

Licence CRRW – IUT de Marne-la-Vallée – Cours de jQuery

- **Contact**

- Courriel : philippe.gambette@gmail.com
- Avant ou après le cours.
- Posez des questions, demandez des exercices d'entrainement !

- **Notes et devoirs**

Travail personnel :

- refaites les exercices de TP mal compris
- pratiquez en expérimentant sur vos projets web

Pendant les cours :

- petits tests de début de cours ou TP
- examen / TP noté lors de la dernière séance

Sources

- Cours de Jean-Loup Guillaume

http://jlguillaume.free.fr/www/documents/teaching/ntw1213/LI385_C5_Jquery.pdf

- Cours de programmation web avancée de Thierry Hamon

<https://perso.limsi.fr/hamon/PWA-20122013/Cours/JQuery.pdf>

- *jQuery : écrivez moins pour faire plus !*, de tit_toinou

<http://openclassrooms.com/courses/jquery-ecrivez-moins-pour-faire-plus>

- *jQuery, Le guide complet*, de Guillaume Allain et Timothy Stubbs

- *Javascript & Ajax pour les nuls*, d'Andy Harris

Introduction



= bibliothèque Javascript
pour simplifier le développement de sites web interactifs

Présenté pour la première fois en janvier 2006 par son créateur John Resig

Première version stable en août 2006

Téléchargeable sur <http://jquery.com/download/>

Avantages

- Puissante
- Légère (94 Ko), avec utilisation possible du fichier sur les serveurs de Google : <https://developers.google.com/speed/libraries/devguide#jquery>
- Multiplateforme : évite les problèmes de compatibilité entre navigateurs
- Gratuite et open source
- Facilite la sélection d'éléments d'une page web
- Facilite l'AJAX
- S'écrit en dehors du code HTML

Principe de jQuery

1. Sélectionner une partie du document.
2. Agir dessus

Objet jQuery = ensemble de noeuds du DOM (Document Object Model)

→ ensemble de balises du document

→ les objets jQuery se créent avec la fonction **jQuery()** abrégée en **\$()** :

- prend en entrée une chaîne de caractères contenant un «sélecteur»
- renvoie en sortie un objet jQuery

Exemple :

1. **\$("div")** renvoie un objet contenant tous les "div" du document.
2. **\$("div").hide()** cache tous les "div" du document.

En pratique

- Insérer le lien vers la bibliothèque jQuery

```
<SCRIPT TYPE="text/javascript"  
SRC="https://ajax.googleapis.com/ajax/libs/jquery/2  
.1.3/jquery.min.js"></script>
```

- Attendre le chargement de la page :

```
<SCRIPT TYPE="text/javascript">  
$ (document) .ready (function () {  
...  
})  
</SCRIPT>
```

- **jQuery ()** s'abrège en **\$ ()**

→ attention aux conflits avec d'autres librairies :

<http://learn.jquery.com/using-jquery-core/avoid-conflicts-other-libraries/>

- **\$ (document) .ready** (function () {...}) s'abrège en **\$ (function () {...})**

Sélection d'éléments

Possibilité de sélectionner :

- par type de bloc
- par identifiant
- par classe
- en combinant les critères
- en filtrant sur les noms d'attributs
- en faisant référence aux positions relatives dans le DOM
- en ne récupérant qu'un seul élément parmi les objets sélectionnés
- en filtrant parmi les objets sélectionnés

Similaire à sélection CSS !

Sélection d'éléments par type de bloc, identifiant, classe

Pour renvoyer toutes les balises :

```
$( "* " )
```

Pour renvoyer tous les <div> de la page :

```
$( "div" )
```

\$("div") .length donne le nombre de div dans la page.

```
// <span id="test">JL</span>
```

```
$( "#test" )
```

```
// <ul class="test">JL</ul>
```

```
$( ".test" )
```


Sélection d'éléments par combinaison de critères

```
// tous les divs de classe main
```

```
$("#div.main")
```

```
// tous les tableaux d'identifiant data
```

```
$("#table#data")
```

```
// objets d'id "content" ou de classe "menu"
```

```
// attention à la position des guillemets
```

```
$("#content, .menu")
```

Sélection d'éléments filtrée

```
// Recherche de p contenant des objets avec classe header  
// pour rendre visibles ces objets  
$("p").find(".header").show();
```

```
// similaire à :  
// $(selecteur, contexte)  
$(".header", $("p")).show();
```

Sélection d'éléments filtrée par numéro d'élément

```
// récupération de tous les éléments + extraction
$("div").get(2)
$("div")[2] // équivalent
// récupération d'un seul élément
$("div").eq(2)
// en partant de la fin
$("div").eq(-2)
```

Sélection d'éléments fondée sur la structure du DOM

Possibilité d'atteindre :

- les fils (>) ;
- tous les descendants (espace) ;
- le (+) ou les (~) frères suivants.

```
<ul>
<li>item 1</li>
<li>item 2</li>
<li class="trois">item 3
<ol><li>3.1</li></ol></li>
<li>item 4
<ol><li>4.1</li></ol></li>
<li>item 5</li>
</ul>
```

```
// cache 4 et 5 :
$('li.trois ~ li').hide();

// cache 4 :
$('li.trois + li').hide();

// cache les <ol> :
$("ul ol").hide();

// ne cache rien :
$("ul > ol").hide();
```

