Université de Paris diderot - Paris 7

Rapport d'exposé

Programmation comparée

Développement Web

Vanilla.js vs. Frameworks

Réalisé par

— Hicham BOUZARA

— Mahdi LARBI

2021/2022

1 Technologies standards du web

1.1 Html

HTML est un langage de balises utilisé pour structurer et donner du sens au contenu web.

Par exemple : définir des paragraphes, titres et tables de données ou encore intégrer des images ou des vidéos dans une page.

HTML est l'abréviation de HyperText Markup Language. Le terme "langage de balisage" signifie que, plutôt que d'utiliser un langage de programmation pour exécuter des fonctions, le HTML utilise des balises pour identifier les différents types de contenu et les objectifs qu'ils servent à la page web.

1.2 Css

CSS est un langage de règles de style utilisé pour mettre en forme le contenu *HTML*. Par exemple : en modifiant la couleur d'arrière-plan ou les polices, ou en disposant le contenu en plusieurs colonnes.

Le langage *HTML* fournit les outils bruts nécessaires pour structurer le contenu d'un site Web. Le *CSS*, quant à lui, permet de styliser ce contenu afin qu'il apparaisse à l'utilisateur de la manière dont il est censé être vu. Ces langages sont séparés afin de garantir que les sites Web sont construits correctement avant d'être reformatés.

1.3 Javascript

Javascript est un langage de programamtion interprété. Un langage de script qui permet d'implémenter des mécanismes complexes sur une page web pour la rendre dynamique et interactive, il permet de manipuler les éléments HTML, modifier le contenu et le style CSS de la page web.

En utilisant *Javascript* on passe d'un code statique à un code dynamique, c'est à dire la capacité de mettre à jour l'affichage d'une page web pour montrer des choses différentes en des circonstances différentes, en générant un nouveau contenu quand nécessaire.

1.3.1 Rendu côté client vs. rendu côté serveur

Dans le contexte du développement Web il existe deux façons de construire une page Web, la première dans le cas où le code *Javascript* est exécuté sur le navigateur de l'utilisateur **Côté client** et le deuxième cas où le code est exécuté sur un serveur en utilisant des langages web comme *PHP, Python* ou même *Javascript* en utilisant l'environnement *Node.js*, le résultat de l'exécution est téléchargé par l'utilisateur sous formmat *HTML* et affiché sur son navigateur.

1.3.2 API du navigateur

Dans le cas où le code *Javascript* est exécuté par le navigateur, ce dernier nous offre également un d'API, comme *API Document Object Element* qui permet de manipuler les éléments HTML et leurs style CSS, *API de géolocalisation* pour récupérer les informations géographiques, *API audio et vidéo* pour gérer les actions multimédia et *API de stockage*.

2 Développement Web - Vanilla.js

Les technologies standard du web sont elles-mêmes des composants nécessaires et suffisants pour créer le côté client de n'importe quelle plate-forme web.

Cependant, avec la tendance croissante et l'adoption de cadres de développement web côté client est né le terme *VanillaJS*, qui désigne un javascript sans bibliothèque ni cadre.

Le nom *VanillaJS* a commencé comme une plaisanterie pour convaincre les parties prenantes de la décision technique controversée d'utiliser un simple javascript sans bibliothèques supplémentaires. Le mot Vanilla signifie en langage familier "non excitant, normal, conventionnel, ennuyeux".

3 Développement Web - Frameworks

Travailler directement avec le DOM nécessite de comprendre de nombreuses choses sur le fonctionnement du DOM : comment créer des éléments, comment modifier leurs propriétés, comment mettre des éléments les uns dans les autres, comment les faire apparaître sur la page. Aucun de ces codes ne gère réellement les interactions avec les utilisateurs, ni ne traite de l'ajout ou de la suppression d'une tâche. Si nous ajoutons ces fonctionnalités, nous devons nous souvenir de mettre à jour notre interface utilisateur au bon moment et de la bonne manière.

Les frameworks JavaScript ont été créés pour faciliter un peu ce genre de travail - ils existent pour offrir une meilleure expérience aux développeurs. Ils n'apportent pas de nouveaux pouvoirs à JavaScript ; ils vous donnent un accès plus facile aux pouvoirs de JavaScript afin que vous puissiez construire pour le Web d'aujourd'hui.

Les avantages des frameworks sont réalisables en JavaScript classique, mais l'utilisation d'un framework élimine toute la charge cognitive liée à la nécessité de résoudre ces problèmes soi-même.

4 Comparaison entre Vanilla.js et Frameworks

Frameworks	Vanilla JS
Outils	
Chaque framework offre des outils qui permettent de faciliter et automatiser les tâches du développer, notamment l'initialisation du projet, les tests et l'analyse du code.	Utiliser que Vanilla js, cela signifie tout construire à partir de zéro pas seulement le code, en particulier la configuration.
Architecture	
La plupart des frameworks sont basés sur des architectures comme MVC, MVVM,...etc, ce qui permet de créer des composants réutilisables, maintenables et extensibles.	L'application est construite à partir de zéro avec un ensemble de conventions qui permettent de réaliser une application performante et scalable mais qui ne sont pas forcément correctes.
Performance	
Les frameworks permettent d'écrire plus du code déclaratif que du code impératif en moins de lignes de code tout en traitant toute la partie complexe du DOM, mais cette abstraction est coûteuse.	Quand il s'agit d'une simple application, trés vite, le choix se porte sur Vanilla.js . Mais quand la complexité augmente les frameworks offrent plusieurs fonctionnalités pour diminuer cette complexité (en plus de l'architecture utilisée).
Boilerplate	
Nécessite l'importation de plusieurs paquets et dépendances. Oblige le développeur à configurer un environnement et une structure de fichiers spécifiques avant de commencer.	Liberté de créer l'ensemble du projet dans un seul fichier HTML. Pas d'importation, de configuration ou de structure de fichier spécifique.
Lisibilité et structure	
Structure prédéfinie : basée sur des composants, gestion du state, separation of concerns. Intégration rapide des nouveaux membres de l'équipe.	Pas de structure prédéfinie, dépend du choix et de l'expertise du développeur. Courbe d'apprentissage élevée pour les nouveaux développeurs dans un projet existant.
Réutilisabilité	
Utilise un DOM virtuel. Peut définir de nouveaux éléments, les combiner et les réutiliser à différents endroits et projets seulement en les appelant.	Ne peut utiliser que des éléments réels, codés en dur. Réutiliser un modèle existant implique de copier tout son code.
Accessibilité	
Il faut souvent utiliser des API avancées pour accéder aux fonctions natives du navigateur.	Accès direct à toutes les fonctions natives du navigateur.