

- Un document HTML est un arbre (DOM). Il peut être décoré avec des propriétés pour le présentation, notamment celles issues de CSS. JavaScript est un langage de script qui permet de manipuler le DOM (et d'autres structures) de manière dynamique.
- Langage interprété par le navigateur.
- Syntaxiquement proche de Java, C,...
- Langage à objets
- Exécution mono-thread (le thread est partagé avec le reste de la page, ou pire avec le reste du navigateur)
- La norme ECMA Script n'est pas toujours pleinement suivie dans tous les navigateurs
- <http://www.w3schools.com/js/default.asp>

- Inclusion directe de JavaScript dans un document HTML, dans `<head>` ou dans `<body>` :
 `<script type="text/javascript">`
 ...
 `</script>`
- Ou bien lien vers un fichier externe (préférable, car mise en cache possible) :
 `<script src="myScript.js"></script>`

- Commentaires : // une ligne de commentaire ou /* plusieurs lignes */
- Variables
 - globales : x = 3
 - locales : var x = 3
- Types
 - nombres : 0, 1, 3.14, 123e5
 - booléens : true, false
 - chaînes : "toto", 'titi'
 - null
 - tableaux : ['rouge', 'vert', true, 8.15]
 - objets : {firstname:"John", lastname:"Doe", id:5566}

- `var joe = {firstname:"John", lastname:"Doe", id:5566}`
- `var joe = new Object() ;`
`joe.firstname="John" ;`
`joe.lastname="Doe" ;`
`joe.id=5566 ;`
`joe.toString() = function() { ... } ;`
- `var tab = ['rouge', 'vert', true, 8.15] ;`
- `var tab = new Array() ;`
`tab[0] = 'rouge' ;`
`tab[1] = 'vert' ;`
`tab[2] = true ;`
`tab[3] = 8.15 ;`

- `if (condition) {...} else {...}`
- `switch (n) { case i: ... break ; case j: ... break ; default: ... }`
- `for (initialization ; condition ; post-iteration) { ... }`
ex. : `for (var i=0 ; i < 10 ; i++) { sum += i ; }`
- `for (variable in object) { ... }`
(parcourt les propriétés de l'objet)
- `while (condition) { ... }`
- `do { ... } while (condition);`
- `try { ... } catch (error) { ... }`

- `function name (inputs) { ... }`
ex. : `function plus1(i) { return i+1 ; }`

- Object
- Array
- String (.length, .replace(...,...), .split(...),...)
- Date (new Date(), .getDay(),...)
- Math (.round(...), max(...), .PI,...)
- RegExp (new RegExp(pattern, modifiers), .exec(), .test())

- Tout le DOM est un objet JavaScript : *document*
- on peut y écrire directement : *document.write('<p>Test</p>')*
- on peut y naviguer comme dans tout autre objet JavaScript : *if (document.body.childNodes.length > 0) ...*
- on peut retrouver les éléments directement
 - par ID : *document.getElementById(...)*
 - par type : *document.getElementsByTagName(...)*
- on peut lire/modifier le code HTML :
document.getElementById('toto').innerHTML='!'
ou toto.appendChild(document.createElement('em'))
titi.appendChild(document.createTextNode(' !'))
- on peut lire/modifier les attributs des éléments HTML :
document.getElementById('toto').title = 'Important !'
- on peut lire/modifier le style : *elem.style.property*
document.getElementById('toto').style.color = 'red'

- Le DOM est un arbre qui réagit aux événements (clic, modification, survol, fin de chargement d'une image,...).
- On peut « écouter » ces événements au niveau de l'arbre entier ou d'un noeud particulier.
- Les événements se propagent dans l'arbre : depuis l'élément origine (this) jusqu'à la racine.

```
<em onclick='this.style.color='red''>vrai</em>  
<em onclick='rougir()'>vrai</em>
```

```
var em = ... ; em.onclick=function(){this.style.color='red'} ;  
var em = ... ; em.onclick=rougir ;
```

```
function rougir() { this.style.color='red' }
```


- souris
 - click, dblclick
 - mouseover, mouseout
 - mouseup, mousedown, mousemove
- clavier : keypress, keyup, keydown
- zones de saisie
 - change, select, reset
 - focus, submit
- fenêtre/objets
 - load, unload
 - resize, scroll
 - error, abort

[http://www.w3schools.com/jsref/
dom_obj_event.asp](http://www.w3schools.com/jsref/dom_obj_event.asp)

- JavaScript Object Notation (JSON) est un format de données basé sur JavaScript
- Format textuel, reprenant les notations de tableaux et d'objets en JavaScript
ex : { "nom" : "Potter", "prenom" : "James", "enfants" : [{ "nom" : "Potter", "prenom" : "Harry" }] }
- Avantage d'échanger des données en JSON : cela revient à échanger des objets natifs, directement exploitables/manipulables dans JavaScript
- <http://www.json.org/>