

Cours N°3 – JavaScript

The most Popular Scripting language

Programmation Web
Master GL - 2019- 2020

Partie II

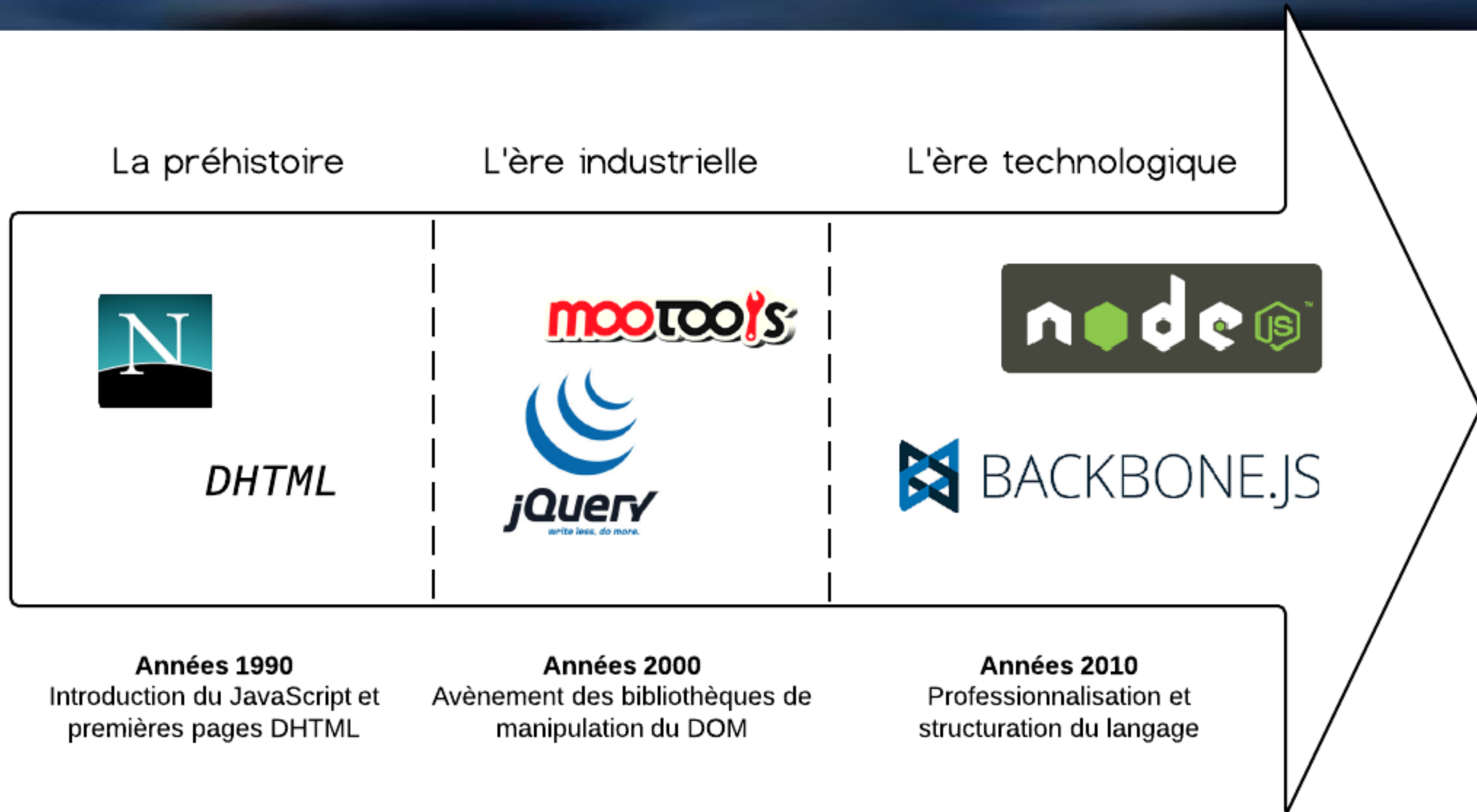
- Evénements et interactions
- jQuery

Section 8: JQuery

\$()

