

# Cours N°3 – JavaScript

The most Popular Scripting language

Programmation Web  
Master GL - 2019- 2020

# Partie II

- Evènements et interactions
- jQuery

# Java script

## JS :

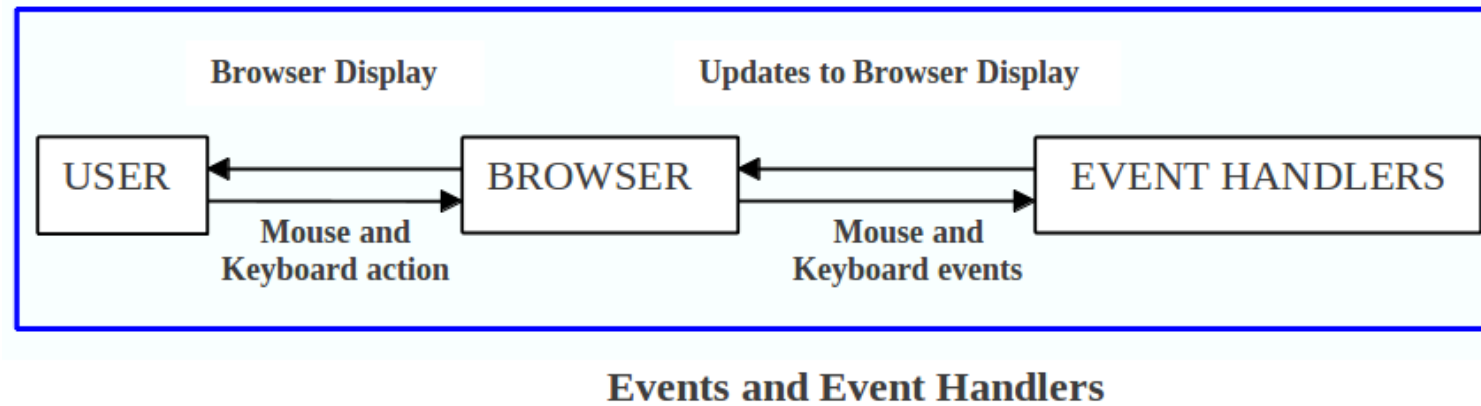
- Un langage évènementiel
- Interagit avec la page à travers une structure appelée « **DOM** » Structures
- Type de données: simple, Tableau, Objet « **var** »
- Conditionnelles, Structures de contrôles «**if, for, while, do/while, for ..in** »
- Fonction « **function, return, this** »

# Introduction aux événements

- Les événements représentent des fonctions appelées en réaction à une action de l'utilisateur : souris, clavier,....
- La fonction répondant à un événement est appelée « gestionnaire d'événement »
- Les événements sont par convention préfixés par « **on** ». Par exemple : « onclick »

# Affectation d'un gestionnaire d'évènement

- L'affectation d'un gestionnaire d'évènement à un objet se fait en renseignant certains attributs de cet objet correspondant à l'évènement
- La valeur de l'attribut correspond à l'appel d'une fonction JS



# Affectation d'un gestionnaire d'évènement

## Objet et Événements associables

Objet	Événements associables
Lien hypertexte	onClick, onMouseOver, onMouseOut
Page du navigateur	onLoad, onUnload
Bouton, Case à cocher, Bouton radio, Bouton Reset	onClick
Liste de sélection d'un formulaire	onBlur, onChange, onFocus
Bouton Submit	onSubmit
Champ de texte et zone de texte	onBlur, onChange, onFocus, onSelect

NB: l'objet **window** (représentant le **navigateur**).

# Affectation de Gestionnaire-Exemple

- Affiche une boîte de dialogue lors du chargement de la page.
- ```
<script language="javascript">  
window.document.OnLoad=alert("Je m'affiche lors  
du chargement de la page");  
</script>
```

```
<div class="vignette" onmouseover="survoler(this)"  
onmouseout="sortir(this)">Survolez-moi</div>
```

- ```
<html><head><title>Les Fonctions et les évènements</title>
<script language="javascript">
function afficheAlert()
{
    alert("Je m'affiche lors du clic sur le bouton Valider") ;
}
</script>
</head><body>

<form id="form" name="form" method="post" action="" >
<p>
<input type="submit" name="Submit" value="Afficher"
onClick="afficheAlert()" />
</p>
</form>

</body></html>
```



- ```
<html><head><title>Les Fonctions et les évènements</title>
<script language="javascript">
function afficheAlert(sButtonName)
{
    alert("Je m'affiche lors du clic sur le bouton "+sButtonName)
}
</script>
</head><body>

<form id="form" name="form" method="post" action="" >
<p>
<input type="submit" name="Submit" value="Afficher"
onClick="afficheAlert(this.value)" />
</p>
</form>

</body></html>
```

```
function survoler(objet) {  
    objet.style.backgroundColor = 'yellow';  
}  
function sortir(objet) {  
    objet.style.backgroundColor = 'lightgreen';  
}  
function click() {  
    alert('la div a ete cliquee');  
}  
function affecterEvenement() {  
    var objet = document.getElementById('myDiv');  
    objet.onclick = click;  
    objet.onmouseover = function (e) { survoler(e.target)};  
    objet.onmouseleave = function (e) { sortir(e.target); }  
}
```

# L'objet « *évènement* »

○ Pour exécuter un code suivant un évènement :

Exécuter directement un code js

Appel d'une fonction à l'intérieur de script

Exécuter directement le code js et influencer par l'utilisation du mot clé **this**

## Avantages:

- Contrôle de saisie (handle user input)
- Réagir simplement et rapidement avec les utilisateurs (user action)
- Réagir simplement et rapidement avec les navigateurs (browser action)

- On écrit un événement à l'intérieur de la balise script
- Forme Générale

```
object.EventName=function() {myScript};
```

