

JQuery et AJAX

Mussab Zneika

CY Cergy Paris Université

- ❑ jQuery est une bibliothèque JavaScript légère, "écrivez moins, faites plus".
- ❑ L'objectif de jQuery est de faciliter l'utilisation de JavaScript sur votre site web.
- ❑ Encapsuler les tâches communes qui nécessitent de nombreuses lignes de code JavaScript pour accomplir dans list des méthodes que on peut appeler avec une seule ligne de code.
- ❑ La bibliothèque jQuery contient les fonctionnalités suivantes:
 - ✓ Manipulation HTML / DOM
 - ✓ Manipulation de CSS
 - ✓ Méthodes d'événement HTML

Principe de jQuery

1. Selectionner une partie du document.

`$()` :

- prend en entree une chaine de caracteres contenant un `<<selecteur>>`
- renvoie en sortie un objet jQuery
- Objet jQuery = ensemble de noeuds du DOM (Document Object Model)

2. Agir dessus

Exemple :

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

Sélecteur

- ❑ Possibilité de sélectionner de manière similaire aux css :
 - ✓ par type de bloc : `$("balise")`
 - ✓ par identifiant (id) : `$("#id")`
 - ✓ par classe (class) : `$(".class")`

```
// <div>Test</div>
$("div")
// <ul id="test">tt</ul>
$("#test")
// <ul class="test">tt</ul>
$(".test")
```

Sélecteurs de hiérarchie

- ❑ 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()
```

Sélecteurs de formulaire

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

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

```
// sélectionner les boutons  
$(":button")
```

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

```
$("input:checked")
```

```
$("input:selected")
```

Sélecteurs de formulaire: Example

```
<html lang="en">
<head>
  <meta charset="utf-8">
  <script src="jquery-3.3.1.min.js"></script>
</head>
<body>
  <select name="valeur" onchange="select1()">
    <option value="1">1</option>
    <option value="2" selected="selected" >2</option>
    <option value="3">3</option>
  </select>
  <script>
function select1()
{alert($("#select option:selected").val());}
</script>

</body>
```

La fonction each

- ❑ Appelle une fonction pour chaque élément sélectionné
 - ✓ `$(this)` = élément courant
 - ✓ `i` - index de l'élément courant

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


Modifier le contenu HTML

.html('[contenu]') : remplacement du contenu d'un element (les balises sont considerees comme des balises)

.text('[contenu]') : remplacement du contenu d'un element en considerant le tout comme du texte (les caracteres < et > des balises sont remplaces par les entites XML (> et <))

.append('[contenu]') : insertion du contenu dans l'element selectionne a la suite des elements existants

.prepend('[contenu]') : insertion du contenu dans l'element selectionne avant les elements existants

Insérer des éléments

- ❑ **append()** insère du contenu à la fin de la sélection
//Ajout d'une espace et de trois astérisques à la suite de chaque titre <h2>
`$('#h2').append(' ***');`
- ❑ **prepend()** insère du contenu au début de la sélection
Ajout de trois astérisques et d'une espace avant chaque titre <h2>
`$('#h2').prepend('*** ');`
- ❑ **before()** insère du contenu avant la sélection
Ajout d'une ligne de séparation horizontale avant le titre <h2>
`$('#h2').before('<hr>');`
- ❑ **after()** insère du contenu après la sélection
Insertion de un sauts de ligne après chaque balise <hr>
`$('#hr').after('
');`

Gestion des attributs

- ❑ Dans chaque balise html, nous retrouvons des attributs différents comme : title, alt, width, height etc. JQuery permet de manipuler ces attributs facilement à l'aide de la fonction **.attr("name","parametre");**

```
// met tous les attributs comme celui du premier bouton
$("button:gt(0)")
.attr("disabled",
$("button:eq(0)").attr("disabled));
```

Modifier des valeurs

val() : permet d'obtenir la valeur des objets
val(valeur) permet de modifier la valeur des objets

jQuery : Gestionnaire d'événements

Associer une fonction à un événement:

1- `$(sel).click(function() {...})`

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

la méthode `on()` a plusieurs avantages. Elle permet de : Limiter l'écriture en associant une même méthode événementielle à plusieurs éléments

```
$('img.grand').mouseenter(traitement1);
```

```
$('img.grand').mousemove(traitement2);
```

```
$('img.grand').on({mouseenter:traitement1, mousemove:traitement2});
```

AJAX et JQuery

- ❑ jQuery fournit plusieurs méthodes pour AJAX.
- ❑ Avec les jQuery AJAX méthodes, vous pouvez demander du texte, du HTML, du XML ou du JSON à un serveur distant en utilisant HTTP Get et HTTP Post
- ❑ Avec JQuery, nous pouvons écrire la fonctionnalité AJAX avec une seule ligne de code.

jQuery load () Méthode

- ❑ La méthode load () nous permet de charger des données à partir d'un serveur et de les placer dans l'élément sélectionné.
- ❑ Syntaxe: `$ (sélecteur) .load (URL, données,callback);`
 - ✓ **URL**: l'URL que nous souhaitons charger.
 - ✓ **Données**: clé / valeur paires de chaîne de requête à envoyer avec la demande.
 - ✓ **callback** : le nom d'une fonction à exécuter une fois la méthode load () terminée.
- ❑ Il est également possible d'ajouter un sélecteur jQuery au paramètre URL.

jQuery load () Méthode: Exemple