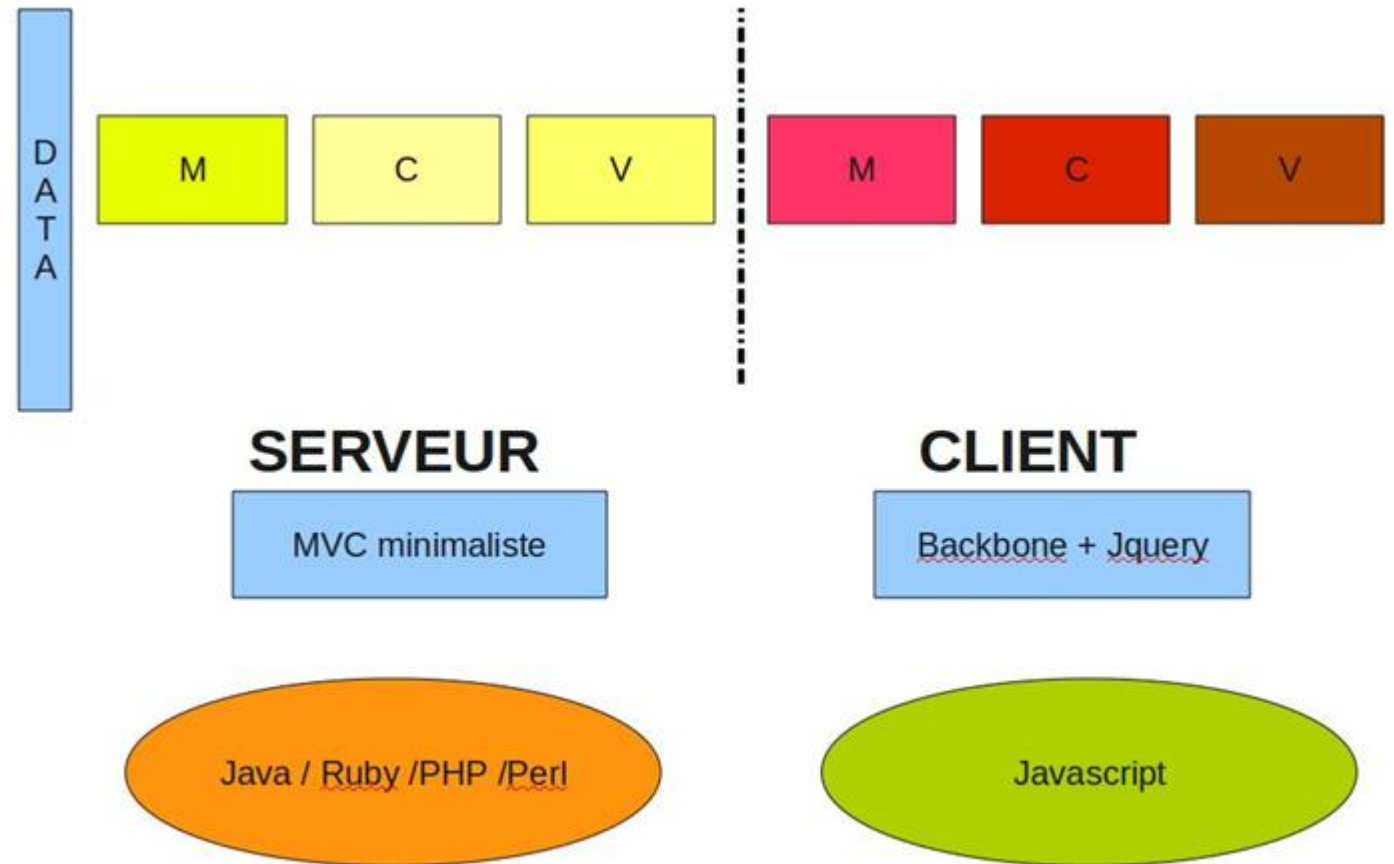
jQuery : écrivez moins
pour faire plus !

- **Deux principaux approches**
 - Manipulation de page via JS
 - Manipulation de page JS + librairie (exemple ., jQuery)



L'apparition des MVC coté client:

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



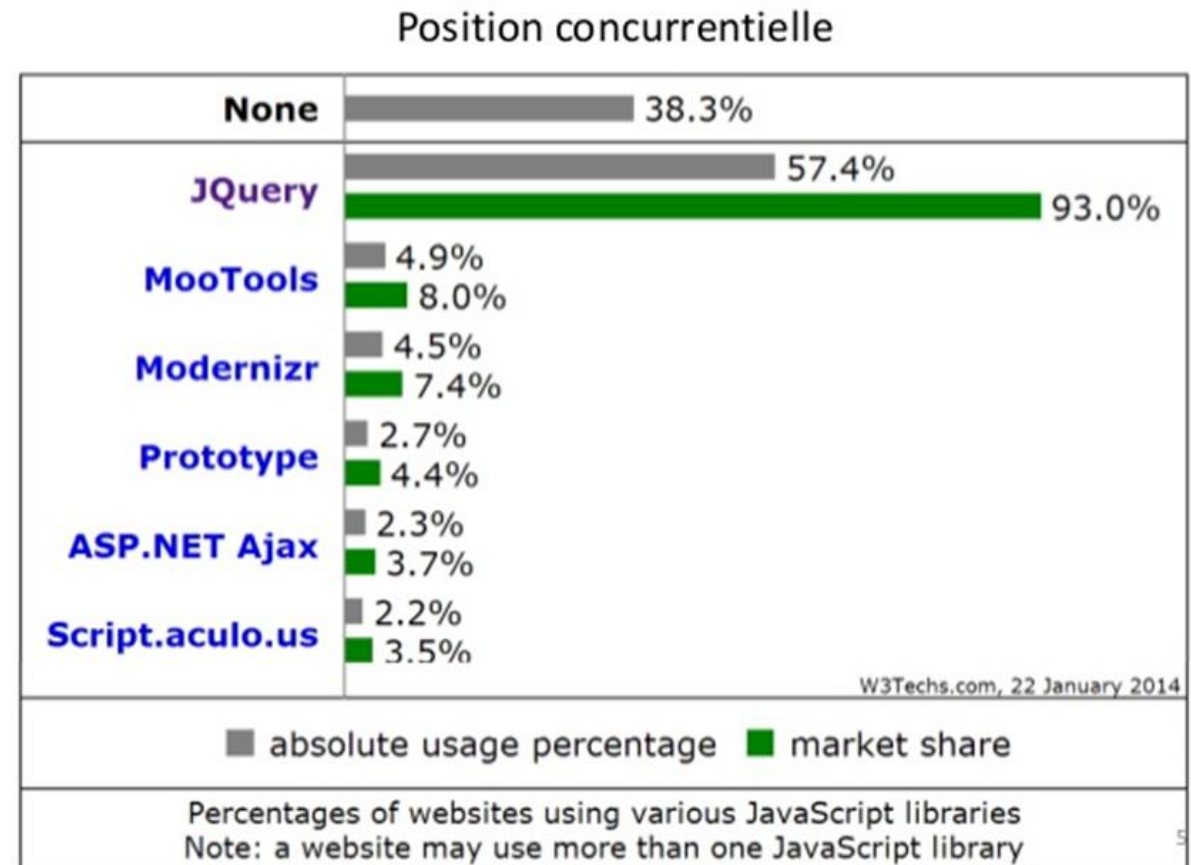
■ Autre Framework

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

Jquery:

Léger, Simple, Apprentissage rapide

- Pas complexe, Taille réduite
- Intégration facile avec divers composants



Pourquoi JQuery !

jQuery est principalement été créé dans le but de :

- manipuler des pages (DOM + CSS) dynamiquement
- prendre en compte des données externes (AJAX)

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 de taille réduite
- Reference : <http://api.jquery.com/>
- Téléchargement dernière version : <http://jquery.com/>