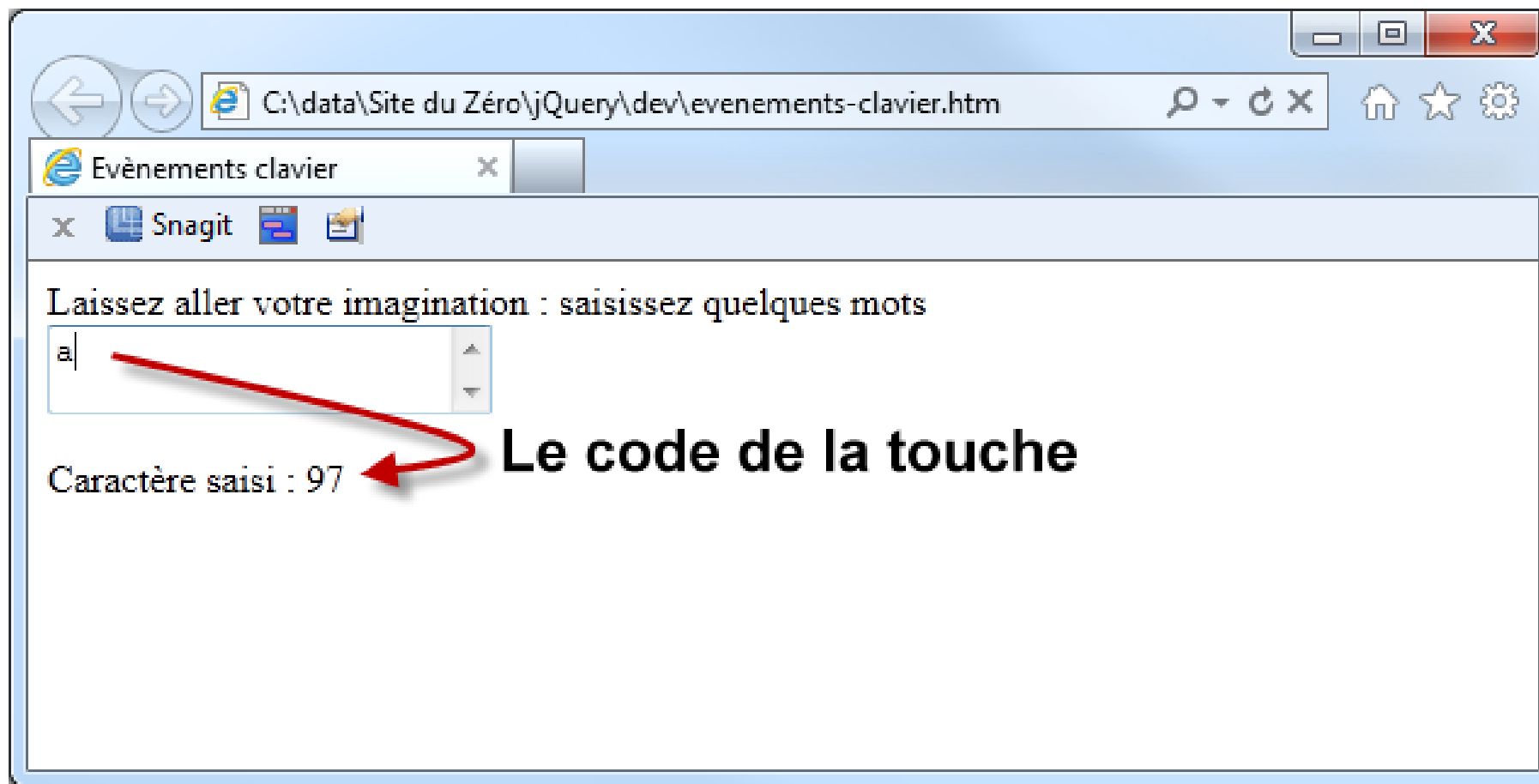
- Exemple:

```
document.getElementById("btnMouseMove").onmouseover = function() {myFunctionOver()};
```

**Event:** est un objet

Propriété	Description
modifiers	Touches spéciales (Par exemple Ctrl ou Alt)
button	Bouton gauche ou droit de la souris (0 ou 2)
pageX, pageY	Coordonnées
cancelable	Indique si l'évènement peut être annulé
target	Objet relatif à l'évènement
charCode	Relatif aux évènements clavier, code caractère
keyCode	Relatif aux évènements clavier, code touche
altKey	Indique si la touche « alt » est pressée
ctrlKey	Indique si la touche « ctrl » est touchée

## Exercice







## Exemple : Authentification dans Windows



Evènement	Description
onclick	Clic simple de la souris
ondblclick	Clic double
onmousedown	Clic de la souris sur un élément
onmouseup	Quand l'utilisateur relache la souris
onmousemove	Quand la souris est déplacée sur un élément
onmouseover	Quand la souris entre dans l'espace d'un élément
onmouseout	Quand la souris sort de l'espace d'un élément

Evènement	Description
onchange	Est provoqué au fur et à mesure que le contenu d'un contrôle est changé
oninput	Provoqué au fur et à mesure du changement du contenu dans une zone de texte
onfocus	Provoqué lorsqu'un contrôle reçoit le focus
onblur	Provoqué lorsqu'un contrôle perd le focus
onsubmit	Lorsque le formulaire est envoyé au serveur



Evènement	Description
onload	Appelé lorsque la page est chargée
onresize	Lorsque la fenêtre est redimensionnée
onerror	Lorsqu'une erreur se produit lors du chargement d'une image
onabort	Lorsque le chargement d'une image est annulé

- Les minuteurs sont des gestionnaires d'évènements qui sont déclenchés à des intervalles de temps définies en millisecondes
- Les minuteurs (timers) déclenchent des actions périodiques sur la page
- La création d'un minuteur se fait en utilisant « ***setInterval*** »
- L'arrêt d'un minuteur se fait en utilisant « ***clearInterval*** »

