# JavaScript

## CM6 : jQuery

Mickaël Martin Nevot

V4.2.1

Cette œuvre de Mickaël Martin Nevot est mise à disposition sous licence Creative Commons Attribution - Utilisation non commerciale - Partage dans les mêmes conditions.

BY-NC-SA

# JavaScript


I. Présentation
II. JS
III. Types/opérateurs
IV. Avancé
V. Ajax
VI. DOM
VII. XHR
VIII. JSON
IX. jQuery
X. JS/Web 2.0

 BY-NC-SA

# JQuery

- Bibliothèque JavaScript très répandue : http://jquery.com
- Open source et compatible avec la plupart des navigateurs
- Un seul fichier

Disponible en version « développement » (lisible) ou « production » (compressé : min)

- Utilisation simplifiée de JavaScript/Ajax et du DOM
- Méthode recommandée d'utilisation de jQuery :

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

Appeler un script externe (chez Google) n'est pas plus lent qu'en local !

(cc) BY-NC-SA

# Fonction jQuery

- Une seule fonction : `$()` (version abrégée de `jQuery()`)
- Peut prendre un paramètre
- Retourne un objet
- Sans paramètre : `$.` (version abrégée de `jQuery.`)
- **Chaînage des méthodes** (l'objet appelant est retourné) :

```
$("anything").parent("p.quote").find("still anything").show();
```

Les méthodes s'appliquent généralement à tous les éléments sélectionnés

(cc) BY-NC-SA

# Sélecteurs jQuery

| Expression | Retour |
|---|---|
| `"#id"` | La balise ayant pour id `id` |
| `"h1"` | Les balises `<h1></h1>` |
| `".class"` | Les balises qui ont la classe `class` |
| `"a, h1, h2"` | Les balises `<a></a>`, `<h1></h1>`… |
| `"*"` | **Toutes** les balises |
| `"a[href]"` | Attribut |
| `":parent"` | Sélecteur spécifique |
| `":visible"` | Filtre |

```
$("#id")…
$(".class")…
$("*")…
$(":visible")…
$("a[href]")…
$("h1#titre")…
```

Les combinaisons sont possibles

Sélecteurs spécifiques

Sélecteurs CSS « classiques » (presque tous les sélecteurs CSS sont utilisables en jQuery)

 (cc) BY-NC-SA

# Méthodes jQuery

```
<div class="demo-cont">
    <div class="demo-box">Démo</div>
    <ul class="level-1">
        <li class="item-1">item 1</li>
        <li>item 2</li>
    </ul>
</div>
…
.selected { color:blue; }
```

La plupart des méthodes ont à la fois une utilisation d'accesseur (*getter*) et de mutateur (*setter*)

- Contenu textuel : `html()`, `text()`

```
$("div.demo-cont").html(); // <div class="demo-box">Démo… </ul>
$("div.demo-cont").text(); // Démo item 1 item 2
```

- CSS : `css()`, `addClass()`, `removeClass()`

```
$("div.demo-box").css("background-color");
$("li.item-1").addClass("selected");
```

Style CSS ou DOM !

- DOM : `parent()`, `first()`, `last()`, `next()`, `children()`

```
$("li.item-1").parent().css("background-color", "white");
$("ul.level-1").first().css("background-color", "black");
$("li.item-1").next().css("background-color", "red");
$("ul.level-1").children().css("background-color", "blue");
```

 (cc) BY-NC-SA

# Méthodes jQuery

```
<span>Hello World !</span>
```

- Manipule : `show()`, `hide()`, `wrap()`, `append()`, `prepend()`

```
$("span").show();
$("span").hide();
$("span").wrap("<em></em>"); // <em><span>Hello World !</span></em>.
$("span").append("<em>Test</em>");
// <span>Hello World !<em>Test</em></span>.
$("span").prepend("<em>Test</em>");
// <span><em>Test</em>Hello World !</span>.
```

- Méthode utilisateur (JavaScript, rappel) :

```
function myFunction(i) {
        return i + "ème : ";
}

$("p").prepend(myFunction);
// ?
```

- Évènements : `.event` (évènement DOM)

 (cc) BY-NC-SA

# Méthodes jQuery

- Itération sur tous les éléments d'un objet : `each()`

```
$("img").each(function(i){
    this.alt = "test" + i + ".jpg"; // Ajoute un attribut alt aux images.
});
$.each([0,1,1,2,3,5,8,13,21], function(n){
    $("body").append(n + "ème nombre de Fibonacci : " + this + "<br />");
});
```

- Ajax : `ajax()`

```
$.ajax({
    url:url,
    data:data,
    success:callback
});
```

- JSON :

```
$.ajax({
    url:url,
    dataType:"json",
    data:data,
    success:callback
});
```

L'évènement DOM `load` n'est levé que lorsque le DOM et les images de la page Web sont chargés (ce qui peut être long). Mieux vaut utiliser à la place la méthode jQuery `$(document).ready(fct)` qui appelle la fonction de traitement `fct` dès que le DOM est chargé

 BY-NC-SA

# Moteur Ajax : sans jQuery

```
function createXHR() {
    try { return new XMLHttpRequest(); } catch(e) {}
    try { return new ActiveXObject("Msxml2.XMLHTTP.6.0"); } catch (e) {}
    try { return new ActiveXObject("Msxml2.XMLHTTP.3.0"); } catch (e) {}
    try { return new ActiveXObject("Msxml2.XMLHTTP"); } catch (e) {}
    try { return new ActiveXObject("Microsoft.XMLHTTP"); } catch (e) {}
    alert("XMLHttpRequest non supporté");
    return null;
}
function callback() {
    if (XHRObj.readyState == 4) {
        if (XHRObj.status == 200) {
            alert("Réponse : " + XHRObj.responseText);
        }
    }
}

let param = "p1=foo&p2=bar";
let XHRObj = createXHR();
XHRObj.open("post", "server.php", true);
XHRObj.onreadystatechange = callback;
XHRObj.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
XHRObj.send(param);
```

(cc) BY-NC-SA

# Moteur Ajax : avec jQuery

```
function callback(res) {
    alert("Réponse : " + res);
}

$.ajax({
    type:"POST",
    url:"server.php",
    data:"p1=foo&p2=bar",
    success:callback
});
```

# Crédits

## Auteur

Mickaël Martin Nevot
mmartin.nevot@gmail.com

Carte de visite électronique

## Relecteurs

Cours en ligne sur : www.mickael-martin-nevot.com