Chapitre 5 : JQuery

Introduction

- C'est une bibliothèque JavaScript open-source et cross-browser
- Manipulation de tout le DOM des pages web
  - Changer/ajouter une classe CSS, Créer des animations,
  - Modifier des attributs, …
- Gestion des événements JavaScript
- Téléchargeable sur http://jquery.com/download/
- Démo des objets : https://jqueryui.com/demos/
- Créée en janvier 2006 par John Resig
- Autre : AngularJS

Principe

1. Sélectionner une partie du document
2. Agir sur cette partie du document
   - Objet jQuery = ensemble de noeuds du DOM (Document Object Model)
     - ensemble de balises du document
     - se crée avec la fonction jQuery() abrégée en $(),
       - Une chaîne de caractères contenant un «sélecteur» en entrée
       - Un objet jQuery en retour

- Exemple :
  - $("div") renvoie un objet contenant tous les "div" du document
  - $("div").hide() cache tous les "div" du document

Syntaxe

- Insertion du script, lien vers la bibliothèque jQuery

```
<script src="jquery-3.3.1.js"></script> <!-- meme rep.-->
```

- Insertion du script, lien vers la bibliothèque jQuery ONLINE : cf. https://code.jquery.com

```
<script src="https://code.jquery.com/jquery-3.3.1.js"
integrity="sha256-2Kok7MbOyxpgUVvAk/HJ2jigOSYS2auK4Pfzbm7uH60="
crossorigin="anonymous"></script>
```

- Attendre le chargement de la page :

```
<script>
$(document).ready(function(){…<!-- manipulation des objets jQuery-->
})</script>
```

Exemple

```
<!doctype html>
<html><head>
    <meta charset="utf-8">
    <title>Demo</title>
</head>
<body>
    <a href="http://jquery.com/">jQuery</a>
    <script src="jquery-3.3.1.js"></script>
    <script>
    $( document ).ready(function() {
        $( "a" ).click(function( event ) {
            alert( "The link will no longer take you to jquery.com" );
            event.preventDefault();
        }); //$(document).ready(function(){…}) s'abrège aussi en $(function(){…})
    });
    </script></body></html>
```

Attention

- Conflit possible avec d'autre bibliothèques : http://learn.jquery.com/using-jquery-core/avoid-conflicts-other-libraries/
  - "That said, there is one caveat: by default, jQuery uses $ as a shortcut for jQuery. Thus, if you are using another JavaScript library that uses the $ variable, you can run into conflicts with jQuery. In order to avoid these conflicts, you need to put jQuery in no-conflict mode immediately after it is loaded onto the page and before you attempt to use jQuery in your page."

Exemple

```
<!-- Putting jQuery into no-conflict mode. -->
<script src="prototype.js"></script>
<script src="jquery.js"></script>
<script>
var $j = jQuery.noConflict();
// $j is now an alias to the jQuery function; creating the new alias is optional.
$j(document).ready(function() {
    $j( "div" ).hide();
});
// The $ variable now has the prototype meaning, which is a shortcut for
// document.getElementById(). mainDiv below is a DOM element, not a jQuery object.
window.onload = function() {
    var mainDiv = $( "main" );
}
</script>
```

Les sélecteurs

- http://api.jquery.com/category/selectors/
- Sélection possible
  - 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

Sélection par type de bloc

- Type: syntaxe : $("nom_balise")
  - Pour renvoyer toutes les balises :

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

  - Pour renvoyer toutes les balises de la page :

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

- Pour renvoyer le nombre de balises dans la page : .length

  ```
  $("div").length
  ```

Sélection par identifiant, classe

- Identifiant : syntaxe : $("#nom_id")

```
// <span id="test">Exemple JQ</span>
$("#test")
```

- Classe : syntaxe : $(".nom_classe")

```
// <ul class="test"> Exemple JQ</ul>
$(".test")
```

Sélection par combinaison de critères

- Exemples :
  - Toutes les balises div de classe main

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

  - Tous les tableaux d'identifiant tablo

    ```
    $("table#tablo")
    ```

  - Les objets avec id "content" ou de la classe "menu"

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

Sélection filtrée

- Exemple : Rechercher les paragraphes <p> contenant des objets possédant la classe header et rendre ces objets visibles

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

Sélection filtrée par numéro d'élément

- Retour d'un élément (index) + extraction : `$("selecteur").get(index)`
  - Exemple :

```
$("div").get(2) // attention 3e élément
$("div")[2] // équivalent
```