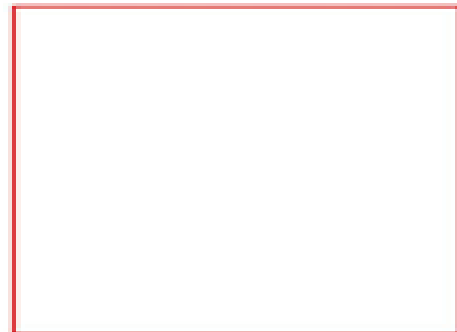
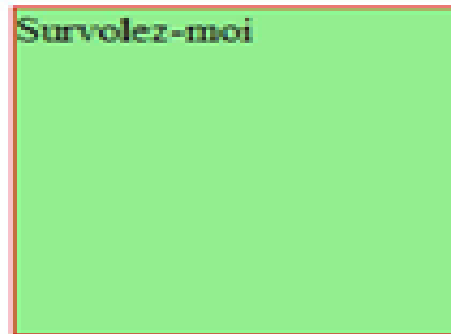
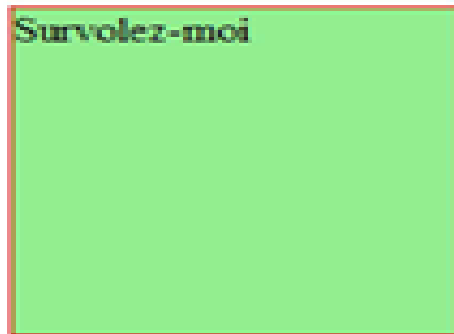
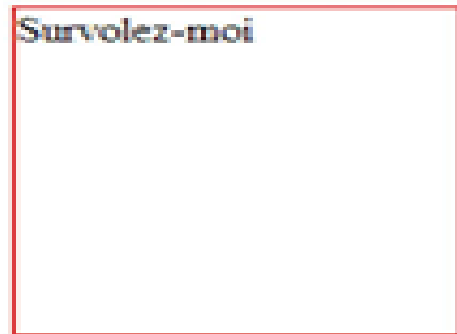
Méthode	Description
<code>open()</code>	Ouvre une nouvelle fenêtre
<code>close()</code>	Ferme la fenêtre
<code>moveTo()</code>	Déplace la fenêtre
<code>resizeTo()</code>	Redimensionne la fenêtre
<code>moveBy()</code>	Déplace la fenêtre par rapport à la position en cours
<code>resizeBy()</code>	Redimensionne par rapport à la taille actuelle

[illegible]



## Page chargée

379,238



Affecter événement

11:29:18

Arrêter

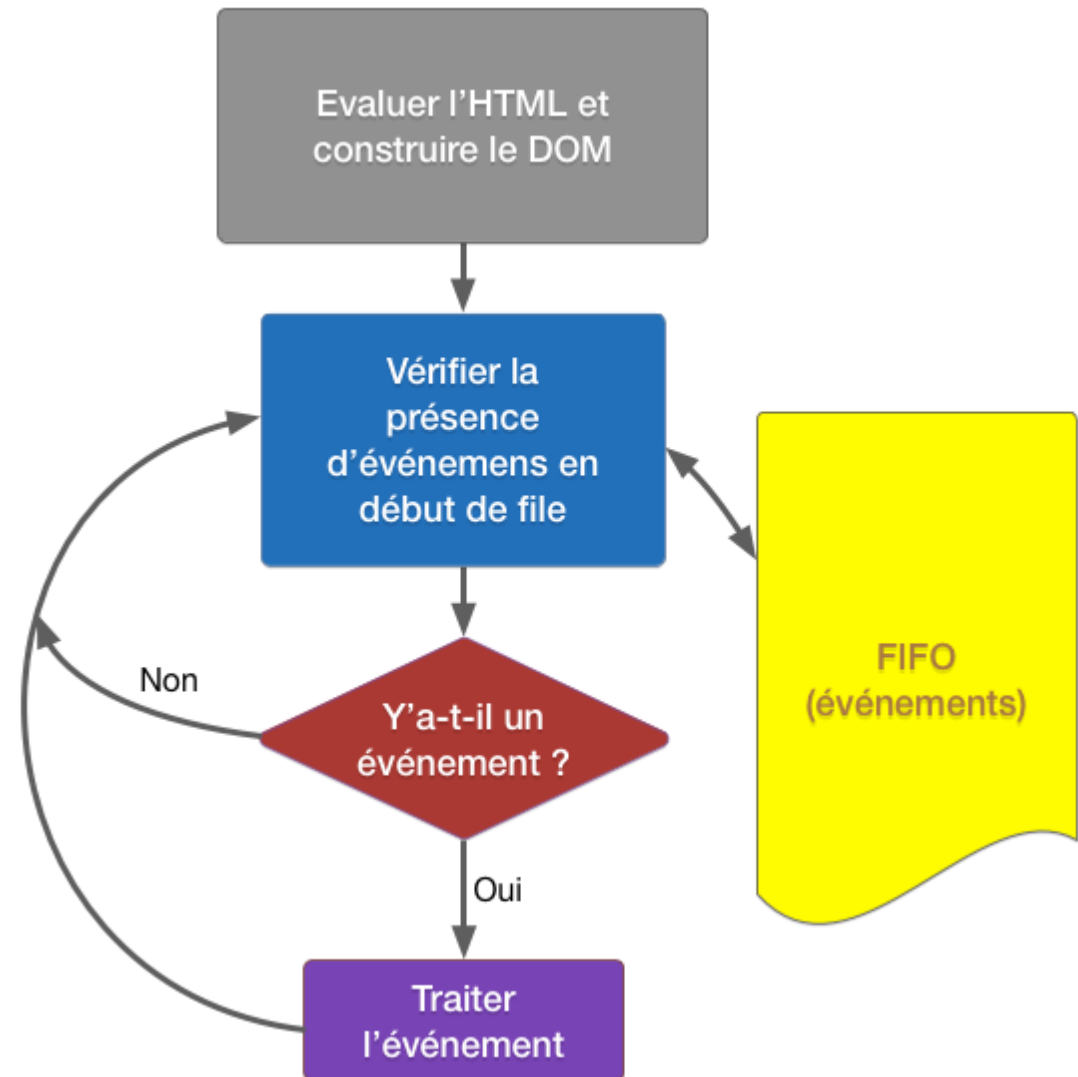
Ouvrir dans une nouvelle fenêtre



# La boucle d'événements

## Le navigateur :

- Mettre en place l'IHM ;
- se mettre dans une boucle d'événements  
en attente qu'un événement se produise ;
- invoquer les actions à exécuter pour chaque événement.



