot-clé `this`.
- **Boucles et conditions** : Utilisation de structures de contrôle pour la logique de l'application.
- **Fonctions** : Écriture de fonctions pour organiser et réutiliser le code.
- **Bootstrap** : Intégration et utilisation de Bootstrap pour le design responsive.
- **Bonnes pratiques** : Utilisation de l'attribut `defer` pour les scripts, validation des entrées utilisateur.

## Instructions

### 1. Initialisation du projet

- Créez un dossier pour votre projet et ajoutez-y les fichiers suivants :
  - `index.html` : la page principale.
  - `style.css` : styles personnalisés si nécessaire.
  - `app.js` : votre code JavaScript.

### 2. Mise en place de Bootstrap

- **Lien CSS** : Dans la section `<head>` de votre `index.html`, ajoutez le lien vers le CSS de Bootstrap.

```
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet"
```

```
integrity="sha384-QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMhY6hW+ALEwIH" crossorigin="anonymous">
```

- **Script JS avec `defer`** : Dans la section `<head>`, ajoutez les scripts JavaScript nécessaires en utilisant l'attribut `defer` pour assurer le chargement correct du DOM avant l'exécution des scripts.

```
<script  
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"  
integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDz0xhy9GkcIdslK1eN7N6jIeHz" crossorigin="anonymous" defer>  
</script>  
  
<script src="app.js" defer></script>
```

### 3. Structure HTML

- Utilisez les composants Bootstrap pour créer une interface utilisateur agréable :
  - Un **en-tête** avec le titre de l'application.
  - Un **formulaire** pour ajouter de nouvelles tâches, avec une validation pour empêcher l'ajout de tâches vides.
  - Une **liste** pour afficher les tâches existantes.

### 4. Fonctionnalités JavaScript

#### a. Validation des entrées

- Avant d'ajouter une tâche, vérifiez que l'entrée utilisateur n'est pas vide ou ne contient pas uniquement des espaces.
- Affichez un message d'erreur sous le champ de saisie si la validation échoue.

#### b. Ajout de tâches

- Capturez l'événement de soumission du formulaire pour ajouter une nouvelle tâche.
- Créez une fonction qui :
  - Empêche le rechargement de la page lors de la soumission du formulaire.
  - Récupère et valide la valeur du champ de texte.
  - Crée un objet tâche avec les propriétés :
    - `id` : un identifiant unique (par exemple, `Date.now()`).
    - `description` : le texte de la tâche.
    - `completed` : statut de la tâche (par défaut `false`).
  - Ajoute l'objet tâche à un tableau de tâches.

- Met à jour l'affichage de la liste des tâches.

### c. Affichage des tâches

- Créez une fonction pour afficher les tâches sur la page en parcourant le tableau de tâches.
- Utilisez le DOM pour créer des éléments HTML dynamiquement.
- Pour chaque tâche, affichez :
  - La description.
  - Un bouton pour marquer comme terminé.
  - Un bouton pour supprimer la tâche.

### d. Marquer comme terminé

- Ajoutez un événement au bouton "Terminé" qui :
  - Modifie la propriété `completed` de la tâche correspondante.
  - Met à jour le style de la tâche (par exemple, texte barré).

### e. Suppression de tâches

- Ajoutez un événement au bouton "Supprimer" qui :
  - Supprime la tâche du tableau.
  - Met à jour l'affichage.

### f. Utilisation de `this`

- Dans vos fonctions d'événements, utilisez `this` pour référencer l'élément qui a déclenché l'événement.
- Assurez-vous de comprendre le contexte dans lequel `this` est utilisé.

## 5. Sauvegarde des tâches (BOM)

- Utilisez le `localStorage` pour sauvegarder le tableau de tâches.
- Chargez les tâches sauvegardées lors du chargement de la page.

## 6. Améliorations supplémentaires (optionnel)

- Implémentez une fonction de filtrage pour afficher toutes les tâches, uniquement les tâches actives ou uniquement les tâches terminées.
- Ajoutez des animations lors de l'ajout ou de la suppression de tâches avec Bootstrap.
- Personnalisez le thème Bootstrap ou ajoutez vos propres styles CSS pour améliorer l'apparence de l'application.

## Annexe : Introduction à Bootstrap

### Qu'est-ce que Bootstrap ?

Bootstrap est un framework front-end open-source qui facilite le développement rapide de sites web responsives et stylés. Il offre une collection de composants CSS et JavaScript prêts à l'emploi.

## Intégration de Bootstrap

### Via CDN

- **CSS** : Ajoutez ce lien dans le `<head>` de votre HTML.

```
<link
  rel="stylesheet"
  href="https://maxcdn.bootstrapcdn.com/bootstrap/5.3.0/css/bootstrap.min.css"
  integrity="sha384-..."
  crossorigin="anonymous"
/>
```

- **JavaScript** : Ajoutez ces scripts dans le `<head>` avec l'attribut `defer`.

```
<script
  src="https://code.jquery.com/jquery-3.6.0.min.js"
  defer
></script>
<script
  src="https://maxcdn.bootstrapcdn.com/bootstrap/5.3.0/js/bootstrap.bundle.min.js"
  integrity="sha384-..."
  crossorigin="anonymous"
  defer
></script>
```

### Téléchargement local

- Vous pouvez également télécharger Bootstrap depuis le site officiel et l'héberger localement.

### Utilisation des composants Bootstrap

#### Grille responsive

- Utilisez le système de grille pour créer des mises en page adaptatives.

```
<div class="container">
  <div class="row">
    <div class="col-md-6">Contenu de la colonne 1</div>
    <div class="col-md-6">Contenu de la colonne 2</div>
  </div>
</div>
```

### Boutons

- Stylisez vos boutons avec des classes prédéfinies.

```
<button class="btn btn-primary">Bouton Principal</button>
```

## Formulaires

- Améliorez l'apparence des formulaires et ajoutez des messages de validation.

```
<form id="taskForm">
  <div class="mb-3">
    <label for="taskInput" class="form-label">Nouvelle Tâche</label>
    <input type="text" class="form-control" id="taskInput" required />
    <div class="invalid-feedback">Veuillez entrer une tâche valide.</div>
  >
</div>
<button type="submit" class="btn btn-success">Ajouter</button>
</form>
```

## Listes et cartes

- Utilisez les composants de listes pour afficher vos tâches.

```
<ul class="list-group" id="taskList">
  <!-- Les tâches seront ajoutées ici dynamiquement -->
</ul>
```

## Ressources supplémentaires

- **Documentation officielle** : [getbootstrap.com](https://getbootstrap.com)
- **Exemples** : Parcourez les exemples fournis pour comprendre comment utiliser les différents composants.

## Conseils pour l'utilisation de Bootstrap

- **Classes utilitaires** : Bootstrap offre de nombreuses classes pour le padding, le margin, le texte, etc.
- **Personnalisation** : Vous pouvez surcharger les styles par défaut en ajoutant vos propres classes CSS.
- **Responsive design** : Testez votre site sur différentes tailles d'écran pour vous assurer qu'il est bien adapté.