

Chapitre 6 :

JQUERY

Introduction

jQuery est une librairie JavaScript.

Nous n'avons alors plus qu'à appeler la bonne méthode pour exécuter tout le code qu'elle contient, d'où un gain de temps substantiel.

Les développeurs qui ont créé jQuery se sont également assurés de la validité parfaite de leur code pour l'ensemble des navigateurs les plus utilisés.

Il est fortement recommandé d'avoir déjà une bonne connaissance et une bonne maîtrise du langage JavaScript et de son fonctionnement général. **La librairie jQuery est très complète et est composée de scripts couvrant plus ou moins toutes les fonctionnalités du JavaScript.**

UTILISER JQUERY

Télécharger jQuery

Pour pouvoir utiliser la bibliothèque jQuery, vous devrez d'abord la télécharger. Pour cela, rendez-vous sur le site [jQuery.com](http://jquery.com).

Utiliser la version non compressé le jQuery 3.1.1). Placer dans le même dossier que les fichiers qui vont utiliser cette librairie pour une plus grande simplicité. Il va ensuite falloir créer un lien entre notre fichier jQuery et les fichiers qui vont l'utiliser. Le lien doit être réalisé au sein d'un élément script, avec l'attribut src.

ExempleJQ1.html

```
<HTML><HEAD>  
  
<TITLE>Jquery Exemple 1</TITLE>  
  
<meta cahrset = 'utf-8'</HEAD>  
  
<body> <h1> JQuery Exemple 1<h1>  
  
    <script src='jquery-V-3.3.1.js'></script>  
  
</BODY>  
  
</HTML>
```

Syntaxe de base de jQuery

Le jQuery possède une syntaxe de base assez simple, logique et intuitive.

Tout d'abord, vous devez savoir que le jQuery va baser tout son fonctionnement sur la fonction `jQuery()`.

Cette fonction est généralement écrite de manière abrégée `$()`.

L'utilisation du signe dollar en JavaScript signifie l'appel à la librairie jQuery.

Cette fonction jQuery va pouvoir accepter différents arguments et va toujours retourner un objet jQuery.

Les arguments peuvent être des sélecteurs CSS ou des fonctions.

L'une des grandes forces du jQuery est que l'on va pouvoir avoir accès aux éléments HTML en utilisant des sélecteurs CSS.

Cela simplifie grandement l'accès au contenu HTML.

La syntaxe de base du jQuery pour accéder aux éléments HTML va donc être de la forme suivante : `$(selecteur)`.

Par exemple, pour accéder à tous les éléments **p** de notre page HTML, nous écrirons en **jQuery : \$('p')**.

Ensuite, nous allons pouvoir appliquer des méthodes à l'objet jQuery créé et retourné faisant référence aux éléments HTML ciblés.

Une grande partie de ce cours va être dédiée à la découverte des méthodes fournies par le jQuery puisque c'est ce qui en fait son principal intérêt.

jQuery est là pour simplifier nos développements en JavaScript.

jQuery va nous fournir des scripts complets JavaScript déjà prêts à l'emploi enfermés dans des méthodes.

Nous n'avons alors plus qu'à appeler la bonne méthode.

Par exemple, jQuery dispose d'une **méthode hide()** qui, comme son nom l'indique, va nous permettre de cacher des éléments HTML.

Pour cacher tous les paragraphes de notre page HTML, nous écrirons donc tout simplement en jQuery : `$('p').hide()`.

ExempleJQ1bis.html :

```
<!DOCTYPE html>

<html>

  <head>

    <title> jQuery</title>

    <meta charset='utf-8'>

  </head>

  <body>

    <h1> jQuery Exemple 2</h1>

    <p>Ce paragraphe n'apparaîtra pas</p>

    <p>Celui-ci non plus</p>


    <!--On inclut la librairie jQuery-->

    <script src='jquery-V-3.3.1.js'></script>

    <!--On écrit notre code JavaScript / jQuery-->

    <script>

      $('p').hide();

    </script>

  </body>

</html>
```

jQuery Exemple 2

Ici, nous réalisons deux opérations en une : on commence par accéder à tous les paragraphes de notre page HTML avec `$('p')` et on applique directement la méthode `hide()` à l'objet `jquery` retourné.

Sélecteurs CSS, objets et méthodes jQuery

Nous allons pouvoir utiliser tous les sélecteurs CSS (incluant les sélecteurs CSS3) en jQuery.

Nous allons tout d'abord évidemment pouvoir utiliser les sélecteurs simples de type « élément », comme le sélecteur `p` par exemple pour accéder aux paragraphes de notre page.