# La boucle d'événements

- La gestion des événements est totalement prise en charge par le navigateur
- Nous avons juste besoin de spécifier les actions, ou les listeners, pour les différents événements qui peuvent être déclenchés (toujours au sein du navigateur).
- Ces événements sont placés dans une FIFO au fur et à mesure qu'ils se produisent, et le navigateur traite ces événements en invoquant les **listeners** associés.
- Une chose à savoir, la boucle d'événements du navigateur est “**mono-threadée**”, ce qui veut dire que chaque événement se trouvant dans la file d'attente, est traité (dans l'ordre). C'est ce que l'on appelle une FIFO. Les événements sont donc traités un par un, à tour de rôle. Voici un schéma très simplifié de ce processus
- Ce principe de listeners — associer une fonction qui sera exécutée plus tard, lorsque l'événement se produira — illustre un mécanisme que l'on appelle *les fonctions de **callback***.

- **Exercice 01: Webmining**

Collecting data with *JavaScript*

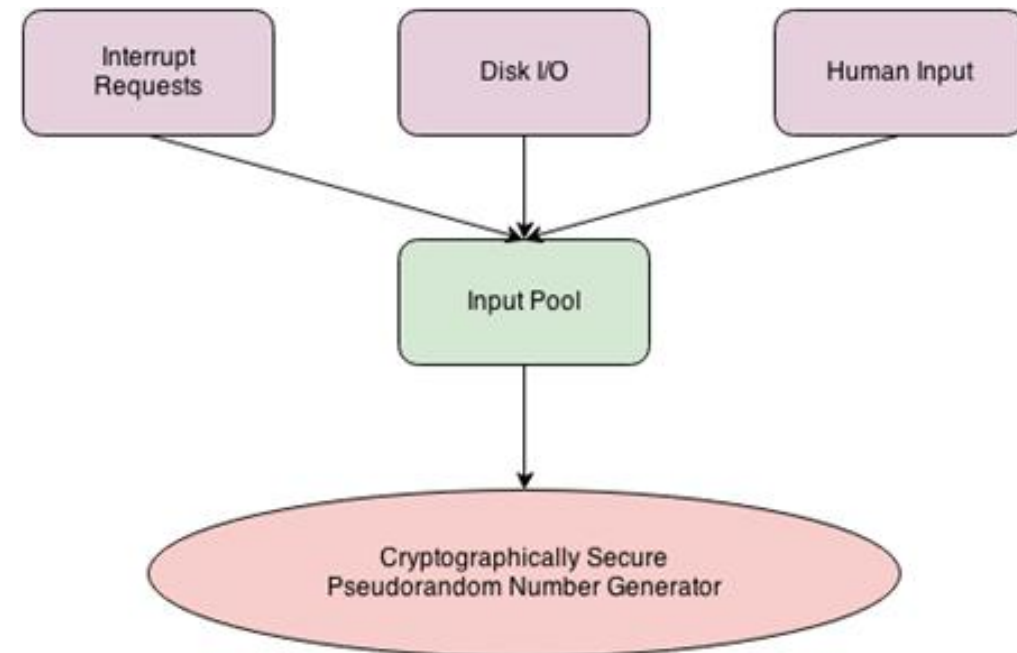
- **Exercice 02: Cryptographie**

Inspiré de : **Entropy and Random Number Generators**

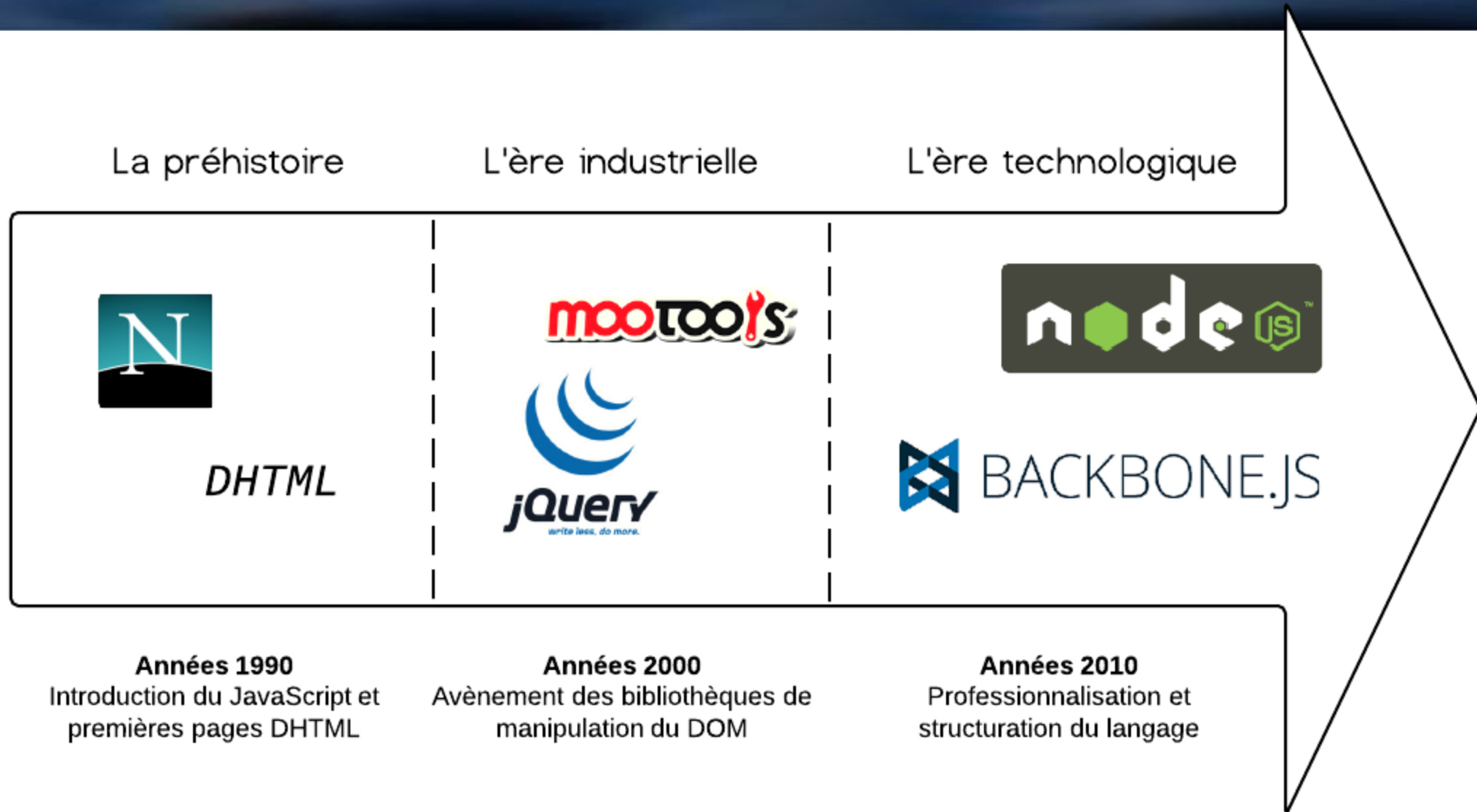
**the problem with our random number generators ?**

