



Fonctionnement du navigateur et DOM

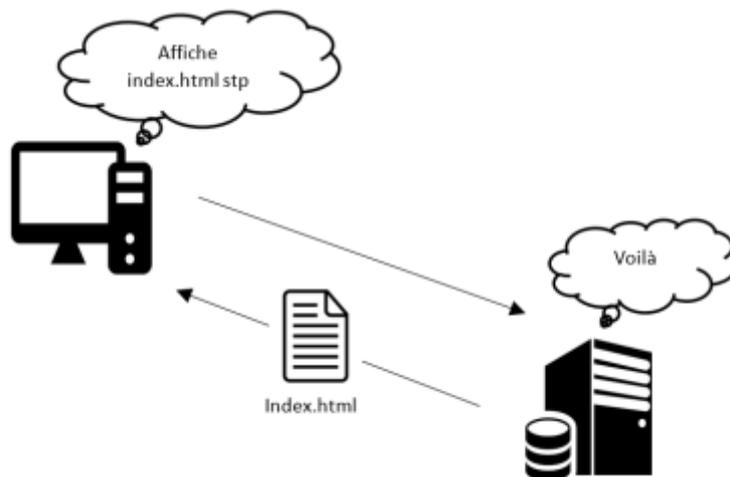


[Mise à jour le 18/11/2020]

- **Sources** et compléments sur **MDN Web Docs**
 - [Référence du DOM](#)
- **Lectures connexes**
 - [JavaScript : les bases](#)
 - [Le DOM et jQuery](#)
- **Mots-clés**
 - Navigateur, HTML, CSS, DOM, noeud, gestionnaire d'évènements.

1. Fonctionnement du navigateur

Le **navigateur** commence par **télécharger** le fichier **HTML** correspondant à l'URL visitée.

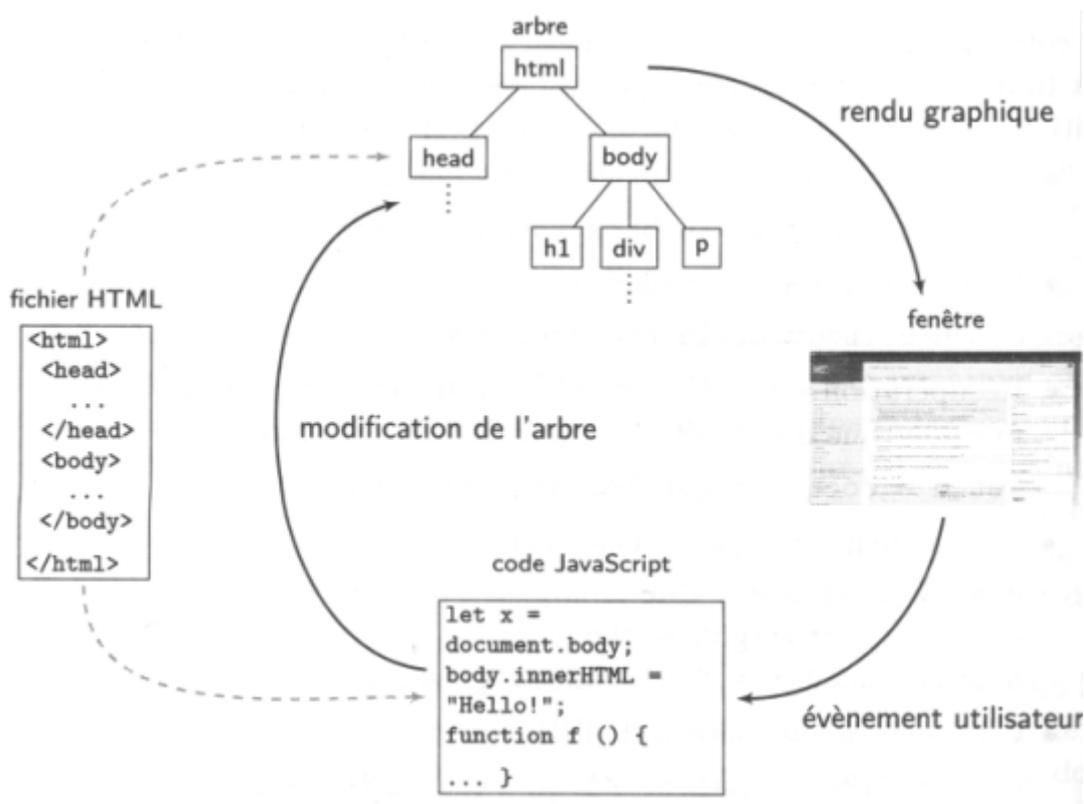


Une fois récupérée, la page est transformée en une **structure de donnée d'arbre**: le **DOM**.

