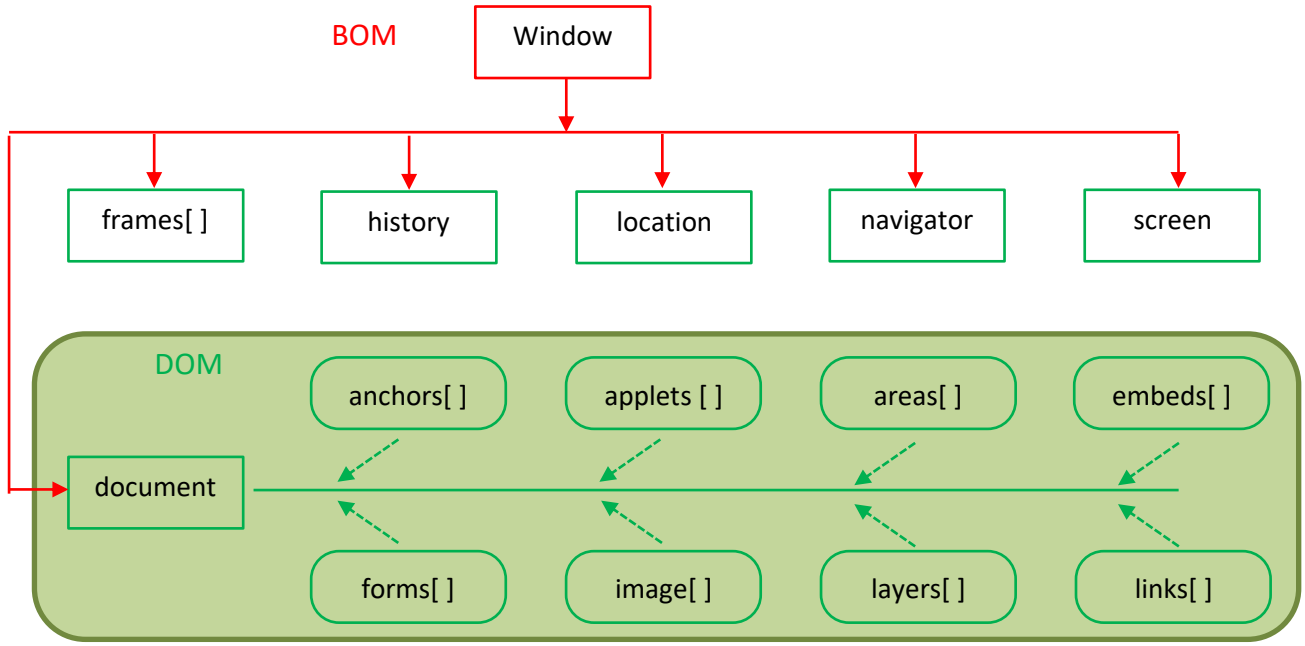
Lycée LOUIS JOUVET Accompagner pour réussir	**Langage JAVASCRIPT** *BOM : Browser Object Model*	***BTS CIEL*** ***Semestre 1*** ***2024_2025***

En JavaScript, tout est objet. L'écran, utilisé pour visionner le site, est un objet. Le navigateur est un objet. Et, le document chargé dans le navigateur est un autre objet. Tous ces objets ont des propriétés et des méthodes. Leurs propriétés permettent d'obtenir des informations sur ces différents objets et leurs méthodes permettent d'interagir avec ces objets.

C'est cette hiérarchie d'objets que l'on appelle le "Browser Object Model" ou BOM. L'objet le plus élevé de la hiérarchie est l'objet window. Tous les autres objets sont intégrés dans cet objet de base, les autres objets sont des propriétés de l'objet window. BOM : Browser Object Model



L'objet frames[ ]

La fenêtre du navigateur peut être divisée en plusieurs parties appelées frames. Chaque partie se comporte comme une fenêtre séparée et peut charger des documents séparés. Le rechargement et la navigation d'une trame n'affectent pas les autres trames. Chaque cadre a un objet windows séparé et chaque objet a une propriété name contenant le nom du cadre. S'il y a plusieurs frames dans une page Web, les objets de plusieurs fenêtres sont indexés dans la collection de frames à la fois par le numéro et le nom du frame. La syntaxe window.frames répertorie les sous-cadres de la fenêtre actuelle en tant qu'objet de type tableau.