Source : [https://calomel.org/entropy\\_random\\_number\\_generators.html](https://calomel.org/entropy_random_number_generators.html)

**Cryptographie** : Générer une clé

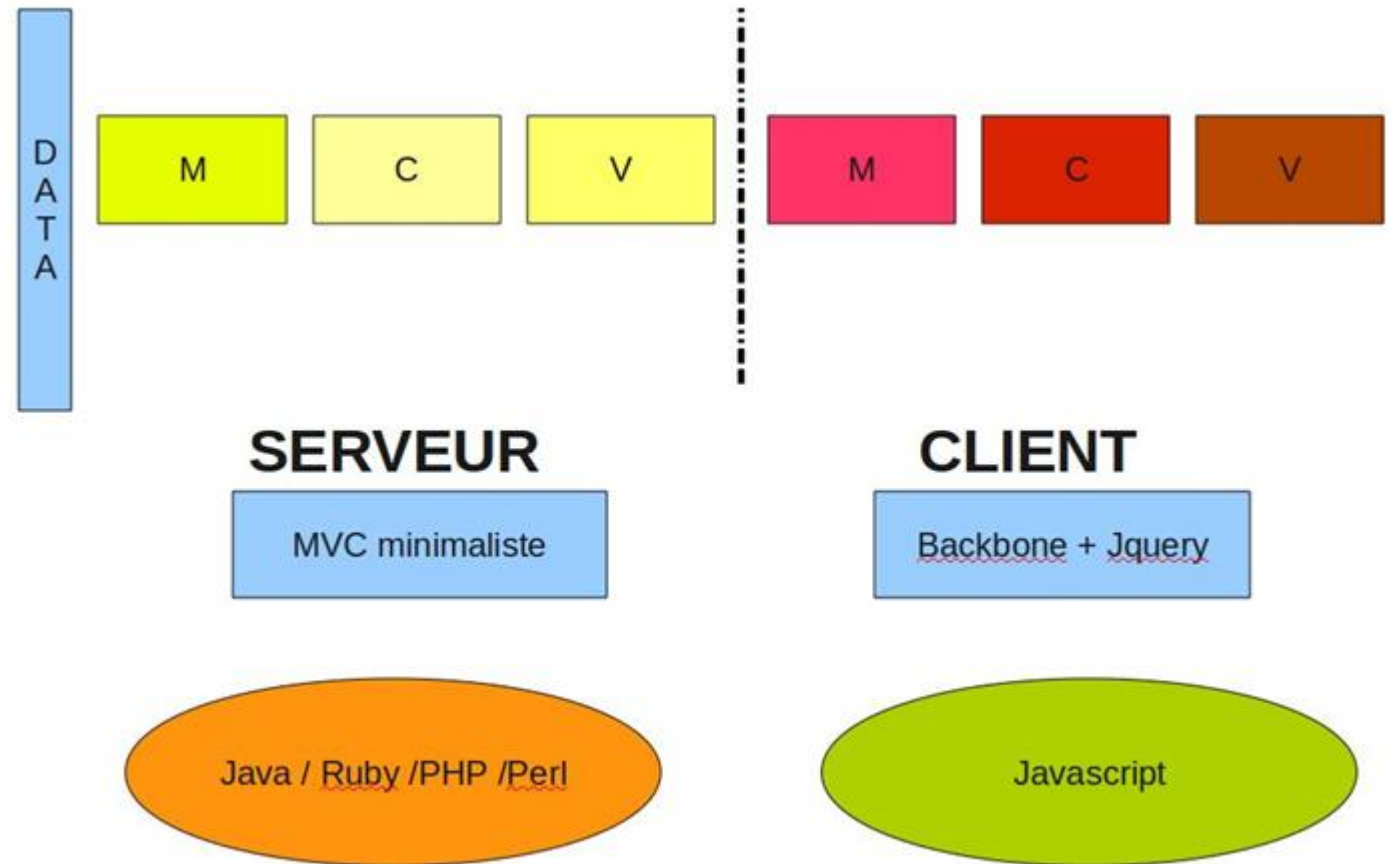


# Section 8: JQuery



## L'apparition des MVC coté client:

- Ces MVC sont forcément en javascript,
- seul langage supporté par les navigateurs.
- Parmi les noms de [MVC javascript](#), [backbone.js](#) est celui le plus cité.



