



Le framework JavaScript du Web 2.0



Introduction

- De JavaScript au Framework jQuery
- Les prérequis :
 - HTML / XHTML
 - Le DOM
 - Les feuilles de style CSS
 - Le JavaScript



Présentation de JQuery

- Utilisation du JavaScript sur le web
- **Write less do more** (Écrire moins pour faire plus)
- Simplicité, Accessibilité, Compatibilité



Démarrer avec jQuery

- Créer son document HTML

`<head> ... </head>`

- Mise en place de la librairie <http://jquery.com>

`<script type="text/javascript" src="jquery.js"></script>`

- Utilisation du CDN Google

`<script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/jquery/1.6.4/
jquery.js"> </script>`



Initier un script jQuery

```
<script type="text/javascript">  
  $(document).ready(function() {  
    // code jQuery  
  });  
</script>
```



Hello World

HTML

- `Un lien`
- `Un lien`

JS

```
$(document).ready(function() {  
    $("a").click(function() {  
        alert("Hello world!");  
    });  
});
```

JavaScript sans jQuery

- `Un lien`
- `Un lien`



Première application jQuery

Code HTML (exo1)

```
<div class="menu">  
  <p class="menu_chapitre">Chapitre 1</p>  
  <div class="menu_point">  
    <a href="#">Point 1</a>  
    <a href="#">Point 2</a>  
    <a href="#">Point 3</a>  
  </div>  
  [...]   
</div>
```



Première application jQuery

CSS

```
<style type="text/css">
body { margin: 10px; font: 0.8em Arial, Helvetica, sans-
  serif;}
.menu { width: 150px;}
.menu_chapitre { padding: 5px; cursor:pointer;position:
  relative; margin:1px; font-weight:bold; background:
  #9cf; border: 1px solid black;}
.menu_point a { display:block; color:black; background-
  color:white; padding-left:30px; text-decoration:none;}
.menu_point a:hover { color: black;
  text-decoration:underline;}
</style>
```




Première application jQuery

JS

```
<script type="text/javascript"
  src="https://ajax.googleapis.com/ajax/libs/jquery/1.6.4/
  jquery.js"> </script>
<script type="text/javascript">
$(document).ready(function() {
  $("div.menu_point").hide();
  $("p.menu_chapitre").click(function() {
    $(this).next("div.menu_point").slideDown(300)
    .siblings("div.menu_point").slideUp("slow");
  });
});
</script>
```



Les sélecteurs en jQuery



Les sélecteurs en jQuery

```
<div id="box" class="boite">Ma boite</div>
```

- Sélection par identifiant : `$ (" #box")`
- Sélection par balise HTML: `$ ("div")`
- Sélection par class CSS: `$ (" .boite")`



Les sélecteurs en jQuery

```
<ul id="liste">  
  <li>un</li>  
  <li>deux</li>  
  <li>trois</li>  
</ul>
```

- Sélection des descendants : `$("#liste li")`
- Sélection des enfants : `$("#liste > li")`
- Sélection du premier élément :
`$("#liste li:first")`



Les sélecteurs en jQuery

Ajouter la classe CSS **red** a la liste `id="orderedlist"`

```
$("#orderedlist").addClass("red");
```

Ajouter la classe CSS **blue** aux enfants de cette liste.

```
$("#orderedlist > li").addClass("blue");
```



Les sélecteurs en jQuery

Ajouter et enlever la classe quand l'utilisateur survole l'élément `` (fonction `hover`)
puis uniquement pour le dernier élément de la liste.

```
$("#orderedlist li:last")  
  .hover(function() {  
    $(this).addClass("green");  
  }, function() {  
    $(this).removeClass("green");  
  }) ;
```



Les sélecteurs en jQuery

En utilisant `<dl id="faq">` imaginez une page de FAQ, où toutes les réponses sont cachées au départ, et affichées lorsque la question est cliquée

```
$('#faq').find('dd').hide().end().find('dt')  
  .click(function() {  
    var answer = $(this).next();  
    if (answer.is(':visible')) {  
      answer.slideUp();  
    } else {  
      answer.slideDown();  
    }  
  }) ;
```



