

JavaScri

Table of Contents

Objectifs	1
Lier un fichier JavaScript au HTML	1
Démonstration	2
Le DOM c'est quoi ?	2
DOM + Javascript = API	2
L'objet de type Window	2
Démonstration	3
L'objet de type Document	3
Les propriétés de l'objet Document	3
Démonstration	3
Les fonctions de l'objet Document pour lire dans le DOM	3
Démonstration	4
Les fonctions de l'objet Document pour Ecrire dans le DOM	4
Démonstration	4
L'objet de type Element	4
Les propriétés de l'objet Element	5
Démonstration	5
Les fonctions de l'objet Element en lecture	5
Démonstration	5
Les fonctions de l'objet Element en écriture	5
Démonstration	6
Démonstration	6
Conclusion	6
Objectif / Niveau	6
A toi de jouer	6

Module 02 - Le DOM

Objectifs

- Savoir utiliser le JavaScript pour rendre une page web interactive.
- Savoir manipuler le DOM.

Lier un fichier JavaScript au HTML

```
...  
  
<script src="module2_2.js"></script>  
</body>  
</html>
```

Démonstration

Lier un fichier JavaScript au HTML

Le DOM c'est quoi ?

Document Object Model

[dom.drawio] | *images/dom.drawio.png*

DOM + Javascript = API

L'API offre tout ce qu'il faut pour manipuler le DOM :

- Des objets
- Des propriétés
- Des méthodes
- Des événements

L'objet de type Window

[dom window.drawio] | *images/dom-window.drawio.png*

NOTE

L'objet de type **console** donne accès à la console de débogage du navigateur. Les spécificités de fonctionnement varient d'un navigateur à l'autre, mais il y a tout de même un ensemble de fonctionnalités qui sont fournies de base.

La propriété en lecture seule **Window.history** renvoie une référence à l'objet History, qui offre la possibilité de manipuler l'historique de session du navigateur (pages visitées dans l'onglet ou le cadre dans lesquels la page courante est chargée).

L'objet de type **Document** représente n'importe quelle page web chargée dans le navigateur et sert de point d'entrée pour accéder au contenu de la page qui est formé par l'arbre du DOM.

Location représente l'emplacement de la page en cours.

Démonstration

L'objet **window** et ses propriétés

L'objet de type Document

L'objet **Document** représente la page web chargée dans le navigateur et sert de point d'entrée pour accéder au contenu de la page qui est formé par l'arbre du DOM.

Les propriétés de l'objet Document

```
* document.documentElement;  
* document.head;  
* document.title;  
* document.body;  
* document.forms;
```

Démonstration

L'objet **document** et ses propriétés

Les fonctions de l'objet Document pour lire dans le DOM

```
* document.getElementById();  
* document.querySelector();  
* document.getElementsByTagName();  
* document.querySelectorAll();
```

La méthode `getElementById()` de `Document` renvoie un objet `Element` représentant l'élément dont la propriété `id` correspond à la chaîne de caractères spécifiée. Étant donné que les ID d'élément doivent être uniques, s'ils sont spécifiés, ils constituent un moyen utile d'accéder rapidement à un élément spécifique.

La méthode `querySelector()` renvoie le premier `Element` dans le document qui correspond au sélecteur spécifié ou au groupe de sélecteurs. Si aucune correspondance n'est trouvée, `null` est retourné.

NOTE

La méthode `Element.getElementsByTagName()` retourne une liste des éléments portant le nom de balise donné. La recherche porte sur le sous-arbre de l'élément spécifié, à l'exception de cet élément lui-même. La liste retournée est live, c'est à dire qu'elle se met à jour automatiquement à chaque changement de l'arbre DOM. Par conséquent, il n'est pas nécessaire d'appeler plusieurs fois `Element.getElementsByTagName()` avec le même élément et les mêmes arguments.

La méthode `querySelectorAll()` de `Element` renvoie une `NodeList` statique représentant une liste des éléments du document qui correspondent au groupe de sélecteurs spécifiés.

Démonstration

L'objet `document` et ses fonctions de lecture

Les fonctions de l'objet `Document` pour écrire dans le DOM

```
* document.write();
* document.createElement();
* document.createAttribute();
* document.createTextNode();
```

Démonstration

L'objet `document` et ses fonctions d'écriture

L'objet de type `Element`

Element est la classe de base la plus générique dont héritent tous les objets qui représentent des éléments d'un **Document**

Les propriétés de l'objet Element

```
* unElement.id;  
* unElement.value;  
* unElement.attributes;  
* unElement.innerText;  
* unElement.innerHTML;  
* unElement.firstChild;  
* unElement.lastElementChild;  
* unElement.nextElementSibling;  
* unElement.previousElementSibling;  
* unElement.parentElement;
```

Démonstration

L'objet **element** et ses propriétés

Les fonctions de l'objet Element en lecture

```
* unElement.getAttribute();  
* unElement.hasAttribute();  
* unElement.hasAttributes();  
* unElement.focus();  
* unElement.getElementsByTagName();  
* unElement.querySelector();
```

Démonstration

L'objet **element** et ses fonctions de lecture

Les fonctions de l'objet Element en écriture

```
* unElement.insertAdjacentElement();  
* unElement.setAttribute();  
* unElement.appendChild();  
* unElement.replaceChild();  
* unElement.removeChild();  
* unElement.removeAttribute();
```

Démonstration

L'objet **element** et ses fonctions d'écriture

Démonstration

FINALE

Conclusion

- Vous savez ce qu'est le DOM
- Vous savez ce qu'est un objet document
- Vous savez ce qu'est un objet element
- Vous savez manipuler le DOM

Objectif / Niveau

- **Essentiel** : Comprendre le fonctionnement du DOM, la fonction getElementById et modifier le contenu d'une balise.
- **Attendu** : Comprendre le fonctionnement du DOM, savoir manipuler le DOM sans difficultés.
- **Avancé** : Comprendre le fonctionnement du DOM, savoir manipuler le DOM sans difficultés et savoir générer l'entiereté du contenu de la balise body en JavaScript.

A toi de jouer

Challenge