# Autres frameworks Javascript

- Prototype : petit, simple et élégant
- **YUI**: La librairie JS de Yahoo
- **Dojo**: une librairie JS complète
- **mooTools**: Idem



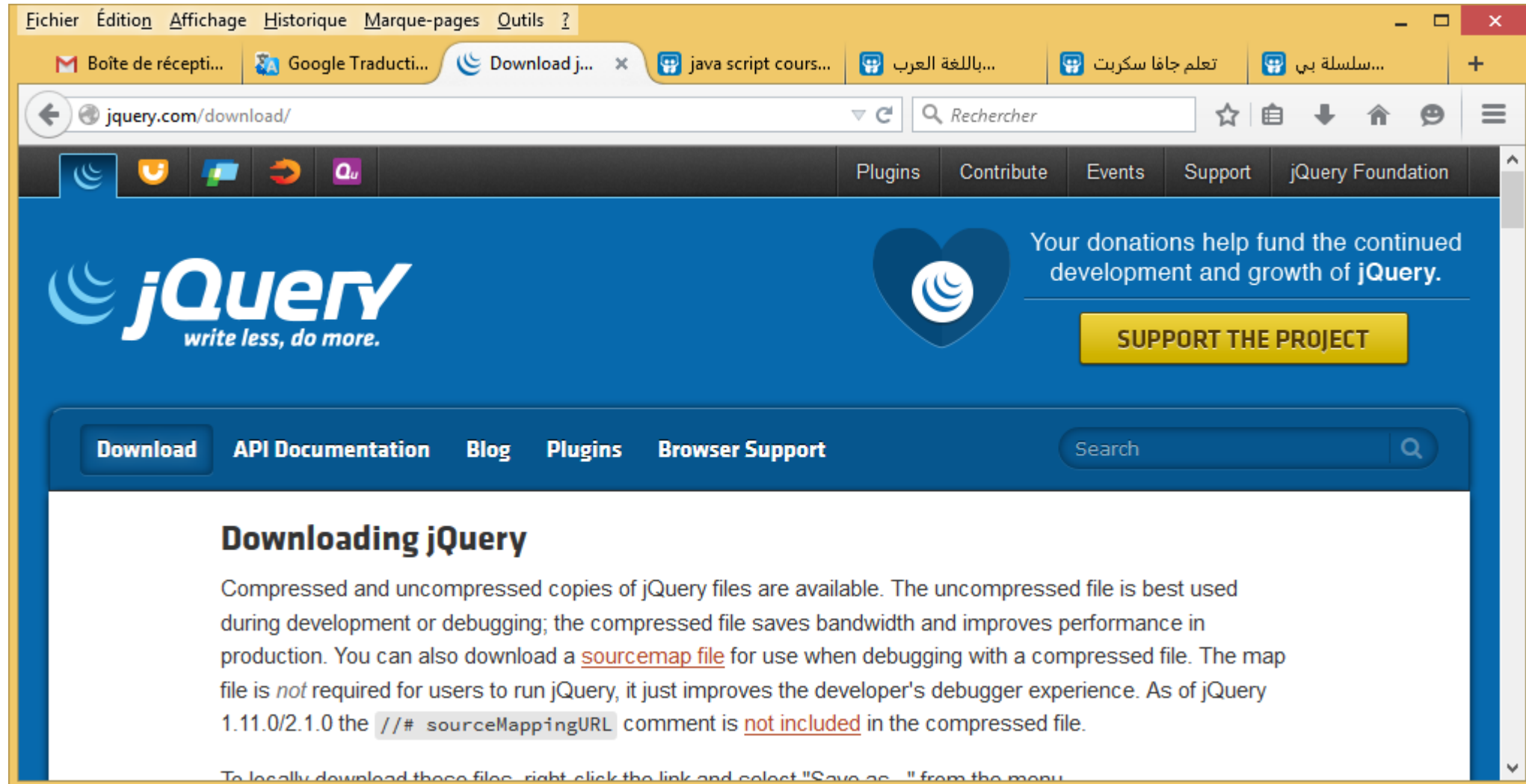
# Différences JQuery

- ☐ Léger
- ☐ Simple
- ☐ Apprentissage rapide
- ☐ Pas complexe
- ☐ Taille réduite
- ☐ Intégration facile avec divers composants

C'est quoi JQuery !

- Bibliothèque JavaScript adoptée par la majorité des industriels : IBM, Microsoft , Amazon, Dell, Twitter et Netflix
- JQuery facilite la manipulation du DOM en appliquant des connaissances HTML, CSS ou JS
- JQuery permet de créer des scripts non obstrusifs
- Reference : <http://api.jquery.com/>
- Téléchargement dernière version : <http://jquery.com/>

- Economie de code (faire plus de choses avec moins de code)
- Code non obstrusif (separation de la structure du traitement)
- Compatibilite avec tous les navigateurs
- Taille insignifiante du code



The screenshot shows a web browser window with the jQuery download page. The browser's address bar displays `jquery.com/download/`. The page features the jQuery logo with the tagline "write less, do more." and a navigation menu with links to "Download", "API Documentation", "Blog", "Plugins", and "Browser Support". A search bar is also present. The main content area is titled "Downloading jQuery" and provides information about compressed and uncompressed files, as well as the source map file. A yellow button labeled "SUPPORT THE PROJECT" is visible on the right side of the page.

Fichier Édition Affichage Historique Marque-pages Outils ?

Boîte de récepti... Google Traducti... Download j... java script cours... بالعربية... تعلم جافا سكربت سلسلة بي...

jquery.com/download/ Rechercher

Plugins Contribute Events Support jQuery Foundation

**jQuery**  
write less, do more.

Your donations help fund the continued development and growth of jQuery.

**SUPPORT THE PROJECT**

**Download** API Documentation Blog Plugins Browser Support Search

## Downloading jQuery

Compressed and uncompressed copies of jQuery files are available. The uncompressed file is best used during development or debugging; the compressed file saves bandwidth and improves performance in production. You can also download a [sourcemap file](#) for use when debugging with a compressed file. The map file is *not* required for users to run jQuery, it just improves the developer's debugger experience. As of jQuery 1.11.0/2.1.0 the `//# sourceMappingURL` comment is not included in the compressed file.

To locally download these files, right click the link and select "Save as..." from the menu.