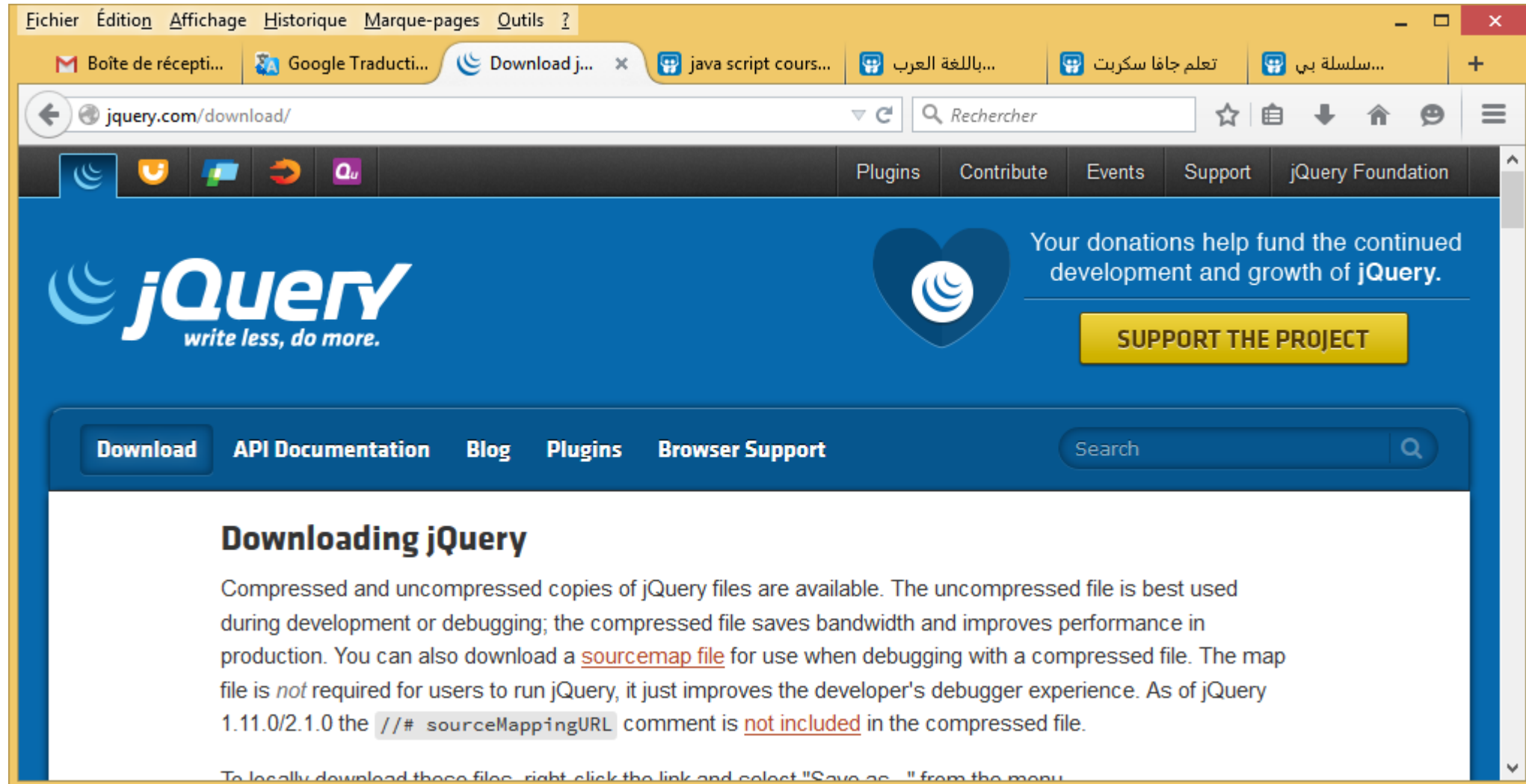
Exemple: code permet de récupérer le nombre en pixels de défilement vertical de la page web (la barre de défilement à droite) :

```
function avoirDefilementVertical() {  
    if (typeof( window.pageYOffset ) == 'number') {  
        // Netscape  
        return window.pageYOffset;
```

```
function avoirDefilementVertical() {  
    return $(document).scrollTop();  
}
```

```
        return document.documentElement.scrollTop;  
    }  
}
```

- Economie de code (faire plus de choses avec moins de code)
- séparation de la structure du traitement
- Compatibilité 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... x java script cours... بالعربية... تعلم جافا سكريبت 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 même répertoire de projet web
- Importer jQuery Dans le head pour pouvoir utiliser ces fonctions
`<script type="text/JavaScript" src=" nom.js"></script>`

- Préférez les serveurs Google
- Google héberge lui aussi les fichiers jQuery :

```
<script type="text/javascript"  
src="http://ajax.googleapis.com/ajax/libs/jquery/1/jquery.min.js"></script>
```

Une autre solution, un peu plus lourde, est d'intégrer les bibliothèques JavaScript et AJAX de Google via <http://www.google.com/jsapi> :

- intégrer les bibliothèques JavaScript suivantes : jQuery, jQueryUI, Prototype, script.aculo.us , MooTools , Dojo, SWFObject , ExtCore...

```
<script type="text/javascript"
src="http://www.google.com/jsapi"></script>
```

```
// Charge la dernière version de jQuery dans 1.xx.
google.load('jquery', '1');
// Charge la dernière version de jQuery dans 1.4x.
google.load('jquery', '1.4');
// Charge la version 1.4.1.
google.load('jquery', '1.4.1');
```


Sélection,
Manipulation et
Création
d'éléments

Fonctions
Utilitaires

Fonction
d'initialisation

Extension avec
les plugins

- **Sélecteurs** : Parcourir le DOM avec les sélecteurs CSS 3 et un support basique de XPath. Filtrer les résultats
- **Elements et attributs** : Changer un éléments ou avoir des informations sur lui
- **Evènements** : Actions utilisateurs ou de modules JS
- **CSS** : Manipuler les styles
- **Animations / Effets** : Changer l'état d'un élément dans l'instant ou sur la durée
- **AJAX** : Requêtes externes à la page
- **Plugins** : Extensions spécifiques au cas par cas
- **Fonctions pratiques** : Détection navigateur, manipulation d'objets JS, etc

Exemples comparatifs entre le JavaScript natif et la bibliothèque jQuery

Programmation Web

```
var element = document.getElementById('elementId'); // Natif
```

```
var $element = $('#elementId'); // jQuery
```

```
var elements = document.getElementsByClassName('elementClass'); // Natif
```

