
Javascript

— BOM: Browser Object Model —

S.Gounane

GI 2019-2020

Introduction

En javascript on trouve deux type d'API

- API intégrées aux navigateurs web. Ces API sont rassemblées dans ce qu'on appelle le BOM (Browser Object Model):
 - l'API **DOM** (Document Object Model): Manipuler le HTML et le CSS d'une page,
 - l'API **Geolocation** : Définir des données de géolocalisation
 - l'API **Canvas**: Dessiner et manipuler des graphiques dans une page ...
- Les API externes, proposées par certains logiciels ou sites:
 - d'API Google Map: Intégrer avec des cartes dans nos pages web
 - l'API Twitter: Afficher une liste de tweets sur un site
 - l'API YouTube qui permet d'intégrer des vidéos sur un site...

BOM

- Le BOM est une super API composée de plusieurs API dont certaines sont composées de plusieurs API et etc.
- A la base du BOM, on trouve l'interface Window qui représente une fenêtre de navigateur contenant un document.
- les objets globaux, variables globales et fonctions globales appartiennent automatiquement à cet objet.
- Window est un objet dit « implicite » : généralement on n'a pas besoin de le mentionner pour utiliser ses méthodes et ses propriétés.

BOM

Les objets suivants appartiennent au BOM et sont tous des enfants de **Window** :

- **Navigator** : représente l'état et l'identité du navigateur ;
- **History** : permet de manipuler l'historique de navigation du navigateur
- **Location**: fournit des informations relatives à l'URL de la page courante ;
- **Screen**: permet d'examiner les propriétés de l'écran qui affiche la fenêtre courante ;
- **Document**: le **DOM** dans son ensemble que nous étudierons en détail dans la suite.

Window: propriétés

L'objet Window possède de nombreuses propriétés:

- ***document, navigator, location*** qui retournent des références aux objets du même nom.
- ***innerHeight, innerWidth***: retournent la hauteur et la largeur de la partie visible de la fenêtre de navigation (la partie dans laquelle le code est rendu).
- ***outerHeight*** et ***outerWidth***: retourner la hauteur et la largeur de la fenêtre du navigateur en comptant les options du navigateur.
-

Window: Méthodes

L'objet Window possède de nombreuses méthodes permettant :

- afficher des boîtes de dialogue
- ouvrir une fenêtre
- Fermer une fenêtre
- redimensionner une fenêtre
- déplacer une fenêtre

Window: Méthodes

- Var f = **open**("url", "id", "width=120, height=200", ...)
- Var rep=**prompt**("message")
- **alert**("message")
- Var rep=**confirm**("message")

Navigator

- L'objet Navigator donne des informations sur le « user agent » (navigateur et préférences enregistrées (langue, etc.))
- Ces informations proviennent de l'utilisateur et ne sont donc totalement fiables.
- Il faut demander une autorisation à l'utilisateur avant de récupérer certaines de ces informations.
- L'objet Navigator est accessible en utilisant la propriété navigator de Window: `var nav=Window.navigator`

Navigator: propriétés

Les propriétés les plus intéressantes sont :

- **language** : la langue définie dans le navigateur ;
- **geolocation** : un objet Geolocation qui peut être utilisé pour définir la localisation de l'utilisateur ;
- **cookieEnabled** : les cookies sont autorisés ou non;
- **platform** : retourne la plateforme utilisée par le navigateur.

Navigator: Méthodes

Les propriétés les plus intéressantes sont :

- **language** : retourne la langue définie dans le navigateur ;
- **geolocation** : retourne un objet Geolocation qui peut être utilisé pour définir la localisation de l'utilisateur ;
- **cookieEnabled** : retourne si les cookies sont autorisés ou non;
- **platform** : retourne la plateforme utilisée par le navigateur.
- **appName** : retourne le nom du navigateur ;
- **appVersion** : retourne la version du navigateur utilisée ;
- **userAgent** : retourne l'en-tête du fichier user-agent envoyé par le navigateur.

Geolocation

Geolocation nous permet de géolocaliser d'un appareil. Cet objet est une propriété de l'objet Navigator. Il met trois méthodes à notre disposition qui ne sont disponibles que dans des contextes sécurisés (HTTPS) pour des raisons de sécurité :

- `getCurrentPosition()`
- `watchPosition()`
- `clearWatch()`

Geolocation

- **getCurrentPosition()** retourne un objet Position qui permet d'obtenir la position actuelle de l'appareil en retournant un objet Position ;
- **watchPosition()** permet de définir une fonction de retour qui sera appelée automatiquement dès que la position de l'appareil change.
- **clearWatch()** est utilisée pour supprimer la fonction de retour passée à watchPosition().

History

- L'objet History permet de manipuler l'historique du navigateur des visiteurs pour la session courante (ex: charger page précédente).
- La propriété history de Window permet d'obtenir une référence à l'objet History.
- L'interface History implémente plusieurs propriétés et méthodes:
 - Length
 - go(n)
 - back() ⇔ go(-1)
 - Forward ⇔ go(1)

Location

- L'interface Location fournit des informations relatives à l'URL d'une page.
- On peut accéder à Location à partir des interfaces **Window** ou **Document**, en utilisant leur propriété location.
- L'objet Location donne accès à plusieurs propriétés et 4 méthodes.
 - Search; pathname; href; hostname; port; protocole ...
 - assign("url")
 - replace("url")
 - reload()

Screen

- L'objet Screen donne accès à des informations concernant l'écran du, comme la taille et la résolution de l'écran.
- Ces informations sont utilisées pour proposer différents affichages à différents visiteurs par exemple.
- On peut récupérer un objet **Screen** en utilisant la propriété screen de **Window**

Var s=window.screen

Screen:propriétés

- ***width*** : retourne la largeur totale de l'écran ;
- ***availWidth*** : retourne la largeur de l'écran moins celle de la barre de tâches ;
- ***height*** : retourne la hauteur totale de l'écran ;
- ***availHeight*** : retourne la hauteur de l'écran moins celle de la barre de tâches ;
- ***colorDepth*** : retourne la profondeur de la palette de couleur de l'écran en bits ;
- ***pixelDepth*** : retourne la résolution de l'écran en bits par pixel.