

05.JavaScript & DOM

8 octobre 2025

Développement web il3

JavaScript & DOM

HE-Arc (DGR) 2024

JavaScript hier

- Page web = HTML (+ CSS + JavaScript)
- Exécuté par le browser (client)
- Interprété, faiblement typé, OO
- Historiquement
 - Depuis Netscape 2 (1995, Brendan Eich)
 - Petites applications exécutées par le navigateur
 - DHTML : rollovers, validation de formulaires, ...

JavaScript aujourd'hui

- Page web = HTML + CSS + **JavaScript**
- Compilation JIT
- HTML5, AJAX, bookmarklets
- One Page Apps
- Implémentations hors-browser
 - Node.js, Spidermonkey, Rhino
 - script d'app (Qt, Notepad++, ...)
- Langage cible de compilateurs : emscripten ¹, WebAssembly ²

1. <https://emscripten.org/>

2. <http://webassembly.org/>

- Embarqué : Espruino ³, robotique : Node Bots ⁴, CylonJS ⁵
- Applications Desktop : Electron ⁶, sciter ⁷

*Script

- ECMAScript : Norme depuis 1997
 - Juin 2024 : ECMA-262 15th edition ⁸
 - Support ⁹ des différentes implémentations
 - Conversions avec BabelJS ¹⁰
- JavaScript : implémentation Firefox (réf. MDN)
- Variantes (à transpiler) :
 - Typescript ¹¹ : variante fortement typée, avec des classes (MS)
 - Coffescript ¹²
 - sucre syntaxique
 - compilé -> js

JavaScript

- Différentes implémentations ¹³ : navigateur, srv, apps, ...
- Permissif : du mauvais code est peu maintenable
 - Design Patterns ¹⁴
 - Bonnes pratiques ¹⁵
- Interface pour scripter le navigateur
 - Accès et modification du contenu via DOM
 - Bookmarklets ¹⁶, exemples ¹⁷
 - Requêtes HTTP (Fetch API, Xml Http Request)
- Développement d'applications complètes, parfois offline
- Langage de script généraliste (paquets npm)

3. <http://www.espruino.com/>

4. <https://nodebots.io/>

5. <https://cylonjs.com/>

6. <https://electronjs.org/>

7. <https://sciter.com/>

8. <https://www.ecma-international.org/publications-and-standards/standards/ecma-262/>

9. <https://compat-table.github.io/compat-table/es2016plus/>

10. <https://babeljs.io/>

11. <https://www.typescriptlang.org/>

12. <http://coffeescript.org/>

13. https://en.wikipedia.org/wiki/List_of_ECMAScript_engines

14. <https://addyosmani.com/resources/essentialjsdesignpatterns/book/>

15. <http://jstherightway.org/>

16. <http://www.howtogeek.com/125846/the-most-useful-bookmarklets-to-enhance-your-browsing-experience/>

17. <http://www.hongkiat.com/blog/100-useful-bookmarklets-for-better-productivity-ultimate-list/>

Caractéristiques du langage

- Orienté Objet par prototype
- Syntaxe proche de C, Java
- Faiblement typé :
 - Pas de déclaration, type déterminé par la dernière affectation
 - Risque : typo => nouvelle variable. Utiliser const et let
- Types :
 - Primitifs : Boolean Null Undefined Number String Symbol
 - Objets : Object Function
- Particularités
 - Prototypes ¹⁸
 - Fermetures ¹⁹
 - Promesses ²⁰ (MDN ²¹, Google ²²)

Fonctions

- Pas de type de retour
- Possibilité de retourner ou non une valeur
- Sans retour, valeur spéciale : undefined
- Pas de surcharge (la dernière définie prime)
- fonction est un type
- Fonctions imbriquées, anonymes
- Fonctions globales :

`escape()`, `unescape()`, `isFinite()`, `isNaN()`,
`parseFloat()`, `parseInt()`, `Number()`, `String()`,
`eval()`, ...

JavaScript dans la page web

- Éléments `<script>` exécutés dans l'ordre de la page
- Conseillé de les placer en fin de page ²³
- Événements (onclick, onerror, onsubmit, ...)

18. https://developer.mozilla.org/fr/docs/Web/JavaScript/Guide/Le_mod%C3%A8le_objet_JavaScript_en_d%C3%A9tail

19. http://www.w3schools.com/js/js_function_closures.asp

20. <https://www.promisejs.org/>

21. https://developer.mozilla.org/en/docs/Web/JavaScript/Reference/Global_Objects/Promise

22. <https://web.dev/articles/promises?hl=fr>

23. <https://stackoverflow.com/questions/1638670/javascript-at-bottom-top-of-web-page>

- Embarqués dans les balises (onXXX)

```
<div id="intro" onclick="change();" />
```

Utiliser DOM

```
<script type="text/javascript">

    document.getElementById("intro").onclick = change;

</script>
```

- Conseillé d’inclure le code (attribut src)

```
<script type="text/javascript" src="script02.js"></script>
```

language="JavaScript" est déprécié et type vaut par défaut text/javascript.

