

Cours 3

Bibliothèque Javascript JQuery

Yacine Bouzidi

Yacine.bouzidi@inria.fr

Plan

- JQuery
 - Introduction
 - **Getter/setter**
 - JQuery et événements
 - JQuery et Ajax

JQuery : introduction

JQuery : qu'est ce que c'est ?

- Bibliothèque Javascript libre et open-source
- Simplifie les interactions DOM/HTML/CSS/Javascript



- Utilisée dans la plupart des applications à interfaces riches
- Développée par **John Resig** (auteur également du plugin CCK)

Ressources

- Téléchargeable sur le site officiel : <http://jquery.com/>
- Documentation et tutoriels :
 - <http://docs.jquery.com/Tutorials>
 - <http://api.jquery.com/>
 - <http://speckyboy.com/2010/03/31/20jquerycheatsheets-docsandreferencesforeveryoccasion/>
- **Refcard** disponible sur le moodle (consultez la !)

Utilisation

- Téléchargez et décompressez le fichier .js
 - Placez le fichier **JQuery.js** dans votre repertoire racine du projet
- Chargez la librairie puis commencez à coder !

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Le titre du document</title>
    <link rel="stylesheet" type="text/css">
  </head>
  <body>
    <!-- code HTML pour structurer le contenu du document -->
    <script src="jquery.js"></script>
    <script src="autre_script.js"></script>
  </body>
</html>
```

- Pour optimiser le chargement, il est préférable de faire appel aux scripts **à la fin du document**

Sélection d'éléments

jQuery est basée sur une seule et unique fonction `jQuery()` ou son alias `$()`.

Cette fonction peut prendre comme paramètres :

- Une fonction, exécutée dès que le DOM est disponible

```
$(function() {  
    // Insérer le code jQuery ici  
});
```

- Un sélecteur CSS ou une balise HTML

```
$('a'); // sélectionne tous les liens hypertextes  
$('#id'); // sélectionne l'élément d'identifiant id  
$('.important'); // sélectionne tous les éléments de classe important
```

Elle retourne un **objet jQuery** assimilable à un **tableau** contenant tous les éléments sélectionnés

Objet JQuery

Cet objet dispose de très nombreuses propriétés et méthodes (**getters et setters**)

- Pour accéder au tableau ainsi qu'à ses propriétés

```
$('#a').length // Retourne le nombre de lien dans le document  
$('#a')[3] // Retourne, si il existe, le quatrième lien du document
```

- Appliquer une méthode à la sélection

```
$('#elem').action() // Applique action() à tous les éléments elem  
$('#a').css('color') // retourne la couleur des liens hypertextes
```

- Une liste exhaustive des méthodes est disponible sur <http://api.jquery.com>

Sélecteurs : types

- Sélecteurs **css/html**

```
$('div') // tous les éléments correspondants aux balises div  
$('.class') // tous les éléments de classe 'class'
```

- Sélecteurs par **attributs**

```
$('[nom]') // éléments ayant un attribut 'nom'  
$('[nom:val]') // éléments ayant un attribut 'nom' de valeur val
```

- Sélecteurs **hiérarchiques**

```
$('parent > enfant') // éléments enfant descendants des éléments parents  
$('enfant1 > enfant2') // éléments enfant2 précédés d'un élément enfant1  
$('li:first-child') // premier élément descendant de l'élément li
```

Sélecteurs : types (suite)

- Pseudo sélecteurs

```
$('#elem:even') // retourne les éléments elem d'indice pair  
$('#elem:gt(3)') // retourne les éléments elem d'indice supérieur à 3  
...
```

- Sélecteurs spécifiques

```
$('#elem:hidden') // éléments elem qui sont cachés  
$('#:visible') // tous les éléments visibles  
...
```

- Sélecteurs formulaires

```
$('#:input') // tous les éléments de type input, textarea, select et button  
$('#:submit') // tous les éléments de type submit  
$('#:checkbox') // tous éléments de type checkbox  
...
```

Parcourir la sélection

Le résultat d'une sélection est donné sous la forme d'un **tableau** qu'il est possible de parcourir à l'aide de la méthode `each()`

```
$('#elem').each(function(index){  
    //Une ou plusieurs instructions JavaScript  
});
```

```
<body>  
  <h1>Introduction</h1>  
  <h1>Corps du texte</h1>  
  <h1>Conclusion</h1>  
  
