

Interactions Homme-Machine



8 – Toolkits Graphiques: JavaScript - JQuery

Plan du cours

1. Les Toolkits graphiques.
2. JavaScript, c'est quoi ?
3. Pur JS, Framework, et librairies
4. Ecrire et exécuter du JavaScript
5. Le noyau JavaScript
6. jQuery - Pur JS Vs. jQuery

Toolkits Graphiques

- Programmation d'IHM – Interactions et interfaces.
- = Bibliothèques permettant le développement et la réalisation des interfaces graphiques.
- Différents toolkits pour différents usages et langages de programmation :
 - ✓ Java – Swing
 - ✓ Python – Tkinter
 - ✓ GTK The Gimp Toolkit
 - ✓ Web – HTML CSS Bootstrap
 - ✓ **JavaScript - JQuery**
 - ✓ C++ - Processing / Arduino / Qt
 - ✓ Etc.

JavaScript, c'est quoi ?

- **JavaScript** (JS), un langage de programmation de scripts principalement utilisé dans les pages web **interactives**, mais pas que.
- Rien à voir avec le langage Java.
- Il a été créé en 1995 par Brendan Eich. Il a été standardisé sous le nom d'ECMAScript en juin 1997. Version 8 en 2020.
- Un langage orienté objet.
- Un langage interprété (i.e. script). Nécessite un interpréteur pour s'exécuter directement (versus. un compilateur).
- Un langage dynamique. Permet de générer du contenu dynamique pour les pages web.



JavaScript, c'est quoi ?

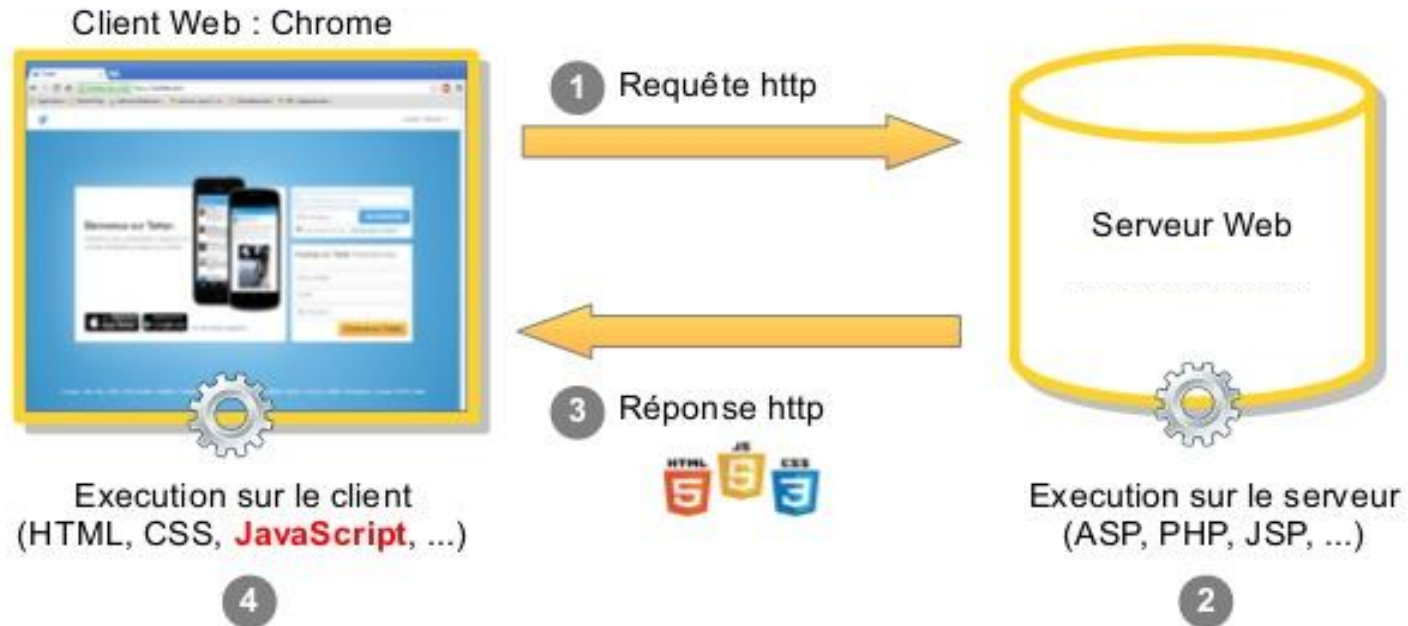
Exemples d'utilisation :

- Sites et applications web dynamiques. Créer des améliorations comme le défilement infini sans recharger la page. Ex : Scroll du mur Facebook.
- Les effets parallaxes. Parallax scrolling. Ex: <https://dogstudio.co/>
Lors du défilement, l'image d'arrière plan et les éléments au premier plan (ex : titres, textes, boutons, etc) bougent à des vitesses différentes. Si les objets devant se déplacent plus vite que l'image de fond, cela crée un effet de profondeur, donnant une impression de 3D.
- Des jeux et des animations dans le navigateur. Ex : Samsung Racer <https://thefwa.com/cases/samsung-racer-s>
- Des applications entières en JS. Ex : Youtube (construit avec AngularJS)
- Modélisation 3D direct dans le navigateur. <https://www.shapespark.com/>

JavaScript, c'est quoi ?



