

Javascript - Séparation de responsabilité

La séparation de responsabilité est un principe de base en programmation. Il consiste à séparer les différentes responsabilités d'un programme en plusieurs parties distinctes. Cela permet de rendre le code plus lisible, plus facile à maintenir et plus facile à tester.

Séparation de responsabilité entre HTML, CSS et JavaScript

En JavaScript, la séparation de responsabilité consiste à séparer le code en plusieurs parties distinctes. Par exemple, on peut séparer le code en plusieurs fichiers, un pour le code JavaScript, un pour le code HTML et un pour le code CSS.

- Le code HTML est responsable de la structure de la page web. C'est lui qui définit les différents éléments de la page, comme les titres, les paragraphes, les images, etc. C'est pour cela que l'on veut utiliser les balises Template pour éviter de mélanger le code HTML et le code JavaScript.
- Le code CSS est responsable de la mise en forme de la page web. C'est lui qui définit les couleurs, les tailles, les marges, etc. C'est pour cela que l'on veut utiliser les classes pour éviter de mélanger le code CSS et le code JavaScript.
- Le code JavaScript est responsable de la logique de la page web. C'est lui qui définit les interactions entre les différents éléments de la page, comme les boutons, les formulaires, etc. C'est pour cela que l'on veut utiliser les fonctions pour éviter de mélanger le code JavaScript et le code HTML ou CSS. On se servira de JavaScript pour ajouter ou enlever des classes ou cloner des balises HTML.

Modules

En JavaScript, pour améliorer la séparation de responsabilité, on peut utiliser des modules. Un module est une partie de code qui regroupe des fonctions et des variables qui ont un lien entre elles. Cela permet de mieux organiser le code et de le rendre plus lisible.

Vous aurez généralement un fichier principal qui servira de chef d'orchestre pour importer les différents modules. Vous pouvez aussi avoir des modules qui importent d'autres modules en fonction de leurs besoins.

Classes JS

Les classes en JavaScript permettent de regrouper des fonctions et des variables qui ont un lien entre elles. Cela permet de mieux organiser le code et de le rendre plus lisible. Les classes sont très utiles pour définir des objets qui ont des propriétés et des méthodes communes.

Elles sont un peu comme un schéma pour créer des objets. Vous pouvez créer plusieurs objets à partir d'une même classe, chacun avec ses propres valeurs pour les propriétés.

IIFE - Fonction anonyme auto-exécutée

Une IIFE (Immediately Invoked Function Expression) est une fonction anonyme qui s'exécute immédiatement après sa définition. Cela permet de créer un contexte isolé pour le code et d'éviter les conflits de noms de variables. Il est souvent utilisé pour encapsuler (protéger) le code principal et éviter les fuites de variables globales.

Pour se souvenir de la syntaxe, commencez par écrire deux paires de parenthèses, puis définissez une fonction anonyme à l'intérieur de la première.

```
(function () {  
    // Le code ici n'est pas accessible de l'extérieur  
})();
```

Modulariser le CSS, le HTML et le JavaScript

Au cours de la formation, vous verrez comment modulariser le CSS, le HTML et le JavaScript pour mieux organiser le code et le rendre plus lisible.

Avantages

- Le code est plus lisible et plus facile à maintenir.
- Les différentes parties du code sont plus faciles à tester.
- Les modules sont réutilisables dans différentes parties de l'application.

Inconvénients

- Beauuuuuucoup de fichiers à gérer.
- Il faut bien réfléchir à l'architecture du projet.
- Il faut bien nommer les fichiers et les dossiers et ne jamais les renommer.