Manipuler les attributs



Manipuler les attributs

- **Ajouter ou supprimer une classe**

```
$ ("p:last").addClass ("selected")
```

```
$ ("p:last").removeClass ("selected")
```

- **Vérifier la présence d'une classe**

```
$ ("#p1").hasClass ("box")
```



Manipuler les attributs

- **Connaître la valeur d'un attribut**

```
$("a").attr("title")
```

- **Ajouter un attribut et sa valeur**

```
$("#photo").attr("alt", "Parc éoliennes" )
```

```
$("img").attr({ src: "hat.gif", alt: "Logo jQuery!" })
```

- **Modifier l'attribut value**

```
$("#input").val("Test")
```



Manipuler les attributs

Ajouter un attribut value et une valeur à la dernière balise input du formulaire form.
Puis afficher cette valeur dans une fenêtre alert lorsque l'utilisateur quitte la zone

```
$("#form input:last").val("jquery  
rocks !")  
.blur(function() {  
    alert($(this).val());  
});
```



Manipuler les attributs

En cliquant sur le premier bouton Ajouter un attribut value et une valeur à la dernière balise input du formulaire form. Puis afficher cette valeur dans une fenêtre alert lorsque l'utilisateur quitte la zone

```
$(".buttons button:first").click(function() {  
    $("#form input:last").val("jquery rocks !");  
});
```

```
$("#form input:last").blur(function() {  
    alert($(this).val());  
});
```



Manipuler les attributs

Intervertir les 2 images en modifiant leur source
en cliquant sur la première photo

Et ajouter à la première photo la classe CSS
`.highlight`

Supprimer l'attribut HTML `height`

Faites disparaître la 2eme photo en la faisant
disparaître avec un effet fondu.



Manipuler les attributs

```
$("#images img:first").click(function() {  
    var source1 = $(this).attr("src");  
    $(this).attr("src", $("#images  
img:last").attr("src")).addClass("highligh  
t");  
    $("#images  
img:last").attr("src", source1);  
    $("#images img").removeAttr("height");  
    $("#images img:last").fadeOut('slow');  
})
```



Manipulation des feuilles de style CSS



Manipulation des feuilles de style CSS

- **Accéder à une propriété de style**
`$("p").css("color");`
- **Modifier les propriétés de style**
`$("p").css({ color: "#FF000",
background: "blue" });`
- **Attribuer des propriétés de style**
`$("p").css("color","red");`
- **Le dimensionnement**

<code>\$("p").height();</code>	<code>\$("p").height(120);</code>
<code>\$("p").width();</code>	<code>\$("p").width(120);</code>



Manipulation des feuilles de style CSS

Le positionnement

- **Déterminer une position**

```
$("#p:first").position();
```

```
$("#p:first").offset();
```

- **Contrôle de la barre de défilement**

```
$("#div").scrollTop(300);
```

```
$("#div").scrollLeft(300);
```



Exercice 1

```
$('#input').click(function(){  
  var texte = $('#p');  
  var taille = texte.css('fontSize');  
  var nombre = parseFloat(taille, 10);  
  var unite = taille.slice(-2);  
  if(this.id == 'plus') {  
    nombre *= 1.2;  
  }  
  else if (this.id == 'moins'){  
    nombre /=1.2;  
  }  
  texte.css('fontSize', nombre + unite);  
});  
});
```



Les événements

- Les événements sont les déclencheurs de l'interactivité apportée par le JavaScript.
- La librairie jQuery met à disposition des gestionnaires d'événements assez similaires à ceux du JavaScript traditionnel.



