



Le Framework jQuery

Frameworks Web côté client

El hadji Mamadou NGUER Enseignant chercheur en Informatique à l'UVS



Séquence 3 : Le Framework jQuery

Objectifs spécifiques : A la suite de cette séquence, l'apprenant doit être capable de:

1. Décrire le Framework jQuery
2. Décrire la syntaxe de jQuery
3. Utiliser jQuery pour rendre dynamique une page Web



Séquence 3 : Le Framework jQuery

Références :

- Le contenu de cette séquence a été conçu principalement sur la base des ressources issues du tutoriel de jQuery du site www.w3schools.com du W3C destiné à l'apprentissage en ligne des technologies Web.
- Les exemples qui y sont présentés sont aussi basés sur ceux de ce tutoriel.

Séquence 3 : Le Framework jQuery

Plan de la séquence :

1. Introduction
2. Installation de jQuery
3. La syntaxe
4. Les sélecteurs
5. Les évènements
6. Les effets
7. jQuery et Html

Introduction

Qu'est-ce que jQuery?

- jQuery est un Framework JavaScript.
- Il sert à rendre beaucoup plus facile l'utilisation de JavaScript sur un site en simplifiant grandement la programmation JavaScript.
- Son apprentissage nécessite comme prérequis HTML, CSS et JavaScript
- Il a plusieurs fonctionnalités dont :
 - La manipulation du HTML/DOM
 - La manipulation du CSS
 - La gestion des événements HTML
 - La gestion des effets et des animations
 - AJAX
 - Et d'autres utilitaires

Introduction

Pourquoi jQuery?

- Il y'a beaucoup d'autres frameworks JavaScript (MooTools , AngularJS, Dojo, Prototype...), mais jQuery semble être la plus populaire, et aussi la plus extensible.
- Bon nombre des plus grandes entreprises sur le Web utilisent jQuery, tels que:
 - Google
 - Microsoft
 - IBM
 - Netflix

Introduction

Exemple:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("p").click(function(){ $(this).hide(); });
});
</script>
</head>
<body>
<p>Cliquez sur moi et je disparaiss.</p>
<p>Cliquez sur moi!</p>
<p>Cliquez sur moi aussi!</p>
</body>
</html>
```



Installation

Ajout de jQuery pour vos pages Web

Il y'a plusieurs façons pour commencer à utiliser jQuery sur votre site web. Vous pouvez:

- Télécharger la bibliothèque jQuery à partir jQuery.com
- Inclure jQuery à partir d'un CDN, comme Google

Télécharger jQuery

Il existe deux versions de jQuery disponibles pour le téléchargement:

- la version de production – c'est une utilisation direct parce qu'il a été comprimé et rapetissé
- La version de développement – c'st pour des tests et pour le développement (code non compressé et lisible)

Les deux versions peuvent être téléchargées à partir jQuery.com .

Installation

Ajout de jQuery pour vos pages Web

La bibliothèque jQuery est un seul fichier JavaScript que vous pouvez référencer avec la balise HTML `<script>` (Mais `<script>` doit être à l'intérieur de la section `<head>`):

`<head>`

`<script src="jQuery-3.3.1.min.js"></script>`

`</head>`

Conseil: Placez le fichier téléchargé dans le même répertoire que les pages où vous souhaitez l'utiliser.

.

Installation

Ajout de jQuery pour vos pages Web

➤ CDN jQuery

Si vous ne souhaitez pas télécharger et héberger jQuery vous-même, vous pouvez l'inclure à partir d'un CDN (Content Delivery Network).

Google et Microsoft héberge jQuery.

Pour utiliser jQuery de Google ou Microsoft, utiliser une des méthodes suivantes.

CDN Google:

```
<head>
```

```
<script src=" https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js "></script>
```

```
</head>
```

CDN Microsoft:

```
<head>
```

```
<script src=" https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js "></script>
```