```
<html>
  <head>
    <script
      src="https://ajax.googleapis.com/ajax/libs/jquery/3.4
        .1/jquery.min.js"></script>

    <script> $(document).ready(function(){ $
      ("#b1").click(function(){ $
        ("#div1").load("file1.txt"); }); });
    </script>

  </head> <body>
    <div id="div1"><h2>jQuery et AJAX </h2>
  </div>
  <button id="b1">Changer le contenu</button> </body>
</html>
```

jQuery load () Méthode: Exemple

File1.txt:

```
<div id="div1"> div1 </div>
```

```
<div id="div2"> div2</div>
```

```
<div id="div3"> dive3</div>
```

```
<html>
```

```
<head>
```

```
<script
```

```
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
```

```
<script> $(document).ready(function(){ $  
("#b1").click(function(){  
$("#div1").load("file1.txt  #div2"); }); });  
</script>
```

```
</head> <body>
```

```
<div id="div1"><h2>jQuery et AJAX </h2>
```

```
</div>
```

```
<button id="b1">Changer le contenu</button> </body> </html>
```


jQuery ajax () Méthode

`$.ajax ({nom: valeur, nom: valeur, ...})`

Nom	Description
dataType	Type de données attendu de la réponse du serveur.
data	Spécifie les données à envoyer au serveur
success(result,status,xhr)	Une fonction à exécuter lorsque la requête s'est terminée avec succès
error(xhr,status,error)	Une fonction à exécuter en cas d'échec de la demande
type	Spécifie le type de demande. (GET ou POST)
Xhr	Une fonction utilisée pour créer l'objet XMLHttpRequest

jQuery \$.ajax Méthode: Exemple

```
<html> <head> <script  
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jque  
ry.min.js">  
</script>
```

```
<script> $(document).ready(function(){ $  
("#b1").click(function()  
{ $.ajax( {url:"file1.txt" ,  
          success:function(results){  
            $("#div1").html(results);}); }); });  
</script>
```

```
</head> <body>  
  <div id="div1"><h2>jQuery et AJAX </h2></div>  
  <button id="b1">Changer le contenu</button>  
</body> </html>
```

jQuery \$.get () Méthode

- ❑ La méthode \$.get () demande des données au serveur avec une requête HTTP GET.
- ❑ Syntaxe: \$.get(URL,callback);

Exemple: \$.get("page1.php", Function (données, état){....});

jQuery \$.get () Méthode : Exemple

```
<html> <head> <script  
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jque  
ry.min.js"></script>
```

```
    <script>  
$(document).ready(function()) {  
$("#b1").click(function()) {  
    $.get("page1.php", function(data, status) {  
        $("#div1").html("Data: " + data + "\n  
nStatus: " + status); }); }); });  
    </script>
```

```
</head> <body>  
    <div id="div1"><h2> Ajax et JQuery ( Get Methode)  
</h2></div> <button id="b1">Changer le contenu</button>  
</body> </html>
```

```
    Page1.php  
    <?php echo "Bonjour M/Mme "; ?>
```

jQuery \$.post () Méthode

- ❑ La méthode \$.post () demande des données au serveur avec une requête HTTP POST .
- ❑ Syntaxe: \$.get(URL, Données,callback);
- ❑ Exemple:

```
$.post ("page1.asp",  
  {  
    nom: "Zneika",  
    ville: "Pontoise"  
  },  
  Function (données, état) {  
    .....  
  });
```

jQuery \$.post () Méthode : Exemple

```
<!DOCTYPE html> <html> <head> <script  
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"  
></script>
```

```
<script> $(document).ready(function()) { $("#b1").click(function()) {  
    $.post("page1.php",  
        { nom: "Zneika", ville: "Pontoise" },  
        function(data, status) {  
            alert("Data: " + data + "\nStatus:  
" + status);  
        }); }  
}); }</script>
```

```
</head> <body> <div id="div1"><h2> Ajax et JQuery ( POST Methode)  
</h2></div> <button id="b1">Changer le contenu</button> </body>  
</html>
```

```
<?php  
$var1 = $_POST["nom"];  
$var2 = $_POST["ville"];  
echo "Bonjour $var1 ville $var2";  
?>
```

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 !
<http://openclassrooms.com/courses/jquery-ecrivez-moins-pour-faire-plus>
- Premiers pas avec AJAX
<https://openclassrooms.com/fr/courses/1631636-simplifiez-vos-developpements-javascript-avec-jquery/1636305-premiers-pas-avec-ajax>
- AJAX Introduction
https://www.w3schools.com/xml/ajax_intro.asp