


Programmation WEB

Introduction à jQuery

Denis Monnerat

monnerat@u-pec.fr 

IUT de Fontainebleau



Sommaire

1. Introduction
2. Sélection
3. Obtention/Modification
4. Méthodes jQuery et valeurs renvoyées
5. jQuery et le dom
6. Gestion des événements
7. Support d'Ajax
8. Effets/animations
9. Plugins

1. Introduction

2. Sélection

3. Obtention/Modification

4. Méthodes jQuery et valeurs renvoyées

5. jQuery et le dom

6. Gestion des événements

7. Support d'Ajax

8. Effets/animations

9. Plugins

jQuery, kesaco ?

jQuery est une bibliothèque, un toolkit (une surcouche de l'api dom) javascript.

jQuery

- permet de parcourir et/ou modifier un document html facilement.
- comprend un support pour la gestion des événements.
- peut rajouter des effets/animations sur les pages.
- facilite l'utilisation d'ajax.

La documentation se trouve à l'url : <http://jquery.com/> 




Un exemple de requête Ajax


```
$.get("http://www.arda/ville.php").done(  
    function (data){  
        console.log(data);  
    });
```


Un très bon guide pour débiter <http://jqfundamentals.com/> 



- Surcouche de jQuery d'éléments d'interface (widgets)
- <http://queryui.com> 



- Surcouche de jQuery d'éléments d'interface (widgets) adaptée aux média tactiles
- <http://querymobile.com> 

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

Versions

- Dernière version jQuery.1.12.
- Il existe une branche 2.x : même api, sans le support de I.E 6,7 et 8.

Utilisation : 2 solutions

- Téléchargement des sources (compressées ou non).
- Utilisation du cdn de jQuery.

```
<script src="//code.jquery.com/jquery-1.11.2.min.js"></script>
```

Sommaire

1. Introduction

2. Sélection

3. Obtention/Modification

4. Méthodes jQuery et valeurs renvoyées

5. jQuery et le dom

6. Gestion des événements

7. Support d'Ajax

8. Effets/animations

9. Plugins

La sélection d'éléments

La sélection se fait de manière analogue aux règles css, à l'aide de la fonction `jQuery`, ou encore l'alias `$()`.

```
jQuery("div")
```

```
jQuery("p #intro > h2")
```

```
$("p #intro > h2 + h3")
```

```
$('input[name*="man"]')
```

- Il existe d'autres possibilités de selections, avec des filtres notamment.
- On peut restreindre la sélection à un contexte.

```
// create a jQuery object from a DOM element  
$( document.body.children[0] );  
  
// create a jQuery object from a list of DOM elements  
$( [ window, document ] );  
  
// make a selection in the context of a DOM element  
var firstBodyChild = document.body.children[0];  
$( 'li', firstBodyChild );  
  
// make a selection within a previous selection  
var paragraph = $( 'p' );  
$( 'a', paragraph );
```



La selection retourne une **collection** jquery. On peut parcourir l'ensemble des éléments DOM de la collection :

"À la main"

```
var divs=$("#div");  
var nb_div=divs.length;  
var div1=divs[0]; /* premier element DOM div */  
var div2=divs[1]; /* etc ... */
```

Avec la méthode each

```
$('#li').each(function(index,el) {  
    alert(index + ' : ' + $(el).text());  
    /* on peut utiliser this également */  
});
```

Sommaire

1. Introduction

2. Sélection

3. Obtention/Modification

3.1 html/text

3.2 valeurs des éléments de
formulaire

3.3 attribut

3.4 classes et css

4. Méthodes jQuery et valeurs
renvoyées

5. jQuery et le dom

6. Gestion des événements

7. Support d'Ajax

8. Effets/animations

9. Plugins

- `html()` : obtient (ou fixe) le contenu html du premier élément d'une collection :

```
$('#div#message').html('<p>Youpi</p>');
```

- `text()` : obtient le texte de chaque élément (et leur descendants) d'une collection.

```
<div class="demo-container">
  <div class="demo-box">Demonstration Box</div>
  <ul>
    <li>list item 1</li>
    <li>list <strong>item</strong> 2</li>
  </ul>
</div>
```

`$('#div.demo-container').text()` renverra la chaîne :
Demonstration Box list item 1 list item 2

