

# Ressource R209

## Initiation au développement Web

---

### Programmation jQuery



IUT de Béziers, dépt. R&T © 2015-2022

<http://www.borelly.net/>

[Christophe.BORELLY@umontpellier.fr](mailto:Christophe.BORELLY@umontpellier.fr)

# Généralités

---

- **jQuery** est une bibliothèque JavaScript libre et multi-plateforme
- Facilite l'écriture de scripts côté client
- Créée par John RESIG (janvier 2006)
- Taille environ 100 Ko (minified)

# Utilisation

---

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <script src="jquery-x.yy.z.min.js"></script>
    <link rel="stylesheet" href="style.css"/>
  </head>
  <body>
    ...
    <script>
      $(document).ready(function(){
        // Le code jQuery se place ici !
      });
    </script>
  </body>
</html>
```

# Syntaxe

---

- **\$(selecteur)** ou **jQuery(selecteur)** permet d'obtenir des objets jQuery auxquels on peut appliquer des méthodes cumulables (toutes les méthodes renvoient un objet jQuery)
- **\$(this)** est l'objet « courant » (souvent utilisé dans des fonctions)
- **\$(document)** représente le document
- **\$(window)** représente la fenêtre du navigateur

# Sélecteurs (1)

---

- Si le sélecteur est une chaîne, il permet de récupérer les éléments comme en **CSS**
- `$("div")` correspond à tous les tags `div` du document
- `$("#msg")` correspond au tag dont l'attribut `id="msg"` du document
- `$(".val")` correspond à tous les tags dont l'attribut `class="val"`

# Sélecteurs (2)

- `$("*")` tous les éléments
- `$("a[href]")` tous les éléments *a* possédant un attribut href
- `$("[size=20]")` tous les éléments possédant un attribut size="20"
- `$("input:radio")` tous les éléments *input* de type="radio"
- `$("p:first")` Le premier paragraphe
- `$("tr:even")` Toutes les lignes paires
- `$("ul li:first-child")` Le premier enfant *li* de toutes les listes *ul*
- `$("p > span")` Tous les *span* à l'intérieur d'un paragraphe *p*
- `$("p + div")` Tous les *div* suivants un paragraphe *p*
- ...

# Objet(s) sélectionné(s)

---

- La propriété **length** permet de connaître le nombre d'éléments sélectionnés par jQuery (s'il y en a plusieurs)
- On peut aussi utiliser la méthode **size()**
- Pour obtenir un *DOMNode* (**pas un objet jQuery**), on peut soit utiliser la méthode **get()** soit se servir des `[]` :

```
var obj=$("#p");  
console.debug("Nombre de paragraphes : "  
              +obj.length); // $(obj).size()  
var p1=$(obj).get(0); // Premier paragraphe  
var p2=obj[2];        // 3ème paragraphe  
...
```

# Attributs

---

- **attr()** utilisée avec un seul argument permet d'obtenir la valeur de l'attribut du premier élément sélectionné

```
var src1=$("#img1").attr("src");
```

- Avec 2 arguments, cela ajoute/modifie l'attribut

```
$("#img1").attr("title","Titre de l'image");
```

- **removeAttr()** permet de supprimer un attribut

```
$("#img1").removeAttr("title");
```



# Les classes

---

- **addClass()** ajoute une valeur à l'attribut *class*
- **removeClass()** enlève la valeur à l'attribut *class*
- **toggleClass()** ajoute/supprime une valeur à l'attribut *class*
- **hasClass()** détermine si la valeur est présente dans l'attribut *class*

```
$( "div" ). toggleClass( "select" );
```

# Style CSS

---

- **css()** permet de récupérer (ou modifier) les attributs CSS.

```
var color=$( "#div1" ).css( "background-color" );  
$( "p" ).css( { "font-family": "Sans-serif",  
                "color": "red",  
                "background-color": color } );
```

# Valeurs

---

- **text()** renvoi le texte concaténé de tous les éléments et descendants (ou change le texte)
- **html()** renvoi tout le contenu HTML (ou change le contenu)
- **val()** renvoi la valeur de l'attribut *value* (ou modifie la valeur)

```
var user=$("#user").val();  
$("#btn").text("Déconnecter "+user);
```

# Propriétés

---

- Utile pour les éléments sélectionnables (checkbox, radio, select, ...)
  - Avant jQuery 1.6, possible différence entre **attr()** et **prop()**.