Une fois l'arbre construit, ses **noeuds** (les éléments du document HTML) sont parcourus; les **règles CSS** sont appliquées et le rendu graphique de la page est effectué.

Dans le même temps, les scripts **JavaScript** présents dans la page sont exécutés. L'une de leurs activités principales est la définition de **gestionnaires d'évènements**. Ces gestionnaires peuvent modifier librement l'**arbre** représentant le document.

Exemple



 Le **fichier original**, stocké ou non sur un serveur Web **n'est pas modifié**.

Les attributs des balises peuvent aussi être modifiés. Parmi ces attributs, la modification de l'attribut style permet de changer le rendu graphique de l'élément. Ces changements sont immédiatement répercutés sur l'affichage.

2. Le DOM

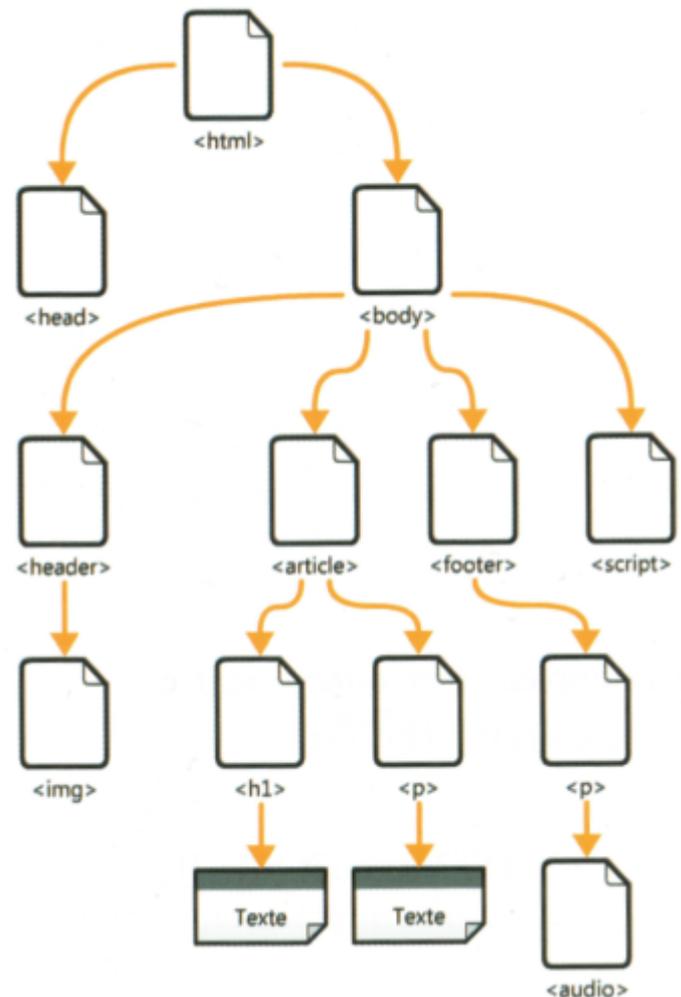
2.1 Présentation

 Le **DOM** (*Document objet Model*) est une **interface de programmation** pour tous les documents et pages web HTML. Il s'agit d'une structure représentant ces **documents** tels qu'ils ont été **interprétés en mémoire par le navigateur**, permettant de parcourir et **modifier leur contenu**, ainsi que leur **présentation visuelle** si des règles CSS sont appliquées.

Au chargement d'un document HTML depuis le réseau ou un fichier local, un "**arbre**" de la page est élaboré via le DOM. Chacun des éléments est considéré comme un **objet** et peut jouer le rôle de

noeud, auquel on peut attacher les qualités de **parent**, **enfant**, **orphelin** ou **frère**. Cette représentation permet d'accéder aux propriétés de chacun de ces objets ou groupes d'objets et aux méthodes associées.

