

jQuery

“The Write Less, Do More, JavaScript Library”

Initiation à jQuery

- **PRESENTATION**

- **jQuery = bibliothèque de fonctions ou librairie Javascript**
(parfois appelée Framework)

- **Autres bibliothèques / Frameworks**

- Prototype & scriptaculous
 - Mootools
 - Mochikit
 - Yahoo! User Interface Library
 - Dojo Toolkit
 - ...

Pourquoi utiliser jQuery ?

- La **syntaxe** de JavaScript n'est pas toujours facile
- Le programmeur rencontre régulièrement des soucis de **compatibilité** avec les différents navigateurs
- Ces 2 points sont souvent corrélés avec la complexité des interactions avec le navigateur (animations graphiques...)

A quoi sert jQuery ?

- Programmer en Javascript avec 2 avantages
 - **Alléger le code**, avec une **syntaxe concise**.
 - S'affranchir des **soucis de compatibilité** avec les navigateurs (en grande partie)
= ce travail est déjà fait par jQuery
- Que sait faire jQuery ?
 - **Manipuler le DOM** (avec des effets graphiques en complément)
 - **Gérer les événements**
 - Envoyer et traiter des requêtes **AJAX**

“The Write Less, Do More, JavaScript Library”

“Instancier un objet” JQuery

- Au préalable télécharger puis inclure la librairie
`<script type= “text/javascript” src=“chemin/jquery.js”></script>`
- JQuery se base sur une fonction principale :

jQuery() ou **\$()**

Fonction \$() ou jQuery() - Paramètres

Paramètres possibles de la fonction :

- **Une fonction**

Elle sera invoquée lorsque le document sera chargé et lorsque le DOM sera utilisable

```
$(function() {  
  // code a exécuter  
  // (éventuellement invocation d'une autre fonction)  
});
```

Fonction \$() ou jQuery() - Paramètres

Paramètres possibles de la fonction :

- Un **sélecteur CSS** (chaîne de caractères) qui permet de faire référence à un noeud du DOM

Par id : var monElt=\$("#monId");

Equivalent à :

var monElt=document.getElementById("monId");

Par balises : var mesElt=\$("#p");

Attention, ceci retournera un tableau d'éléments.

Equivalent à :

var monElt=document.getElementsByTagName("p");

Par classes : var mesElt= \$(".maclasse");

Même remarque, ceci retournera un tableau d'éléments

Fonction \$() ou jQuery() - Paramètres

Paramètres possibles de la fonction :

- **Sélecteurs CSS (suite)**

Remarque : il est possible de faire des sélections selon la hiérarchie des éléments

var mesElts=\$("#menu li");

- **Des filtres spécifiques à jQuery**

var monElt=\$("p:eq(2)");

On sélectionne le 2e paragraphe

var mesElts =\$(":hidden");

On sélectionne les éléments cachés

(Se reporter à la documentation jQuery)

Fonction \$() ou jQuery() - Méthodes

jQuery propose de nombreuses méthodes applicables à une sélection

- **Manipulation du DOM**
- **Modification du style CSS**
- **Gestion d'événements**
- **Effet visuels**
- **Requêtes AJAX**
- ...

Fonction \$() ou jQuery() - Méthodes

- **Pour appliquer une méthode :**

`$("Selection").methode();`

- **Il est possible de “chainer” l'invocation des méthodes :**

`$("Selection").methode1().methode2();`

Manipulation du DOM

- **Modifier le contenu HTML d'un élément**
`$("#monP").html("Ce que je veux");`
- **Récupérer le contenu HTML d'un élément**
`var contenu=$("#monP").html();`
- **Modifier le contenu texte d'un élément**
`$("#monP").text("Ce que je veux");`
- **Récupérer le contenu texte d'un élément**
`var contenu=$("#monP").text();`

Manipulation du DOM

- **Fixer la valeur d'un champ de formulaire**
`$("#monInput").val("Ce que je veux");`
- **Récupérer a valeur d'un champ de formulaire**
`var valeur=$("#monInput").val();`

NB :

A verifier selon les types de champs de formulaire

Manipulation du DOM

- **Mais également**
 - Ajouter du contenu HTML avec **append**
 - Insérer avant, après avec **after** et **before**
 - Remplacer avec **replaceWith**
 - Supprimer des éléments enfants avec **empty**
 - Supprimer des éléments avec **remove**
 - Cloner des éléments avec **clone**
 - etc.

Manipulation des attributs

- **Obtenir la valeur d'un attribut**
`var monHref = $("#monLien").attr("href");`
- **Fixer la valeur d'un attribut**
`$("#monLien").attr("href","http://site.ext");`
- **jQuery permet également d'enlever un attribut avec `removeAttr`, ajouter une classe avec `addClass`, la supprimer avec `removeClass`**

Manipulation des styles CSS

- **Obtenir la valeur d'une propriété**
`var c = $("#monDiv").css("background-color");`
- **Fixer la valeur d'une propriété**
`$("#monP").css("color","red");`
- **Gérer le positionnement, avec `offset`, `scrollTop`, `scrollLeft`**
- **Gérer les dimensions avec par exemple `height` et `width`**

Gestion d'événements

- **Invoquer une fonction lorsque le DOM est prêt à être manipulé**

```
$(document).ready(function () {  
    $("#monP").text("DOM prêt");  
});
```

- **Lier un événement à un élément**

```
$("#monP").bind("click", function(evtObject){  
    $("#monP").text(evtObject.type);  
});
```


Gestion d'événements

- Gestionnaires d'événements « standards »

```
$("#monP").click(function () {  
    $(this).text(evtObject.type);  
});
```

```
$("#monInput").focus(function () {  
    $(this).css("background-color", "#ff0000");  
});
```

etc.

Effets

- `show()` permet de montrer des éléments
La méthode accepte en paramètre une durée (en ms) et une éventuelle fonction appelée en callback après l'animation
- `hide()` fonctionne de la même manière pour cacher des éléments
- `toggle()` permet d'inverser la visibilité d'éléments
- jQuery propose également des effets de `fading`, `sliding`

AJAX

- jQuery permet d'écrire simplement des requêtes AJAX, avec la méthode GET ou POST

```
$.ajax({  
  type: "GET",  
  url: "some.php",  
  data: "nom=Toto&prenom=Titi",  
  success: function(reponse){  
    alert("Réponse du serveur : " + reponse);  
  }  
});
```

AJAX

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

AJAX

- Des événements sont disponibles lors d'une requête AJAX
 - success
 - error
 - etc.
- Des options permettent
 - de faire des requêtes synchrones ou asynchrones
 - De fixer le type de données attendues en retour (xml, html, script, json, text)
 - etc.

PLUGINS

- **Il existe de nombreux [plugins](#) disponibles pour jQuery = [extensions](#) étendant les possibilités de jQuery**
 - **Nous utiliserons [Fancybox](#) en TD pour gérer l'affichage de documents ou d'images en pop up**
 - **Nous expérimenterons les fonctions drag & drop de [jQuery UI](#) (bibliothèque proposant de nombreux widgets et un gestionnaire de thèmes d'interface)**

Demo

Démonstration de développement rapide
avec jQuery