- Décompresser le dossier dans le répertoire de la fonction
- Importer jQuery Dans le head pour pouvoir utiliser ces fonctions  
`<script type="text/JavaScript" src=" nom.js"></script>`

Sélection,  
Manipulation et  
Création  
d'éléments

Fonctions  
Utilitaires

Fonction  
d'initialisation

Extension avec  
les plugins

- Pour manipuler JQuery, il faut passer par un objet « **JQuery** »
- Pour rendre la syntaxe plus compacte, « **\$** » est un alias sur cet objet
- Le traitement initial et l'affectation des événements se fait dans la fonction d'initialisation



// syntaxe 1

```
$(function ()  
    {  
        //  
        traitement  
    });
```

// syntaxe 2

```
$(document).ready(function  
    () {  
        // traitement  
    });
```

- Les sélecteurs JQuery permettent de sélectionner un ou plusieurs éléments HTML
- Les sélecteurs renvoient des ensembles enveloppés
- Pour identifier les éléments, JQuery utilise une syntaxe proche de celle des sélecteurs CSS avec quelques petites spécificité

# Exemples de sélecteurs - Suite

Sélecteur	Description
*	Tous les éléments
div	Tous les div
#monId	Sélectionne l'élément dont l'id est « monId »
p span.maClasse	Tous les span dont la classe est « maClasse » et qui sont descendants d'un paragraphe
div,p	Tous les div et tous les paragraphes
li:first	Premier li
li:odd	Li pairs
li:nth-child(3n+1)	1 <sup>er</sup> , 4 <sup>ème</sup> , ...

# Exemples de sélecteurs - Suite

Sélecteur	Description
eq(n)	Sélectionne le nème élément (1 <sup>er</sup> démarre par 0)
lt(n)	Les n premiers, après le nème, les éléments sont ignorés
gt(n)	Les derniers après le nème élément
input[type=checkbox]	Toutes les cases à cocher
input[type=checkbox][checked]	Toutes les cases à cocher qui sont cochées
:animated	Les éléments qui sont en cours d'animation
:enabled	Éléments activés
:disabled	Éléments désactivés



# Création d'éléments HTML

- L'alias **\$** permet aussi de créer des balises HTML
- Par exemple `$ (« <p>Mon par</p> »)` ou `$ (« <div> »)`

# Ajout d'éléments DOM

Sélecteur	Description
<code>appendTo(elt)</code>	Ajoute des éléments à un ensemble
<code>insertAfter(elt)</code>	Insère après
<code>insertBefore(elt)</code>	Insère avant
<code>html(elt)</code>	Change le HTML interne d'un élément

Fonction	Description
size()	Renvoie le nombre d'éléments d'un ensemble
get(n)	Renvoie l'nème élément de l'ensemble
eq(n)	Renvoie un ensemble enveloppé contenant au plus le nème élément
first()	Renvoie un ensemble enveloppé contenant au plus le premier élément
last()	Renvoie un ensemble enveloppé contenant au plus le dernier élément
toArray()	Convertit l'ensemble en un tableau Javascript
index(elt)	Cherche elt dans l'ensemble puis renvoie sa position ou -1 s'il n'est pas trouvé
add(elt)	Ajoute des éléments à l'ensemble
filter(expression)	Filtre les éléments à filtrer
slice(debut,fin)	Renvoie un sous-ensemble de l'ensemble original



# Gestion d'un ensemble enveloppé - Suite

Fonction	Description
each(fonction)	Fonction appelée pour chaque fonction
children(elts)	Renvoie tous les descendants des éléments d'un ensemble
closest(elts)	Renvoie les éléments les plus proches
find(selecteur)	Cherche les éléments conformes au sélecteur
is(selecteur)	Indique si un élément de l'ensemble est conforme au sélecteur

Fonction	Description
<code>attr(nom_attribut)</code>	Donne la valeur de l'attribut du premier élément
<code>attr(nom,valeur)</code>	Change la valeur d'un attribut
<code>data(nom)</code>	Renvoie une donnée stockée sur un élément
<code>data(nom,valeur)</code>	Change la valeur d'une donnée stockée sur un élément
<code>removeData(nom)</code>	Supprime une donnée stockée
<code>addClass(nom)</code>	Ajoute une classe CSS à un élément
<code>removeClass(nom)</code>	Supprime une classe CSS d'un élément
<code>toggleClass(nom)</code>	Ajoute ou supprime une classe CSS d'un élément
<code>hasClass(nom)</code>	Indique si un élément de l'ensemble a une classe
<code>css(nom,valeur)</code>	Applique directement un style css à un élément

Fonction	Description
<code>width()</code> , <code>width(valeur)</code> , <code>height()</code> , <code>height(valeur)</code>	Renvoie ou change la hauteur ou la largeur des éléments d'un ensemble
<code>text()</code> ou <code>text(valeur)</code>	Renvoie ou change le texte d'un élément
<code>wrap(html)</code>	Envelopper les éléments dans un conteneur
<code>remove(selecteur)</code>	Supprime les éléments de la page
<code>empty()</code>	Vide les éléments
<code>clone()</code>	Clone les éléments
<code>val()</code> ou <code>val(valeur)</code>	Renvoie ou modifie la valeur d'un contrôle de formulaire
<code>hide()</code>	Cacher les éléments
<code>show()</code>	Montre des éléments précédemment cachés
<code>toggle()</code>	Cache / Montre l'élément

# Gestion des évènements

- JQuery permet de simplifier l'affectation des evenements aux elements et elimine tous les problemes d'incompatibilite
- JQuery permet d'affecte un evenement de plusieurs fagons et de le desaffecter
- JQuery peut meme affecter des evenements a des elements qui n'existent pas encore