JS, une technologie client **Front-end**

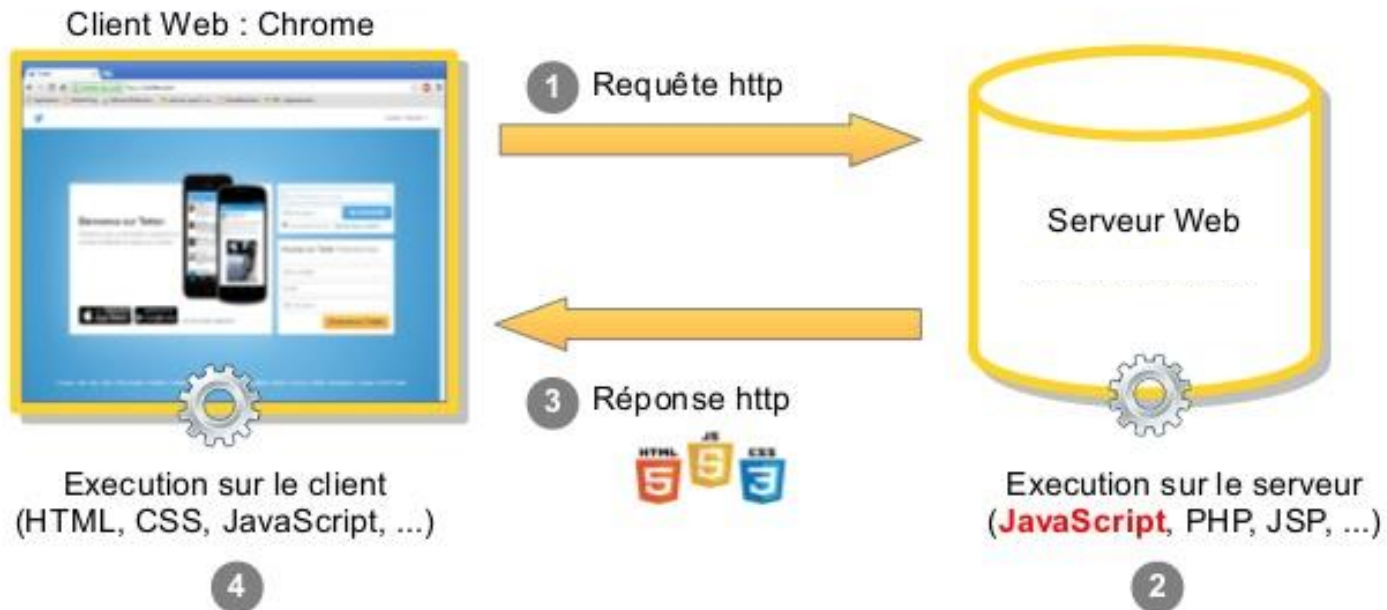


JavaScript, c'est quoi ?



JS, une technologie serveur

Back-end



JavaScript, c'est quoi ?

Frameworks et librairies JS:

Utiliser du **JS pur**, seul. Mais pas adapté pour de grosses applications.

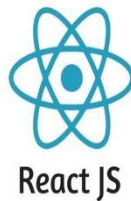
Utiliser une **librairie** ou « bibliothèque » JavaScript : est un ensemble de fichiers de code JavaScript homogènes (= qui se concentrent sur un aspect particulier du langage) qu'on va devoir télécharger pour les utiliser. Ces fichiers de code contiennent des structures de code prêtes à l'emploi qu'on va pouvoir utiliser immédiatement pour gagner du temps en développement.

Utiliser un **framework** ou « cadre de travail » est relativement similaire dans son but à une « super librairie ». Les framework vont également nous fournir un ensemble de codes tout prêts pour nous faire gagner du temps en développement. La grande différence entre un framework et une librairie réside dans l'inversion du contrôle : lorsqu'on télécharge une librairie, on peut l'utiliser comme on le souhaite en intégrant ses éléments à nos scripts tandis que pour utiliser un framework il faut respecter son cadre (ses règles).

JavaScript, c'est quoi ?

Frameworks et librairies JS:

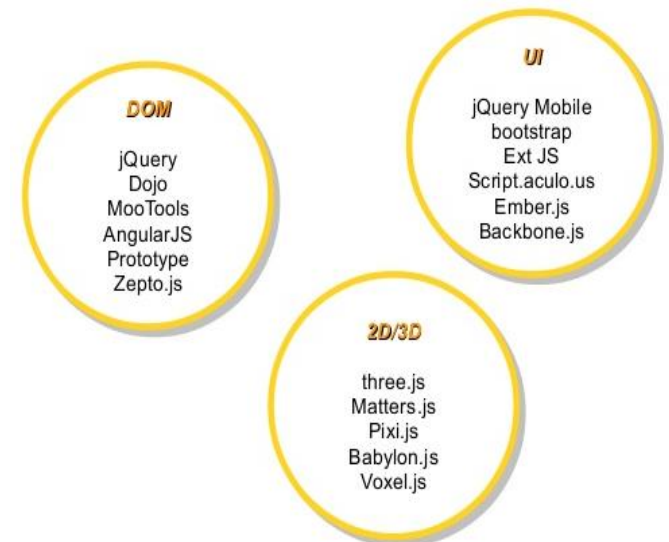
- Front-end : **Angular JS, Reat JS, et JQuery**
- Back-end : **Node JS**



Angular JS



Best JavaScript Frameworks



JavaScript, c'est quoi ?



AngularJS vs jQuery

Which companies use these tools?



AngularJS



Google



Lyft



Udemy



Snapchat



jQuery



Uber



Twitter



reddit



Slack



Walmart

DOW JONES

ebay

intuit

PayPal

NETFLIX

LinkedIn

The New York Times























Microsoft

UBER

YAHOO!

Kingfisher

JavaScript, c'est quoi ?

 React JS <small>Major Users</small>	 ANGULARJS <small>Major Users</small>
 facebook  Instagram	 Google  SONY
 airbnb  WhatsApp	Forbes 
 NETFLIX  UBER	 upwork  HBO
 Dropbox  BBC	 GM  You Tube
 CLOUDFLARE  ATlassian  FLIPBOARD	 PayPal  freelancer iStock.

JavaScript, c'est quoi ?

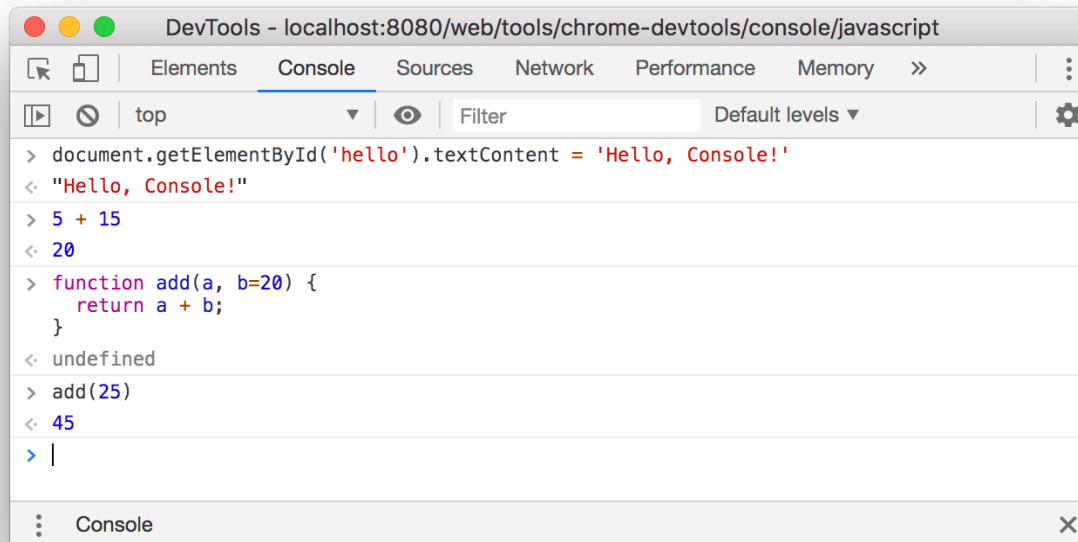
Navigateur et JavaScript:

- Chaque navigateur intègre un **interpréteur** de JS, plus ou moins performant:
 - ✓ SpiderMonkey (Firefox), V8 (Google Chrome), Chakra (Internet Explorer), SquirrelFish (Safari).
- Permet un niveau d'**interactivité** plus riche qu'avec de l'HTML simple:
 - ✓ Certains traitements simples (ex: contrôle des saisies utilisateur) peuvent être réalisés par le navigateur plutôt que par le serveur
 - ✓ Un document HTML/CSS chargé dans le navigateur peut être modifié dynamiquement sans recharger la page.