  <script src="jquery.js"></script>  
  <script>  
    $(function() {  
      $('h1').each(function(index){  
        this.text('Titre'+ (index+1));  
      });  
    });  
  </script>  
</body>
```

Conversion JQuery/DOM

Attention : un objet JQuery n'est pas un objet Javascript DOM !

Les deux objets `o1,o2` suivants **ne sont pas** de même type

```
var o1 = $('#id');  
var o2 = document.getElementById(id)
```

Besoin de méthodes de **conversion** JQuery/DOM

- Javascript DOM --> JQuery

```
var varJS = 'Voici un text';  
var varJQ = $(varJS);
```

- JQuery --> Javascript DOM

```
var aJS = $('a').get();  
aJS[0].href
```

JQuery : Getters et Setters

Accès et modification des éléments

- Accès/modification des **propriétés CSS** avec la méthode `css()`

```
var couleur = $('#titre').css('color');  
$('#titre').css('color', 'red');
```

- Accès/modification des **attributs** avec la méthode `attr()`

```
$('#image').attr('src');  
$('#image').attr('src', 'logo.gif');  
$('#image').attr({src: 'logo.gif', alt: 'Logo de la société'});  
$("a").attr({target: function(){...}});
```

- Accès/modification des **champs de formulaires** avec la méthode `val()`

```
$('#champ').val();  
$('#date').val('Today')
```

- Accès/modification du **contenu des balises** avec méthodes `text()`, `html()`

```
$('#p').text();  
$('#p').text('ceci est le nouveau paragraphe');
```

Modification du DOM avec JQuery

JQuery permet également d'interagir avec le DOM, en particulier

- rajouter/supprimer **des attributs, des classes...**

```
addClass() //ajoute une classe dans les éléments sélectionnés ;  
removeClass() //supprime une classe des éléments sélectionnés ;  
toggleClass() //ajoute la classe si elle n'existe pas et la supprime sinon  
removeAttr() //supprime un attribut des balises le contenant  
...
```

- rajouter/remplacer/supprimer **des éléments du DOM**

```
remove(); // supprime un élément du DOM  
replaceWith(); // remplace l'élément sélectionné par un autre  
append(); // insère du contenu à la fin de la sélection  
prepend(); // insère du contenu au début de la sélection  
before(); // insère du contenu avant la sélection  
...
```

Effets

- Cacher ou rendre visibles des éléments

```
show(speed,callback) // Affiche progressivement un/des éléments  
hide(speed,callback) // Dissimule progressivement un/des éléments  
toggle(speed,callback) // Inverser une propriété
```

- Animer des éléments

```
animate(params,option) // Permet de définir une animation personnalisée  
scrollTop(val)  
...
```


Associer des données aux éléments

On peut **stocker** des données textuelles dans des éléments du DOM avec la méthode `data()`

```
$.data(el, 'nom', nom_don: don);  
$.data('#id', 'Var_id', donnee1: valeur1);  
var v = $.data('#id', 'Var_id').donnee1;
```

Méthode utile par exemple lors de **l'édition** d'un élément pour stocker dans celui-ci son état initial et le récupérer a posteriori:

```
var nextElmt = $("<textarea>" + contenuInitial +  
"</textarea>").data("initValue",contenuInitial);  
$(this).replaceWith(nextElmt);  
  
$("textarea").each(function (){  
    // $(this) Le textarea courant !  
    var initValue = $(this).data("initValue");  
    $(this).replaceWith('<p class="editable">' + initValue + "</p>");  
});
```

JQuery et événements

JQuery : événements

- Méthode principale :

```
$('#elem').event(function() {  
    // gestion de l'événement lorsqu'il se produit  
})
```

```
<script src="jquery.js"></script>  
<script>  
    $(function() {  
        $('h1').mousemove(function() {  
            alert('passage sur la balise <h1>');  
        });  
        $(window).scroll(function() {  
            alert('Utilisation de la roulette dans le document');  
        });  
    });  
</script>
```

- Permet de purifier le code HTML !

JQuery : événements (suite)

- Événements de la souris