The type attribute gives the language of the script or format of the data. [...] The default, which is used if the attribute is absent, is “text/javascript”.

HTML5 : script²⁴

Unobstrusive JS²⁵

- Séparation JS...

```
document.addEventListener("DOMContentLoaded", function() {
    document.getElementById('date').addEventListener("change", validateDate);
});
```

- ...et HTML

```
<input type="text" name="date" id="date" />
```

- Dégradation élégante
 - Alternatives pour un browser ne supportant pas JS
- Accessibilité
 - Les fonctionnalités restent accessibles en cas d’erreur
- Utilisabilité
 - Le script doit faire gagner du temps, pas distraire

It is an incredibly popular mistake to use load where DOMContentLoaded would be much more appropriate, so be cautious.

MDN : DOMContentLoaded²⁶

24. <https://www.w3.org/TR/html5/scripting-1.html#the-script-element>

25. https://en.wikipedia.org/wiki/Unobtrusive_JavaScript

26. <https://developer.mozilla.org/en/docs/Web/Events/DOMContentLoaded>

Node.js²⁷ / Deno²⁸

- Node.js : une implémentation hors navigateur
 - environnement d'exécution + bibliothèques
 - event driven, non-blocking IO -> scalable
 - V8 engine
 - scripts exécutables sans navigateur
 - npm²⁹ : gestionnaire de paquets
 - gulp : make js
- Exemples³⁰ d'applications
 - gulp, grunt, bower, yarn
 - browserify
 - serveur http
 - express, cordova, forever, dev, pm2, karma, sails, phantomjs
- Tuto³¹, Playground³²

DOM

- Document Object Model
- Représentation arborescente de la page
- Accessible depuis objet JS document
- Possibilité d'accéder au contenu de la page :
 - Lecture
 - Modification
 - Ajout
- JS peut donc modifier le contenu d'une page

DOM³³

```
<html>
<head>
  <title>My title</title>
</head>
<body>
```

27. <https://nodejs.org>

28. https://www.reddit.com/r/node/comments/nx9qqr/deno_vs_nodejs_a_comparison_you_need_to_know/

29. <https://www.npmjs.com>

30. <https://colorlib.com/wp/npm-packages-node-js/>

31. <https://www.tutorialspoint.com/nodejs/index.htm>

32. <https://runkit.com>

33. <http://bioub.github.io/dom-visualizer/>

```
<h1>A heading</h1>
<a href="#">Link text</a>
</body>
</html>
```