JavaScript, c'est quoi ?

Ecrire et Exécuter du JavaScript:

1. Directement dans un navigateur web : **Console** développeur
 - Sur Chrome par exemple : press Ctrl+Shift+J ou aller sur Paramètres => Plus d'outils => Outil de développement => Onglet Console
2. **Console** en ligne : <https://jsconsole.com/>



JavaScript, c'est quoi ?

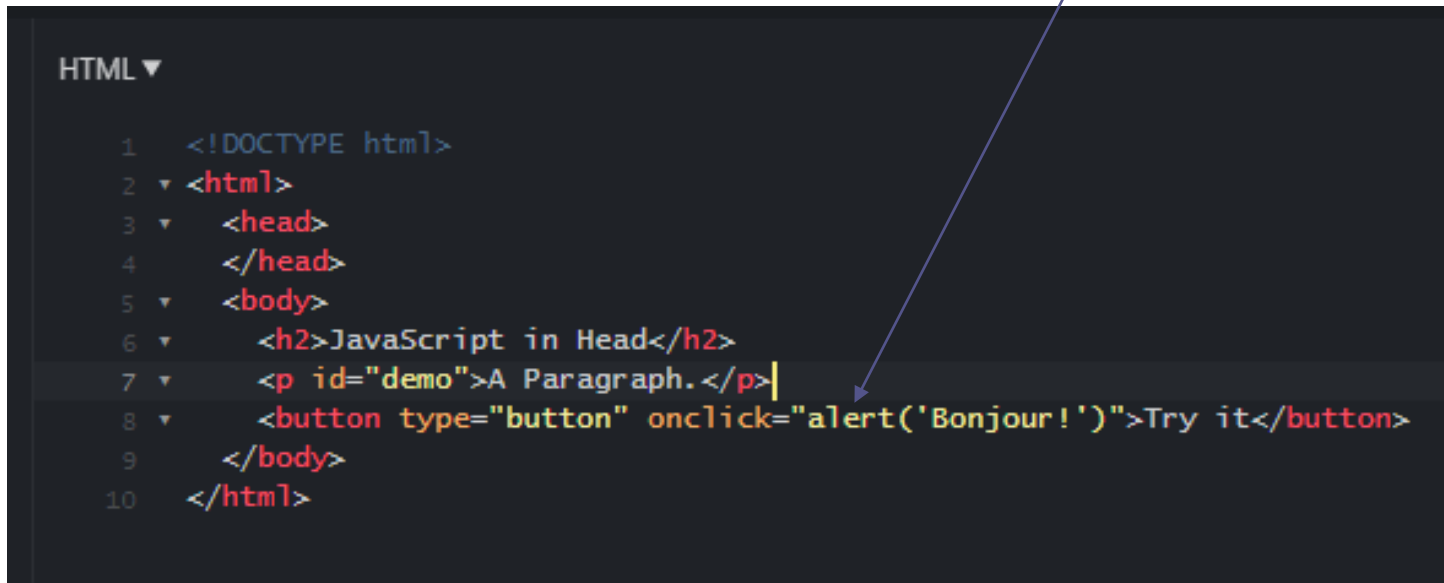
Ecrire et Exécuter du JavaScript:

3. Dans le **HTML**. En utilisant la balise html **<script ></script>**
 - Les attributs de cette balise sont :
 - ✓ **type** : indique le type de contenu (appelé aussi type MIME). La valeur est typiquement "text/javascript". N'est plus obligatoire.
 - ✓ **charset** : indique le jeu de caractères utilisé.
 - ✓ **src** : indique que le code se situe dans un fichier externe.
 - On va pouvoir placer du code JavaScript à trois endroits différents :
 - a) Directement dans la balise ouvrante d'un élément HTML ;
 - b) Dans un élément script, au sein d'une page HTML ;
 - c) Dans un fichier séparé externe contenant exclusivement du JavaScript et portant l'extension **.js**.

JavaScript, c'est quoi ?

Ecrire et Exécuter du JavaScript:

3. Dans le HTML. En utilisant la balise html **<script ></script>**
 - a. Directement dans la balise ouvrante d'un élément HTML. Ex :

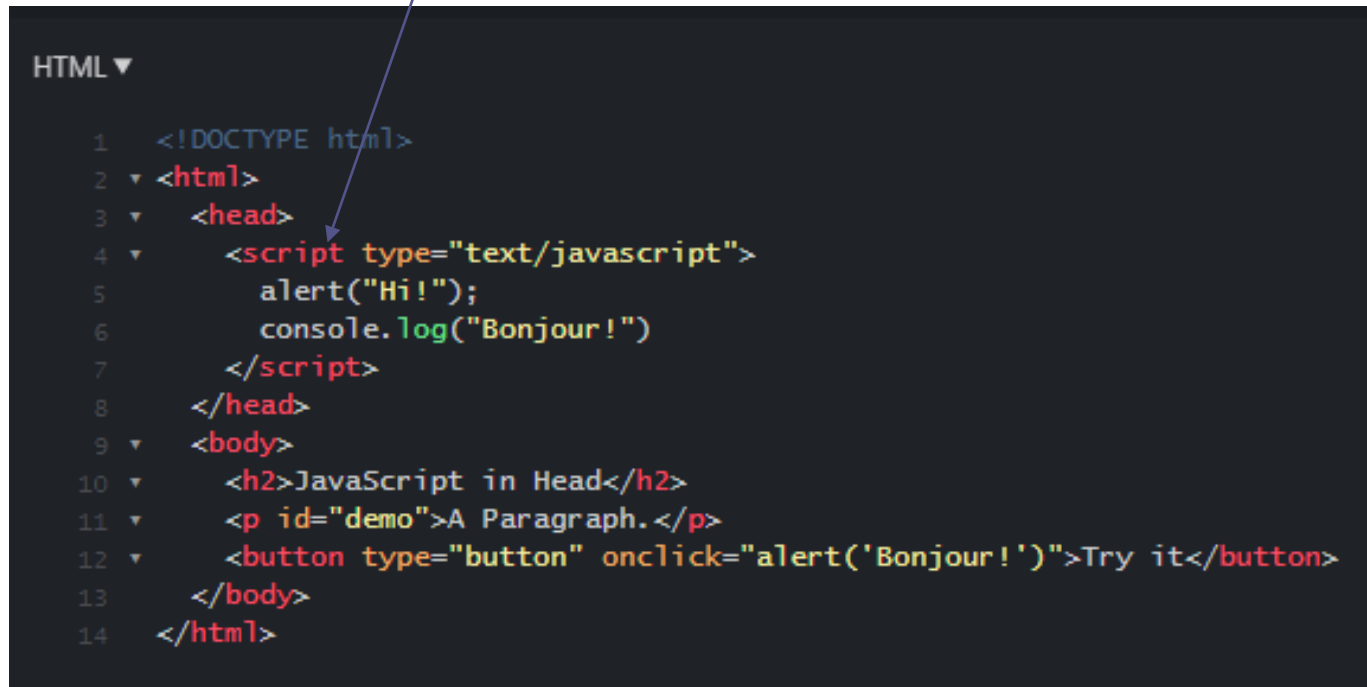


```
HTML ▼  
1 <!DOCTYPE html>  
2 <html>  
3 <head>  
4 </head>  
5 <body>  
6 <h2>JavaScript in Head</h2>  
7 <p id="demo">A Paragraph.</p>  
8 <button type="button" onclick="alert('Bonjour!')">Try it</button>  
9 </body>  
10 </html>
```