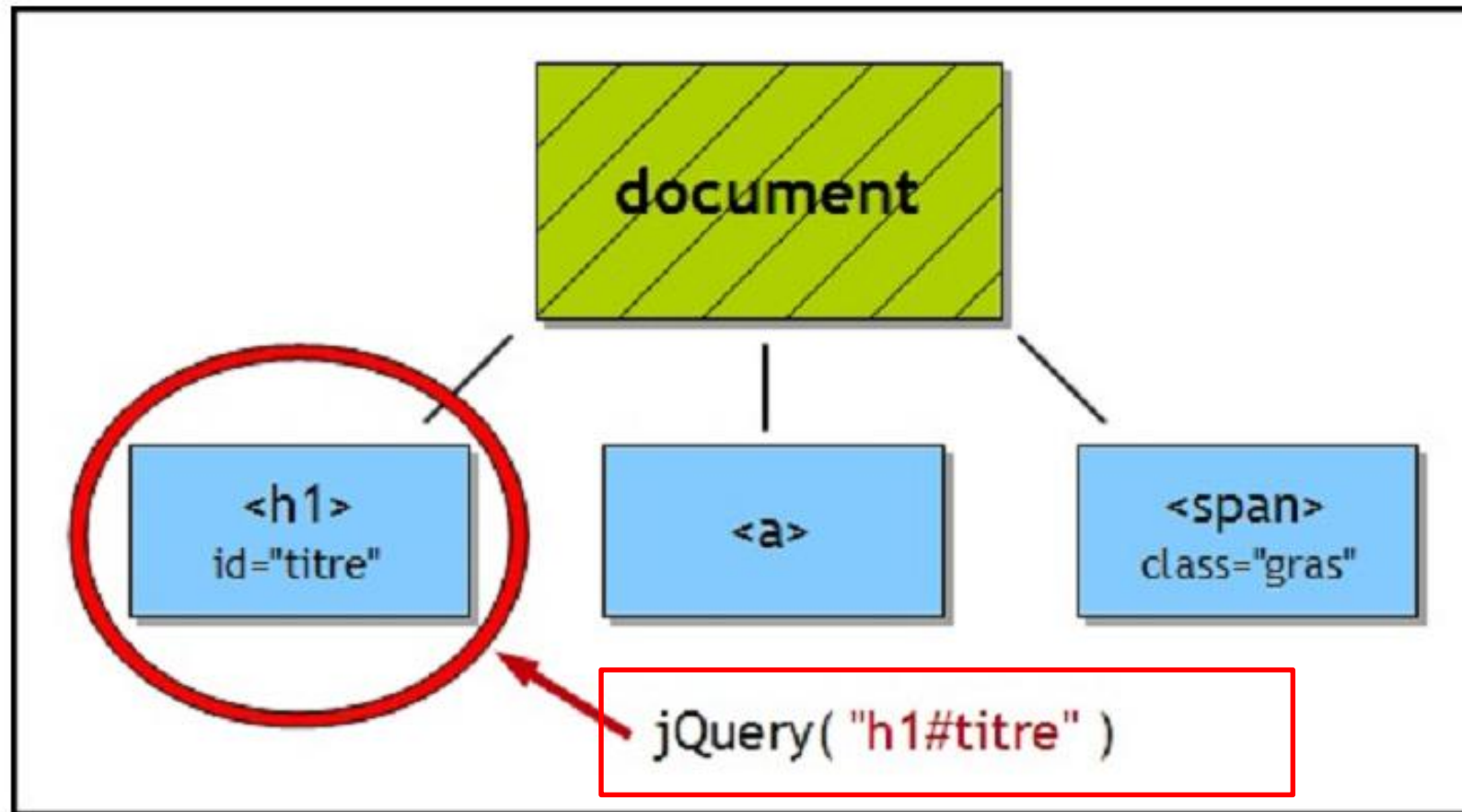
```
var $elements = $('.elementClass');
```

```
document.getElementById("MyElement").onclick = function()  
{ alert("welcome"); }
```

```
$("#MyElement").click( function()  
{ alert("welcome"); } )
```

Exemple de sélecteurs basiques: **CSS**

| Expression | Retour |
|----------------|--------------------------------------|
| #titre | la balise ayant pour id "titre" |
| h1 | les balises h1 |
| .gras | les balises qui ont la classe "gras" |
| a,h1,h2 | les balises a, h1 et h2 |
| * | toutes les balises |



Sélection d'un élément par la fonction principale de jQuery

- Une fois que la fonction retourne t'on objet jQuery, on en fait quoi ?
 - Les méthodes sont des fonctions associées à des objets .
 - Les méthodes jQuerys ont donc des fonctions qui ne s'utilisent que sur des objets jQuery.