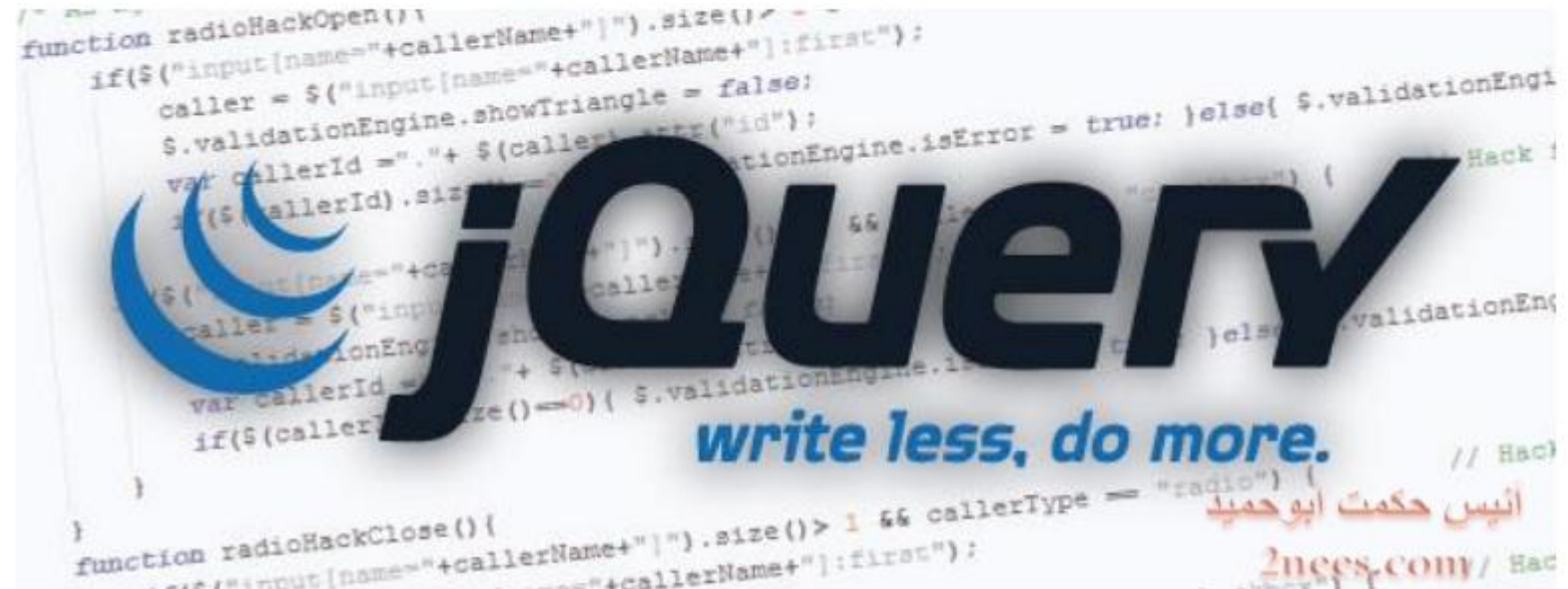
Fonction	Description
click, hover	Des alias pour le clic ou le survol de la souris
bind(nom, fonction)	Associe une fonction à un évènement
unbind(nom)	Supprime un gestionnaire d'évènement
one(nom, fonction)	Gestionnaire d'évènement qui ne s'exécute qu'une seule fois
on(evt,selecteur,fonction )	Attache un évènements aux éléments sélectionnés (s'applique aussi aux éléments créés dynamiquement)
trigger(evt,donnees)	Déclenche manuellement un évènement

- Certaines fonctions comme «show » ou « hide » peuvent etre animees en ajoutant une duree d'execution
- La fonction « animate » permet de changer des proprietes d'une maniere animee
- Il est interessant de combiner JQuery avec les animations et les transformations CSS3



- Hide Effect
- Show Effect
- Highlight Effect
- Puff Effect
- Effect Pulsate
- Scale Effect
- Effect Shake
- Size Effect
- Effect Switch C
- Toggle Effect

## jQuery Effects





# Bibliographie

- Java Script Reference, W3C Schools
- Sam's Teach Yourself, HTML, CSS, and JavaScript, 2012
- JQuery in Action, Bear Bibeault, Yehuda Katz, Mann, 2010
- JQuery reference ([www.JQuery.com](http://www.JQuery.com))

# Example

# Accéder à la balise à l'intérieur d'un ... `<body>`

```
$(document).ready(function(){  
  $("?????").....  
});
```

????? >>> balise quand l'on peut les manipuler

\* : veut dire les objets de la page

```
$(document).ready(function(){  
  $("p").click(function(){  
    $(this).hide();  
  });  
});
```

# Ecrire une fonction à l'intérieur de jquery

```
$(document).ready(
```

```
function () {} );
```

OR

```
$(document).ready(function(){
```

```
})
```

# Ecrire une fonction à l'intérieur de jQuery

2ème écriture

```
$(function(){  
  // jQuery methods go here...  
})
```

3ème écriture

```
jQuery(function($){  
  //jQuery methods go here...  
});
```

# Accéder à la balise à l'intérieur d'un ... `<body>`

Accéder à un élément :

class ou id

```
$(document).ready(function(){  
  $(".nameOfclass").click(function(){  
    $(this).hide();  
  });  
});
```

Class (.) et id #

# Example

# hide Method

Cacher les objets dans la page internet :

```
$(document).ready(function(){  
  $(".1").click(function(){  
    $(this).hide(5000);  
  });  
});
```

Propriété : vitesse

```
$(this).hide(5000);  
$(this).hide("slow");  
$(this).hide("fast");
```



# show method

Meme principe que la methode hide

```
$(document).ready(function(){  
  $(".a").show(5000);  
});
```

# show And hide method

exécution séquentielle

```
$(document).ready(function(){  
  $(".a").show(5000).hide(2000).show(3000);  
})
```

# delay Method

Pour temporiser l'exécution

```
$(document).ready(function(){  
  $(".a").show(2000).delay(4000).hide(2000);  
});
```

# fadeOut Method

Visualiser un texte par dégradation

On peut déterminer le temps

```
$(document).ready(function(){  
  $(".a").fadeOut(2000);  
});
```

# fadeOut Method

- Au contraire de fadeIn

```
$(document).ready(function(){  
  $(".a").fadeOut(5000);  
});
```