Exemple de code HTML5 et arborescence correspondante



*.html

```
<html>
<head></head>

<body>
  <header>
    <img>
  </header>
  <article>
    <h1>Un titre</h1>
    <p>Un paragraphe</p>
  </article>
  <footer>
    <p>
      <audio></audio>
    </p>
  </footer>
  <script>
```

```
</script>  
</body>  
</html>
```

2.2 Modification du DOM en live !

Pour illustrer la notion d'objet dans le DOM ou a des fins de test, il est possible de modifier le DOM en entrant du code directement dans la **console JavaScript** du navigateur. On commence par créer le fichier HTML ci-dessous.

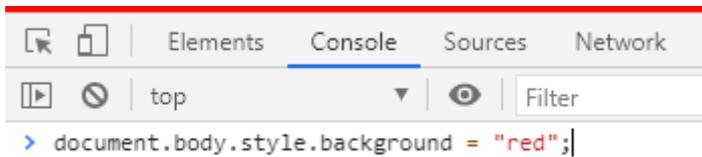
[testdom.html](#)

```
<!DOCTYPE html>  
<html>  
<head>  
  <meta charset="UTF-8">  
  <title>Modification du DOM dans la console du navigateur</title>  
</head>  
<body>  
  <div id="hello">Bonjour !</div>  
</body>  
</html>
```

On affiche la page dans un navigateur et on ouvre la **console JavaScript** avec **F12** ou **Ctrl-Maj-I** dans Chrome.



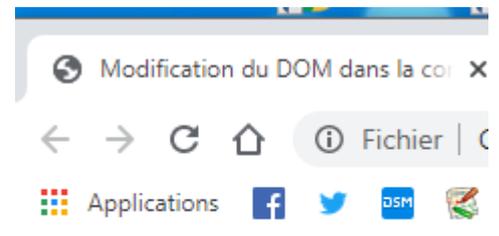
- *Changement de la couleur de fond de la page*
En entrant le code ci-dessous dans la console, suivi de Entrée, on observe le changement de couleur de la fenêtre du navigateur.

A screenshot of the Chrome DevTools Console. The "Console" tab is selected. The command `> document.body.style.background = "red";` has been entered and executed. The console shows the command and its execution. The background of the browser window is now red.

- *Réinitialisation de la couleur par défaut*

*.js

```
document.body.style.background = "";
```



Bonjour !

- *Autres manipulations*

*.js

```
// Récupération d'un élément et modification du style
let d = document.getElementById("hello");
d.style.color="blue";
d.style.fontWeight = "bold";

// Changement du contenu de l'élément d
d.innerHTML="Bonjour tout le monde !"
```

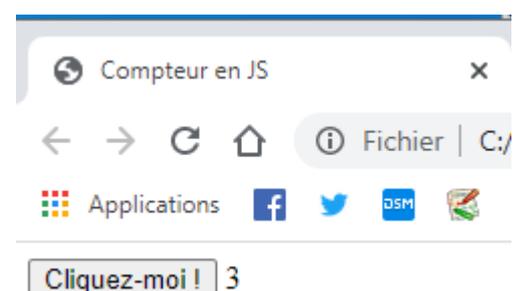
2.3 Le langage JavaScript pour modifier le DOM



Pour modifier les pages Web sans les recharger dans le navigateur, on modifie le DOM avec du code **JavaScript**. Ce code est placé dans des balises HTML `<script>`.

Les navigateurs contiennent un interpréteur JavaScript qui lit les instructions situées entre les balises `<script>` et `</script>` et les traduit en différents types d'actions sur la page Web.