Exemple:

La méthode: **html()**

```
$('#titre'); // Sélectionne notre balise mais ne fait rien.  
alert($('#titre').html()); // Affiche le contenu "J'aime les  
frites."  
$('#titre').html('Je mange une pomme'); // Remplace le contenu  
("J'aime les frites.") par "Je mange une pomme".  
$('#titre').html($('#title').html()); // Remplace le contenu par le  
titre de la page (contenu dans la balise <title>).
```

- 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  
    });
```

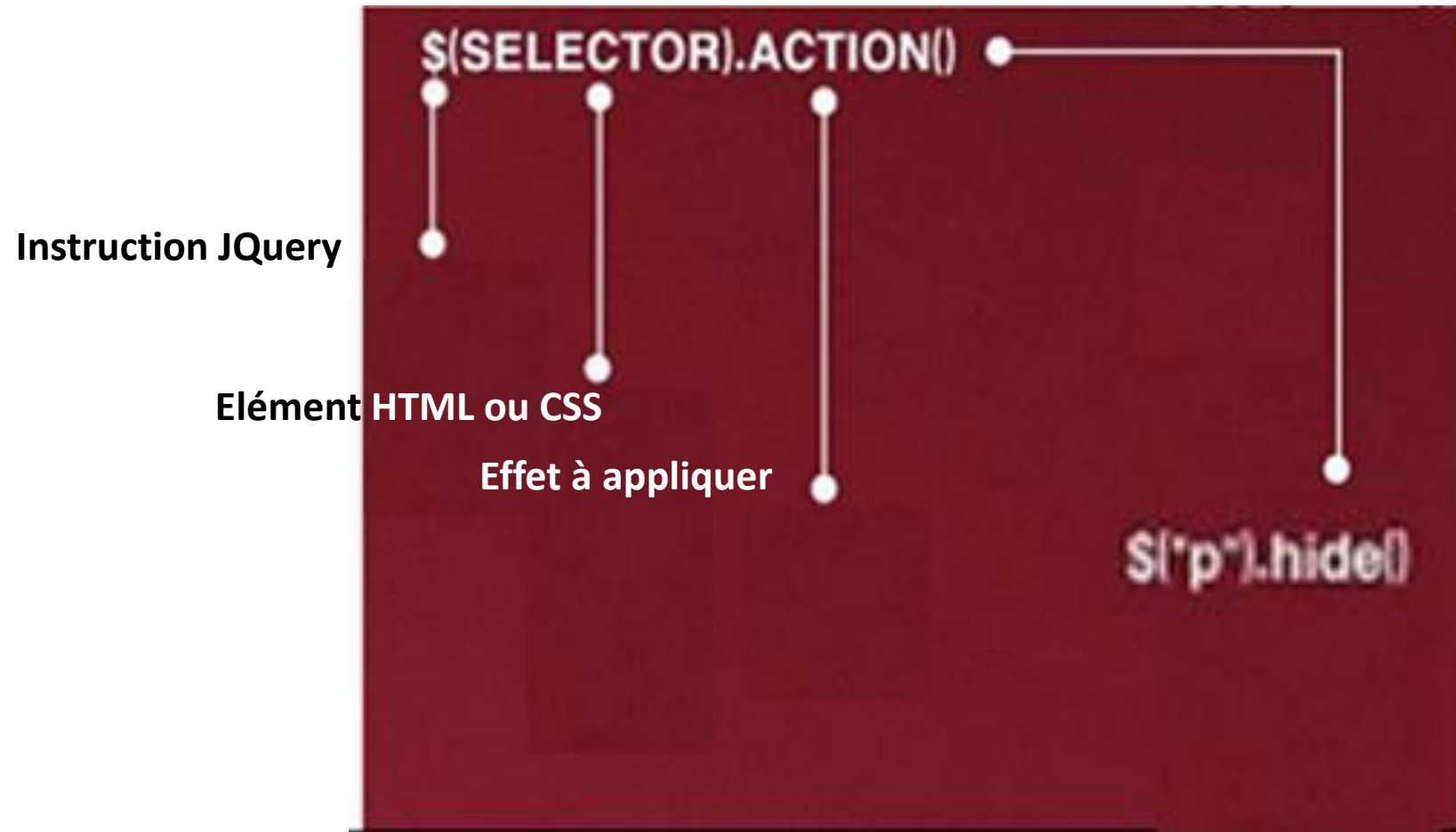
// syntaxe 2

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

// syntaxe 3

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

- Philosophie de jQuery
`$('Trouver du contenu').EnFaireQuelqueChose();`



| Instruction | Rôle |
|--|--|
| <code>\$(s).hide();</code> | Visualiser |
| <code>\$(s).show();</code> | cache |
| <code>var x=\$('#element').size();</code> | Extraire le nombre d'occurrence d'un élément |
| <code>\$(function(){code});</code> | Définir une fonction qui fonctionne aléatoirement dans la page |
| <code>\$('#a#add').click(function(){ \$('added').appendTo('ul'); });</code> | Ajouter un élément a al liste ul quand on clique sur un lien a nommé add |
| <code>\$('#element').remove();</code> | Supprimer un élément de la page |
| <code>\$('#element code').appendTo('part of page');</code> | Ajouter un élément à la page |
| <code>\$('#element').fadeOut(time);</code> | Effet de fermeture |
| <code>\$('#element').fadeIn(time);</code> | Effet d'ouverture |