Sélection d'éléments fondée sur la structure du DOM

Possibilité de sélectionner de manière plus précise :

- frère, enfants, parents
- utilisation de fonctions

```
// frère suivant  
.next(expr)
```

```
// frère précédent  
.prev(expr)
```

```
// frères  
.siblings(expr)
```

```
// enfants  
.children(expr)
```

```
// père  
.parent(expr)
```

Autres sélecteurs

```
// premier paragraphe
```

```
p:first
```

```
// dernier élément de liste
```

```
li:last
```

```
// quatrième lien
```

```
a:nth(3) ou a:eq(3)
```

```
// paragraphes pairs ou impairs
```

```
p:even or p:odd every
```

```
// tous les liens à partir (greater than)
```

```
// du quatrième ou avant (lower than)
```

```
a:gt(3) or a:lt(4)
```

```
// liens qui contiennent le mot click
```

```
a:contains('click')
```

Sélecteurs de visibilité

```
// si l'élément est visible  
$("div:visible")  
  
// sinon  
$("div:hidden")
```

Sélecteurs de formulaire

```
// sélectionner les cases à cocher
$("input:checkbox")

// sélectionner les boutons radio
$("input:radio")

// sélectionner les boutons
$("button")

// sélectionner les champs texte
$("input:text")

$("input:checked")

$("input:selected")

$("input:enabled")

$("input:disabled")
```


Sélecteurs de formulaire

```
<select name="valeur">
```

```
<option value="1">1</option>
```

```
<option value="2" selected="selected">2</option>
```

```
<option value="3">3</option>
```

```
</select>
```

```
$("#select option:selected").val()
```

Fonction foreach

Appelle une fonction pour chaque élément sélectionné :

- `$(this)` : élément courant
- `i` : index de l'élément courant

```
$("table tr")  
  .each(function(i) {  
    if (i % 2)  
      $(this).addClass("odd");  
  })  
);
```

Modifier le contenu HTML

[sélecteur].html (' [contenu] ') : remplacement du contenu d'un élément (les balises sont considérées comme des balises)

[sélecteur].text (' [contenu] ') : remplacement du contenu d'un élément en considérant le tout comme du texte (les caractères < et > des balises sont remplacés par les entités XML (> et <))

[sélecteur].after (' [contenu] ') : insertion du contenu après l'élément sélectionné

[sélecteur].before (' [contenu] ') : insertion du contenu avant l'élément sélectionné

[sélecteur].append (' [contenu] ') : insertion du contenu dans l'élément sélectionné à la suite des éléments existants

[sélecteur].prepend (' [contenu] ') : insertion du contenu dans l'élément sélectionné avant les éléments existants

Modifier le contenu HTML

`[sélecteur].wrap ('<balise></balise>')` : insertion des balises passées en paramètre **de part et d'autre** de l'élément

`[sélecteur].wrapInner ('<balise></balise>')` : insertion des balises passées en paramètre **de part et d'autre des enfants** de l'élément

`[sélecteur].unwrap ()` : suppression de la balise parent

Possibilité de combiner les modifications les unes à la suite des autres :
`$("#div").html("Hello jQuery").wrapInner ('')`

Attention à la lisibilité !

Possibilité de récupérer du contenu d'un autre objet pour le passer en paramètre :

```
$("#div.a").html ($("#div.c").html ());
```

→ met le contenu du `div.c` dans le `div.a`

Récupérer ou modifier les propriétés CSS

Récupération de la valeur de l'attribut CSS d'un élément :

[sélecteur].css('color') renvoie la couleur de l'élément

Attribution d'une valeur à l'attribut CSS d'un élément :

[sélecteur].css('color', 'red') attribue la couleur rouge

Attribution d'une valeur à l'attribut CSS des éléments de classe CSS «id» en fonction de leur valeur actuelle à l'aide d'une fonction :

```
var tailleActuelle =  
    parseInt($('.id').css("font-size"));  
$('.id').css("font-size", function() {  
    return tailleActuelle+10;  
});
```

augmente de 10 points la taille de police de caractères des éléments de classe «id»

Attribution d'un ensemble de valeurs à un ensemble d'attributs CSS d'un élément :

[sélecteur].css({'border': '1px solid black', 'color': 'red'}) attribue la couleur rouge à l'élément et lui ajoute une bordure noire.

Récupérer ou modifier la classe CSS

[sélecteur].addClass('cla')

Ajoute la classe CSS `cla` à l'élément.

[sélecteur].removeClass('cla')

Retire la classe CSS `cla` à l'élément.

[sélecteur].toggleClass('cla')

Ajoute la classe CSS `cla` à l'élément s'il ne l'a pas, la lui retire sinon.

[sélecteur].hasClass('cla')

Est vrai si l'élément a la classe CSS `cla`, faux sinon.

→ Renvoie `true` si l'élément a la classe CSS `cla`, `false` sinon.