Quelques exemples :

```
frameList = window.frames; // renvoie la liste des cadres.
window.frames[0] // accès aux cadres par numéro.
window.frames['iframe_name']  // accéder à l'iframe par son nom.
window.frames[index].location // modifie la page Web dans la position 'index' à l'emplacement attribué.
window.length // donne le nombre d'éléments iframe dans la fenêtre actuelle.
```

L'objet history

Il permet de manipuler l'historique du navigateur pour la session courante et de charger par exemple la page précédente.
Lorsqu'on parle "d'historique" ici on parle de la liste des pages visitées au sein de l'onglet ou fenêtre ou de la frame dans laquelle la page actuelle est ouverte.

Les principales propriétés et méthodes:

- La propriété length qui retourne le nombre d'éléments dans l'historique (en comptant la page actuelle), c'est-à-dire le nombre d'URL parcourues durant la session ;
- La méthode go() qui nous permet de charger une page depuis l'historique de session. On va lui passer un nombre en argument qui représente la place de la page qu'on souhaite atteindre dans l'historique par rapport à la page actuelle (-1 pour la page précédente et 1 pour la page suivante par exemple) ;
- La méthodeback() qui nous permet de charger la page précédente dans l'historique de session par rapport à la page actuelle. Utiliserback() est équivalent à utiliser go(-1) ;
- La méthode forward() qui nous permet de charger la page suivante dans l'historique de session par rapport à la page actuelle. Utiliser forward() est équivalent à utiliser go(1).

Exemple :

```
function hgo(){
    history.go(-2);
}
```

L'objet location

L'interface location fournit des informations relatives à l'URL (c'est-à-dire la localisation) d'une page.
On peut accéder à location à partir des interfaces Window ou Document, en utilisant leur propriété location respective c'est-à-dire Window.location et Document.location.

Les principales propriétés :

- hash, qui retourne la partie ancre d'une URL si l'URL en possède une ;
- search, qui retourne la partie recherche de l'URL si l'URL en possède une ;
- pathname, qui retourne le chemin de l'URL précédé par un / ;
- href, qui retourne l'URL complète ;
- hostname, qui retourne le nom de l'hôte ;
- port, qui retourne le port de l'URL ;
- protocole, qui retourne le protocole de l'URL ;
- host, qui retourne le nom de l'hôte et le port relatif à l'URL ;
- origin, qui retourne le nom de l'hôte, le port et le protocole de l'URL.

3 commandes intéressantes :

- La méthode assign() qui va charger une ressource à partir d'une URL passée en argument ;
- La méthode reload() qui va recharger le document actuel ;
- La méthode replace() qui va remplacer le document actuel par un autre disponible à l'URL fournie en argument.

La différence fondamentale entre les méthodes assign() et replace() est qu'en utilisant assign(), on peut revenir en arrière pour revenir sur notre page de départ car celle-ci a été sauvegardée dans l'historique de navigation ce qui n'est pas le cas si on choisit d'utiliser replace().

Exemple :

```
function assigne(){
    location.assign('http://btssnnewton.free.fr');
```

```
}
```

L'objet navigator

Il va donner des informations sur le navigateur ainsi que sur les préférences enregistrées (langue, etc.). C'est également ce qu'on appelle le "user agent".

Les principales propriétés :

- language : retourne la langue définie dans le navigateur ;
- geolocalisation : retourne un objet geolocalisation qui peut être utilisé pour définir la localisation de l'utilisateur;
- cookieEnabled : cookieEnableddétermine si les cookies sont autorisés ou non et retourne true ou false ;
- platform : retourne la plateforme utilisée par le navigateur

Exemple :

```
window.alert('Langue du navigateur : ' + navigator.language);
```

L'objet screen

L'objet Screen donne accès à des informations concernant l'écran, comme la taille ou la résolution de l'écran par exemple.
On peut utiliser ces informations pour proposer différents affichages à différents utilisateurs.

Les principales 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.

Exemples:

```
alert("Dimensions totales de l'écran : " + screen.width + '*' + screen.height);
```