

JAVASCRIPT

Etude - Librairie JQUERY

Equipe 3

Blum Frédéric – Leboeuf Lise – Merk Rodolphe – Costes Matthieu

Présentation de la librairie JQuery

jQuery est une bibliothèque JavaScript qui porte sur l'interaction entre JavaScript (comprenant AJAX) et HTML, et a pour but de simplifier des commandes communes de JavaScript. JQuery se caractérise par un ensemble de fonctions qui permettent d'offrir une alternative à la programmation JavaScript de façon uniforme sur les navigateurs les plus courants et permet par exemple de manipuler aisément le DOM, de créer des animations etc... mais surtout de gagner du temps dans le développement des applications : « write less, do more ».

La librairie est sous licence GPL et MIT, et donc complètement réutilisable sur des travaux professionnels. De plus la librairie à l'avantage d'être compatible avec d'autre librairie javascript.

La première version date de janvier 2006 et fut principalement développée par John Resig.

Quelques caractéristiques :

- **C'est une bibliothèque puissante.** Le système jQuery réalise toutes sortes de tâches impressionnantes pour simplifier l'écriture de votre JavaScript.
- **Elle est légère.** Il faut inclure une référence à votre bibliothèque dans chaque fichier qui l'utilise. La bibliothèque jQuery fait 26 ko(dans sa version compressée), une taille inférieure à certains fichiers image. Elle n'a donc aucun impact significatif sur la vitesse de chargement.
- **Elle prend en charge un mécanisme de sélection flexible.** jQuery simplifie et développe le mécanisme *document.getElementById*, essentiel pour la manipulation du DOM.
- **Elle dispose d'un excellent support d'animation.** Vous pouvez utiliser jQuery pour afficher et masquer, déplacer et glisser des éléments.
- **Elle rend les requêtes AJAX évidentes.** Vous allez être surpris par la facilité d'AJAX avec jQuery.
- **Elle possède un mécanisme d'évènement amélioré.** JavaScript dispose d'un support très limité pour les évènements. jQuery offre un outil très puissant pour ajouter un gestionnaire d'évènement à presque tous les éléments.
- **Elle fournit un support multiplateforme.** La bibliothèque jQuery tente de gérer des questions de compatibilité de navigateur pour vous. Ainsi, vous n'avez pas à vous soucier d'éventuels problèmes de navigateurs.
- **Elle prend en charge les composants d'interface utilisateur.** jQuery propose une bibliothèque d'interface utilisateur puissante qui compte des outils que HTML n'a pas, comme les contrôles glisser-déposer, les sliders et les calendriers.
- **Elle est évolutive.** jQuery possède une bibliothèque d'extension qui accepte tous types de fonctionnalités optionnelles, y compris de nouveaux composants et outils tels que l'intégration de son, les galeries d'images, les menus, etc.
- **Elle introduit de nouvelles idées de programmation.** jQuery est l'outil idéal pour découvrir des idées intéressantes comme la programmation fonctionnelle et les objets chaînables.
- **Elle est gratuite et open-source.** jQuery est disponible en licence open-source, ce qui signifie que son utilisation est gratuite et que vous pouvez la consulter et la modifier si vous le souhaitez.
- **Elle est tout de même classique.** Si vous décidez d'utiliser une autre bibliothèque AJAX, vous pourrez y exploiter les enseignements acquis dans jQuery.

I) Installation

jQuery existe sous forme d'un fichier JavaScript contenant toutes les fonctions de base de 179 ko (26 ko dans sa version compressée). Actuellement la version 1.4.4 est disponible ainsi qu'une version 1.5 Beta.

Deux types de fichier peuvent donc être choisis :