Les événements

- **Au clic de la souris** : `$("p").click();`
- **Au double clic** : `$("div").dblclick();`
- **Le focus** : `$("p").focus();`
- **La perte du focus** : `$("input").blur();`
- **La barre de défilement** : `$(window).scroll();`
- **Le bouton de la souris** :
`$("p").mousedown(); $("p").mouseup();`
- **Le déplacement du curseur** :
`$("div").mousemove();`



Méthodes ou gestionnaires d'événements avancés

- **Lier un événement à un objet :**
`$("#a#lien3").bind("mouseover mouseout", function(e){
$(this).text("-- " + $(this).text() + " --"); });`
La méthode `unbind()` supprime les actions associées à un événement par la méthode `bind()`.
- **Exécuter une fonction une seule fois :**
`$("#p").one("click", function(){ alert($(this).text()); });`
- **Déclencher un événement particulier:**
`$("#p").trigger("click");`
- **Déléguer un événement :**
`$("#p").live("click", function(){ alert($(this).text()); });`



Méthodes propres à jQuery

- **Au passage de la souris**
hover(fonction 1, fonction 2)
- **Basculer d'une situation à l'autre**
toggle (fonction 1, fonction 2)



Les effets

- **Afficher et cacher**

`$('#p').show('slow'); $('#p').hide('fast');`

- **fonction de rappel (callback) (optionnel) :
à exécuter à la fin de l'effet.**

`$('#p').show('slow', function(){alert("Fin");});`





Les effets

- **Glisser verticalement**
`$('#div').slideDown('fast'); $('#div').slideUp('fast');`
- **Réaliser un effet de fondu**
`$('#p:first').fadeIn(4000);`
`$('#p:first').fadeOut(4000);`
- **Fondu vers une opacité spécifiée**
`$('#p:first').fadeTo('slow', 0.33);`
- **Basculer d'un effet à l'autre**
`$("#p").toggle();`
- `$("#p").toggle(function(){`
 `$(this).addClass("selected");`
- `},function(){`
 `$(this).removeClass("selected");`
- `});`
- `$('#div').slideToggle('fast');`



Les effets - Créer une Animation

- ```
$('#clickme').click(function() {
 $('#book').animate({
 opacity: 0.25,
 left: '+=50'
 })
 , 5000, function() {
 // Animation complete.
 });
});
```

# Traverser le DOM

- **Accéder aux éléments du DOM**

`$("div").children()` - enfants immédiats

`$("span").parent()` - parents immédiats

`$("li").parents()` - tous les parents

`$("div").siblings()` - frères immédiats

`$("td").prev()` - frère immédiat précédent

`$("td").prevAll()` - frères immédiats précédents

`$("div").find("p")` - éléments descendants

répondant aux conditions du sélecteur spécifié.



# Manipuler le DOM

- **Insérer à l'intérieur :**  
A la fin : `$("p").append("<b>Hello</b>")`  
Au debut : `$("p").prepend("<b>Hello</b>")`
- **Insérer à l'extérieur**  
`$("p").after("<b>Hello</b>")`  
`$("p").before("<b>Hello</b>")`
- **Supprimer un élément**
- `$("p").remove()`



# Manipuler le DOM

- **Entourer un élément**  
`$("p").wrap("<div class='wrap'></div>");`
- **Remplacer un élément**  
`$("#div1").replaceWith("<div>Nouvelle division</div>");`
- **Remplacer le contenu de l'élément**  
`$("#div1").html("nouveau contenu")`
- **Copier un élément**  
`$("p").clone()`



# Filtrer le DOM

- **Le filtrage des éléments du DOM**  
`$("div").filter(".selected")` filtre de recherche  
`$("div").filter(".selected, :first")` multiples filtres de recherche
- **Trouver un élément déterminé**  
`$("p").eq(1)` : sélectionne le second paragraphe
- **Trouver un élément selon un critère**  
`$(":checkbox").parent().is("form")`



# AJAX vu par jQuery

- AJAX (*Asynchronous JavaScript and XML*)
- Nouvelle conception de page web – Web 2.0
- Extrême concision du code et la facilité de mise en œuvre



# Les requêtes AJAX raccourcies

- **Charger un fichier**  
`$("div").load("test.htm")`

Permet de charger d'une façon extrêmement simple, un fichier contenant du code HTML.



