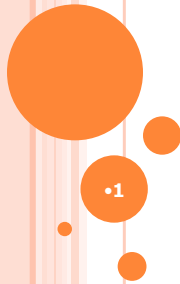


INTRODUCTION À JQUERY



- Une bibliothèque javascript open-source et cross-browser;
- Elle permet de traverser et manipuler très facilement l'arbre DOM des pages web;
- JQuery permet par exemple de changer/ajouter une classe CSS, créer des animations, modifier des attributs, etc. ;
- Gérer les événements javascript ;
- Faire des requêtes AJAX simplement.

UNE SIMPLE BIBLIOTHÈQUE À IMPORTER

- Disponible sur le site de JQuery
<http://jquery.com/>

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

- Ou directement sur Google code

```
<script type="text/javascript"
  src="http://ajax.googleapis.com/ajax/libs/jquery/1/jquery
  .min.js">
</script>
```

•3

LA FONCTION JQUERY()

- jQuery repose sur une seule fonction :
jQuery() ou \$().
- C'est une fonction JavaScript
- Elle accepte des paramètres
- Elle retourne un objet

•4

EXEMPLE DE MANIPULATION DU DOM

```
<html>
<script type="text/javascript"
  src="http://ajax.googleapis.com/ajax/libs/jquery/1/jquery.min.js">
</script>
<body>
<div id="monDiv">Bonjour</div>
<a href="#"
  onClick="$('#monDiv').hide();">
disparition</a>
</body>
</html>
```

•5

JQUERY: UN OBJET

- les méthodes s'appliquent généralement à tous les éléments sélectionnés
 - `$('.classe').hide();`
 - `$('.classe').show();`
- de nombreuses méthodes utilitaires
 - parcourir le DOM: `.parent()`, `.next()`, `.children()`, `.parents()`
 - ajouter ou retirer des classes CSS: `addClass`, `removeClass`
 - manipuler: `append`, `wrap`, `prepend`
- Intérêt fondamental: la plupart des méthodes de l'objet retournent l'objet lui-même
 - on peut chaîner les appels !
 - `$('elem').parent().find('autreElem').show();`
 - Cette propriété est extrêmement puissante !

•6

JQUERY RESSOURCES INTÉRESSANTES

- Site officiel
 - <http://www.jquery.com>
- Learning Center
 - <http://docs.jquery.com/Tutorials>
 - <http://www.learningjquery.com/>
 - <http://15daysofjquery.com/>
 - <http://www.roseindia.net/ajax/jquery/>
- Support
 - <http://docs.jquery.com/Discussion>
 - <http://www.nabble.com/JQuery-f15494.html> mailing list archive
 - irc.freenode.net irc room: #jquery
- Documentation
 - http://docs.jquery.com/Main_Page
 - <http://www.visualjquery.com>
 - <http://jquery.bassistance.de/api-browser/>
- jQuery Success Stories
 - http://docs.jquery.com/Sites_Using_jQuery

7

- Main Page (http://docs.jquery.com/Main_Page)
- Getting Started
 - [Downloading jQuery](#)
 - [How jQuery Works](#)
 - [Frequently Asked Questions](#)
 - [Tutorials](#)
 - [Using jQuery with Other Libraries](#)
 - [Variable Types](#)
- Plugins
 - [Plugin Repository](#)
 - [Plugin Authoring](#)
- Support
 - [Frequently Asked Questions](#)
 - [Forums and Chat](#)

•8

- [Submit a Bug Report](#)
- [Commercial Support](#)
- **About jQuery**
 - [Contributors](#)
 - [History of jQuery](#)
 - [Sites Using jQuery](#)
 - [Browser Compatibility](#)
 - [License](#)
 - [Donate](#)
- **jQuery API Reference**
 - [jQuery Core](#)
 - [Selectors](#)
 - [Attributes](#)
 - [Traversing](#)

•9

- [Manipulation](#)
- [CSS](#)
- [Events](#)
- [Effects](#)
- [Ajax](#)
- [Utilities](#)
- [Internals](#)
- There are a number of [alternative resources](#) for browsing the API.
- **jQuery UI Reference**
 - Interactions
 - [Draggable](#)
 - [Droppable](#)
 - [Sortable](#)
 - [Selectable](#)

•10

- [Resizable](#)
- Widgets
 - [Accordion](#)
 - [Datepicker](#)
 - [Dialog](#)
 - [Progressbar](#)
 - [Slider](#)
 - [Tabs](#)
- [Effects](#)
- Theming
 - [Theming Overview](#)
 - [Theming API](#)
 - [ThemeRoller](#)

•11

BASE

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Demo</title>
    <script src="jquery.js"></script>
    <script>
  </head>
  <body>

    </script>
  </body>
</html>
```

•12

DÉCLENCHER DU CODE LORSQUE LA PAGE EST CHARGÉE

Sans JQuery

```
window.onload = function(){ alert("Bienvenu"); }
```

Avec JQuery

```
$(document).ready(function(){ // Votre code});
```

Afficher un message lorsqu'un lien est cliqué.