```
if ( $( "#chk1" ).prop( "checked" ) ) ...
```

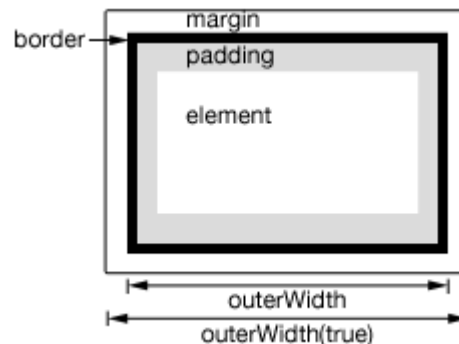
- Autre écriture avec **is()** :

```
if ( $( "#chk1" ).is( ":checked" ) ) ...
```

- Effacement avec **removeProp()**

# Tailles

- **width()** et **height()** renvoient (ou modifient) la largeur ou la hauteur de l'élément
- **innerWidth()** et **innerHeight()** intègrent le padding mais pas border
- **outerWidth()** et **outerHeight()** intègrent le padding et border (margin en option)



# Position

---

- **offset()** renvoie (ou modifie) la position dans le document
  - Le résultat (ou le paramètre) est un objet avec les propriétés **left** (valeur x) et **top** (valeur y)

```
var pos=$("#div").offset();  
$("#div").offset({left:50,top:80});
```

- **position()** renvoie (ou modifie) la position **relativement** au parent

# Modifications

---

- Opération sur les éléments sélectionnés :
  - **after()**, **before()**, **append()**, **prepend()**
  - **replaceWith()**, **wrap()**, **wrapAll()**, **wrapInner()**
  - **clone()** copie l'élément (en option, copie aussi les données et événements jQuery)
  - **remove()**, **detach()** (garde les associations jQuery en vue d'une réinsertion)
  - **empty()** vide le contenu
- ```
$ ( "#msg" ) . append ( "Hello<br/>" );
```

# Opérations inverses

---

- Opération sur les éléments spécifiés en paramètre :
- **insertAfter(), insertBefore(), appendTo(), prependTo()**
- **replaceAll()**

```
$(" <br /> ").appendTo( "#msg" );
```

```
$(" <img /> ").text( "src", "pdf.jpg" )  
                  .insertAfter( "p" );
```

```
$(" <h3 /> ").replaceAll( "li" );
```



# Événements

---

- On peut définir les actions réalisées pour un événement particulier (Voir aussi **on()**) :
- Souris : **click()**, **dblclick()**, **mouseenter()**, **mouseleave()**, ...
- Clavier : **keypress()**, **keydown()**, **keyup()**
- Formulaires : **submit()**, **change()**, **focus()**, **blur()**, ...
- Fenêtre : **load()**, **unload()**, **resize()**, **scroll()**, ...

# Examples

---

```
$("#target").click(cbClick);
$("#target2").click(function() {
    console.log("Click sur "+$(this).text());
});
$("#target3").click(function(){
    cbClick3($(this));
});
...
function cbClick() {
    console.log("Click !");
}
function cbClick3(obj) {
    console.log("Click "+$(obj).text());
}
```

# Objet event

---

- Permet d'obtenir la position de la souris lors de l'événement :

```
$(window).click(function(event) {  
    console.log("Pos. : "+  
                event.pageX+", "+event.pageY);  
});
```

- D'autres propriétés sont accessibles suivant le contexte :
  - altKey, bubbles, button, cancelable, charCode, **clientX**, **clientY**, ctrlKey, currentTarget, data, detail, eventPhase, metaKey, offsetX, offsetY, originalTarget, pageX, pageY, relatedTarget, screenX, screenY, shiftKey, target, view, which, ...

# filtrage

---

- **filter()**, **has()**, **eq()**, **not()**, **slice()** permettent de réduire le nombre d'éléments sélectionnés :

```
var ligne1sur2=$("tr").filter(":even");
```

- On peut sélectionner en fonction de l'indice quand la fonction renvoi « vrai » :

```
var ligne1sur3=$("tr").filter(  
    function(index) {return index%3===0;});
```

# Recherche

---

- **find()** permet de rechercher dans les descendants
- **first()**, **last()**, **next()**, **prev()** renvoi le premier, le dernier, le suivant ou le précédent élément
- **parent()**, **parents()**, **children()**, **siblings()**

```
var objs=$("#div").find("span.code")  
                .next()  
                .children();
```

# Itérations

---

- **each()** permet d'appliquer une fonction à chaque élément sélectionné :

```
$( "div" ).each( function( ) {  
    console.log( "ID : " + $( this ).attr( "id" ) );  
} );
```

- Possibilité d'utiliser 2 arguments (this==element) :

```
$( "div" ).each( function( index, element ) {  
    . . .  
} );
```

# Animations

---

- **hide()**, **show()** et **toggle()** cache ou affiche les éléments sélectionnés
- **slideUp()**, **slideDown()**, **slideToggle()** idem avec un effet de glissement
- **fadeIn()**, **fadeOut()**, **fadeTo()**, **fadeToggle()** idem avec un effet de transparence
- **animate()** permet de modifier les propriétés CSS (possibilité de +=, -=, etc)

# Références

---

- <https://jquery.com/>
- <http://api.jquery.com/>
- <http://www.w3schools.com/jquery/>
- <http://www.w3schools.com/jquery/tryse1.asp>