L'une des grandes forces du jQuery est que l'on va également pouvoir utiliser les sélecteurs d'attributs ou des sélecteurs complexes afin de cibler précisément certains éléments dans notre page.

Par exemple, on va pouvoir sélectionner tous les éléments portant la class `para`, en écrivant `$('.para')` en jQuery.

On peut utiliser n'importe quel le sélecteur CSS pour accéder à nos éléments HTML en jQuery.

Une fois un élément sélectionné, vous devez savoir que le jQuery va retourner un objet jQuery faisant référence à l'élément en question. C'est ce qui nous permet d'appliquer des méthodes directement à cet objet.

Comme la fonction jQuery renvoie toujours un objet jQuery, nous allons pouvoir pratiquer ce qu'on appelle le « chaînage » de méthodes.

Pour chaîner des méthodes, nous allons utiliser la syntaxe suivante :
`$(selecteur).methode1().methode2().methode3()`.

Dans ce cas là, la méthode « methode1 » va s'exécuter d'abord, puis la méthode « methode2 » va s'exécuter immédiatement après, et enfin la méthode « methode3 ».

ExempleJQ3.html :

Voici l'écriture complète en jQuery utilisant **l'évènement ready** :

```
<!DOCTYPE html>

<html>

  <head><title> jQuery Exemple 3</title> <meta charset='utf-8'> </head>

  <body>

    <h1>Exemple 3</h1>

    <p class='para'>Ce paragraphe n'apparaîtra pas</p>

    <p>Pas de problème pour moi !</p>

    <script src='jquery-V-3.3.1.js'></script>

    <!--On écrit notre code JavaScript / jQuery-->

    <script>

      //On utilise ready en jQuery

      $(document).ready(function(){

        /*On peut par exemple cacher les éléments possédant un attribut class = 'para'*/

        $('.para').hide();

      });

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

Gestion des évènements en JQuery

Un évènement correspond à un moment précis où une action se produit.

L'action peut venir de l'utilisateur, dans le cas par exemple d'un clic sur un élément de la page ou encore du navigateur comme c'est le cas lors du chargement d'une page.

En JavaScript, nous allons pouvoir exécuter du code et effectuer des actions lors de ces évènements en créant des gestionnaires d'évènements.

Le JavaScript met notamment à notre disposition la méthode **addEventListener()** qui nous permet de réagir à l'évènement de notre choix.

Un gestionnaire d'évènements est constitué de deux choses : du nom de l'évènement qu'il doit gérer et généralement d'une fonction à exécuter lors du déclenchement de l'évènement.

Gestion d'un évènement en jQuery : l'évènement click

En jQuery, nous allons utiliser la méthode `click()` pour gérer les évènements de type click.

On va déjà pouvoir appliquer `click()` sans argument sur un objet jquery faisant référence à un élément pour réceptionner un évènement.

Exemple réceptionner un évènement click sur les paragraphes de notre page :

ExempleJQ4click.html

```
<!DOCTYPE html>

<html> <head> <title> jQuery</title><meta charset='utf-8'> </head>

<body>

  <h1>Exemple 4 jQuery click</h1>


  <p>Un premier paragraphe : click pour disparaitre</p>

  <p>Un deuxième paragraphe : click pour disparaitre</p>

  <script src='jquery-V-3.3.1.js'></script>

  <!--On écrit notre code JavaScript / jQuery-->

  <script>

    $(document).ready(function(){

      $('p').click(function(){

        $(this).hide();

      });

    });

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


La méthode jQuery dbleclick()

La méthode dbleclick() va nous permettre de réagir à un évènement de type « double clic ».

ExempleJQ4dbclick.html

```
<!DOCTYPE html>

<html> <head>      <title> jQuery</title> <meta charset='utf-8'> </head>

  <body>

    <h1>Exemple double clique</h1>

    <p>Un premier paragraphe :disparait avec double click </p>

    <p>Un deuxième paragraphe:  disparait avec double click</p>

    <script src='jquery-V-3.3.1.js'></script>

    <script>

      $(document).ready(function(){

        $('p').dblclick(function(){

          $(this).hide();

        });

      });

    </script>

  </body>

</html>
```

Exemple double clique

Un premier paragraphe : disparaît avec double click

Un deuxième paragraphe: disparaît avec double click

Les méthodes `mouseenter()` et `mouseleave()`

Les méthodes jQuery `mouseenter()` et `mouseleave()` vont nous permettre de répondre aux événements « la souris passe sur un élément » et « la souris sort d'un élément ».