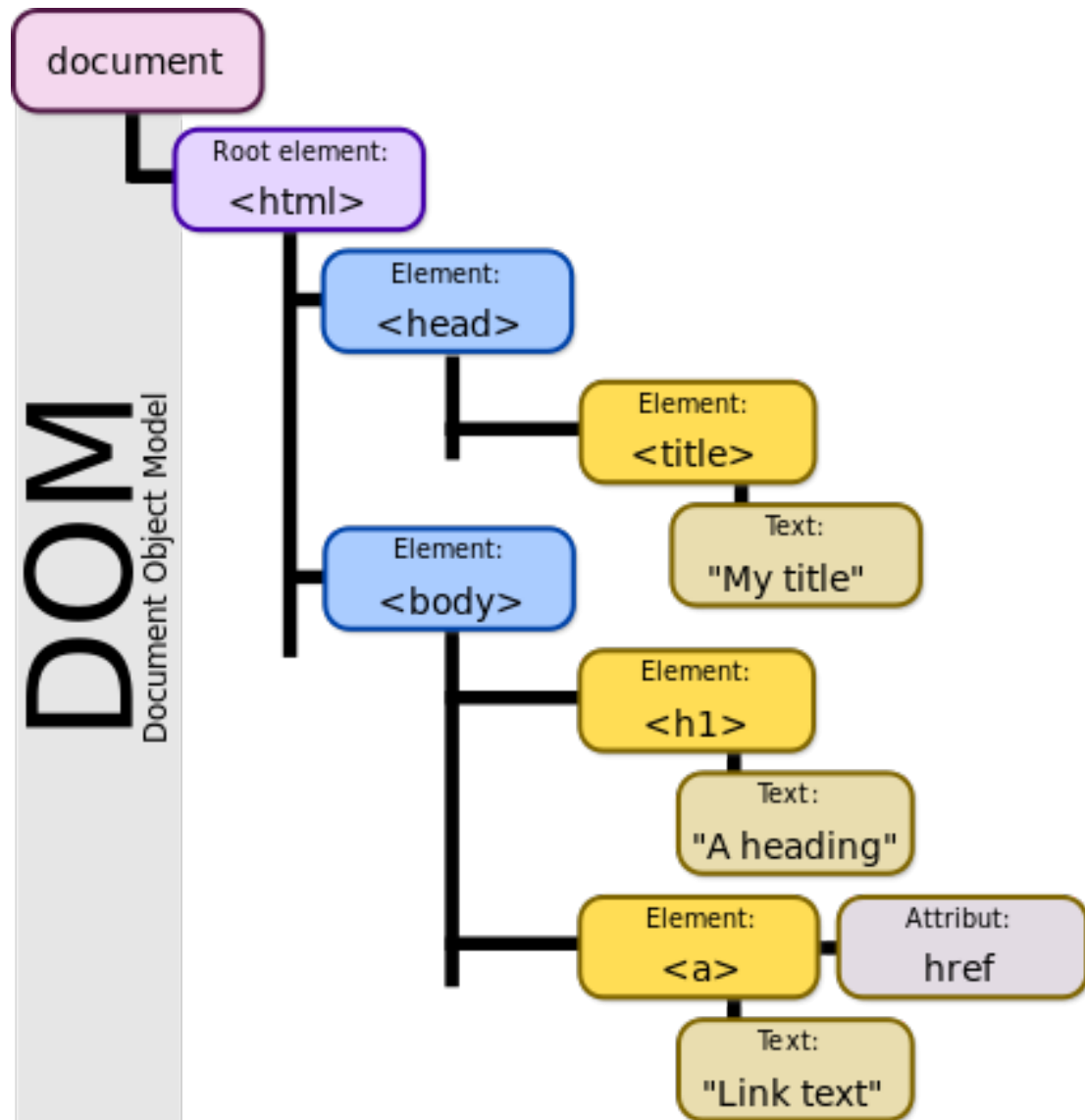


FIGURE 1 – DOM tree

L'objet Document

- Trouver ou modifier des éléments
- Méthodes de Document

```
querySelector(), querySelectorAll(),  
getElementById(), getElementsByTagName(), getElementsByClassName(),  
createElement(), createTextNode()
```

- Méthodes de Node (appel depuis nœud parent)

```
insertBefore(child), appendChild(child),  
removeChild(child), replaceChild(new,old)
```

Ajouter un noeud

```
function addNode() {  
    var inText = document.getElementById("textArea").value;  
    var newText = document.createTextNode(inText);  
  
    var newGraf = document.createElement("p");  
    newGraf.appendChild(newText);  
  
    var docBody = document.getElementsByTagName("body")[0];  
    docBody.appendChild(newGraf);  
}
```

- Création du nouveau nœud :
 - newText contient le texte à ajouter
 - newGraf est un élément p qui contient le texte
- Ajout du nœud comme une feuille de body :
 - Sélection du parent (le premier noeud body)
 - Ajout du nouveau nœud depuis son parent

Supprimer un nœud

```
function delNode() {  
    var allGraf = document.getElementsByTagName("p");  
  
    if (allGraf.length > 1) {  
        var lastGraf = allGraf.item (allGraf.length-1);  
        lastGraf.parentNode.removeChild(lastGraf);  
    }
```

```

    }
    else {
        console.error("Nothing to remove!");
    }
}

```

- Sélection du nœud à supprimer :
 - allGraf contient tous les éléments p
 - lastGraf contient le dernier du tableau allGraf
- Suppression :
 - Suppression du nœud sélectionné depuis son parent³⁴

Insérer un nœud

```

function insertNode() {
    var newText = document.createTextNode("New Text");
    var newGraf = document.createElement("p");
    newGraf.appendChild(newText);

    var divMod = document.getElementsByTagName("div")[0];
    var allGraf = divMod.getElementsByTagName("p");
    var oldGraf = allGraf.item(0);           // position

    divMod.insertBefore(newGraf, oldGraf);
}

```

- Création du nouveau nœud :
 - allGraf contient tous les éléments p
 - lastGraf contient le dernier du tableau allGraf
- Insertion :
 - Recherche du parent
 - Recherche du frère gauche
 - Insertion depuis le parent

Avec jQuery

- Création et ajout :

```

var noeud = $('<p>Nouveau texte</p>'); // create node
$("body").append(noeud);                // après le dernier fils

```

34. <https://developer.mozilla.org/en-US/docs/Web/API/Node/parentNode>

- Sélection et Suppression :

```
var noeud = $("p"); // select node(s)
noeud.remove();
```

Références

- Une réintroduction à JavaScript ³⁵
- How does it feel to learn JS in 2016 ³⁶
- Référence MDN ³⁷
- Tutoriels The Modern JS Tuto ³⁸ w3schools ³⁹ LearnJS ⁴⁰
- Outils de développement Chrome et Firefox (F12, Ctrl+Shift I)
- Visualisation du DOM ⁴¹
- Outils web
 - JSFiddle ⁴²
 - JSLint ⁴³

Sources

-
- 35. https://developer.mozilla.org/fr/docs/Web/JavaScript/Une_r%C3%A9introduction_%C3%A0_JavaScript
 - 36. <https://hackernoon.com/how-it-feels-to-learn-javascript-in-2016-d3a717dd577f>
 - 37. <https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference>
 - 38. <https://javascript.info/>
 - 39. <http://www.w3schools.com/js/>
 - 40. <https://learnjavascript.online/>
 - 41. <http://bioub.github.io/dom-visualizer/>
 - 42. <https://jsfiddle.net/>
 - 43. <http://www.jshint.com/>