

Programmation web sur client

TD3 - Le DOM

Matthieu Nicolas

Préambule

Afin de m'aider à m'organiser dans mon travail, je souhaite réaliser une simple application web me permettant de gérer une liste de tâches à réaliser.

Pour cela, nous allons utiliser JS pour manipuler le DOM, en ajoutant et modifiant dynamiquement des éléments. Dans le cadre de ce TP, nous n'utiliserons pas de framework JS "lourds" tels que Angular, React ou Vue.js. Par contre, vous êtes libres d'utiliser des frameworks "légers" tels que Foundation, Bootstrap ou Semantic UI pour obtenir un résultat plus élégant facilement.

Exercice 1 - TODO

Dans le fichier `/models/todo.js`,

1. Écrire la classe `Todo`. Une instance de `Todo` possède deux propriétés, `title` et `deadline`, qui représentent respectivement l'intitulé de la tâche et la date limite pour l'effectuer. Le constructeur de la classe `Todo` prend ces 2 paramètres pour créer une instance.

Exercice 2 - Pur HTML

Dans le fichier `/index.html`,

1. Afficher, en pur HTML, plusieurs tâches codées en dur. L'intitulé de chaque tâche ainsi que sa date limite doivent être clairement affichés.

Exercice 3 - `todoAsHTML()`

Dans le fichier `/main.js`,

1. Créer plusieurs instances de `Todo` en dur.
2. Écrire la fonction `todoAsHTML()`. Cette fonction prend une instance de `Todo` en paramètre et renvoie un élément HTML qui correspond à la représentation de la tâche en HTML.
3. Utiliser la fonction `todoAsHTML()` pour afficher les tâches instanciées précédemment en ajoutant leur représentation au DOM.

Exercice 4 - Suppression d'une tâche complétée

De temps en temps, j'arrive à compléter une tâche. Je souhaite alors la retirer de la liste, afin de pouvoir me concentrer sur les tâches restantes. Pour que ce ménage soit rapide à effectuer, on souhaite supprimer les tâches simplement en cliquant dessus.

Dans le fichier `/main.js`,

1. Ajouter un `EventListener` à la liste des tâches. Cet `EventListener` est déclenché lorsqu'un clic est effectué sur une tâche. Lors de son déclenchement, l'`EventListener` retire la tâche indiquée de la liste.

Exercice 5 - Ajout d'une tâche

Vider la liste des tâches, c'est bien. Mais il faudrait aussi pouvoir en ajouter des nouvelles.

Dans le fichier `/index.html`,

1. Ajouter un formulaire permettant de saisir une nouvelle tâche.

Dans le fichier `/main.js`,

1. Ajouter un `EventListener` au formulaire. Cet `EventListener` est déclenché lorsque le formulaire est validé. Lors de son déclenche, l'`EventListener` instancie un nouveau `Todo` à partir des informations du formulaire et ajoute cette nouvelle tâche à l'interface