- Retour d'un jQuery (index) + extraction : `$("selecteur").eq(index)`
  - Exemple :

```
$("div").eq(2)
```

- Rmq : Si index<0, on part de la fin

Sélection avec la structure du DOM

- Possibilité d'atteindre :
  - les fils (>)
  - tous les descendants (espace)
  - le (+) ou les (~) frères suivants

Sélection avec la structure du DOM

- Exemple :

```
<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>
```

```
// cacher 4 et 5
$('li.trois ~ li').hide();
// cacher 4
$('li.trois + li').hide();

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

// ne cache rien
$("ul > ol").hide();
// pour cacher les ol ici
//=> $(ul>li>ol).hide();
```

Sélection avec 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 élément : `selecteur:first`
  - Ex. paragraphe `p:first`
- Dernier élément : `selecteur:last`
  - Ex. `li:last`
- Nième élément : `selecteur:nth(numero)` ou `selecteur:eq(numero)`
  - Ex. quatrième lien `a:nth(3)` ou `a:eq(3)`
- Les éléments pairs ou impairs : `selecteur:even` ou `selecteur:odd` `every`
  - Ex. `p:even` ou `p:odd  every`

Autres sélecteurs

- Tous les sélecteurs après (greater than) le ième ou avant (lower than) : `selecteur:gt(numero)` `selecteur:lt(numero)`
  - Ex. `a:gt(3)` `a:lt(4)`
- Tous les sélecteurs qui contiennent le mot « mot » `selecteur:contains('mot')`
  - Ex. Liens qui contiennent le mot click `a:contains('click')`

Sélecteurs de visibilité

- Montrer un élément : `$("selecteur:visible")`
  - Ex. `$("div:visible")`
- Cacher un élément : `$("selecteur:hidden")`
  - Ex. `$("div:hidden")`

Sélecteurs de formulaire

- Sélectionner les cases à cocher

```
$("input:checkbox")
```

- Sélectionner les boutons radio

```
$("input:radio")
```

- Sélectionner les boutons

```
$(":button")
```

Sélecteurs de formulaire

- Sélectionner les champs texte suivant le contexte éventuel

```
$(":text")
$("input:checked")
$("input:selected")
$("input:enabled")
$("input:disabled")
```

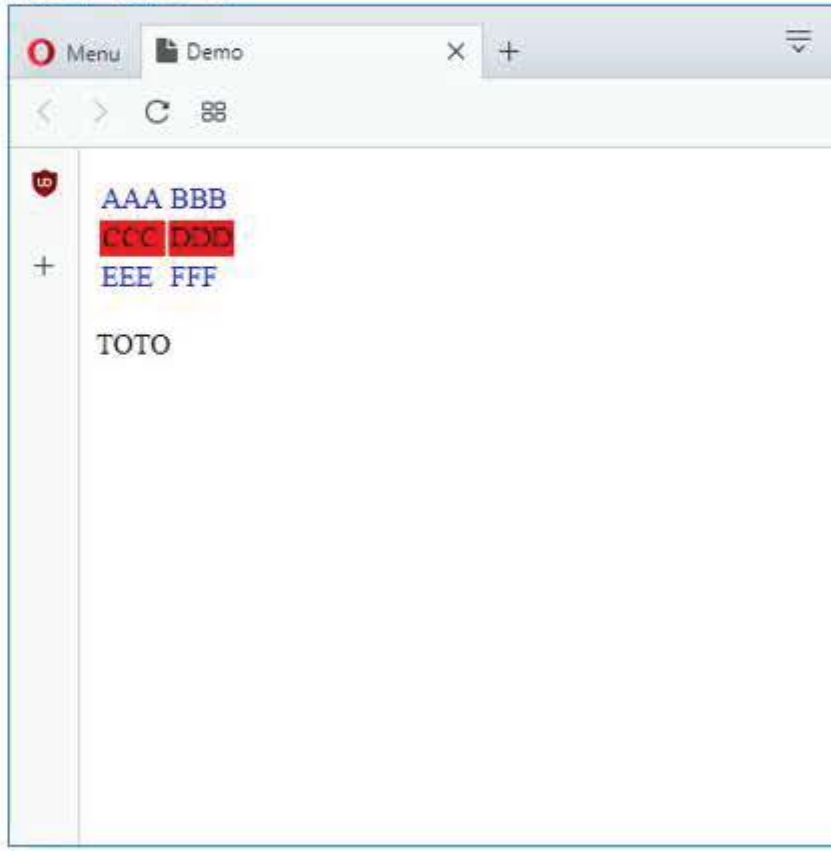
Exemple

```
<select name="valeur">
<option value="1">1</option>
<option value="2" selected="selected">2</option>
<option value="3">3</option>
</select>
<script>alert($("select option:selected").val())</script>
<!- ici affiche…2 dans la boite alert-->
```