Exemple



*.html

```
<!DOCTYPE html>
```

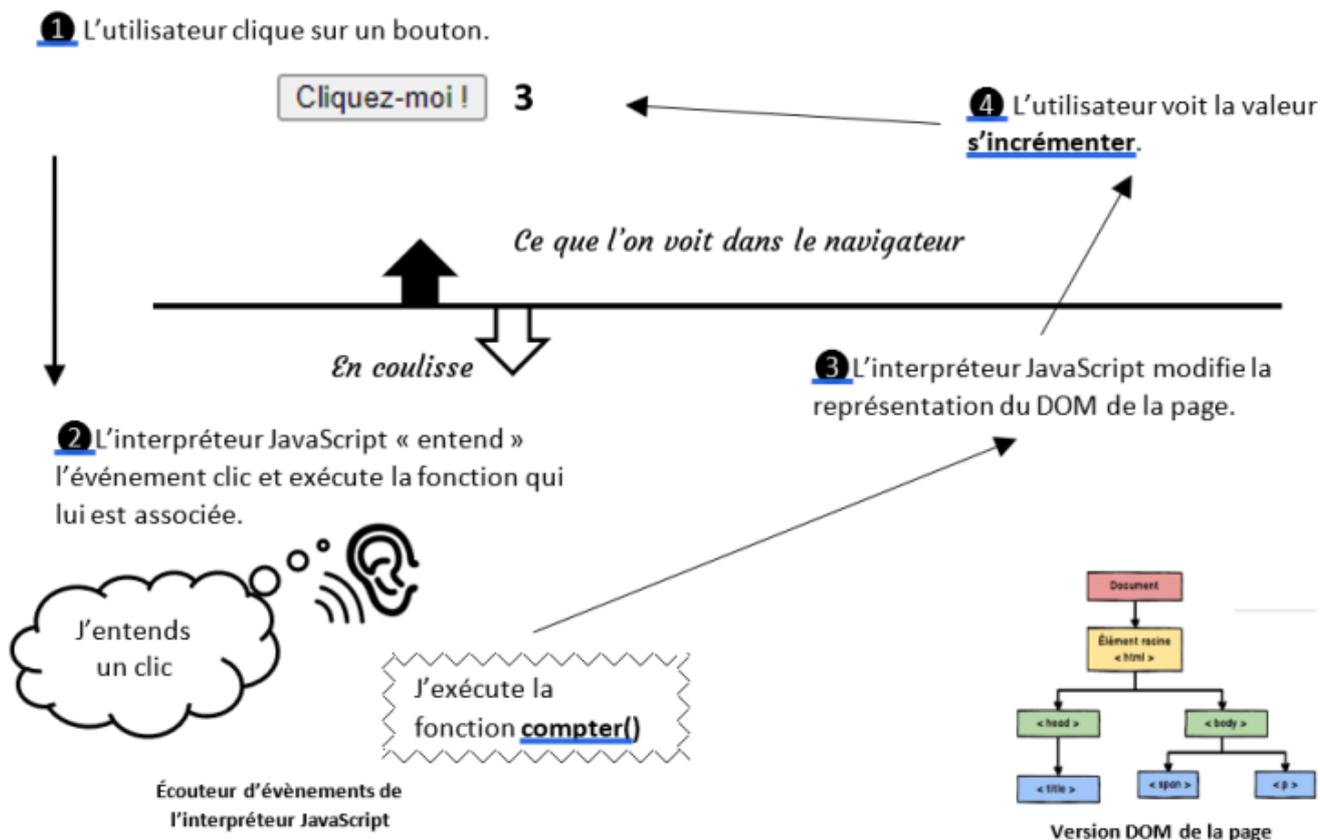
```
<html>

<head>
  <meta charset="UTF-8">
  <title>Compteur en JS</title>
  <script type="text/javascript">
    let compteur = 0;
    function compter() {
      compteur = compteur + 1;
      let v = document.getElementById("valeur");
      v.innerHTML = compteur;
    }
  </script>
</head>

<body>
  <button onclick="compter();">Cliquez moi !</button>
  <span id="valeur">0</span>
</body>

</html>
```

• Fonctionnement



 Le **JavaScript** est un langage de programmation sous forme de **scripts**¹⁾. C'est



également un langage **événementiel**²⁾ et **orienté objet**.

Pour aller plus loin

[JavaScript : JavaScript et JQuery : les bases](#)

A retenir



Les sites Web offrent des interfaces **interactives** grâce à du code écrit en langage **JavaScript**. Ce code est **exécuté dans le navigateur**. Le code JavaScript est intégré au sein d'un fichier HTML avec des balises `<script>`. Le code JavaScript permet d'associer à des éléments HTML des **gestionnaires d'évènements**. Ces derniers sont des fonctions JavaScript qui sont exécutées lorsqu'un évènement choisi (clic de souris, pression sur une touche, etc.) se déclenche pour l'élément concerné. L'attribut `id` permet de **référencer des éléments HTML** dans du code JavaScript.

1)

Sans phase de compilation du code source vers un langage de plus bas niveau

2)

À l'écoute des évènements : clic de souris, etc.

From:

<https://webge.fr/dokuwiki/> - **WEBGE Wikis**

Permanent link:

<https://webge.fr/dokuwiki/doku.php?id=web:dom&rev=1628666367>

Last update: **2021/08/11 09:19**