jquery-1.4.4.js ou jquery-1.4.4.min.js (version compressée).

```
1  /*!
2  * jQuery JavaScript Library v1.4.4
3  * http://jquery.com/
4  *
5  * Copyright 2010, John Resig
6  * Dual licensed under the MIT or GPL Version 2 licenses.
7  * http://jquery.org/license
8  *
9  * Includes Sizzle.js
10 * http://sizzlejs.com/
11 * Copyright 2010, The Dojo Foundation
12 * Released under the MIT, BSD, and GPL Licenses.
13 *
14 * Date: Thu Nov 11 19:04:53 2010 -0500
15 */
16 (function( window, undefined ) {
17
18 // Use the correct document accordingly with window argument (sandbox)
19 var document = window.document;
20 var jQuery = (function() {
21
22 // Define a local copy of jQuery
23 var jQuery = function( selector, context ) {
24 // The jQuery object is actually just the init constructor 'enhanced'
25 return new jQuery.fn.init( selector, context );
26 },
27
28 // Map over jQuery in case of overwrite
29 _jQuery = window.jQuery,
30
31 // Map over the $ in case of overwrite
32 _$ = window.$,
33
34 // A central reference to the root jQuery(document)
35 rootjQuery,
36
37 // A simple way to check for HTML strings or ID strings
38 // (both of which we optimize for)
39 quickExpr = /^(?:[<]*(<[\w\W]+>)[>]*$|#[\w\-\+]*$)/,
40
41 // Is it a simple selector
42 isSimple = /^[^:#[\.\,]*$/ ,
43
44 // Check if a string has a non-whitespace character in it
45 rnotwhite = /\S/,
46
```

jquery-1.4.4.js

```

1  /!*
2  * jQuery JavaScript Library v1.4.4
3  * http://jquery.com/
4  *
5  * Copyright 2010, John Resig
6  * Dual licensed under the MIT or GPL Version 2 licenses.
7  * http://jquery.org/license
8  *
9  * Includes Sizzle.js
10 * http://sizzlejs.com/
11 * Copyright 2010, The Dojo Foundation
12 * Released under the MIT, BSD, and GPL Licenses.
13 *
14 * Date: Thu Nov 11 19:04:53 2010 -0500
15 */
16 (function(E,B)(function ka(a,b,d){if(d===B$a.nodeType===1)(d=a.getAttribute("data-"+b);if(
typeof d=="string"){try{d=d=="true"?true:d=="false"?false:d=="null"?null:!c.isNaN(d)?
parseFloat(d):Ja.test(d)?c.parseJSON(d):d}catch(e){}c.data(a,b,d)}else d=B}return d}function U
){return false}function ca(){return true}function la(a,b,d){d[0].type=a;return c.event.handle.
apply(b,d)}function Ka(a){var b,d,e,f,h,l,k,o,x,r,A,C=[];f=[];h=c.data(this,this.nodeType?
"events":"_events_");if(typeof h=="function")h=
h.events;if(!(!a.liveFired===this||!h||h.live||a.buttons&&a.type=="click"))(if(a.namespace)A=
RegExp("(^|\\.|"+a.namespace.split("\\.").join("\\.|(?!\\.)*\\.?$)"+"\\.?$)");a.liveFired=this;var J
h.live.slice(0);for(k=0;k<J.length;k++){h=J[k];h.origType.replace(X,"")===a.type?f.push(h.
selector):J.splice(k--,1)}f=c(a.target).closest(f,a.currentTarget);o=0;for(x=f.length;o<x;o++){
=f[o];for(k=0;k<J.length;k++){h=J[k];if(r.selector===h.selector&&(!A||A.test(h.namespace)))
{L=r;elem=e=null;if(h.preType=="mouseenter"){
18 h.preType=="mouseleave"}(a.type=h.preType;e=c(a.relatedTarget).closest(h.selector)[0])if(!e||e
!==l)C.push({elem:l,handleObj:h,level:r.level})}}o=0;for(x=C.length;o<x;o++){f=C[o];if(d&&f.
level>d)break;a.currentTarget=f.elem;a.data=f.handleObj.data;a.handleObj=f.handleObj;A=f.
handleObj.origHandler.apply(f.elem,arguments);if(A===false||a.isPropagationStopped()){d=f.level
if(A===false)b=false;if(a.isImmediatePropagationStopped())break}return b}function Y(a,b){
return(a&&a!="*"?"a+"."+:"")+b.replace(La,
19 " ").replace(Ma,"$")}function ma(a,b,d){if(c.isFunction(b))return c.grep(a,function(f,h){return
!b.call(f,h,f)===d});else if(b.nodeType)return c.grep(a,function(f){return f===b===d});else if
typeof b=="string"){var e=c.grep(a,function(f){return f.nodeType===1});if(Na.test(b))return c.
filter(b,e,d);else b=c.filter(b,e)}return c.grep(a,function(f){return c.inArray(f,b)>=0===d})}
function na(a,b){var d=0;b.each(function(){if(this.nodeName===(a[d]&&a[d].nodeName)){var e=c.
data(a[d++],f=c.data(this,
20 e);if(e&&e.events){delete f.handle;f.events={};for(var h in e)for(var l in e[h])c.event.add(
this,h,e[h][l],e[h][l].data)}}}function oa(a,b){b.src?c.ajax({url:b.src,async:false,dataType:
"script"}):c.globalEval(b.text||b.textContent||b.innerHTML||"");b.parentNode&&b.parentNode.
removeChild(b)}function oa(a,b,d){var e=b=="width"?a.offsetWidth:a.offsetHeight;if(d=="border"
)return e;c.each(b=="width"?Pa:Qa,function(){d||e-=parseFloat(c.css(a,"padding"+this))||0;if
d=="margin")e+=parseFloat(c.css(a,
21 "margin"+this))||0;else e-=parseFloat(c.css(a,"border"+this+"Width"))||0})return e}function da