Méthode `val()`

Getter

```
// Get the value from a dropdown select  
$( "select.foo option:selected" ).val();  
// // Get the value from a dropdown select even easier  
$( "select.foo" ).val();  
// // Get the value from a checked checkbox  
$( "input:checkbox:checked" ).val();  
// // Get the value from a set of radio buttons  
$( "input:radio[name=bar]:checked" ).val();
```

Méthode `val(string| [] |function)`

Getter

```
$("#input:text").val("coucou");
```

```
$( "input" ).on( "blur", function() {  
    $( this ).val(function( i, value ) {  
        return value.toUpperCase();  
    });  
});
```

Méthode attr().

```
$('#greatphoto').attr('title', 'Photo by Kelly Clark');  
$('#greatphoto').attr({  
  alt: 'Beijing Brush Seller',  
  title: 'photo by Kelly Clark'  
});  
$('#titre').removeAttr('id');
```


Classes et propriétés css

```
$("#p:last").addClass("selected highlight");  
$("#intro").removeClass("red");  
$("#intro").toggleClass("green");  
$("#p").css("color", "#ff0000");  
$("#p").css({"color": "#ff0000", "font-size": "1em"});
```

Sommaire

1. Introduction

2. Sélection

3. Obtention/Modification

4. Méthodes jQuery et valeurs renvoyées

5. jQuery et le dom

6. Gestion des événements

7. Support d'Ajax

8. Effets/animations

9. Plugins



`$()` renvoie une collection jquery. (La plupart) des méthodes s'appliquent sur tous les objets de la collection. On peut restreindre le résultat avec la méthode `first()`.



Certaines méthodes ne s'appliquent qu'au premier objet de la collection. Exemple :

Méthode s'appliquant sur le premier objet

```
var h = $('div#titre').height();  
var src = $('img.photo').attr('src');  
var p = ('$p:last').html();  
var email = $('input#email').val();
```

Il est (souvent) possible d'enchaîner les méthodes (elles retournent souvent la même collection).

```
$('div.titre').addClass('selected').show();
```

Sommaire

1. Introduction

2. Sélection

3. Obtention/Modification

4. Méthodes jQuery et valeurs renvoyées

5. jQuery et le dom

5.1 Parcours

5.2 Insertion

5.3 Remplacement, Effacement

6. Gestion des événements

7. Support d'Ajax

8. Effets/animations

9. Plugins

Il existe de nombreuses méthodes de parcours :

```
parent, next, prev, nextAll, parents, sibling, children,  
closest, find
```

Méthode closest

```
var input=$('#compte input[type=text]');  
input.each(function(i,e){  
  if ($(e).val() == ""){  
    $(e).closest('.control-group').addClass('error');  
  } else  
    $(e).closest('.control-group').removeClass('error');  
});
```

Insertion à l'intérieur

```
$('.inner').append('<p>Test</p>');  
$("p").append(document.createTextNode("Hello"));  
$('.container').append($('h2')); /* ajoute un element existant  
('<p>Test</p>').appendTo('.inner');  
$('.container').prepend($('h2'));  
('<p>Test</p>').prependTo('.inner');
```

Insertion à l'extérieur

```
$('.inner').after('<p>Test</p>');  
$('.container').before($('h2'));  
('<p>Test</p>').insertAfter('.inner');  
$("p").insertBefore("#foo");
```

Remplacement

```
<div class="container">  
  <div class="inner first">Hello</div>  
  <div class="inner second">And</div>  
</div>
```

après `$('#<h2>New heading</h2>').replaceAll('.inner');`

```
<div class="container">  
  <h2>New heading</h2>  
  <h2>New heading</h2>  
</div>
```

```
$('#div.second').replaceWith('<h2>New heading</h2>');
```


Destruction

```
$('.hello').empty(); /* enleve tous les fils */  
$('.hello').remove(); /* enleve le noeud et tous ses fils */  
var p = $("p").detach(); /* idem remove, mais l'objet jquery  
                           est sauvegardé et peut être  
                           réutilisé */
```