```
</head>
```

Installation

Ajout de jQuery pour vos pages Web

➤ CDN jQuery

L'avantage d'utiliser jQuery des CDN de Google ou Microsoft:

Beaucoup d'utilisateurs ont déjà téléchargé jQuery de Google ou de Microsoft en visitant un autre site. En conséquence, il sera chargé dans la mémoire cache quand ils visitent votre site, ce qui conduit à accélérer le temps de chargement.

Aussi, la plupart des CDN s'assurent qu'à chaque fois qu'un utilisateur demande un fichier au serveur, de le servir à partir d'un serveur qui lui est le plus proche. Ce qui conduit également à accélérer le temps de chargement.

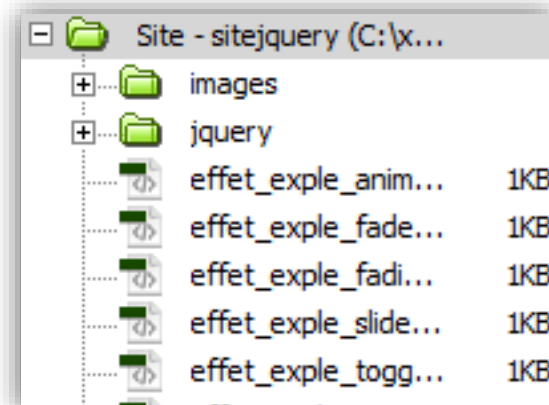
Installation

Dossier des exemples du cours :

- Pour une meilleur compréhension du cours, nous vous recommandons de tester les exemples de page situés dans le dossier sitejquery téléchargeable à l'adresse :

<https://drive.google.com/drive/folders/1-11T-7n2VeT-1Kuhd2AWfuy4Tcuf3RAz?usp=sharing>

- Ce dossier contient :
 - Le dossier du Framework jQuery
 - Un dossier images
 - Une page pour chaque sélecteur, évènement, effet et méthodes de manipulation de code HTML et de CSS.



La syntaxe

Avec jQuery on sélectionne des éléments HTML puis on effectue des "actions" sur eux.

La syntaxe jQuery est fait sur mesure pour faire **la sélection** des éléments HTML et pour effectuer certaines **actions** sur l'élément (s).

La syntaxe de base est: **\$ (selecteur).action()**

- Le signe \$ pour définir/accéder à jQuery
- Le (*selecteur*) pour trouver les éléments HTML
- action() est l'action à appliquer sur les éléments

Exemples:

\$(this).hide() - cache l'élément courant.

\$("p").hide() -. Cache tous les éléments <p>.

\$(".test").hide() -. Cache tous les éléments avec class = "test".

\$("#test").hide() -. Cache l'élément avec id = "test".

La syntaxe

Le Document Ready Event

Vous avez sans doute remarqué que toutes les méthodes jQuery dans nos exemples, sont à l'intérieur d'un Document Ready Event :

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

```
// méthodes jQuery ici...
```

```
});
```

- C'est pour empêcher tout code jQuery de s'exécuter avant que le document ne soit complètement chargé.
- Il est une bonne pratique de mettre le code JavaScript dans l'entête du document pour avoir cela.

La syntaxe

Le Document Ready Event

Il existe aussi une méthode encore plus courte que le Document Ready Event:
\$(function(){

// méthodes jQuery ici...

});

Utilisez la syntaxe que vous préférez. Nous pensons que le Document Ready Event est plus facile à comprendre à la lecture du code.

Les sélecteurs

- Les sélecteurs jQuery vous permettent de sélectionner et de manipuler les éléments HTML.
- Ils sont utilisés pour trouver les éléments HTML par leur id, leur classe, leur type, leurs attributs, leur valeur d'attributs, etc.
- Il est basé sur les sélecteurs CSS, et en plus, il a quelques sélecteurs personnalisés propres.
- Tous les sélecteurs jQuery commencent avec le signe du dollar et les parenthèses: \$ ().

Les sélecteurs

Le sélecteur d'élément

Il sélectionne des éléments sur la base du nom de l'élément.

Vous pouvez sélectionner tous les éléments `<p>` sur une page comme ceci:
`$("p")`

Exemple: Click sur un bouton pour cacher tous les éléments `<p>`:

```
<!DOCTYPE html>
```

```
<html><head><script src=" https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js  
></script>
```

```
<script>
```

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

```
  $("button").click(function(){
```

```
    $("p").hide();
```

```
  });
```

```
});
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<h2>This is a heading</h2>
```

```
<p>This is a paragraph.</p>
```

```
<p>This is another paragraph.</p>
```

```
<button>Click me</button>
```

```
</body>
```

```
</html>
```

Les sélecteurs

Le sélecteur #id

Il utilise l'attribut id d'une balise pour trouver l'élément spécifique.

Un id doit être unique dans une page, vous devez donc utiliser le sélecteur de #id lorsque vous souhaitez trouver un seul élément unique.

Pour trouver un élément avec un ID spécifique, écrire un caractère dièse, suivie par l'id de l'élément HTML: \$("#test")

Exemple : click sur un bouton pour cacher l'élément d'id = "test" :

```
$(document).ready(function(){  
    $("button").click(function(){  
        $("#test").hide();  
    });  
});
```

Les sélecteurs

Le sélecteur .class

Il trouve les éléments appartenant à une classe spécifique.

Pour trouver des éléments avec une classe spécifique, écrire « . » suivi du nom de la classe:

```
$(".test")
```

Exemple: Click sur un bouton pour cacher les éléments de "test" :

```
$(document).ready(function(){  
    $("button").click(function(){  
        $(".test").hide();  
    });  
});
```



Les sélecteurs

Autres exemples de seleteurs

Syntaxe	Description
<code>\$("*")</code>	Sélectionne tous les éléments
<code>\$(this)</code>	Sélectionne l'élément courant
<code>\$("p.intro")</code>	Selectionne tous les elements <code><p></code> de classe "intro"
<code>\$("p:first")</code>	Selectionne le premier element <code><p></code>
<code>\$("ul li:first")</code>	Selectionne le premier element <code></code> du premier <code></code>
<code>\$("ul li:first-child")</code>	Selectionne le premier element <code></code> de chaque <code></code>
<code>\$("[href]")</code>	Selectionne tous les elements ayant un attribut href
<code>\$("a[target='_blank']")</code>	Selectionne tous les elements <code><a></code> avec un attribut target ayant la valeur "_blank"
<code>\$("a[target!='_blank']")</code>	Selectionne tous les elts <code><a></code> avec avec un attribut target ayant une valeur different de "_blank"
<code>\$(":button")</code>	Selectionne tous les elts <code><button></code> et <code><input></code> de type "button"
<code>\$("tr:even")</code>	Selectionne tous les evenements sur les elements <code><tr></code>
<code>\$("tr:odd")</code>	Selectionne tous les elements <code><tr></code> pair

Les événements

jQuery est fait pour répondre aux événements dans une page HTML.

Qu'est ce qu'un événement?

- Toute action que les visiteurs peuvent effectuer sur une page Web est appelée événement.
- Un événement représente le moment précis où quelque chose se passe.

Exemple:

le survole d'une souris sur un élément

la sélection d'un bouton radio

Le clic sur un élément

Les événements

Voici quelques événements DOM

Souris	Boutons clavier	Formulaire	Document/Fenêtre
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload

Les évènements

Syntaxe des méthodes d'événement

La plupart des événements DOM ont une méthode jQuery équivalent.

Pour attribuer un événement clic à tous les paragraphes sur une page, vous pouvez faire ceci: `$("p").click();`

Il reste à définir ce qui doit se produire lorsque l'événement se déclenche. Vous devez passer une fonction à l'événement:

```
$("p").click(function(){  
    // action à faire!!  
});
```

Les événements

Les méthodes d'événement les plus utilisées

Chaque méthode attache une fonction de gestionnaire d'événement à un élément HTML.

- **`$(Document).ready ()`**

Elle permet d'exécuter une fonction lorsque le document est totalement chargé.

- **`click()`**

Elle attache une fonction de gestionnaire d'événement à un élément HTML.

La fonction est exécutée lorsque l'utilisateur clique sur l'élément HTML.

Exemple

```
$("#p").click(function(){  
    $(this).hide();  
});
```


Les évènements

Les méthodes d'événement les plus utilisées

- **dblclick()**

Elle est exécutée lorsque l'utilisateur double-clique sur l'élément HTML.

Exemple

```
$("#p"). dblclick(function(){  
    $(this).hide();  
});
```

- **mouseenter()**

Elle est exécutée lorsque le pointeur de la souris entre dans l'élément.

Exemple

```
$("#p"). mouseenter (function(){  
    $(this).hide();  
});
```



Les événements

Les méthodes d'événement les plus utilisées

- **mouseleave ()**

Elle est exécutée lorsque le pointeur de la souris quitte l'élément HTML:

Exemple

```
$("#p1").mouseleave(function(){  
    alert("Au revoir! Vous venez de quitter p1!");  
});
```

- **mousedown ()**

Elle est exécutée lorsque le bouton gauche de la souris est enfoncé, tandis que la souris est sur l'élément HTML:

Exemple

```
$("#p1").mousedown(function(){  
    alert("Souris enfoncée sur p1!");  
});
```

Les évènements

Les méthodes d'événement les plus utilisées

- **mouseup ()**

Elle est exécutée lorsque le bouton gauche de la souris est enfoncé, tandis que la souris est sur l'élément HTML:

Exemple

```
$("#p1").mousedown(function(){  
    alert("Souris enfoncée sur p1!");  
});
```

- **focus ()**

Elle est exécutée lorsque le champ du formulaire prend le focus:

Exemple

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

Les évènements

Les méthodes d'événement les plus utilisées

- **hover ()**

Elle prend deux fonctions et est une combinaison `mouseenter ()` et `mouseleave` (méthodes).

La première fonction est exécutée lorsque la souris entre dans l'élément, et la seconde fonction est exécutée lorsque la souris quitte l'élément HTML:

Exemple

```
$("#p1").hover(function(){  
    alert("Vous êtes entré dans p1!");  
},  
function(){  
    alert("Au revoir! Vous venez de quitter p1!");  
});
```

Les événements

Les méthodes d'événement les plus utilisées

- **blur ()**

Elle est exécutée lorsque le champ de formulaire perd le focus:

Exemple

```
$("#input").blur(function(){  
    $(this).css("background-color","#ffffff");  
});
```

Pour une référence complète des événement de jQuery

http://www.w3schools.com/jQuery/jQuery_ref_events.asp

Les évènements

Les méthodes d'événement les plus utilisées

- **blur ()**

Elle est exécutée lorsque le champ de formulaire perd le focus:

Exemple

```
$("#input").blur(function(){  
    $(this).css("background-color","#ffffff");  
});
```

Pour une référence complète des événement de jQuery

http://www.w3schools.com/jQuery/jQuery_ref_events.asp

Les effets

Les effets sont Hide, Show, Toggle, Slide, Fade et Animate

- Les effets Hide et Show

Avec jQuery, on peut afficher ou masquer les éléments HTML avec la les méthodes hide () et show () :

Exemple:

```
$("#hide").click(function(){  
    $("#p").hide();  
});
```

```
$("#show").click(function(){  
    $("#p").show();  
});
```

Les effets

- **Les effets Hide et Show**

Syntaxe:

```
$(selector).hide(speed,callback);
```

```
$(selector).show(speed,callback);
```

Le paramètre de vitesse optionnel spécifie la vitesse pour masquer / afficher, et peut prendre les valeurs suivantes: "slow", "fast", ou milliseconds.

Le paramètre option optionnel callback est une fonction à exécuter après la méthode hide () ou show () soit complète.

Exemple:

```
$("#button").click(function(){  
    $("#p").hide(1000);  
});
```


Les effets

- **L'effet toggle**

Avec jQuery, vous pouvez basculer entre les méthodes `hide()` et `show()` avec la méthode `toggle()`.

Éléments représentés sont cachés et éléments cachés sont affichés:

Exemple

```
$("#button").click(function(){  
    $("#p").toggle();  
});
```

Les effets

- L'effet toggle

Syntaxe:

```
$(selector).toggle(speed,callback);
```

Le paramètre de vitesse en option peut prendre les valeurs suivantes: "slow", "fast", ou milliseconds.

Le paramètre optionnel callback est une fonction à exécuter après que toggle () soit complète.

Les effets

- L'effet fondu (fade)

Avec jQuery on peut estomper les éléments dans et hors de la visibilité avec les modes de fondu suivantes:

- ✓ fadeIn ()
- ✓ fadeOut ()
- ✓ fadeToggle ()
- ✓ fadeTo ().

Les effets

- **L'effet fondu (fade) : la méthode fadeIn ()**

La méthode jQuery fadeIn () est utilisé pour estomper un élément caché.

Syntaxe:

```
$(selector).fadeIn(speed,callback);
```

Le paramètre optionnel speed spécifie la durée de l'effet. Il peut prendre les valeurs suivantes: "slow", "fast", ou milliseconds.

Le paramètre optionnel callback est une fonction à exécuter après la disparition complète.

L'exemple suivant illustre la méthode fadeIn () avec des paramètres différents:

Exemple

```
$("#button").click(function(){  
    $("#div1").fadeIn();  
    $("#div2").fadeIn("slow");  
    $("#div3").fadeIn(3000);  
});
```

Les effets

- L'effet fondu (fade) : la méthode `fadeIn()`

Exemple

```
<body>
```

```
<p>Illustration de fadeIn() avec différents paramètres.</p>
```

```
<button>Click </button><br><br>
```

```
<div id="div1" style="width:80px;height:80px;display:none;background-color:red;"></div><br>
```

```
<div id="div2" style="width:80px;height:80px;display:none;background-color:green;"></div><br>
```

```
<div id="div3" style="width:80px;height:80px;display:none;background-color:blue;"></div>
```

```
</body>
```

```
</html>
```

Les effets

- **L'effet slide (Diaporama)**

Les méthodes de slide permettent de glisser les éléments vers le haut et le bas.

jQuery possède les méthodes de diapositives suivants:

- `slideDown()`
- `slideUp ()`
- `slideToggle ()`

Syntaxe:

`$(selector).slideDown(speed,callback);`

Le paramètre optionnel `speed` spécifie la durée de l'effet. Il peut prendre les valeurs suivantes: "slow", "fast", ou `milliseconds`.

Le paramètre optionnel `callback` est une fonction à exécuter après la disparition complète.

Les effets

- **L'effet Animation**

La méthode jQuery animate () vous permet de créer des animations personnalisées.

Syntaxe:

```
$(selector).animate({params},speed,callback);
```

Le paramètre obligatoire *params* définit les propriétés CSS à animer.

Le paramètre optionnel *speed* spécifie la durée de l'effet. Il peut prendre les valeurs suivantes: "slow", "fast", ou milliseconds.

Le paramètre optionnel *callback* est une fonction à exécuter après la disparition complète.

Les effets

- **L'effet Animation**

Exemple: Déplacement d'un élément <div> vers la droite, jusqu'à une distance de 250px:

```
$("#button").click(function(){  
    $("#div").animate({left: '250px'});  
});
```

NB: Par défaut, les éléments HTML ont une position statique, et ne peuvent être déplacés. Pour les déplacer, il faudra régler la propriété CSS position à relative, fixed ou absolute!

Animer des propriétés multiples :

```
$("#button").click(function(){  
    $("#div").animate({  
        left: '250px', opacity: '0.5',  
        height: '150px',  
        width: '150px'  
    });  
});
```


Les effets

- **L'effet Animation**

Est-il possible de d'animer toutes les propriétés de CSS avec la méthode animate ()?

Oui, presque! Cependant, il y'a une chose importante à retenir: tous les noms de propriété doivent être de la forme CamelCase lorsque c'est utilisé avec la méthode animate ():

Exemple : On dit écrire paddingLeft lieu de padding-left, marginRight la place de la marge-droite, et ainsi de suite.

Cependant, l'animation de couleur n'est pas inclus dans le noyau de la bibliothèque de jQuery. Si vous souhaitez animer la couleur, vous devez télécharger le plug-in d'animation de couleur à partir de [jQuery.com](http://jquery.com).

Les effets

- **L'effet Animation**

Utiliser des valeurs relatives : la valeur est alors par rapport à la valeur actuelle de l'élément. Ceci est réalisé en plaçant += ou -= devant la valeur:

Exemple

```
$("#button").click(function(){  
    $("#div").animate({  
        left: '250px',  
        height: '+=150px', width: '+=150px'  
    });  
});
```

Utilisation des valeurs prédéfinies: On peut même spécifier la valeur de la propriété avec "show", "hide", ou "toggle":

Exemple