```

jquery-1.4.4.min.js

Il peut être inclus dans toute page web en plaçant entre les balises <head> et </head> le code suivant :

Il faudra ensuite intégrer le code suivant dans la balise <head>

```
<script type="text/javascript" src="jquery-1.4.4.js"></script>
```

Ou

```
<script type="text/javascript" src="jquery-1.4.4.min.js"></script>
```

Il s'agira ensuite d'appeler le fichier script.js qui contiendra le script d'exécution

```
<script type="text/javascript" src="script.js"></script>
```

II) Principe de programmation

a. La fonction jquery()

La programmation jQuery repose sur l'utilisation d'une seule fonction : `jquery()` et peut être appelée de trois manières différentes :

- via la fonction `$`. Cette fonction est chaînable (elle retourne l'objet appelant). Par exemple :

```
$("#div.test").add("p.quote").addClass("blue").slideDown("slow");
```

- via le préfixe de fonction `$.`; par exemple :

```
$.each([1,2,3], function() {  
    document.write(this + 1);  
});
```

- via le préfixe `jquery`. Par exemple :

```
jquery(document).ready(function() {  
    //Quand le document est chargé on exécute une fonction  
    jquery("#container").load("mapage.php");  
    jquery("#container").fadeOut("slow");  
    jquery("#div_a_effacer").slideUp("fast");  
});
```

Cette fonction accepte des paramètres, et retourne un objet, appelé "objet jQuery".

Il est possible de passer des paramètres tels que :

- Une fonction

Dans ce cas, jQuery va charger cette fonction lorsque la page sera chargée par le navigateur.

Permettant d'éviter le classique `window.onload = function(){}` qui attend que la page et les éléments qu'elle contient soient complètement chargés, ce qui peut être très long.

Processus :

1. Le fichier html est téléchargé.
2. Il est lu par le navigateur.
3. Le navigateur commence à télécharger les éléments auxquels le fichier html fait référence (images, css...).
4. Le code situé dans `$(function(){})` est exécuté dès que possible.
5. Lorsque tout est téléchargé, le code situé dans `window.onload = function(){}` est exécuté, en dernier.

Théoriquement, ce code devrait même être exécuté avant l'affichage, mais en réalité le navigateur peut commencer à afficher la page avant qu'elle ne soit complètement téléchargée.

De plus, les téléchargements ne se font pas un par un, car le navigateur n'attend pas qu'une image arrive pour en télécharger une autre.

Voilà qui facilite le côté non-intrusif du Javascript : Si un élément ne doit apparaître qu'au survol de la souris, par exemple, il faut bien que cet élément soit visible lorsque Javascript n'est pas activé : il peut alors simplement être placé dans le code de la page, et dissimulé en Javascript au chargement, pour réapparaître au survol...

On peut l'utiliser en stockant la fonction dans une variable pour la passer ensuite à jQuery :

```
var ma_fonction = function()
{
// Tout ce qui est ici sera exécuté au chargement du DOM.
};
$(ma_fonction);
```

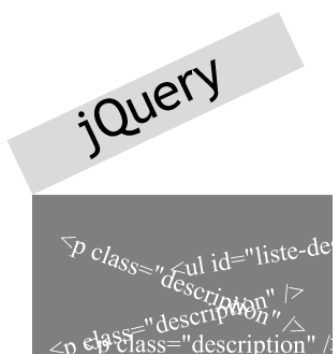
Ou, plus directement, en passant simplement une fonction anonyme à jQuery :

```
$(function()
{
// Tout ce qui est ici sera exécuté au chargement du DOM.
});
```

- Un élément ou un tableau d'éléments DOM

En lui passant des éléments du DOM, la fonction `$()` va nous renvoyer un objet jQuery qui va contenir les éléments qui lui ont été passés, et sur lesquels on va pouvoir user de la puissance de jQuery.

