



LA MANU

L'ÉCOLE DES MÉTIERS DU NUMÉRIQUE
Manufacture de compétences



SUPPORT APPRENANT

JQuery



CONFIDENTIEL

*Ce document est strictement confidentiel et ne doit pas être diffusé
sans accord préalable écrit*

JQuery

Ce que l'on a vu...



JQuery est une bibliothèque du langage JavaScript. En développement, une bibliothèque correspond à un ensemble d'instructions permettant d'enrichir et de faciliter le développement d'applications. La bibliothèque jQuery a été créée en 2006 par John RESIG et comme son slogan l'indique, elle permet d'écrire moins de code tout en faisant plus de choses. Elle permet de parcourir et modifier le DOM (Document Object Model, soit le document HTML représenté sous forme d'objets), de manipuler des feuilles de style CSS, de faciliter l'utilisation de l'Ajax, ainsi que d'utiliser des plugins. Le langage JavaScript est utilisé par la majorité des sites web, ainsi l'utilisation d'une bibliothèque telle que jQuery peut être bien pratique. JQuery est la distribution la plus répandue, ce qui en fait une bibliothèque pérenne. Elle a également l'avantage de faire abstraction des différences d'implémentation du JavaScript selon les navigateurs, permettant ainsi d'avoir un site compatible avec un grand nombre de navigateurs.

La bibliothèque jQuery commence à être un peu moins utilisée, cependant elle est encore très répandue du fait notamment qu'elle est utilisée par de nombreux plugins ou frameworks. En effet, s'il est nécessaire d'inclure jQuery à son projet pour utiliser Bootstrap par exemple, autant utiliser la bibliothèque.

Pour utiliser jQuery il faut inclure au document HTML le fichier script jQuery contenant toutes les instructions de la bibliothèque. Ce script est à placer à la fin de la balise **<body>** avant votre propre fichier script. Sans cela, le code jQuery mis dans votre script ne pourra pas être interprété correctement. Comme pour le framework Bootstrap, la bibliothèque jQuery peut être ajoutée au projet de 2 façons :

- Soit en utilisant le CDN (Content Delivery Network) notamment disponible à l'adresse suivante : <http://code.jquery.com/>
- Soit en téléchargeant le fichier jQuery à ajouter au projet, disponible à l'adresse : <https://jquery.com/download/>

La bibliothèque jQuery peut être appelée dans le script via la fonction **jQuery** ou via son alias **\$**. Pour faire appel à une méthode jQuery on utilisera la syntaxe suivante **\$(sélecteur).méthode()** (ex : `$('#text').hide()` ;). L'utilisation des sélecteurs (les mêmes qu'en CSS) permet de simplifier le code par rapport au JavaScript comme le montre les exemples du tableau suivant :

JavaScript	jQuery
<code>var text = document.getElementById('text');</code>	<code>var text = \$('#text');</code>
<code>var text = document.getElementsByClassName('text');</code>	<code>var text = \$('.text');</code>
<code>document.getElementById('lastname').value ;</code>	<code>\$('#lastname').val() ;</code>
<code>document.getElementById('text').style.display = 'none' ;</code>	<code>\$('#text').hide() ;</code>
<code>document.querySelectorAll('p').style.display = 'none';</code>	<code>\$(p).hide() ;</code>

Tout comme en JavaScript, il est possible de déclarer des fonctions assignées à des événements. Pour cela il n'y aura pas besoin de passer par un attribut d'évènement à ajouter dans le document HTML ou par des écouteurs d'évènements. Il faudra utiliser la syntaxe suivante :

```
$(sélecteur).événement(function () {  
    instructions  
});
```

Une bonne pratique lorsqu'on utilise jQuery consiste à mettre son code dans la fonction suivante :

```
$(document).ready(function () {  
    Fonctions/instructions  
})
```

Cela permet que les fonctions déclarées dans cette fonction s'exécutent seulement une fois que tous les éléments du DOM (Document Object Model) aient été correctement chargés, évitant ainsi qu'une fonction jQuery ne tente d'accéder à un élément HTML qui ne serait pas encore chargé.

Quelques méthodes...