```
$("#button").click(function(){  
    $("#div").animate({  
        height: 'toggle'  
    });  
});
```

Les effets

- **L'effet Animation**

Utilise la fonctionnalité de file d'attente

Par défaut, jQuery est livré avec la fonctionnalité de file d'attente pour les animations.

Cela signifie que si vous écrivez des animations multiples l'un après l'autre, jQuery crée une file d'attente «interne» à ces appels de méthode. Ensuite, elle les appelle une par une.

Exemple

```
$("#button").click(function(){  
    var div = $("#div");  
    div.animate({height: '300px', opacity: '0.4'}, "slow");  
    div.animate({width: '300px', opacity: '0.8'}, "slow");  
    div.animate({height: '100px', opacity: '0.4'}, "slow");  
    div.animate({width: '100px', opacity: '0.8'}, "slow");  
});
```

Les effets

- **L'effet Animation**

Utilise la fonctionnalité de file d'attente

L'exemple ci-dessous déplace l'élément <div> vers la droite, puis augmente la taille de la police du texte:

Exemple

```
$("#button").click(function(){  
    var div = $("#div");  
    div.animate({left: '100px'}, "slow");  
    div.animate({fontSize: '3em'}, "slow");  
});
```



Les effets

- **L'effet Stop Animation**

La méthode **stop** () de jQuery est utilisée pour arrêter une animation ou un effet avant qu'il ne termine.

La méthode **stop** () fonctionne sur toutes les fonctions d'effet de jQuery, y compris le glissement (slide), la décoloration (fade) et les animations personnalisées.

Syntaxe: \$(selector).stop(stopAll,goToEnd);

Le paramètre optionnel **stopAll** spécifie si également les animations dans la file d'attente doivent être arrêtées ou non. Par défaut il est à **false**, ce qui signifie que seul l'animation active sera arrêtée, les autres animations de la liste se poursuivent.

Le paramètre optionnel **gotoEnd** spécifie si l'animation en cours doit terminer immédiatement. Par défaut sa valeur est **false**.

Donc, par défaut, la méthode **stop** () arrête l'animation courant effectuée sur l'élément sélectionné.

Les effets

- L'effet Stop Animation

Exemple :

```
<script>
$(document).ready(function(){
  $("#flip").click(function(){
    $("#panel").slideDown(5000);
  });
  $("#stop").click(function(){
    $("#panel").stop();
  });
});
</script>

<style>
#panel, #flip {
  padding: 5px;  font-size: 18px;
  text-align: center;
  background-color: #555;
  color: white; border: solid 1px #666;
  border-radius: 3px;}

#panel {
  padding: 50px;
  display: none;
}
</style>
</head>

<body>

<button id="stop">Stop sliding</button>

<div id="flip">
Click to slide down panel</div>
<div id="panel">Hello world!</div>

</body>
</html>
```

Les effets

- L'effet Stop Animation: Exemple 2

```
<script>
$(document).ready(function(){
  $("#start").click(function(){
    $("#div").animate({left: '100px'}, 5000);
    $("#div").animate({fontSize: '3em'}, 5000);
  });


  $("#stop").click(function(){
    $("#div").stop();
  });

  $("#stop2").click(function(){
    $("#div").stop(true);
  });

  $("#stop3").click(function(){
    $("#div").stop(true, true);
  });
});
</script>
```

```
<body>

<button id="start">Start</button>
<button id="stop">Stop</button>
<button id="stop2">Stop all</button>
<button id="stop3">Stop but finish</button>
<div
style="background:#98bf21;height:100px;width:
200px;position:absolute;">Hello
</div>
</body>
</html>
```



Les effets

- **La fonction de rappel (callback)**

Les instructions JavaScript sont exécutés ligne par ligne. Cependant, avec les effets, la prochaine ligne de code peut être exécutée même si l'effet n'est pas fini. Cela peut créer des erreurs.

Pour éviter cela, vous pouvez créer une fonction de rappel.

Une fonction de rappel est exécutée après l'effet en cours est terminée.

Syntaxe : \$ (sélecteur) .hide (vitesse, rappel);

Exemple avec rappel

```
$("#button").click(function(){
    $("#p").hide("slow", function(){
        alert("Le paragraphe est caché");
    });
});
```

Exemple sans rappel

```
$("#button").click(function(){
    $("#p").hide(1000);
    alert("Le paragraphe est caché");
});
```



Les effets

- **L'enchaînement des actions**

Avec jQuery, on peut enchaîner les actions / méthodes.

L'enchaînement permet d'exécuter plusieurs méthodes jQuery (sur le même élément) au sein d'une seule déclaration.

La méthode de chaînage

Jusqu'à présent, nous avons écrit les déclarations une à la fois (l'un après l'autre).

Cependant, il existe une technique appelée chaînage, qui nous permet d'exécuter des commandes multiples, l'un après l'autre, sur le même élément.

Pour enchaîner sur une action, on l'ajoute simplement à l'action précédente.

L'exemple suivant enchaîne ensemble les méthodes `css()`, `slideUp()`, et `slideDown()`.

L'élément "p1" change d'abord à rouge, puis il glisse vers le haut, puis vers le bas:

Ex: `$("#p1").css("color", "red").slideUp(2000).slideDown(2000);`

Les effets

- **L'enchaînement des actions**

Plusieurs appels de méthode peuvent être ajoutés si nécessaire.

Ce qui fait que la ligne de code pourrait devenir assez longue.

Mais comme jQuery n'est pas très stricte sur la syntaxe; on peut formater la ligne en y mettant des sauts de ligne et des indentations comme dans l'exemple suivant, .

Cela fonctionne également très bien:

Exemple