```
window.onload = function(){
```

```
// Fonctions du genre document.getElementById('balise') qui  
marchent,  
// on peut accéder aux éléments.  
};
```

syntaxe à peu près similaire

La fonction principale jQuery() ou \$()

```
$ (function () {  
  // On peut accéder aux éléments.  
  // $('#balise') marche.  
});
```

Accéder au 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();  
  });  
});
```

Accéder au balise à l'intérieur d'un ... <body>

Utiliser: Class (.) et id (#)

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

Remarque: Les éléments sélectionnés via jQuery sont retourné sous la forme d'un tableau (collection d'éléments ou ensembles enveloppés)

- 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 » |
| div,p | Tous les div et tous les paragraphes |
| li:first | Premier li |
| li:odd | Li pairs |
| li:nth-child(3n+1) | 1 ^{er} , 4 ^{ème} , ... |

Exemples de sélecteurs - Suite

| Sélecteur | Description |
|-------------------------------|---|
| eq(n) | Sélectionne le nème élément (1 ^{er} 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 |
| | |
| | |
| | |
| | |

```
$("#MyDiv").html("myHTML")  
$("body").append("myText")
```

| 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

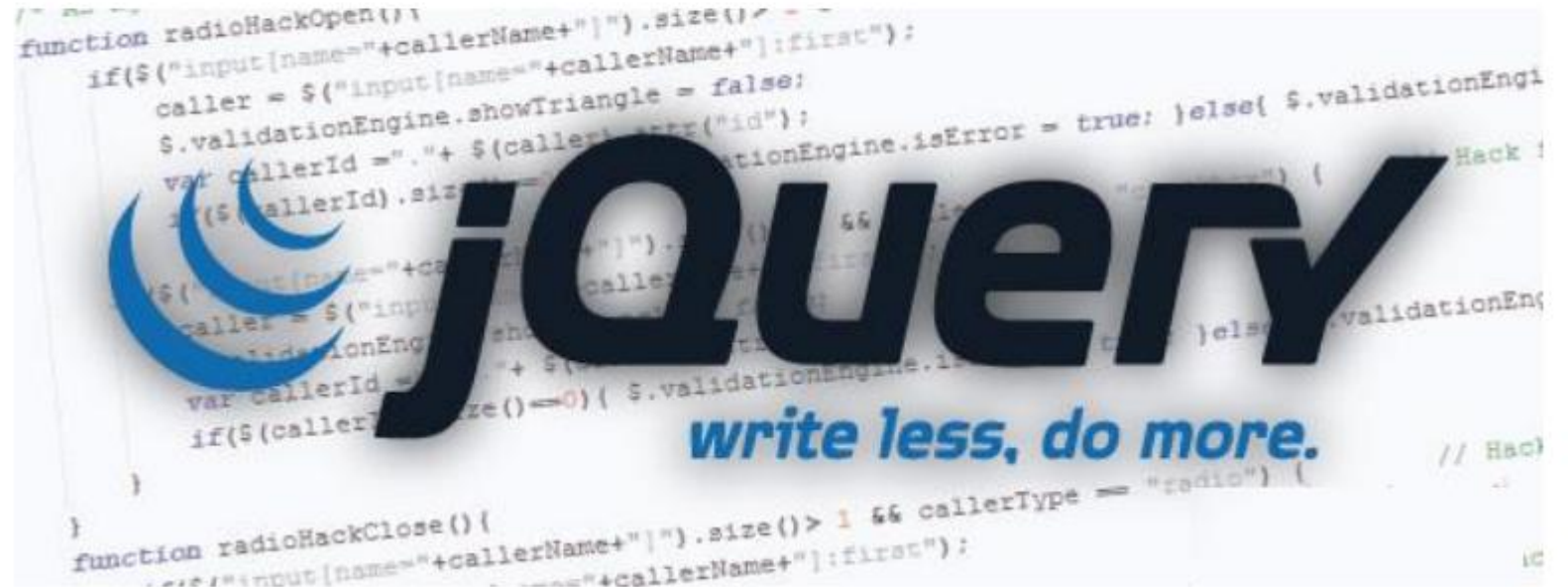
- JQuery permet de simplifier l'affectation des evenements aux elements et elimine tous les problemes d'incompatibilité
- JQuery permet d'affecte un evenement de plusieurs facons et de le desaffecter