Fonction for each

- Appelle une fonction pour chaque élément sélectionné :
  - `$(this)` : élément courant
  - `i` : index de l'élément courant
- Exemple :

```
<table>     <tr> <td>AAA</td> <td>BBB</td> </tr>
            <tr> <td>CCC</td> <td>DDD</td> </tr>
            <tr> <td>EEE</td><td>FFF</td></tr>
</table>
<p>TOTO</p>
<script>$( "table tr" ).each(function(i){  if (i % 2) {
(this).style.backgroundColor="red";} //en fond rouge
else
{(this).style.color="blue";} //en bleu
});</script>
```



Modifier le contenu HTML

- `.html('[contenu]')` : remplacement du contenu d'un élément
- `.text('[contenu]')` : remplacement du contenu d'un élément en considérant comme du texte (les caractères < et > des balises sont remplacés par > et <
- `.after('[contenu]')` : insertion du contenu après l'élément sélectionné
- `.before('[contenu]')` : insertion du contenu avant l'élément sélectionné

Modifier le contenu HTML

- `.append('[contenu]')` : insertion du contenu dans l'élément sélectionné à la suite des éléments existants
- `.prepend('[contenu]')` : insertion du contenu dans l'élément sélectionné avant les éléments existants

Modifier le contenu HTML

- `.wrap('<balise></balise>')` : insertion des balises passées en paramètre de part et d'autre de l'élément
- `.wrapInner('<balise></balise>')` : insertion des balises passées en paramètre de part et d'autre des enfants de l'élément
- `.unwrap()` : suppression de la balise parent
- Possibilité de combiner les modifications les unes à la suite des autres :
  - Ex. `$("div").html("Hello jQuery").wrapInner('<b></b>')`

Modifier le contenu HTML

- Possibilité de récupérer du contenu d'un autre objet pour le passer en paramètre :
  - Exemple : `$("div.1").html($("div.2").html());`
    `// Met le contenu de div.2 dans le div.1`

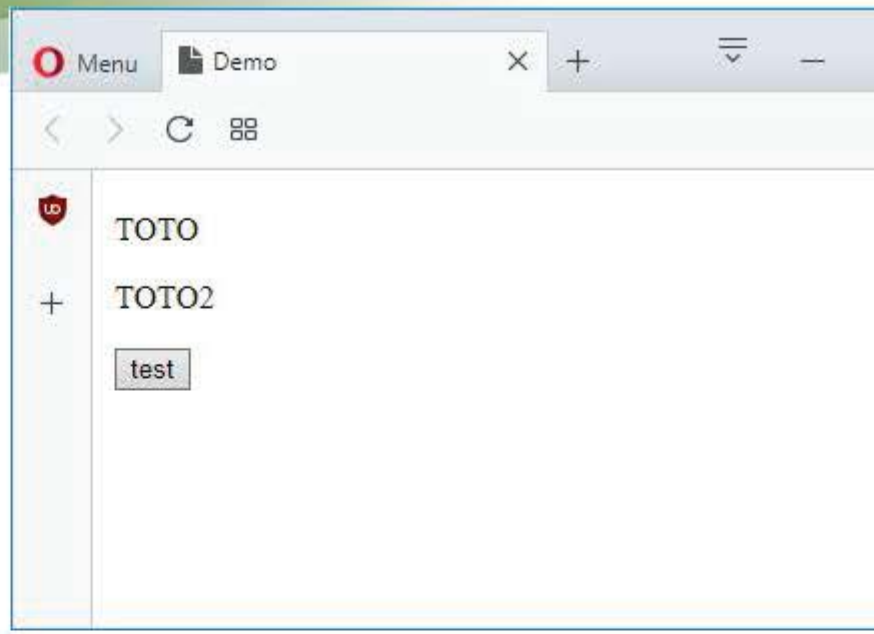
Récupérer ou modifier les propriétés CSS