```
$("#p1").css("color", "red")  
    .slideUp(2000)  
    .slideDown(2000);
```

jQuery supprime les espaces supplémentaires et exécute les lignes ci-dessus comme une seule ligne de code.

jQuery et Html

- jQuery contient des méthodes puissantes pour changer et manipuler des éléments et attributs HTML.
- Il est doté d'un ensemble de méthodes DOM (Document Object Model) connexes qui permettent d'accéder et de manipuler facilement les éléments et attributs de HTML.
- Le DOM définit une norme pour l'accès aux documents HTML et XML:

"Le Document Object Model (DOM) de W3C est une plate-forme et un interface en langage neutre qui permet aux programmes et aux scripts d'accéder dynamiquement et de mettre à jour le contenu, la structure et le style d'un document".

jQuery et Html

- **Obtenir le contenu - texte (), html () et Val ()**

Les trois méthodes simples suivants permettent d'obtenir le contenu texte:

- **text ()** retourne le contenu texte des éléments sélectionnés
- **html ()** retourne le contenu des éléments sélectionnés (y compris les balises HTML)
- **val ()** retourne la valeur des champs de formulaire

Exemple:

```
$("#btn1").click(function(){  
    alert("Text: " + $("#test").text());  
});  
$("#btn2").click(function(){  
    alert("HTML: " + $("#test").html());  
});
```

```
$("#btn2").click(function(){  
    alert("HTML: " + $("#test").val());  
});
```

jQuery et Html

- **Obtenir la valeur d'un attribut- attr ()**

La méthode jQuery **attr ()** permet d'obtenir la valeur d'un attribut.

L'exemple suivant permet d'obtenir la valeur de l'attribut href d'un lien:

Exemple

```
<script>
$(document).ready(function(){
    $("button").click(function(){
        alert($("#ugb").attr("href"));
    });
});
</script>
</head>
```

```
<body>

<p><a href="http://www.ugb.sn"
id="ugb">www.gb.sn</a></p>

<button>Afficher la valeur de href</button>

</body>
</html>
```

jQuery et Html

- Définir le contenu - `text ()`, `html ()` et `Val ()`

Les trois méthodes simples suivantes permettent de définir le contenu:

- **text ()** qui définit le contenu texte des éléments sélectionnés
- **html ()** qui définit le contenu des éléments sélectionnés (y compris les balises HTML)
- **val ()** qui définit la valeur des champs de formulaire

Exemple:

```
$("#btn1").click(function(){  
    $("#test").text("Bonjour !");  
});  
$("#btn2").click(function(){  
    $("#test").html("<b>Hello world!</b>");  
});
```

```
$("#btn3").click(function(){  
    $("#test").val("Bonjour");  
});
```

jQuery et Html

- **Fonction de rappel pour texte (), html () et Val ()**

Les trois méthodes text (), html () et Val () de jQuery peuvent être utilisées avec une fonction de rappel.

La fonction de rappel a deux paramètres: l'indice de l'élément courant dans la liste des éléments sélectionnés et la valeur d'origine. Vous retournez ensuite la chaîne que vous souhaitez utiliser comme nouvelle valeur de la fonction.

L'exemple suivant utilise text () et html () avec une fonction de rappel:

Exemple:

```
$("#btn1").click(function(){  
    $("#test1").text(function(i, origText){  
        return "Old text: " + origText + " New text: Hello world!  
        (index: " + i + ")";  
    });  
});
```

jQuery et Html

- **Définir les attributs attr ()**

La méthode attr () est également utilisée pour définir / modifier les valeurs d'attribut.

Exemple

```
$("#button").click(function(){  
    $("#ugb").attr("href", "http://www.ugb.sn");  
});
```

Elle permet également de définir plusieurs attributs en même temps.

Exemple

```
$("#button").click(function(){  
    $("#ugb").attr({  
        "href" : "http://www.ugb.sn",  
        "title" : "Université de Saint-Louis"  
    });  
});
```


jQuery et Html

- **Fonction de rappel pour attr ()**

La méthode attr () de jQuery peut être utilisée avec une fonction de rappel.

La fonction de rappel a deux paramètres: l'indice de l'élément courant dans la liste des éléments sélectionnés et la valeur d'origine. Vous retournez ensuite la chaîne que vous souhaitez utiliser comme nouvelle valeur de la fonction.

L'exemple suivant utilise attr () avec une fonction de rappel:

Exemple:

```
$("#button").click(function(){  
    $("#ugb").attr("href", function(i, origValue){  
        return origValue + "/sat";  
    });  
});
```

jQuery et Html

- **Ajout du contenu HTML**

On se penche sur quatre méthodes jQuery utilisées pour ajouter du contenu HTML:

- **append ()** – Insère du contenu à la fin des éléments sélectionnés
- **prepend ()** – Insère du contenu au début des éléments sélectionnés
- **after ()** - insère du contenu après les éléments sélectionnés
- **before ()** - Insère du contenu avant les éléments sélectionnés

Exemple:

```
$("p").append("Du texte à ajouter à la fin.");  
$("p").prepend("Du texte à ajouter au début");  
$("p").after("Du texte à ajouter après.");  
$("p").before("Du texte à ajouter avant.");
```

NB: Plusieurs textes séparés par des virgules peuvent être ajoutés.



jQuery et Html

- **Suppression d'élément**

Pour supprimer des éléments et du contenu, il existe principalement deux méthodes jQuery:

- Remove () - Supprime l'élément sélectionné (et ses éléments enfants)
- Empty () - Supprime les éléments enfants de l'élément sélectionné

Exemple:

```
$("#div1").remove();  
$("#div1").empty();
```

- **Filtrer les éléments à supprimer**

La méthode remove () peut prendre un paramètre pour filtrer les éléments à supprimer. Le paramètre peut être l'une des syntaxes du sélecteur jQuery.

Exemple:

\$("#p").remove(".test"); supprime tous les élt <p> de class = "test":

\$("#p").remove(".test, .demo"); supprime tous les éléments <p> avec class = "test" et class = "demo"

jQuery et Html

- **Manipulation du CSS**

jQuery dispose de plusieurs méthodes de manipulation CSS dont :

- **AddClass ()** - Ajoute une ou plusieurs classes aux éléments sélectionnés
- **RemoveClass ()** - Supprime des classes d'éléments sélectionnés
- **ToggleClass ()** - Bascule entre l'ajout et la suppression de classes d'éléments sélectionnés
- **Css ()** - Définit ou renvoie l'attribut de style

- **Exemple de feuille de style CSS**

```
.important {  
    font-weight: bold;  
    font-size: xx-large;  
}
```

```
.blue {  
    color: blue;  
}
```

jQuery et Html

- Manipulation du CSS

La méthode `addClass()`

L'exemple suivant montre comment ajouter des attributs de classe à différents éléments. On peut sélectionner plusieurs éléments, lors de l'ajout de classes:

Exemple :

```
$("#button").click(function(){  
    $("h1, h2, p").addClass("blue");  
    $("div").addClass("important");  
});
```

On peut également spécifier plusieurs classes dans la méthode `addClass()`:

Exemple :

```
$("#button").click(function(){  
    $("#div1").addClass("important blue");  
});
```

jQuery et Html

- Manipulation du CSS

La méthode `removeClass()`

Elle permet de supprimer un attribut de classe spécifique de différents éléments:

Exemple :

```
$("#button").click(function(){  
    $("#h1, h2, p").removeClass("blue");  
});
```

La méthode `toggleClass()`

Elle permet de basculer entre l'ajout et la suppression de classes des éléments sélectionnés:

Exemple :

```
$("#button").click(function(){  
    $("#h1, h2, p").toggleClass("blue");  
});
```

jQuery et Html

Manipulation du CSS

- La méthode **css()**

Elle définit ou retourne la valeur d'une ou plusieurs propriétés de style.

Retourner la valeur d'une propriété CSS

Syntaxe: `css(" propertyname ");`

Exemple : Il retourne la couleur d'arrière-plan du premier l'élément p.

```
$("p").css("background-color");
```

Définir une propriété CSS

Syntaxe : `css(" propertyname ", " value ");`

Exemple : Pour définir la couleur d'arrière-plan pour tous les éléments p:

```
$("p").css("background-color", "yellow");
```

jQuery et Html

Manipulation du CSS

- La méthode `css()`

Définir plusieurs propriétés CSS

Syntaxe : `css({" propertyname ":" value "," propertyname ":" value ",...});`

L'exemple suivant définit une couleur d'arrière-plan et une taille de police pour tous les éléments p:

Exemple :

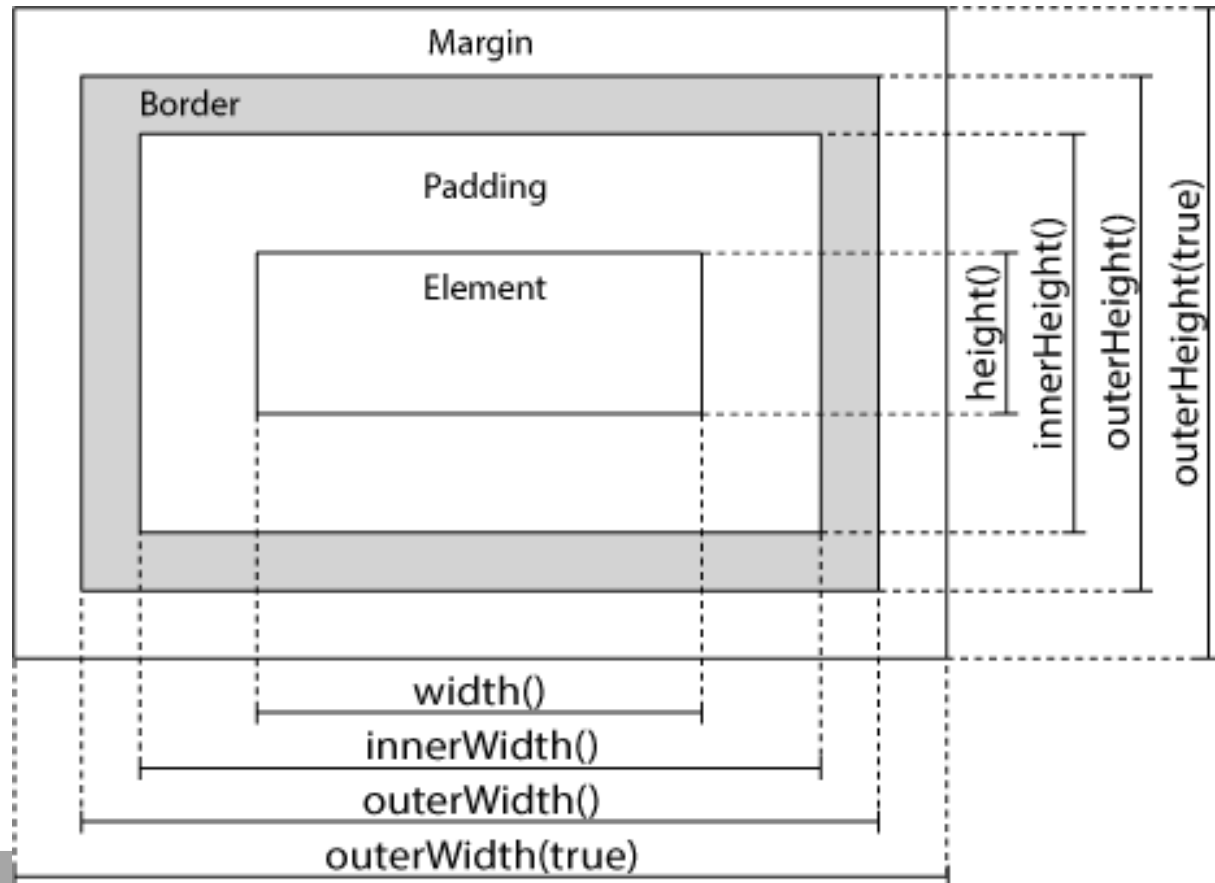
```
$("p").css({"background-color": "yellow", "font-size": "200%"});
```


jQuery et Html

Les dimensions

Avec jQuery, il est facile de travailler avec les dimensions des éléments et la fenêtre du navigateur en utilisant les méthodes suivantes :

- **width()**
- **height()**
- **innerWidth()**
- **innerHeight()**
- **outerWidth()**
- **outerHeight()**



jQuery et Html

Les dimensions

- Les méthodes **width ()** et **height ()**
 - La méthode **width ()** définit ou renvoie la largeur d'un élément (à l'exclusion du remplissage, de la bordure et de la marge).
 - La méthode **height ()** définit ou renvoie la hauteur d'un élément (à l'exclusion du remplissage, de la bordure et de la marge).

Exemple

```
$("#button").click(function(){  
    var txt = "";  
    txt += "Width: " + $("#div1").width() + "</br>";  
    txt += "Height: " + $("#div1").height();  
    $("#div1").html(txt);  
});
```



jQuery et Html

Les dimensions

- Les méthodes `innerWidth ()` et `innerHeight ()`
 - La méthode `innerWidth ()` renvoie la largeur d'un élément (y compris le remplissage).
 - La méthode `innerHeight ()` renvoie la hauteur d'un élément (y compris le remplissage)

Exemple

```
$("#button").click(function(){  
    var txt = "";  
    txt += "Inner width: " + $("#div1").innerWidth() + "<br>";  
    txt += "Inner height: " + $("#div1").innerHeight();  
    $("#div1").html(txt);  
});
```

jQuery et Html

Les dimensions

- Les méthodes `outerWidth ()` et `outerHeight ()`
 - La méthode `outerWidth ()` renvoie la largeur d'un élément (y compris le remplissage et la bordure).
 - La méthode `outerHeight ()` renvoie la hauteur d'un élément (y compris le remplissage et la bordure).

Exemple

```
$("#button").click(function(){  
    var txt = "";  
    txt += "Outer width: " + $("#div1").outerWidth() + "</br>";  
    txt += "Outer height: " + $("#div1").outerHeight();  
    $("#div1").html(txt);  
});
```

jQuery et Html

Les dimensions

- Les méthodes `outerWidth ()` et `outerHeight ()`
 - La méthode `outerWidth (true)` renvoie la largeur d'un élément (y compris le rembourrage, la bordure et la marge).
 - La méthode `outerHeight (true)` renvoie la hauteur d'un élément (y compris le remplissage, la bordure et la marge).

Exemple

```
$("#button").click(function(){  
    var txt = "";  
    txt += "Outer width (+margin): " + $("#div1").outerWidth(true) + "</br>";  
    txt += "Outer height (+margin): " + $("#div1").outerHeight(true);  
    $("#div1").html(txt);  
});
```

jQuery et Html

Les dimensions

- Largeur d'un document et de la fenêtre du navigateur

Exemple

```
$("#button").click(function(){  
    var txt = "";  
    txt += "Document width/height: " + $(document).width();  
    txt += "x" + $(document).height() + "\n";  
    txt += "Window width/height: " + $(window).width();  
    txt += "x" + $(window).height();  
    alert(txt);  
})
```

Exemple : (la largeur et la hauteur d'un élément <div> spécifié)

```
$("#button").click(function(){  
    $("#div1").width(500).height(500);
```

