

Introduction DOM, BOM, CORE et Window

Qu'est-ce qu'une API ?

Une API (application programming interface) est un ensemble d'identifiants qu'un programme offre à d'autres programmes pour interagir avec lui.

Il faut voir une API comme un ensemble de fonctions, objets, méthodes etc qui sont accessibles par d'autres programmes pour pouvoir l'utiliser sans avoir à se préoccuper des détails de l'implémentation.

Une API vient avec une documentation qu'on appelle documentation d'API . Elle liste toutes les fonctionnalités disponibles et comment s'en servir.

Qu'est-ce qu'une Web API ?

Une Web API est une API destinée à des programmes sur des serveurs Web et / ou dans des navigateurs Web .

Les navigateurs ont des dizaines d'API qui se complexifient d'année en année.

Il existe une API Bluetooth, une Network Information API (par exemple pour accéder à la vitesse de connexion de l'utilisateur), une API pour obtenir des informations sur la batterie... des API pour la communication réseau (TCP, Fetch, XMLHttpRequest, Streams ...).

L'objectif d'un développeur n'est pas du tout de connaître toutes les API Web .

Premièrement car ce n'est pas possible, il y en a des nouvelles tout le temps, et elles ne sont pas forcément standardisées (vous pouvez avoir cinq API différentes pour faire environ la même chose sur Chrome, Mozilla, Edge, Opera etc).

Deuxièmement, car cela ne sert à rien : chaque programme a des besoins spécifiques et si vous ne travaillez pas sur une application mobile il n'y a pas d'intérêt à apprendre à utiliser l'API de gestion de la batterie des mobiles.

Troisièmement, car une fois que vous savez lire une documentation d'API cela ne vous posera pas de difficultés d'aller la lire.

En revanche, certaines API très utilisées, répondant à des problématiques universelles doivent être connues.

Par exemple, comment envoyer des requêtes Ajax et comment sauvegarder des informations sur le navigateur sont des problématiques que rencontrent quasiment toutes les applications. Il faut donc les maîtriser ! Nous allons donc voir les API essentielles ensemble au cours de la formation.

Qu'est-ce que l'objet window ?

Window est l'objet global dans un navigateur. Il représente la fenêtre de l'onglet utilisé par votre application.

Il y a donc autant d'objet window que d'onglet ouvert. Mais vous ne pouvez accéder qu'à un seul objet window, celui qui a ouvert votre site Web, pour des raisons de sécurité évidentes.

Window contient de nombreuses propriétés dont le document qui représente le DOM.

Ces propriétés sont extrêmement diverses et permettent par exemple de connaître les dimensions de la fenêtre, ou d'obtenir un objet cryptographique pour générer un nombre aléatoire utilisable en cryptographie.

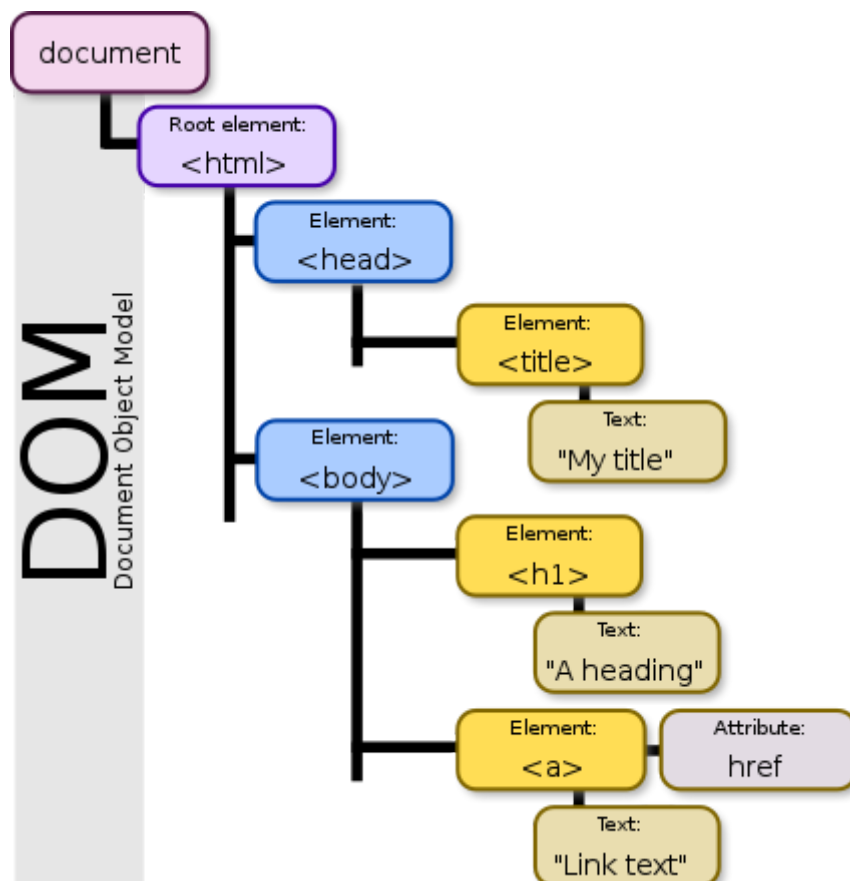
Nous verrons ces propriétés au fur et à mesure de nos besoins, mais comme il y en a des centaines cela n'a aucun intérêt d'en faire une liste.

Qu'est-ce que le DOM ?

Le DOM (Document Object Model) est une API permettant d'interagir avec des documents.

Ces documents peuvent être au format HTML, XML ou SVG mais ce qui nous intéresse est le HTML.

Le DOM est un arbre de noeuds qui représente le document HTML.



Ces noeuds sont des parties de la page Web et cette API permet de les manipuler comme on le souhaite : les ajouter, les supprimer, les modifier, les déplacer etc.

Le DOM est fondamental car c'est ce qui permet de manipuler la page avec du JavaScript notamment. Sans DOM vous ne pourriez avoir que des pages statiques qui ne peuvent pas être modifiées et donc qui ne peuvent pas être interactives.

Le DOM contient plusieurs blocs, ceux relatifs aux documents HTML sont :

Premièrement, les parties **Core et HTML** permettent de former l'objet représentant n'importe quel document HTML . Ils contiennent toutes les interfaces pour définir les `Element` , les `Node` , les `Attribut` etc.

Deuxièmement, la partie **Views** permet de manipuler le document. Elle fournit des fonctions pour accéder, supprimer, déplacer, ajouter des éléments.

Troisièmement, la partie **Event** permet d'ajouter des `scripts` qui sont exécutés lors d'interactions spécifiques de l'utilisateur, qui sont des événements.

Quatrièmement, la partie **Style** permet de modifier dynamiquement le style `CSS` de la page.

Cinquièmement, la partie **Traversal and Range** permet de parcourir dynamiquement le document et de gérer des fragments de document.

Sixièmement, la partie **XPATH** permet d'accéder au document en utilisant le langage `XPATH` .