JavaScript, c'est quoi ?

Ecrire et Exécuter du JavaScript:

- Dans le HTML. En utilisant la balise html **<script ></script>**
- b. Dans une balise script, au sein d'une page HTML. Ex :



```
HTML ▼
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <script type="text/javascript">
5      alert("Hi!");
6      console.log("Bonjour!")
7  </script>
8  </head>
9  <body>
10 <h2>JavaScript in Head</h2>
11 <p id="demo">A Paragraph.</p>
12 <button type="button" onclick="alert('Bonjour!')">Try it</button>
13 </body>
14 </html>
```


JavaScript, c'est quoi ?

Ecrire et Exécuter du JavaScript:

- Dans le HTML. En utilisant la balise html **<script ></script>**
- b. Dans une balise script, au sein d'une page HTML.
- Vous pouvez placer n'importe quel nombre de balises scripts dans un document HTML.
- Les scripts peuvent être placés dans la <body>, ou dans la section <head> d'une page HTML, ou dans les deux.
- Vous pouvez écrire et tester JS dans le HTML en utilisant n'importe quel éditeur de texte comme Sublime Text.
- Ou en ligne : comme **JSFiddle** : <https://jsfiddle.net/>

JavaScript, c'est quoi ?

Ecrire et Exécuter du JavaScript:

- Dans le HTML. En utilisant la balise html **<script ></script>**
- c. Dans un fichier séparé externe contenant exclusivement du JavaScript et portant l'extension **.js**. External File. Ex :

The screenshot shows a code editor with two panels. The top panel, titled 'HTML', contains the following code:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4   </head>
5   <body>
6     <h2>JavaScript in Head</h2>
7     <p id="demo">A Paragraph.</p>
8     <button type="button">Try it</button>
9     <script src='main.js'></script>
10  </body>
11 </html>
```

The bottom panel, titled 'JavaScript + No-Library (pure JS)', contains the following code:

```
1 alert('Bonjour!')
2
```

Two blue arrows originate from the text 'External File. Ex :'. One arrow points to the `<script src='main.js'></script>` tag in the HTML code. The other arrow points to the `main.js` file name in the file explorer on the right, which is labeled 'Fichier main.js'.

JavaScript, c'est quoi ?

- Il est préférable d'utiliser des **fichiers JS externes** pour des raisons de :
 - ✓ Maintenabilité : le code JavaScript peut être rassemblé dans un unique répertoire.
 - ✓ Caching : un fichier JS partagé par deux pages ne sera téléchargé qu'une seule fois.
 - ✓ Lisibilité.
 - ✓ Rapidité d'exécution.
- Il est recommandé de **placer les balises script juste avant la balise fermante de l'élément body**, après tout code HTML.

JavaScript, c'est quoi ?

JavaScript Display Possibilities

- JavaScript peut "afficher" les données (*output*) de différentes manières:

- ✓ Écriture dans un élément HTML, en utilisant *innerHTML*.

A tester : https://www.w3schools.com/js/tryit.asp?filename=tryjs_output_dom

- ✓ Écriture dans la sortie HTML à l'aide de *document.write ()*.

A tester : https://www.w3schools.com/js/tryit.asp?filename=tryjs_output_write

- ✓ Écrire dans une boîte d'alerte, en utilisant *window.alert ()*.

A tester : https://www.w3schools.com/js/tryit.asp?filename=tryjs_output_alert

- ✓ Écriture dans la console du navigateur, à l'aide de *console.log ()*.

A tester : https://www.w3schools.com/js/tryit.asp?filename=tryjs_output_console

Le noyau JavaScript

JavaScript est formé de trois composants:

1. **Le Core ECMAScript**, qui est défini dans l'édition ECMA-262, et qui fournit les fonctionnalités centrales du langage.
2. **DOM** (Document Object Model) qui fournit les fonctionnalités pour interagir avec le contenu HTML d'une page web.
3. **BOM** (Browser Object Model) qui fournit les fonctionnalités permettant de manipuler et d'interagir avec le navigateur.



Le noyau JavaScript

Le **Core ECMAScript**, qui est défini dans l'édition ECMA-262, et qui fournit les fonctionnalités centrales du langage.

- Définit ce qui suit, comme tout autre langage de programmation :

- JS Comments
- JS Variables
- JS Operators
- JS Arithmetic
- JS Assignment
- JS Data Types
- JS Functions
- JS Objects
- JS Events
- JS Strings
- JS String Methods
- JS Numbers
- JS Number Methods
- JS Arrays

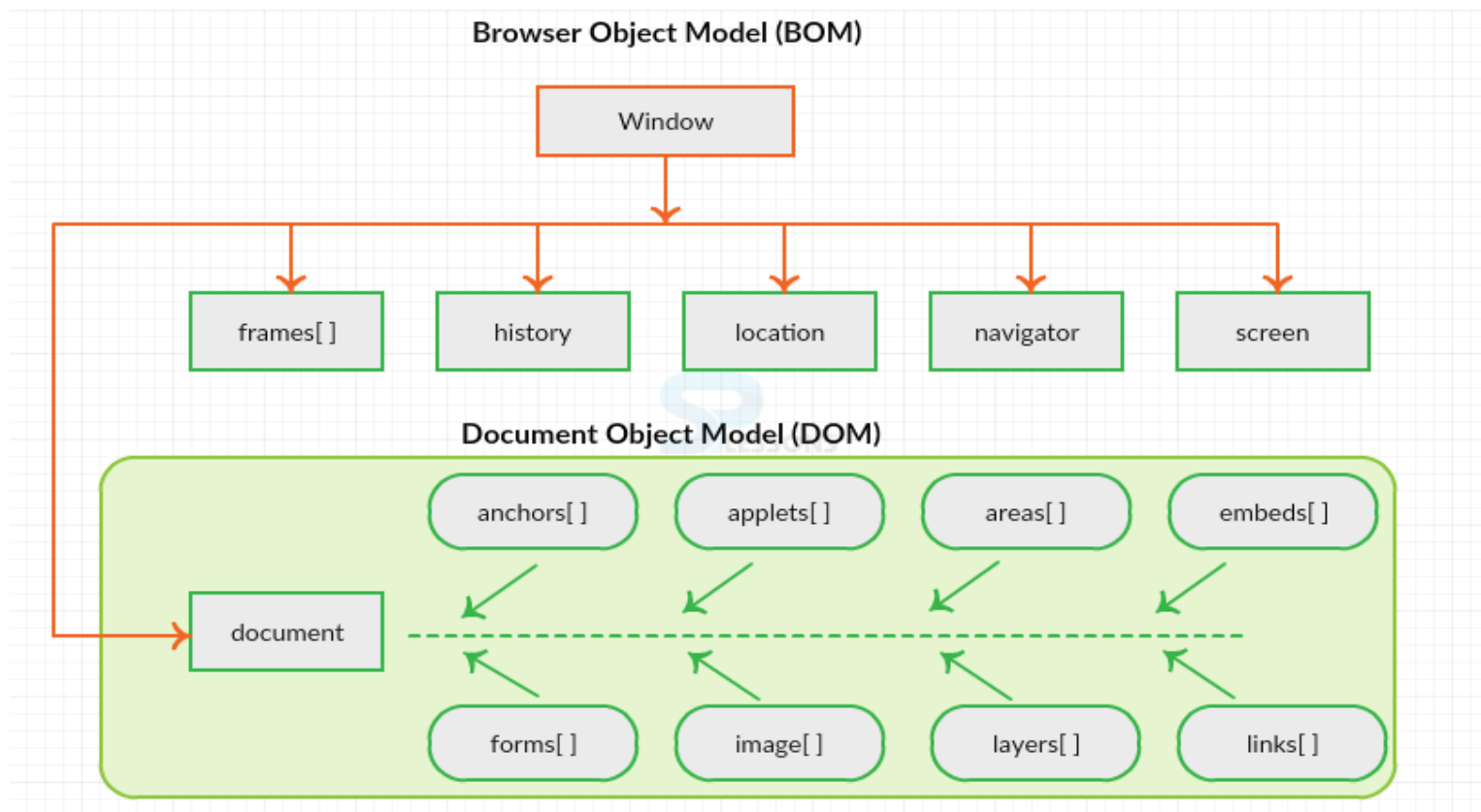
- JS Math
- JS Random
- JS Booleans
- JS Comparisons
- JS Conditions
- JS Switch
- JS Loop For
- JS Loop While
- JS Break
- JS Type Conversion
- JS Bitwise
- JS RegExp
- JS Errors
- JS Scope
- JS Hoisting
- JS Strict Mode
- JS this Keyword
- JS Let
- JS Const

- JS Arrow Function
- JS Classes
- JS Debugging
- JS Style Guide
- JS Best Practices
- JS Mistakes
- JS Performance
- JS Reserved Words
- JS Versions
- JS Version ES5
- JS Version ES6
- JS JSON
- JS Forms

Le noyau JavaScript

Le **BOM** (Browser Object Model) qui fournit les fonctionnalités permettant de manipuler et d'interagir avec le navigateur.

Arborescence :



Le noyau JavaScript

Le **BOM** (Browser Object Model) qui fournit les fonctionnalités permettant de manipuler et d'interagir avec le navigateur.

Window: L'objet *window* représente la fenêtre du navigateur web.

- Dans un navigateur utilisant des onglets, chaque onglet contient son propre objet Window.
- Possède de nombreuses méthodes et propriétés dont notamment des propriétés *document*, *navigator*, *location* qui retournent des références aux objets du même nom.
- Quelques **propriétés** : `window.innerHeight`, `window.innerWidth`, `window.outerHeight` et `window.outerWidth`
- Quelques **méthodes** : `window.open()`, `window.close()`, `window.moveTo()`, `window.resizeTo()`.

Le noyau JavaScript

Le **BOM** (Browser Object Model) qui fournit les fonctionnalités permettant de manipuler et d'interagir avec le navigateur.

Window: Exemple :

Try it : https://www.w3schools.com/js/tryit.asp?filename=tryjs_win_inner