ExempleJQ6souris.html

```
<!DOCTYPE html>

<html> <head>      <title> jQuery</title>      <meta charset='utf-8'>

    <style>

        .p1{ border: 1px solid black; }

        .p2{          border: 1px solid red; }

    </style>  </head><body>

    <h1>Gestion d'évènements jQuery Exemple Souris</h1>

    <p class='p1'>Passez pour apparaitre...</p>

    <p class='p2'>... et j'apparais !</p>

    <script src='jquery-V-3.3.1.js'></script>

    <script>

        $(document).ready(function(){

            $('.p2').hide();

            $('.p1').mouseenter(function(){

                $('.p2').show();

            });

            $('.p1').mouseleave(function(){

                $('.p2').hide();

            });

        });  </script>  </body></html>
```

Gestion d'évènements en jQuery Passez sur moi...

Gestion d'évènements jQuery Exemple Souris

Passez pour apparaitre...

... et j'apparais !

La méthode `hover()`

Pour répondre à la fois à l'évènement `mouseenter` et à `mouseleave`, on peut utiliser la méthode jQuery `hover()` qui va être une combinaison des méthodes `mouseenter()` et `mouseleave()`.

Cette méthode va donc prendre deux fonctions en arguments : la première va être exécutée lorsque la souris va entrer dans un élément, tandis que la seconde va être exécutée lorsque la souris va sortir de l'élément.

Nous allons reprendre notre exemple précédent et l'adapter en utilisant `hover()` :

```
<!DOCTYPE html>

<html><head><title>jQuery</title>    <meta charset='utf-8'>

    <style>

        .p1{

            border: 1px solid black;

        }

        .p2{

            border: 1px solid red;

        }

    </style>

</head>

<body>

    <h1>Gestion d'évènements Hover </h1>

    <p class='p1'>Passez pour apparître...</p>

    <p class='p2'>... et j'apparais !</p>

    <script src='jquery-V-3.3.1.js'></script>
```

```
<script>

$(document).ready(function(){

    //On cache p2 par défaut
    $('#p2').hide();

    $('#p1').hover(
        function(){$('#p2').show();},
        function(){$('#p2').hide();}
    );

});

</script>

</body>

</html>
```

Les méthodes `mousedown()` et `mouseup()`

Les méthodes jQuery `mousedown()` et `mouseup()` vont respectivement nous permettre de gérer les événements « le bouton de la souris est enfoncé sur un élément » et « bouton de souris relâché ».

Exemple8JQmouse.html

```
<!DOCTYPE html>

<html>

  <head>

    <title> jQuery</title>

    <meta charset='utf-8'>

    <style>

      .p1{

        border: 1px solid black;

      }

      .p2{

        border: 1px solid red;

      }

    </style>

  </head>

  <body>

    <h1>Gestion d'évènements en jQuery mouse un down</h1>

    <p class='p1'>rester enfoncé la souris .....</p>

    <p class='p2'>... et j'apparais !</p>

    <!--On inclut la librairie jQuery-->

    <script src='jquery-V-3.3.1.js'></script>

    <script>

      $(document).ready(function(){
```

```
//On cache p2 par défaut
$('.p2').hide();

$('.p1').mousedown(function(){
    $('.p2').show();
});

$('.p1').mouseup(function(){
    $('.p2').hide();
});

});

</script>

</body>

</html>
```

Les méthodes jQuery `keydown()`, `keyup()` et `keypress()`

Les méthodes `keydown()`, `keyup()` et `keypress()` vont nous permettre de répondre à des événements relatifs au clavier de l'utilisateur.

Plus précisément, `keydown()` va gérer l'évènement « on presse une touche du clavier » tandis que `keyup()` va gérer l'évènement contraire, à savoir de relâchement d'une touche du clavier (qui était enfoncée).

La méthode `keypress()` va elle nous permettre de gérer l'évènement « maintien d'une touche du clavier pressée ».

Ces trois méthodes vont être très utilisées dans le cadre de la création de jeux par exemple ou avec les formulaires HTML, afin d'aider l'utilisateur.

```
<!DOCTYPE html>

<html>

  <head>

    <title> jQuery</title>

    <meta charset='utf-8'>

  </head>

  <body>

    <h1>Gestion d'évènements en jQuery clavier keydown keyup</h1>

    <label for='prenom'>Entrez votre prénom</label>

    <input type='text' id='prenom' name='prenom'>

    <script src='jquery-V-3.3.1.js'></script>

    <script>

      $(document).ready(function(){

        //Si une touche est enfoncée, le champ est bleu

        $('#prenom').keydown(function(){

          $(this).css('background-color', '#69C');

        });

        //Si on relâche la touche, le champ devient orange

        $('#prenom').keyup(function(){

          $(this).css('background-color', '#C72');

        })

      });

    </script>

  </body>

</html>
```


</script>

</body>

</html>