| 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 être animées en ajoutant une durée d'exécution
- La fonction « animate » permet de changer des propriétés d'une manière animée
- Il est intéressant 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 Class
- Toggle Effect
-

jQuery Effects



Examples

Cacher les objets dans la page :

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

Propriété : vitesse

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

Même principe que la méthode hide

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


Exécution séquentielle(Chaînage des méthodes)

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

Pour temporiser l'exécution

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

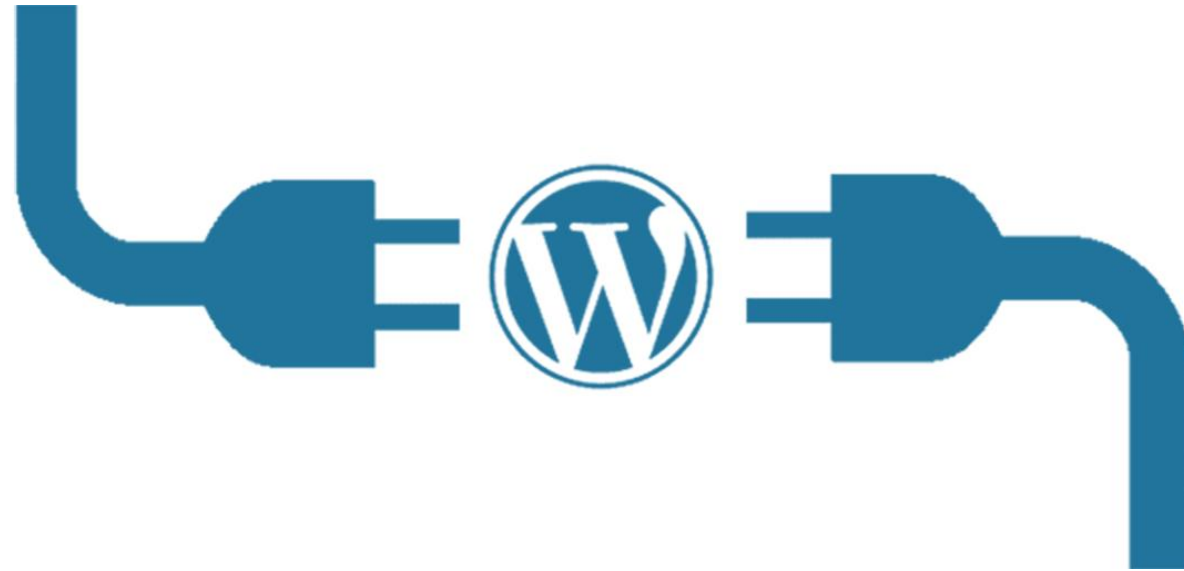
Visualiser un texte par dégradation

On peut déterminer le temps

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

- Au contraire de fadeIn

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



ColorBox

- Slideshow plugin

tablesorter

- Table formatting & row sorting

ListNav

- Filter long lists

That's Just Typical..

1. Download the plugin from its site.
 - Depending on the plugin you can have 1 or more files to install.
2. Copy the plugin, and any of its dependencies to your server.
3. If needed call css `<link href="plugin.css" />`
4. Call jQuery `<script src="jQuery.js">`
5. Call the plugin `<script src="jQuery.pluginname.js">`
6. Initialize plugin `$("#mypluginContainer").pluginInit();`

plugins.jquery.com

Exercice 01 :

Construisez une page html avec une image et un champ texte dans lequel on affiche le nombre de clics sur le bouton.

Exercice 02 A l'aide des fonction JQuery répondez aux questions suivantes :

- 1) Modifiez la vitesse du diaporama.
- 2) Modifiez la vitesse des transitions.
- 3) Modifiez le mode de transition.
- 4) Modifiez le style d'affichage des légendes.

Exercice 03 : Utilisation du plugin Camera slideshow

Objectif de cet exercice : exploiter le plugin Camera slideshow.

Ce plugin est disponible sur le site suivant : www.pixedelic.com/plugins/camera

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)