```
HTML ▼
1  <!DOCTYPE html>
2  <html>
3  <head>
4  </head>
5
6  <body>
7    <p id="demo"></p>
8    <script>
9      var w = window.innerWidth;
10     var h = window.innerHeight;
11     var x = document.getElementById("demo");
12     x.innerHTML = "Browser inner window width: " + w + ", height: " + h + ".";
13   </script>
14 </body>
15
16 </html>
```

Le noyau JavaScript

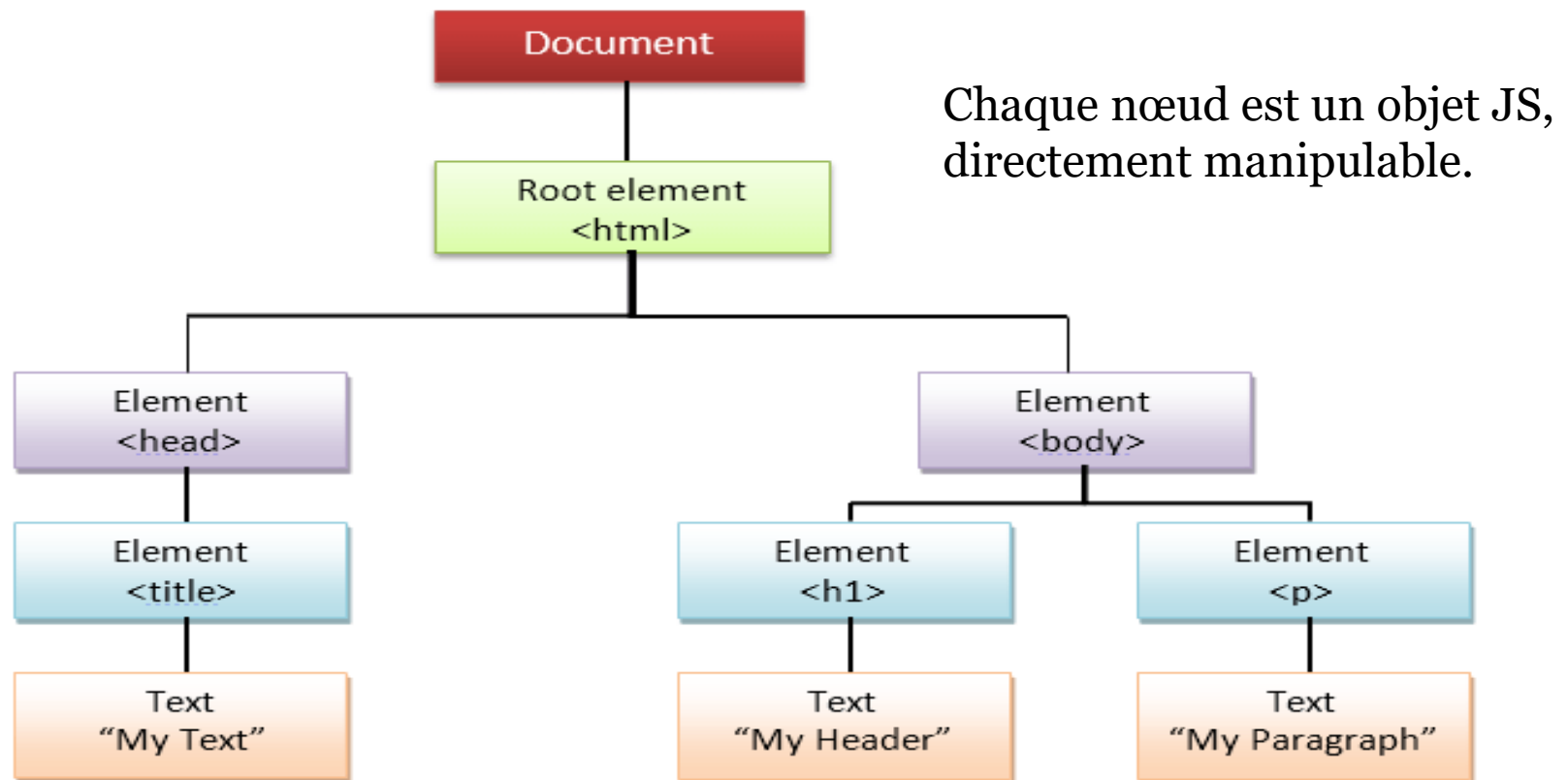
Les objets suivants appartiennent au **BOM** et sont tous des enfants de **Window** :

- L'objet **Navigator** qui représente l'état et l'identité du navigateur ; **Ex**: navigator.appName, navigator.platform, etc.
- L'objet **History** qui permet de manipuler l'historique de navigation du navigateur; **Ex**: history.back(), history.forward(), etc.
- L'objet **Location** qui fournit des informations relatives à l'URL de la page courante ; **Ex**: location.href, location.pathname, location.reload(), etc.
- L'objet **Screen** qui nous permet d'examiner les propriétés de l'écran qui affiche la fenêtre courante ; **Ex**: screen.width, screen.height, etc.
- L'objet **Document** et le DOM.

Le noyau JavaScript

Le DOM (Document Object Model) qui fournit les fonctionnalités pour manipuler et interagir avec le contenu XML et HTML d'une page web.

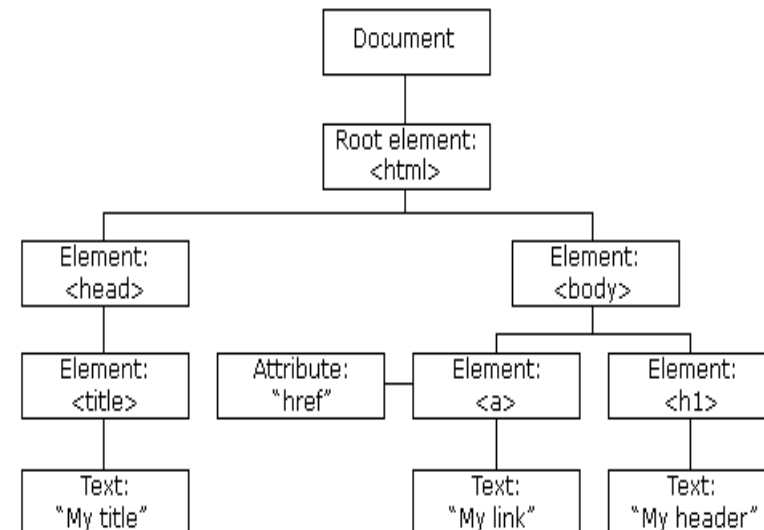
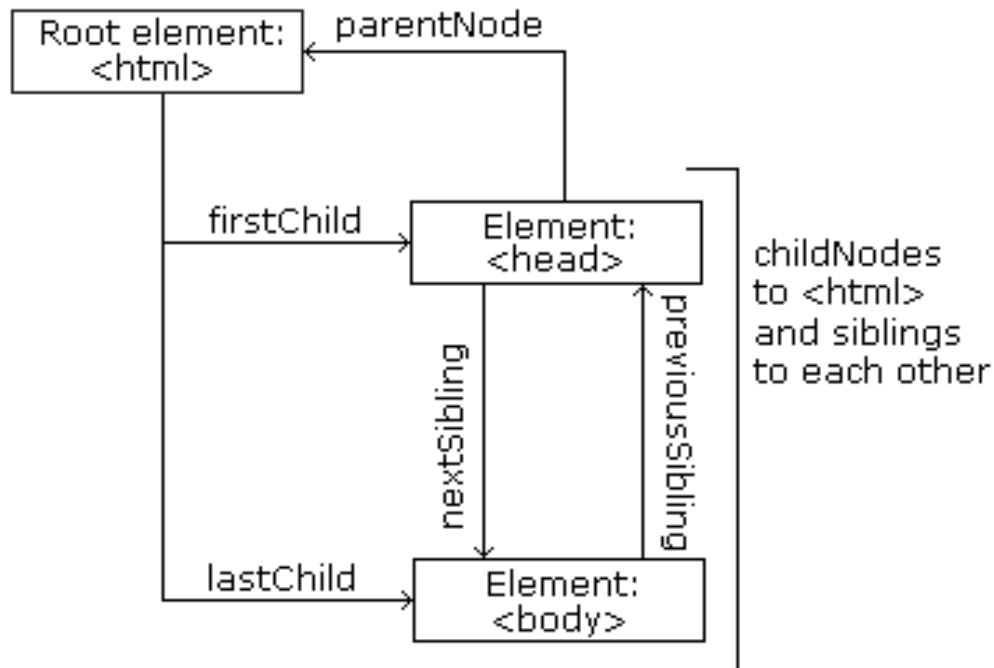
Arborescence d'objets:



Le noyau JavaScript

Le **DOM** (Document Object Model) qui fournit les fonctionnalités pour manipuler et interagir avec le contenu XML et HTML d'une page web.

Arborescence d'objets: [DOM Nodes Navigation](#)



Le noyau JavaScript

Le **DOM** (Document Object Model) qui fournit les fonctionnalités pour manipuler et interagir avec le contenu XML et HTML d'une page web.

- Le DOM HTML est un modèle d'objet standard et une interface de programmation (API) pour HTML.
- Il définit:
 - Les **éléments** HTML comme objets
 - Les **propriétés** de tous les éléments HTML
 - Les **méthodes** pour accéder à tous les éléments HTML
 - Les **événements** pour tous les éléments HTML
- En d'autres termes: le DOM HTML est un standard pour savoir comment obtenir, modifier, ajouter ou supprimer des éléments HTML.

Le noyau JavaScript

Le **DOM** (Document Object Model) qui fournit les fonctionnalités pour manipuler et interagir avec le contenu XML et HTML d'une page web.

Exemple : **élément**: rechercher et récupérer des éléments HTML; différentes manières possibles :

- Recherche d'éléments HTML par **id** : `document.getElementById("ido1")`
- Recherche d'éléments HTML par nom de **balise**:
`document.getElementsByTagName("div")`
- Recherche d'éléments HTML par nom de **classe** :
`document.getElementsByClassName("container")`
- Recherche d'éléments HTML par des sélecteurs **CSS**:
`document.querySelectorAll("div.container")`

Le noyau JavaScript

Le **DOM** (Document Object Model) qui fournit les fonctionnalités pour manipuler et interagir avec le contenu XML et HTML d'une page web.

Exemple : **méthode**: *getElementById()*, qui permet de récupérer et d'accéder à un élément (balise) HTML avec son **id**.

Try it : https://www.w3schools.com/js/tryit.asp?filename=tryjs_dom_method