```
click() // Clic gauche  
dblclick() // Double-clic  
mousedown() // Appui sur le bouton gauche ou droit de la souris  
mouseenter() ou mouseover() // Début de survol de l'élément  
mouseleave() ou mouseout() // Arrêt de survol de l'élément  
mousemove() // Déplacement du pointeur au-dessus de l'élément  
mouseup() // Relâchement du bouton gauche ou droit  
scroll() //Utilisation de la roulette
```

- Événements du clavier

```
keydown() // Appui sur une touche du clavier  
keyup() // Relâchement d'une touche du clavier préalablement enfoncée  
keypress() // Maintien d'une touche du clavier enfoncée
```

- Événement éléments

```
focus() // Réception de focus par l'élément  
blur() // Perte de focus par l'élément  
change() // Modification d'un élément  
load(), unload() // Chargement complet de la page ou passage à une autre
```

Evenements (suite)

Les deux instructions suivantes sont equivalentes

- `$(elem).click(function() {...})`
- `$(elem).on('click', function() {...})`

La méthode `on()` permet la gestion de **plusieurs** événements

```
// méthodes standard
$('img').mouseenter(function() { ... })
$('img').mousemove(function() { ... })
// méthode on()
$('img').on('mouseenter mousemove', function() { ... })
```

Possibilité de création d'événements **personnalisés**

```
$('#element').on('monEvenement', function() {
  alert('Ceci est mon propre evenement !');
});
// declenchement de l'evenement
$('#element').trigger('monEvenement');
```


JQuery et Ajax

JQuery et Ajax

- Ajax est beaucoup plus **simple** à utiliser avec JQuery !
- Utilisation de la méthode `load()`

```
$('#elem').load('page_cible', function() {  
    //une ou plusieurs instructions  
});
```

- L'élément `elem` est **mis à jour** avec le résultat de `page_cible`
- Une fois la mise à jour faite, la fonction de rappel est exécutée

Méthode load() : exemple

```
<p id="par">Eh ! Dans ce monde y a deux type d'hommes :  
ceux qui sont armée et ceux qui creuse.</p>  
<button id="maj">Mise à jour du paragraphe</button>  
<script src="jquery.js"></script>  
<script>  
  $(function() {  
    $('#maj').click(function() {  
      $('#par').load('page_cible.html', function() {  
        alert('Le paragraphe a été mise à jour');  
      });  
    });  
  });  
</script>
```

Fichier page_cible

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="UTF-8">  
  </head>  
  <body>  
    Il essaie de faire de l'humour, écoutez ça c'est pathétique  
  </body>  
</html>
```

Chargement d'une partie de la cible

- Il est possible d'obtenir **une partie seulement** de page cible
- Faire suivre le nom du fichier cible de l'élément que l'on souhaite obtenir

```
$('#elem').load('page_cible elem2', function() {  
    //une ou plusieurs instructions  
});
```

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="UTF-8">  
  </head>  
  <body>  
    <p id="par1">Il essaie de faire de l'humour, c'est pathétique</p>  
    <p id="par2">On se lève tous pour la lumière</p>  
  </body>  
</html>
```

```
$('##par').load('page_cible.html par2', function() {  
    alert('Le paragraphe a été mise à jour');  
});
```

Passage de paramètres à la cible

Utilisation de la méthode `load()`

- Première possibilité : émuler GET()

```
$('#elem').load('login.php','nom=' + $('#nom').val());
```

- Seconde possibilité : émuler POST()

```
$('#elem').load('login.php',{nom:'yacine',password: 'serieux !'});
```

Utilisation des méthodes JQuery `$.get()` et `$.post()`

```
// méthode get()  
$.get(url, données, function() {  
    // Une ou plusieurs instructions  
});  
  
// méthode post()  
$.post(url, données, function() {  
    // Une ou plusieurs instructions  
});
```

Chargement de script Javascript

- JQuery permet également de charger de manière asynchrone un script .js ainsi que des donnée JSON
- JSON (**JavaScript Object Notation**) : format de données textuelle adapté au Javascript

```
{  
  "nom": "M Bison",  
  "age": "inconnu",  
  "ville": "Thaïlande",  
  "Pouvoir": "electricité"  
}
```

- Cela se fait avec les méthodes `getscript()` et `getJSON()`

```
$.getScript('url', function() {  
  // Une ou plusieurs instructions  
});  
  
$.getJSON('fichier.json', function(donnees) {  
  // Une ou plusieurs instructions  
});
```