```
$(document).ready(function()
{
    $("a").click( function()
    {
        alert("Vous avez cliqué ici!");
    }
    );
});
```

•13

* \$("a") est un sélecteur jQuery, ici, il sélectionne tous les éléments de type lien <a>.

* \$ est un alias pour la classe jQuery : \$() fabrique un nouvel objet jQuery.

* La fonction click() appelée dans la suite est une méthode de l'objet jQuery. Elle associe un événement à tous les éléments sélectionnés et exécute la fonction fournie quand l'événement se déclenche.

Le code suivant génère le même message :

```
<a href="#" onclick="alert ('Vous avez cliqué ici!');">Lien</a>
```

La différence :

* Nous n'avons pas besoin d'écrire un événement onclick pour chaque élément ;

* Nous obtenons une séparation claire entre la structure (HTML) et le comportement (JavaScript) de la même manière que votre feuille de style CSS est dissociée de votre code HTML.

•14

Les sélecteurs

Les sélecteurs permettent de sélectionner un ou plusieurs éléments du DOM, ces derniers pouvant être manipulés comme tout autre objet. JQuery permet la sélection des éléments grâce à une fonction spéciale nommée « \$ ».

`$(document)` // Extension de l'objet document à la classe jQuery.

`$('*')` // Sélectionne tous les éléments.

`$('#monDiv')` // Sélectionne l'élément ayant l'ID "monDiv".

`$(p.first')` // Sélectionne les éléments <p> ayant la classe "first".

`$(p[title])` //Sélectionne les éléments <p> ayant un attribut "title".

`$(p[title="Bonjour"])` // Sélectionne les éléments <p> dont l'attribut title est "Bonjour".

`$(p[title!="Bonjour"])` // Sélectionne les éléments <p> dont l'attribut title n'est pas "Bonjour".

•15

`$(p[title^="H"])` // Sélectionne les éléments dont l'attribut title commence par "H".

`$(p[title$="H"])` // Sélectionne les éléments dont l'attribut title fini par "H".

`$(p[title*="H"])` // Sélectionne les éléments dont l'attribut title contient "H".

`$(ul, ol, dl)` // Sélectionne les éléments , et <dl>

`$(div .desc)` // Sélectionne les éléments ayant la classe "desc" descendants (au sens CSS) d'éléments <div>.

`$(div > .enfant)` // Sélectionne les éléments ayant la classe "enfant" enfants d'éléments <div>.

`$(label + input)` // Sélectionne les éléments <input /> dont l'élément précédent (dans le DOM) est <label>.

`$(#debut ~ div)` // Sélectionne les éléments <div> frères se situant après l'élément dont l'id est "debut".

•16

La sélection par filtre

Filtres

// La notation [a | b] signifie que l'on applique le filtre a ou le filtre b.

\$(div:first) // Sélectionne le premier élément <div>.

\$(div:last) // Sélectionne le dernier élément <div>.

\$(div:not(.ok)) // Sélectionne les <div> n'ayant pas la classe "ok".

\$(div:[even | odd]) // Sélectionne les éléments <div> de rang [pair | impair] (le premier rang est 0).

\$(div:[eq | lt | gt](n)) // Sélectionne le ou les éléments <div> de rang [égal | inférieur | supérieur] à n.

\$(header) // Sélectionne les éléments <h1>.

\$(animated) // Sélectionne les éléments actuellement animés.

•17

\$(div:contains('dvp')) // Sélectionne les éléments <div> contenant le texte "dvp" (sensible à la casse !)

\$(div:empty) // Sélectionne les éléments <div> vides.

\$(div:has(p)) // Sélectionne les éléments <div> ayant un descendant <p>.

\$(div:parent) // Sélectionne les éléments <div> ayant des enfants (y compris les noeuds texte).

\$(div:nth-child([n | even | odd | equation])) // Les enfants de <div> [de rang n | pairs | impairs | résultat de].

\$(div:[first-child | last-child]) // Les éléments [premier | dernier] enfants d'un élément <div>.

\$(div:only-child) // Les éléments qui sont les seuls enfants d'un élément <div>.

•18

Les éléments de formulaire

jQuery possède quelques filtres spécifiques à la sélection d'éléments de formulaires en fonction de leur nature ou de leur état.

Éléments de formulaires

`$( 'input' )` // Tous les éléments `<input />`, `<textarea>`, `<select>` et `<button>`.

`$( ':text|password|radio|checkbox|submit|image|reset|button|file|hidden' )` // Les `<input />` du type choisi.

`$( ':enabled|disabled|checked|selected' )` // Les `<input />` possédant l'attribut indiqué.

QUELQUES EXEMPLES

```
$("#monDiv").css("border", "3px solid red");
```

```
$("#div").css("border", "3px solid red");
```

```
$(".maClasse").css("border", "3px solid red");
```

```
$("#div.monStyle, la classemyClasse").css("border", "3px solid red");
```

```
$("#maTable td:eq(2)").css("color", "#FFC488");
```

```
$("#maTable td:eq(4)").css("color", "#A6FFA6");
```

TD #0	TD #1	TD #2
TD #3	TD #4	TD #5
TD #6	TD #7	TD #8