```
HTML ▼
1  <!DOCTYPE html>
2  <html>
3  <head>
4  </head>
5
6  <body>
7    <p id="demo"></p>
8    <script>
9      var w = window.innerWidth;
10     var h = window.innerHeight;
11     var x = document.getElementById("demo");
12     x.innerHTML = "Browser inner window width: " + w + ", height: " + h + ".";
13   </script>
14 </body>
15
16 </html>
```

Le noyau JavaScript

Le **DOM** (Document Object Model) qui fournit les fonctionnalités pour manipuler et interagir avec le contenu XML et HTML d'une page web.

Exemple : propriété: *innerHTML*, qui change le contenu d'une balise HTML

Try it : https://www.w3schools.com/js/tryit.asp?filename=tryjs_dom_method

```
HTML ▼
1  <!DOCTYPE html>
2  <html>
3  <head>
4  </head>
5
6  <body>
7    <p id="demo"></p>
8    <script>
9      var w = window.innerWidth;
10     var h = window.innerHeight;
11     var x = document.getElementById("demo");
12     x.innerHTML = "Browser inner window width: " + w + ", height: " + h + ".";
13   </script>
14 </body>
15
16 </html>
```

Browser inner window width: 414, height: 268.

Le noyau JavaScript

Le DOM

Événements:

- Un évènement est provoqué par une action de l'utilisateur en interagissant avec l'IHM ou du navigateur lui-même.
- Une **fonction** appelée en réponse à un évènement se nomme un écouteur (event handler ou **event listener**). Souvent, leur nom commence par *on* comme par exemple **onclick** ou **onload**.
- Définit par : un **type d'évènement**, les **actions** à exécuter au déclenchement de l'évènement, et un **lien** entre le composant et les actions.

Un composant :

Cliquez-moi !

Un type d'évènement :

onclick

Des actions :

```
document.getElementById("bouton1").classList.remove("btn-info");  
document.getElementById("bouton1").classList.add("btn-success");  
alert("Modification réussie du bouton1");
```

Le noyau JavaScript

Le DOM

Événements:

Évènement	Élément(s) html concerné(s)
onLoad	BODY, FRAMESET, OBJECT
onUnload	BODY et FRAMESET
onError	IMG, OBJECT, BODY et FRAMESET
onAbort	BODY et FRAMESET
onSelect	INPUT et TEXTAREA
onChange	INPUT, SELECT et TEXTAREA
onSubmit	FORM
onReset	FORM
onFocus	LABEL, INPUT, SELECT, TEXTAREA et BUTTON
onBlur	LABEL, INPUT, SELECT, TEXTAREA et BUTTON
onResize	BODY
onScroll	BODY
onClick	Quasiment tout
onMouseOver	Quasiment tout
onContextMenu	Quasiment tout

Le noyau JavaScript

Événements: Exemple 1: onmouseover et mouseout – Inline Event Attribute

Try it : https://www.w3schools.com/js/tryit.asp?filename=tryjs_events_mouseover

The screenshot displays a web browser interface with a dark theme. On the left, the 'HTML' panel shows the following code:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 </head>
5
6 <body>
7 <div onmouseover="mOver(this)" onmouseout="mOut(this)"
8 style="background-color:#D94A38;width:120px;height:20px;padding:40px;">
9   Mouse Over Me
10 </div>
11 <script>
12   function mOver(obj) {
13     obj.innerHTML = "Thank You"
14   }
15   function mOut(obj) {
16     obj.innerHTML = "Mouse Over Me"
17   }
18 </script>
19 </body>
20 </html>
```

On the right, the 'CSS' panel is empty, showing only a line number '1'. Below the code panels, a red rectangular box is visible with the text 'Mouse Over Me' in white. Two blue arrows point from the text 'Événements:' in the title to the 'onmouseover' and 'onmouseout' attributes in the HTML code. Another blue arrow points from the text 'Exemple 1:' to the 'mOver' function definition in the script.

