

ProgWeb 2 – CM6

Objectif

Conclure le module en listant à titre indicatif des notions de JavaScript encore plus avancées, et en présentant une ouverture sur l'écosystème autour de ce langage.

I/ Quelques notions en plus...

A ce stade, vous êtes pas mal équipés pour effectuer des actions dynamiques sur vos pages web via le langage JavaScript. Sachez juste qu'il y a encore beaucoup à apprendre par rapport à ce langage et qu'on peut aller beaucoup plus loin. Le JavaScript, malgré ses quelques particularités qu'il faut connaître et comprendre, est un vrai langage mature et très fortement adopté permettant de faire énormément dans le monde du développement d'aujourd'hui. Voici une liste non exhaustive des choses que nous n'avons pas vraiment abordées dans ce cours :

- L'utilisation de fonctions de localStorage et sessionStorage pour stocker localement des données avec une structure simple et droit au but
- L'utilisation de l'API IndexedDB permettant de travailler avec des concepts de base de données directement dans le navigateur
- Quelques opérateurs avancés :
 - L'optional chaining pour accéder à des propriétés sur des objets potentiellement null ou undefined pour prévenir des erreurs
 - L'utilisation d'opérateurs avancés tel que le « Nullish coalescing operator »
- L'utilisation de « this » dans des objets
- Les décorateurs de fonctions
- Les prototypes – le mécanisme interne au JS qui constitue les objets
- Les classes – permettant de faire des objets de façon plus « moderne et familière » en utilisant les prototypes sous le capot
- Les fonctions génératrices
- Les proxys
- Une façon différente de travailler avec des scripts JS en créant des « modules » (Comme typiquement sur un projet React par exemple)
- L'utilisation de surcouches au JavaScript tel que TypeScript qui permet d'ajouter une notion de « typage » fort à notre code
- Et d'autres fonctions encore...

Il ne faut pas être intimidé face à toutes ces notions. Certaines sont plus utiles que d'autres, mais si vous connaissez bien déjà tout ce que nous avons vu dans les précédents CM de ce module, vous avez déjà une bonne base en JS que vous allez naturellement cultiver dans la suite de vos études et votre parcours professionnel si vous vous dirigez vers un poste où le JS est dominant. Comme je l'ai souvent répété dans ce module, la ressource <https://javascript.info/> est excellente pour creuser toutes les notions de JS qu'on n'a pas vu ensemble ici.

II/ L'écosystème JavaScript en général

Tel que spécifié en début de module, le JavaScript vient de loin, et a transcendé son utilité de « simple langage de script pour rajouter de l'interaction de base sur des pages web ». Même si il continue à se traîner sa legacy, le JavaScript permet aujourd'hui de créer de véritables applications dans le domaine du web en étant assez performant pour la plupart des besoins, une barrière d'entrée relativement basse, mais en étant assez évolué pour faire des choses très complexes lorsque l'on connaît toutes les possibilités associées au langage.

La communauté JS est devenue énorme au point où énormément d'outils se sont rajoutés à l'écosystème pour permettre de concevoir des applications toujours plus complexes.

Le JavaScript via des outils comme NodeJS permettent d'utiliser le JavaScript en dehors d'un contexte navigateur, au point où il est possible de faire une application entière en utilisant du JavaScript en Front et en Back. Il est même possible via des frameworks comme Electron de faire des applications « natives » pour PC basé sur les technologies web.

Enfin, l'écosystème du JavaScript a grandi de façon exponentielle, et très rapidement au cours de la dernière décennie. La prolifération des technologies annexes, de l'intégration plus forte sur toutes les échelles de la stack technique, et la multiplication des frameworks, librairies, et méthodologies sont un témoin évident de cette évolution. Bien que ceci puisse changer du jour au lendemain, il n'y a aucun signe de ralentissement en vue.

En bref, votre exposition à tout cet écosystème sera inévitable si vous partez sur le domaine du front-end, ou n'importe quel métier où le JavaScript est inclus dans la stack technique.

Petite lecture conclusive par rapport à ce paragraphe : <https://css-tricks.com/ooooops-i-guess-were-full-stack-developers-now/>