```
$("#monDiv p").append("<b>jQuery est cool</b>");

$("p").after("<b>Ca Pète!!!</b>");

$("#monDiv p").before("<b>Jimi Hendrix, Electric Ladyland</b>");

$("div.foo").append("Hello World!").css("color","red");
```

Equivalent à :

- `$("div.foo").append("Hello World!");`
- `$("div.foo").css("color","red");`

Utilisation des sélecteurs et des événements

jQuery nous propose 2 approches afin de sélectionner des éléments.

Pour essayer certains de ces sélecteurs, sélectionnons et modifions, par exemple, une liste ordonnée `<ol>` insérée dans une page.

Tout d'abord, sélectionnons la liste elle-même. Cette liste a comme id « listeOrdonnee ». En JavaScript classique, vous pourriez la sélectionner avec `document.getElementById("listeOrdonnee")`. Avec jQuery, cela devient :

```
$(document).ready( function()
{
    $("#listeOrdonnee").addClass("rouge");
}
);
```

Admettons que notre feuille de style propose une classe « rouge » qui ajoute simplement un fond rouge. Donc, si vous ne rechargez pas la page dans votre navigateur, vous devriez voir que la première liste ordonnée obtient un fond rouge. Travaillons maintenant avec les enfants de cette liste.

```
$(document).ready( function()
{
    $(">li").addClass("bleu");
}
);
```

Ce dernier sélectionne tous les enfants `<li>` de l'élément avec l'id « listeOrdonnee » et leur ajoute la classe « bleu ».

Nous voulons ajouter et enlever la classe quand l'utilisateur survole l'élément , mais seulement pour le dernier élément de la liste.

```
$(document).ready( function()
{
    $("#listeOrdonnee li:last").hover( function()
    {
        $(this).addClass("Rouge");
    },
    function()
    {
        $(this).removeClass("Rouge");
    }
    );
});
```

•23

Quelques exemples

Determiner si une checkbox est cochée

```
if ($('#total').attr('checked')) {
    //Traitement si cochée
}
else {
    //Traitement si non cochée
}
```

Intercepter le bouton submit d'un formulaire :

```
$(document).ready(function() {  
    $('#ok').submit(function() {  
        if ($('#login').val() == '') {  
            alert ('Entrer un login');  
            return false;  
        }  
    })  
});
```



Effacer le contenu d'un champs de texte lorsqu'il a le focus

```
<input name="nom" type="text" id="nom"  
      value="Entrez votre nom">
```

```
$('#nom').focus(function() {  
    var elem = $(this);  
    elem.val('');  
});
```



```
function testHideShow() {  
    $('#effet').hide("slow");  
    $('#effet').show("slow");  
}
```

•27

Utilisation très simple d'AJAX

```
$.ajax({  
    type: "GET",  
    url: "test.htm",  
    error:function(msg){  
        alert( "Erreur !: " + msg );  
    },  
    success:function(data){  
        //affiche le contenu du fichier dans le conteneur dédié  
        $('#contenu_fichier_ajax').text(data);  
    });
```

•28

Envoyer des données au serveur et notifier l'utilisateur lorsque c'est terminé

```
$.ajax({  
  type: "POST",  
  url: "prog.php",  
  data:  
    "nom=Carole&emplacement=Laval",  
  success: function(){  
    alert( "Données enregistrées");  
  }  
});
```

•29

Charger et exécuter un fichier JavaScript

```
$.ajax({  
  type: "GET",  
  url: "test.js",  
  dataType: "script"  
});
```

•30

COMPLET

```
var requete = $.ajax({
  url: "script.php",
  type: "POST",
  data: "nom=Carole&emplacement=Laval",
});

requete.done(function(msg) {
  $("#log").html( msg );
});

request.fail(function() {
  alert( "Requête échouée " );
});
```

•31

```
<html>
<head>
<script type="text/javascript" src="jquery-1.3.1.min.js"></script>
<script type="text/javascript">
$(document).ready(function(){ // le document est chargé
$("#a").click(function(){ // on selectionne tous les liens et on définit une action quand on
clique dessus
page=$(this).attr("href")); // on recupere l' adresse du lien
$.ajax({ // ajax
url: page, // url de la page à charger
cache: false, // pas de mise en cache
success:function(html){ // si la requête est un succès
afficher(html); // on execute la fonction afficher(donnees)
},
error:function(XMLHttpRequest, textStatus, errorThrows){ // erreur durant la requete
}
});
return false; // on desactive le lien
});
function afficher(donnees){ // pour remplacer le contenu du div contenu
$("#contenu").empty(); // on vide le div
$("#contenu").append(donnees); // on met dans le div le résultat de la requête ajax
}
```

•32



```
</script>
</head>
<body>
<a href="contenu1.html">contenu1.html</a>

<a href="contenu2.html">contenu2.html</a>
<div id="contenu">
Contenu avant utilisation des liens
</div>
</body>
</html>
```