Le noyau JavaScript

Événements: Exemple 2: onclick - Utilisation de HTML DOM

Try it : https://www.w3schools.com/js/tryit.asp?filename=tryjs_events2

The screenshot shows a web development tool interface with two panels: HTML and CSS. The HTML panel contains the following code:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 </head>
5
6 <body>
7 <p>Click "Try it" to execute the displayDate() function.</p>
8 <button id="myBtn">Try it</button>
9 <p id="demo"></p>
10 <script>
11   document.getElementById("myBtn").onclick = displayDate;
12   function displayDate() {
13     document.getElementById("demo").innerHTML = Date();
14   }
15 </script>
16 </body>
17 </html>
```

The CSS panel is empty, showing only a line number 1. Below the code panels, there is a text instruction: "Click 'Try it' to execute the displayDate() function." and a button labeled "Try it".

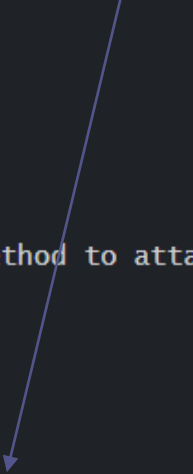
Le noyau JavaScript

Événements: Exemple 3: onclick - Utilisation d'Event Listeners

Try: https://www.w3schools.com/js/tryit.asp?filename=tryjs_addEventListener_displayDate

HTML ▼

```
2 <html>
3 <head>
4 </head>
5
6 <body>
7 <h2>JavaScript addEventListener()</h2>
8 <p>This example uses the addEventListener() method to attach a click
9 event to a button.</p>
10 <button id="myBtn">Try it</button>
11 <p id="demo"></p>
12
13 <script>
14     document.getElementById("myBtn").addEventListener("click", displayDate);
15     function displayDate() {
16         document.getElementById("demo").innerHTML = Date();
17     }
18 </script>
19 </body>
20 </html>
```



jQuery - Pur JS Vs. jQuery

- **jQuery** a été créé en 2006 par John Resig. Il a été conçu pour gérer les incompatibilités du navigateur et simplifier la manipulation HTML DOM, la gestion des événements, les animations, et Ajax.
- Pure JS Vs. jQuery :

jQuery

```
var myElement = $("#id01");
```

Try it Yourself »

JavaScript

```
var myElement = document.getElementById("id01");
```

Try it Yourself »

- Ne pas oublier d'ajouter jQuery au document HTML en ajoutant :
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js">

Pur JS Vs. jQuery

Try it: https://www.w3schools.com/js/js_jquery_selectors.asp

| Pur JS | jQuery |
|---|--|
| <pre>var myElement =
document.getElementById("ido1");</pre> | <pre>var myElement = \$("#ido1");</pre> |
| <pre>var myElements =
document.getElementsByTagName("p");</pre> | <pre>var myElements = \$("p");</pre> |
| <pre>var myElements =
document.getElementsByClassName("intro");</pre> | <pre>var myElements = \$(".intro");</pre> |
| <pre>var myElements =
document.querySelectorAll("p.intro");</pre> | <pre>var myElements = \$("p.intro");</pre> |

Pur JS Vs. jQuery

Try it: https://www.w3schools.com/js/js_jquery_elements.asp

Pur JS	jQuery
<pre>// Supprimer un élément HTML document.body.removeChild(document.get ElementById("ido1"));</pre>	<pre>// Supprimer un élément par id \$("#id").remove();</pre>
<pre>// Récupérer l'inner text d'un élément var myText = myElement.textContent myElement.innerText;</pre>	<pre>// Récupérer l'inner text d'un élément HTML var myText = myElement.text();</pre>
<pre>//Cacher un élément HTML myElement.style.display = "none";</pre>	<pre>//Cacher un élément HTML myElement.hide();</pre>
<pre>// Changer le CSS d'un élément HTML myElement.style.fontSize = "35px";</pre>	<pre>myElement.css("font- size","35px");</pre>
<pre>var myElement.innerHTML = "<p>Hello World</p>";</pre>	<pre>var myElement.html("<p>Hello World</p>");</pre>

Javascript

Aller plus loin :

- Exemples code Javascript à tester – w3schools :

https://www.w3schools.com/js/js_examples.asp

- MOOC complet en ligne – Youtube PrimFX :

<https://www.youtube.com/playlist?list=PLEagTQfI6nPPVSKoYo2p8Cf8eijcyz5t9>

- Projet 1 – Simple Work ToDoList with JS

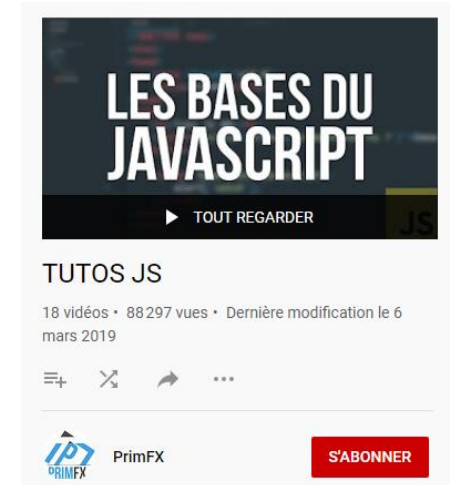
Code source : <https://codepen.io/JohnPaulFich/pen/MXmzzM>

- Projet 2 – Lolcat Clock with JS

Code source : <https://codepen.io/codifiedconcepts/pen/bwgxRq>

- D'autres projets JS : <https://code-projects.org/c/languages/project/jsprojects/>

Ex : <https://code-projects.org/todo-list-web-app-in-javascript-with-source-code/>

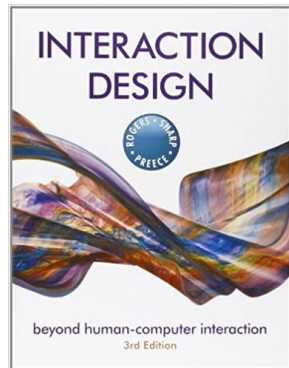


Références



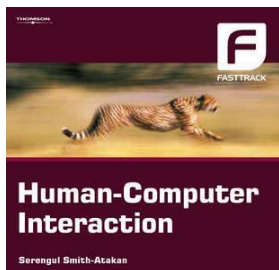
Designing Interactive Systems: A Comprehensive Guide to HCI, UX and Interaction Design, 3rd Edition

- ✓ Auteur : David Benyon
- ✓ Éditeur : Pearson
- ✓ Edition : 2013



Interaction Design: beyond human-computer interaction (3rd edition)

- ✓ Auteur : Yvonne Rogers, Helen Sharp & Jenny Preece
- ✓ Éditeur : Wiley
- ✓ Edition : 2011



The FastTrack to Human-Computer Interaction

- ✓ Auteur : Serengul Smith-Atakan
- ✓ Éditeur : Thomson Learning
- ✓ Edition : 2006

Références

Cours:

<https://perso.liris.cnrs.fr/fabien.duchateau/docs/ens/LIFIHM/programmation-interactions.pdf>

W3Schools : <https://www.w3schools.com/js/default.asp>

Cours : <http://www.cril.univ-artois.fr/~paris/cours/javascript/slidesJavascriptFr.pdf>

Cours : <https://www.pierre-giraud.com/javascript-apprendre-coder-cours/>

Cours : <https://fr.slideshare.net/OlivierLeGoar/cours-javascript-60125613>

Template Projet : <https://www.free-css.com/free-css-templates/page252/theevent>