# Exemple de la fonction AJAX

## JS

```
$("#bouton1").click(function() {
 $("#div").load("option1.htm", function(){
 alert("Fichier Option1 charge");
 });
});
$("#bouton2").click(function() {
 $("#div").load("option2.htm ", function(){
 alert("Fichier Option2 charge");
 });
});
```

## HTML

```
<button id="bouton1">Option 1</button>
<button id="bouton2">Option 2</button>
```

## Fichier option1.htm & option2.htm

### **fichier option1.htm**

```
Option 1
```

### **fichier option2.htm**

```
Option 2
```





# Les requêtes AJAX raccourcies

- **Charger qu'en cas de modification**  
`$("#div").loadIfModified("test.htm")`
- **Chargement de fragments de page**  
`$('#result').load('ajax/test.html #container');`  
Permet de charger dans le container #result une partie de la page test.html, représenté par le #container



# Les requêtes AJAX raccourcies

- **Charger selon la méthode GET ou POST**
- `$("#myDiv").load("myScript.php", {var:x, var2:y, var3:z});`  
GET par défaut dans la method load sauf si des arguments sont fournis POST

```
$.get("test.html", function(data){ $("#resultat").html(data); });
```

```
$.post("test.html", function(data){ $("#resultat").html(data);});
```

- **Dans ce cas nécessité de traiter les données reçues – `html()`**



# Charger un script

- `$.getScript("test.js");`  
Charge un script JavaScript du serveur en utilisant la méthode HTTP GET et exécute celui-ci.



# La requête AJAX complète

- **Réalise une requête HTTP asynchrone (AJAX)**
  - > renvoi un objet XMLHttpRequest

```
$.ajax({
 url: "test.htm",
 success: function(data) {
 $("#resultat").html(data);
 },
});
```



# La requête AJAX complète

- **Autre paramètre de cette requête**
- `dataType` (optionnel) : spécifie le format des données qui seront renvoyées par le serveur (xml, html, json ou script). Les types disponibles sont :
  - `"xml"` : retourne un document XML qui pourra être traité par jQuery.
  - `"html"` : retourne du code Html au format texte.
  - `"script"` : évalue la réponse en JavaScript et retourne cette dernière au format texte.
  - `"json"` : évalue la réponse en JSON et retourne un objet JavaScript.



# Exemple du traitement en XML

- ```
$.ajax({
  type: "GET",
  url: "sites.xml",
  dataType: "xml",
  success: function(xml) {
    (xml).find('site').each(function()) {
      var id = $(this).attr('id');
      var title = $(this).find('title').text();
      var url = $(this).find('url').text();
      $('<div class="items" id="link_'+id+'"></div>').html('<a
href="'+url+'">'+title+'</a>').appendTo('#page-wrap');
    });
  } });
```



Les événements associés à la requête AJAX

- **ajaxSend()**
fonction exécutée avant l'envoi de la requête
- **ajaxStart()**
fonction à exécuter lorsqu'une requête débute.
- **ajaxStop()**
fonction exécutée a la fin de la requête
- **ajaxSuccess()**
fonction exécutée à chaque fois qu'une requête AJAX se sera terminée avec succès
- **ajaxComplete()**
- **ajaxError()**
- fonction qui sera exécutée si la requête AJAX échoue



Les événements associés à la requête AJAX

- `$('#message').ajaxError(function() {
 $(this).append('Erreur de la
 requête!
').show();
});
$('a').click(function() {
 $('#resultat').load('x.html');
});`



Utilisation des plugins

- **jQuery UI**

<https://ajax.googleapis.com/ajax/libs/jqueryui/1.8.16/jquery-ui.min.js>

- Sur une page faites apparaître les différents Widgets