Méthodes JQuery	Description
<code>\$('#sélecteur').hide();</code>	Masque le sélecteur.
<code>\$('#sélecteur').show();</code>	Affiche le sélecteur.
<code>\$('#sélecteur').val();</code>	Définit ou retourne la valeur du sélecteur.
<code>\$('#sélecteur').html();</code>	Définit ou retourne le contenu du sélecteur (y compris le balisage HTML).
<code>\$('#sélecteur').attr('attribut');</code>	Retourne la valeur de l'attribut du sélecteur.
<code>\$('#sélecteur').append('contenu');</code>	Insère un contenu à la fin du sélecteur (y compris le balisage HTML).
<code>\$('#sélecteur').prepend('contenu');</code>	Insère un contenu au début du sélecteur (y compris le balisage HTML).
<code>\$('#sélecteur').remove();</code>	Supprime le sélecteur et son contenu.
<code>\$('#sélecteur').empty();</code>	Vide uniquement le contenu du sélecteur.
<code>\$('#sélecteur').addClass();</code>	Ajoute une classe (ou plusieurs) au sélecteur.
<code>\$('#sélecteur').click(function() { instructions });</code>	Fonction exécutée au clic sur le sélecteur.
<code>\$('#sélecteur').dblclick(function() { instructions });</code>	Fonction exécutée au double-clic sur le sélecteur.
<code>\$('#sélecteur').mouseenter(function() { instructions });</code>	Fonction exécutée à l'entrée du pointeur de la souris sur le sélecteur.
<code>\$('#sélecteur').mouseleave(function() { instructions });</code>	Fonction exécutée à la sortie du pointeur de la souris sur le sélecteur.
<code>\$('#sélecteur').hover(function() { instructions }, function() { instructions });</code>	Combinaison des méthodes mouseenter() et mouseleave() ; 1 ^{ère} fonction s'exécute à l'entrée du pointeur de la souris sur le sélecteur et 2 ^{ème} fonction s'exécute à la sortie du pointeur de la souris sur le sélecteur.
<code>\$('#sélecteur').keypress(function() { instructions });</code>	Fonction exécutée à l'appuie (et relâche) d'une touche du clavier sur le sélecteur.
<code>\$('#sélecteur').keydown(function() { instructions });</code>	Fonction exécutée à l'appuie (sans relâche) d'une touche du clavier sur le sélecteur.
<code>\$('#sélecteur').keyup(function() { instructions });</code>	Fonction exécutée au relâchement d'une touche du clavier sur le sélecteur.
<code>\$('#sélecteur').blur(function() { instructions });</code>	Fonction exécutée quand le sélecteur perd le focus.
<code>\$('#sélecteur').focus(function() { instructions });</code>	Fonction exécutée quand le sélecteur a le focus.
<code>\$('#sélecteur').on('événement', function() { instructions });</code>	Fonction exécutée lors de l'évènement sur le sélecteur.

Liens utiles

<https://jquery.com/>
<https://openclassrooms.com/en/courses/1567926-un-site-web-dynamique-avec-jquery>
<https://openclassrooms.com/en/courses/1631636-simplifiez-vos-developpements-javascript-avec-jquery>
<https://openclassrooms.com/en/courses/3504441-introduction-a-jquery>
<https://learn.jquery.com/>
<https://www.w3schools.com/jquery/>
<https://www.grafikart.fr/tutoriels/jquery>
<https://developer.mozilla.org/fr/docs/Glossaire/jQuery>
<https://www.alsacreations.com/astuce/lire/916-librairie-javascript-jquery-script.html>

Bonnes pratiques

Rappel des principales bonnes pratiques	
<input type="checkbox"/>	Utilisation des simples quotes.
<input type="checkbox"/>	Utilisation de l'anglais est obligatoire pour les noms de variables, des class, des id...
<input type="checkbox"/>	Mettre le script dans un fichier à part et l'inclure en bas de page HTML avant la balise fermante </body>.
<input type="checkbox"/>	Attention à l'ordre des balises script, mettre le fichier Script.js en dernier.
<input type="checkbox"/>	Utiliser les commentaires.
<input type="checkbox"/>	Vérifier que la saisie utilisateur correspond à ce qui est attendu.
<input type="checkbox"/>	Pour une meilleure compréhension du code, il faut indenter son code.