Sommaire

1. Introduction

2. Sélection

3. Obtention/Modification

4. Méthodes jQuery et valeurs renvoyées

5. jQuery et le dom

6. Gestion des événements

7. Support d'Ajax

8. Effets/animations

9. Plugins

Fonctions reflexes dédiés à un type d'événements

```
$('#a:first').click(function(ev){
    $(this).css("color","red");
    return false; // ou ev.preventDefault();
});
$("p").click(function () {
    $(this).slideUp();
});
$(document).ready(function(){alert("le dom est prêt");});
$(function(){alert("le dom est prêt");});
```

Méthode générique on()

```
$("#dataTable tbody tr").on("click", function(event){
    alert($(this).text());
});
$("#dataTable tbody").on("click", "tr", function(event){
    alert($(this).text());
});
```

Sommaire

1. Introduction

2. Sélection

3. Obtention/Modification

4. Méthodes jQuery et valeurs renvoyées

5. jQuery et le dom

6. Gestion des événements

7. Support d'Ajax

8. Effets/animations

9. Plugins

Chargement d'un fichier

Méthode load

```
/* on charge directement le contenu  
 * depuis un fichier */  
$('#result').load('ajax/test.html');
```

Requête get.

Méthode get

```
$.get("test.cgi", { name: "John", time: "2pm" },  
  function(data){  
    alert("Data Loaded: " + data);  
  });  
/* on peut utilisergetJSON si le resultat est a format json */
```

Requête post

Méthode post

```
$.post("test.php", $("#testform").serialize());
```

Les données sont automatiquement encodées en données de formulaires.

```
$.post( "test.php", { name: "John", time: "2pm" } );
```

```
$.post( "test.php", { 'choices[]': [ "Jon", "Susan" ] } );
```

Méthode générique ajax

Méthode ajax

```
$.ajax({  
  type: "POST",  
  url: "some.php",  
  data: "name=John&location=Boston",  
  success: function(msg){  
    alert( "Data Saved: " + msg );  
  }  
});  
  
/* beaucoup d'autres paramètres  
 * et réglages possibles  
 */
```


Promesses

```
$.ajax( "example.php" )  
  .done(function() { alert("success"); })  
  .fail(function() { alert("error"); })  
  .always(function() { alert("complete"); });
```

La méthode ajax retourne une promesse :

```
$.ajax({url: "/ServerResource.txt"}).then(  
  [successFunction1, successFunction2, successFunction3],  
  [errorFunction1, errorFunction2]);
```

Chaînage

`then` permet de modifier le résultat d'une promesse et de le passer à un autre callback (chaînage).

```
$.ajax("https://randomuser.me/api")  
  .then(function(data){  
    return data.results[0];  
  })  
  .then(function(user){  
    console.log(user);  
  })
```

Rendez-vous

```
var p1 = $.ajax("https://randomuser.me/api")
  .then(function(data){
    return data.results[0];
  })
var p2 = $.ajax("https://randomuser.me/api")
  .then(function(data){
    return data.results[0];
  })
$.when(p1,p2).done(function(user1,user2){
  console.log(user1);
  console.log(user2);
})
```

Sommaire

1. Introduction

2. Sélection

3. Obtention/Modification

4. Méthodes jQuery et valeurs renvoyées

5. jQuery et le dom

6. Gestion des événements

7. Support d'Ajax

8. Effets/animations

9. Plugins

Effets particuliers ...

```
$('#h1').hide('slow');  
$('#h1').slideDown('fast');  
$('#h1').fadeOut(1000);
```

que l'on peut enchaîner.

```
$('#h1').fadeIn(1000).slideUp();
```

Méthode générale animate

```
$("#go").click(function(){
  $("#block").animate({
    width: "+=70px",
    opacity: 0.4,
    marginLeft: "0.6in",
    fontSize: "3em",
    borderWidth: "10px"
  }, 1500 );
});
```

Sommaire

1. Introduction

2. Sélection

3. Obtention/Modification

4. Méthodes jQuery et valeurs renvoyées

5. jQuery et le dom

6. Gestion des événements

7. Support d'Ajax

8. Effets/animations

9. Plugins

Plugins

- Il en existe des centaines !
- On peut facilement en écrire soi-même :

```
$.fn.hideLinks = function()  
{  
  this.find('a[href]').hide();  
  return this;  
}  
$('p').hideLinks();
```