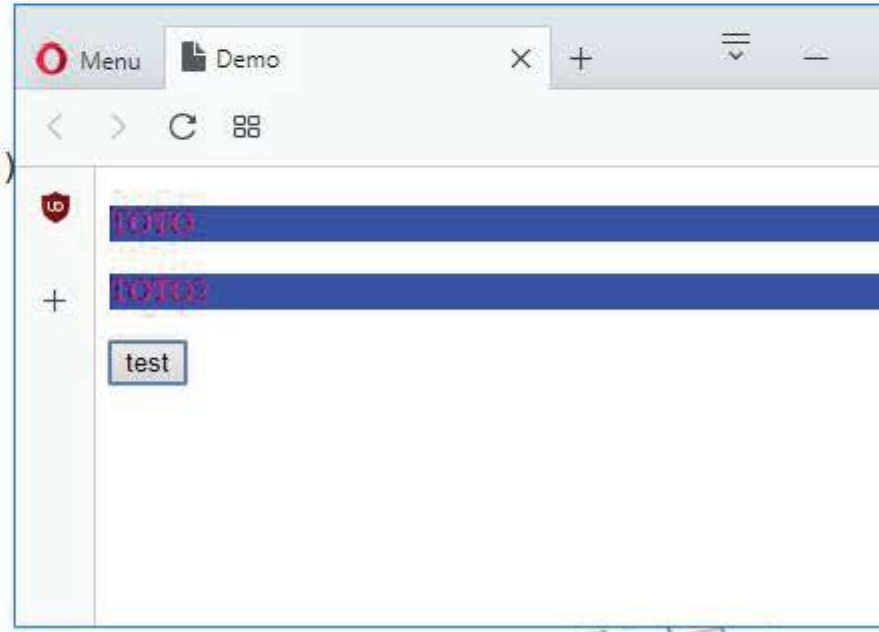
- Récupération de la valeur de l'attribut CSS d'un élément :
  `.css('propriété')`
  - Ex. `.css('color')` renvoie la couleur de l'élément
- Attribution d'une valeur à l'attribut CSS d'un élément :
  `.css('propriété', 'valeur')`
  - Ex. `.css('color','red')` attribue la couleur rouge à l'élément
- Attribution de plusieurs propriétés CSS d'un élément :
  `.css({'propriété1' : 'valeur1', 'propriété2' : 'valeur2' …})`

Exemple

- Exemple :

```
<head>
…
<title>Demo</title>
<script> function test()
{
//$("#test").css('color', 'red');
$("p").css({'color': 'red', 'background-color' : 'blue'})
}
</script></head>
<body>
<p id="test">TOTO</p>
<p id="test2">TOTO2</p>
<form>
   <input type="button" value="test" onClick="test()">
</form></body></html>
```





Récupérer ou modifier les propriétés CSS

- Modifier la valeur d'un attribut CSS pour les éléments de type « id» en fonction de leur valeur actuelle à l'aide d'une fonction
  - Exemple : modification de la taille avec +10 par rapport taille actuelle

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

Récupérer ou modifier la classe CSS

- Attribue la classe CSS laClasse à l'élément

```
.addClass('laClasse')
```

- Supprime la classe CSS laClasse à l'élément

```
.removeClass('laClasse')
```

- Attribue la classe CSS laClasse à l'élément qui ne la possède pas ou sinon lui retire

```
.toggleClass('laClasse')
```

- Retourne l'appartenance à la classe CSS laClasse (true ou false)

```
.hasClass('laClasse')
```

jQuery UI

- Un ensemble de composants graphiques http://jqueryui.com/download :
  - un noyau (Core)
  - des « comportements » (interactions)
    - draggable : pour glisser-déplacer un élément : http://jqueryui.com/demos/draggable/
    - droppable : pour « déposer » un élément : http://jqueryui.com/demos/droppable/
    - resizable : pour redimensionner un élément : http://jqueryui.com/demos/resizable/
    - selectable : pour sélectionner des éléments à la souris : http://jqueryui.com/demos/selectable/
    - sortable : pour trier des éléments : http://jqueryui.com/demos/selectable/

jQuery UI

- des « widgets »
  - « accordéon » : http://jqueryui.com/demos/accordion/
  - « calendrier » : http://jqueryui.com/demos/datepicker/
  - boîte de dialogue : http://jqueryui.com/demos/dialog/
  - barre de progression : http://jqueryui.com/demos/progressbar/
  - curseur : http://jqueryui.com/demos/slider/
  - onglets : http://jqueryui.com/demos/tabs/
- des « effets »
  - http://jqueryui.com/demos/effect/
  - Clignotement, disparition, apparition, éclatement, transition…
- Une collection de thèmes : http://jqueryui.com/themeroller/