Un peu comme une grosse boîte magique, dans laquelle on va mettre tout ça.



Il est possible de passer un élément, ou un tableau d'éléments à jQuery. Il va sans dire que le code suivant doit être placé dans le « chargeur » jQuery, vu juste au-dessus, pour être exécuté :

```
var mon_element = document.getElementById("id_de_mon_element_html");
var mon_objet_jquery = $(mon_element);
```

- Une chaîne de caractères (query)

En passant une chaîne de caractères à la fonction `$()`, nous allons lui demander de nous retourner un objet jQuery rempli d'éléments, sans même avoir à sélectionner préalablement les objets DOM.

Plusieurs possibilités permettent de sélectionner les éléments désirés : les sélecteurs CSS de niveaux 1 à 3, XPath, ainsi que quelques sélecteurs jQuery.

- Les sélecteurs

jQuery propose deux approches pour sélectionner des éléments. La première utilise une combinaison de sélecteurs CSS et XPath passés comme chaîne de caractères au constructeur jQuery (comme par exemple `$("div > ul a")`). La seconde se sert de différentes méthodes de l'objet jQuery. Les deux approches peuvent être combinées.

- *Les sélecteurs jQuery*

Il existe des sélecteurs spécifiques à jQuery qui ne trouvent pas d'équivalences en CSS. Ils sont toutefois très pratiques. Par exemple :

Sélection des éléments impairs :

```
$("div p:odd");
```

Sélection des éléments inférieurs à (less than) 4 :

```
$("div p:lt(4)");
```

Ou encore la sélection des éléments d'un formulaire (input, select, textarea, button) :

```
$("div p:input");
```

[Documentation sur les sélecteurs jQuery](#)

- *Les sélecteurs CSS*

Prenons cette structure HTML :

```
<div id="fixe">
  <h2>Je suis un titre, j'ai de l'importance.</h2>
  <p class="ique">Ca va les chevilles là-haut ?</p>
  <p>Lorem ipsum sont des mots qui vont très bien ensemble, très bien ensemble.</p>
</div>
```

Pour sélectionner le premier élément <p>.
Voici différentes possibilités :

```
// Sélection des éléments ayant pour classe "ique", dans l'élément ayant pour id "fixe"
var mon_objet_jquery = $("#fixe .ique");

// Sélection du premier élément <p> de l'élément <div>.
var mon_objet_jquery = $("div p:first-child");

// Sélection des éléments <p>, enfants directs d'un élément <div> et ayant pour classe "ique".
var mon_objet_jquery = $("div > p.ique");
```

Le langage CSS offre suffisamment de flexibilité pour permettre une sélection fine et souple.

jQuery va lire cette phrase et la transformer en sélection DOM, mixer le tout et retourner précisément le résultat de la requête dans un objet jQuery !

Exemple : On souhaite sélectionner tous les éléments p ayant pour classe « ique ». On peut alors utiliser la méthode `getElementsByName("p")`, qui va nous retourner un tableau d'éléments p. Nous pourrons ensuite trier ce tableau à l'aide d'une boucle, pour ne garder que ceux ayant pour classe « ique ».

Avec jQuery, une ligne suffit : `$("#p.ique");`

Il est bien sûr possible, comme en CSS, d'utiliser plusieurs sélecteurs à la fois :

```
var mon_objet_jquery = $("#fixe .ique, #fixe :last-child");
```

[Documentation sur les sélecteurs CSS](#)

- *XPath*

jQuery ajoute à cela une implémentation basique de [XPath](#).

Voici comment sélectionner les éléments p enfants de body, lui-même enfant de html :

```
$("#/html/body//p");
```

[Documentation XPath de jQuery](#)

Attention, il est possible de [combiner les différents types de sélecteurs](#) dans une même requête !

b. Les méthodes

Pour utiliser un objet jQuery un nombre important de méthodes sont disponibles dans la librairie pour utiliser les objets jQuery : manipulation du DOM, des CSS, des évènements et autres effets visuels.

