

## Leçon 4.2 – HTML et jQuery

s1 -----

s2 -----

Dans cette leçon, nous introduisons le framework jQuery. On appelle framework JavaScript une bibliothèque de fichiers JavaScript, rassemblant des fonctions qui permettent de simplifier les instructions, synthétiser certaines tâches répétitives, faire les tests qui adaptent le code aux différents navigateurs, etc. Nous allons voir comment utiliser JavaScript et très brièvement sur quelques exemples en quoi la syntaxe JavaScript est simplifiée.

s3 -----

jQuery est donc un ensemble de fonctions qui permettent notamment de manipuler aisément les éléments du DOM, c'est-à-dire de la page HTML et le code CSS associé.

jQuery permet notamment la création d'animations et d'effets, d'effectuer des opérations complexes sur les chaînes de caractères, de mettre en place aisément une communication client-serveur (à travers l'architecture AJAX), etc.

jQuery est vaste : nous n'allons voir ici que quelques fonctionnalités ou méthodes qui donneront un avant-goût de sa puissance.

s4 -----

Comment mettre en oeuvre jQuery ?

Une première solution est de télécharger la dernière version du framework. jQuery apparaît sous la forme d'un fichier jquery.js, le nom de ce fichier étant éventuellement enrichi avec un numéro de version. On déplace ce fichier à l'intérieur du dossier contenant les autres fichiers de notre projet en cours. Il faut ensuite déclarer ce fichier dans l'en-tête du fichier HTML comme un fichier JavaScript standard présent localement.

La seconde façon est de déclarer jQuery en fournissant une localisation sur le web. Cette dernière façon de faire présente l'avantage pour l'internaute de ne devoir télécharger le fichier qu'une seule fois (le fichier sera dans le cache du navigateur) et de toujours disposer de la dernière version du framework.

s5 -----

Voyons sur un exercice comment utiliser jQuery.

Reprenons le jeu des pays qui a été développé dans la partie précédente et modifions l'en-tête du fichier HTML pour pouvoir utiliser jQuery

Ensuite, dans le fichier JS : changeons les instructions de type `document.getElementById` de quelque chose

par  
`$('# quelque chose ')`...

Testons pour voir si cette simplification apportée au code fonctionne.

s6 -----

Que se passe-t-il ?

jQuery repose sur une seule fonction : `jQuery()` ou plus simplement `$()`. Comme toutes les fonctions JavaScript, cette fonction accepte des paramètres, et retourne un objet. Si on passe une chaîne de caractères comme précédemment, on accède à un objet du DOM, c'est-à-dire de la page HTML. C'est ce que nous venons de faire en écrivant `$('#div1')`, l'utilisation du symbole `#` pour identifier un élément est la même que pour un fichier de style CSS.

On peut passer comme argument une classe donnée, par exemple `classe1`.

En écrivant `$(".classe 1)` on cible tous les éléments du DOM possédant cette classe.

Plus largement, et c'est ce qui fait l'intérêt de jQuery, on va pouvoir passer comme arguments des sélecteurs complexes suivant la même logique que celle décrite dans la leçon sur CSS : ici on cible tous les éléments de type paragraphe et de classe `classe1` contenus dans le conteneur `div1` et on utilise la méthode `hide()` qui va rendre invisibles tous ces éléments. Une écriture donc très efficace que vous pouvez tester avec le petit exemple de code HTML proposé.

On peut aussi passer l'ensemble d'une fonction

```
$(function(){  
  // ...  
});
```

et jQuery va exécuter cette fonction une fois que la page sera chargée par le navigateur. C'est la raison pour laquelle on encapsule souvent l'ensemble du code JavaScript à l'intérieur des accolades précédentes lorsqu'on veut être sûr d'éviter des problèmes entre JavaScript et HTML au chargement de la page. Cette instruction est l'équivalent de l'instruction `document.onload()` utilisée dans la leçon précédente.

s7 -----

Ceci nous amène au deuxième exercice : on copie le code JavaScript écrit précédemment (attention : pas les fonctions) à l'intérieur de `window.onload()` à l'intérieur de l'instruction suivante

```
$(function){
```

on n'oublie pas de refermer correctement la parenthèse et l'accolade qui viennent ainsi d'être ouvertes : l'accolade pour les instructions de la fonction, la parenthèse pour l'objet jQuery

Troisième exercice : testons les méthodes : `after()` et `before()` qui permettent d'introduire du contenu après et avant un conteneur existant (mais à l'extérieur de ce dernier).

s8 -----

Ensuite, nous allons ajouter un timer.

Pour ce faire, on utilise la fonction JavaScript `setInterval`. Cette fonction, une fois lancée, déclenche à intervalles réguliers une autre fonction.

Le code est très simple

on initialise une variable `timerIni` à zéro.

setInterval lance à intervalles réguliers - ici toutes les 1000 millisecondes - une fonction, ici la fonction que nous avons appelée Horloge.

Que fait cette fonction Horloge ? Tout simplement, elle incrémente la variable timerIni et met à jour l'élément heure du document HTML

N'oubliez pas de créer cet élément avec l'identifiant heure dans le fichier HTML !

Une remarque : cet exercice nous donne l'occasion d'introduire la fonction text() qui est un tout petit peu différente de la fonction html(). Tandis que la fonction html() traite une chaîne de caractères comme une chaîne HTML contenant éventuellement des balises, la fonction text() traite une chaîne comme du texte : toute mise en forme à l'aide de balise va disparaître ; ne demeurera que le texte brut.

Une fois que vous aurez créé ce timer, je vous propose d'ajouter un test : après 60 secondes, on masque le bouton Valider (on utilisera la méthode jQuery hide()) et on affichera directement la solution.

Enfin, nous allons ajouter des instructions à la fonction reset qui démasquera le bouton Valider (on utilisera la méthode jQuery show()) et relancera le timer

C'est à vous !

s9 -----

Pour finir cet aperçu très rapide de jQuery, nous vous proposons un deuxième exercice : toujours dans le jeu des pays, on souhaite créer les listes déroulantes grâce à une boucle et en utilisant la méthode append() qui insère un contenu à l'intérieur d'un conteneur après le dernier élément trouvé (on parle de dernier enfant du conteneur parent).

n'oubliez pas de modifier dynamiquement autant de fois qu'il y a de pays les attributs id et name de ces nouvelles listes

C'est à vous !

s10 -----

À titre d'information, un petit mot sur jQuery UI. jQuery UI complète jQuery en offrant des méthodes facilitant grandement la réalisation d'interface utilisateur comprenant notamment des onglets, des accordéons, des menus, des sélecteurs de date, etc.

jQuery UI sort du cadre de ce module, mais si cela vous intéresse vous trouverez plusieurs tutoriels sur Internet.

s11 -----

Voici donc pour cette courte introduction à jQuery. Cette leçon clôt également la première partie de ce module à l'occasion de laquelle nous avons introduit HTML, CCS et JavaScript.

Je vous donne rendez-vous dans la deuxième partie de ce module, consacrée à HTML5 et à quelques-unes des nouveautés qu'il apporte.

À très bientôt !