Si on désire par exemple masquer progressivement les paragraphes d'une page, on utilise :

```
$("#p").fadeOut();
```

Soit sans passer de paramètre comme cela, soit en précisant une durée en milli-secondes ou encore à l'aide des chaînes `slow`, `fast` et `normal`.

Deux autres exemples :

```
$("#li").html("lorem <strong>ipsum</strong> dolor");  
$("#menu a").click(function()  
{  
  window.open($(this).attr("href"));  
  return false;  
});
```

Le dernier, combine des fonctions JavaScript "classiques" au code jQuery et transforme l'objet courant `this` en objet jQuery avec `$(this)`.

L'ensemble des méthodes et leurs documentations sont disponibles sur [Visual jQuery](#).

c. Callback

Certaines fonctions (comme la fonction `fadeOut()`) acceptent une autre fonction en paramètre. Cette dernière sera exécutée à la fin de l'exécution de la première, appelé un **callback**.

Cela permet par exemple de créer des animations :

```
$(".test").fadeOut("slow",function(){  
  
  $(this).fadeIn("slow");  
  
});
```

d. Chainabilité

Les méthodes jQuery retournant un objet jQuery permet de « chaîner » les fonctions. Ainsi au lieu d'écrire :

```
$(".toto").append(" un texte");  
  
$(".toto").css("border","1px solid red");  
  
$(".toto").addClass("titi");
```

```
$(".toto").removeClass("toto");
```

Il est possible de simplifier par l'exécution du code par :

```
$(".toto").append(" un texte").css("border","1px solid red").addClass("titi").removeClass("toto");
```

e. Utilitaires

Parmi les méthodes jQuery on trouve quelques fonctions très pratiques qui seront employées assez souvent.

length et size

Ces deux techniques sont équivalentes (mais attention à la syntaxe qui diffère). Elles retournent le nombre d'éléments d'un objet jQuery.

C'est très pratique pour tester la présence d'un élément dans la page également.

```
if($("#form").length) { // ... } $("img").size();
```

III) Les plugins

jQuery à l'avantage de proposer de nombreux plugins permettant d'intégrer directement de nombreuses fonctionnalités. Il sera sûrement nécessaire d'adapter quelques parties du code (structure HTML, JS ou encore CSS) par rapport à l'utilisation voulus mais les plugins offrent un gain de temps important pour certaines fonctionnalités avancées.

<http://plugins.jquery.com/>

IV) Exemples d'application potentiels

jQuery offre de nombreuses applications potentiels tel que :

- La transformation des liens d'une page
- Faciliter la géo-localisation
- Révéler le code d'une page
- Mettre en place des slideshow, bandeau d'actualité, des galeries d'images dynamiques
- Charger une image dans un document
- Mettre en place une interface d'édition
- Mettre en place des blocs déroulants

V) Tutoriaux

Il existe de très nombreux tutoriaux pour jquery :

http://docs.jquery.com/Tutorials#Tutoriaux_en_Fran.C3.A7ais

VI) Liens

Liens à visiter :

<http://www.futurecolors.ru/jquery>

<http://www.siteduzero.com/tutoriel-3-160891-jquery-ecrivez-moins-pour-faire-plus.html>

<http://jqueryui.com>

Ressource :

<http://fr.wikipedia.org/wiki/JQuery>

<http://jquery.developpeur-web2.com>

<http://www.jquery.info/spip.php?article21>

<http://visualjquery.com/>

<http://babylon-design.com/apprendre-et-comprendre-jquery-1-3/>

<http://www.lesintegristes.net/2007/06/11/inauguration-et-introduction-a-jquery/>

Référence :

<http://jquery.com>

<http://docs.jquery.com>

<http://www.jquery.info>

<http://babylon-design.com/apprendre-et-comprendre-jquery-1-3/>

<http://jquery.com/>

- documentation de jQuery :

http://docs.jquery.com/Main_Page

- Documentation sélecteurs CSS :

http://docs.jquery.com/DOM/Traversing/Selectors#CSS_Selectors

- Documentation XPath de jQuery :

http://docs.jquery.com/DOM/Traversing/Selectors#XPath_Selectors

- Documentation des sélecteurs propres à jQuery :

http://docs.jquery.com/DOM/Traversing/Selectors#Custom_Selectors

- Combiner les différents types de sélecteurs :

http://docs.jquery.com/DOM/Traversing/Selectors#Using_CSS_and_XPath_Together