**JQUERY CE QU’IL FAUT SAVOIR**

Définition

jQuery est ce qu’on appelle une « librairie » ou une « bibliothèque » JavaScript. Le rôle d’une librairie, en informatique, est de simplifier l’utilisation d’un certain langage de programmation en fournissant un ensemble de codes déjà prêts à l’emploi.

Pourquoi utiliser

Utiliser jQuery va donc pouvoir être très avantageux pour nous à deux niveaux différents : on va d’une part gagner beaucoup de temps de développement en utilisant les outils mis à notre disposition par cette librairie et d’autre part être assuré de la validité de notre code.

En effet, les créateurs de jQuery s’occupent de la validité parfaite de leur code pour l’ensemble des navigateurs les plus utilisés.

En utilisant jQuery, nous n’avons donc plus à nous soucier de la compatibilité inter-navigateurs et interversions de notre code puisque les créateurs de jQuery ont déjà réfléchi à ce problème à notre place.

L’utilité de jQuery

* La manipulation du DOM HTML et CSS
* La gestion d’événements
* La création d’effets et d’animations
* L’envoi de requêtes Ajax

**jQuery est-il une alternative au JavaScript ?**

jQuery ne doit pas être vu comme une alternative au JavaScript mais plutôt comme un ajout : nous allons utiliser jQuery au côté du JavaScript classique dans nos scripts.

Par ailleurs, jQuery ne dispense pas et ne doit pas remplacer l’apprentissage du JavaScript ni être vu comme un « raccourci » pour apprendre le JavaScript.

Au contraire : pour tirer avantage de jQuery, il est fortement recommandé d’avoir déjà une bonne connaissance et une bonne maîtrise du langage JavaScript et de son fonctionnement général.

De même, jQuery n’est pas non plus une solution miracle qui va faire tout le travail à votre place. Cette bibliothèque est très bien faite et c’est un outil très puissant, mais il vous faudra tout de même vous impliquer et réfléchir afin de créer des scripts performants et bien construits.

## Rappels sur l’AJAX

Le terme AJAX est l’abréviation de “Asynchronous JavaScript and XML”. L’AJAX n’est pas un langage de programmation mais correspond plutôt à un ensemble de techniques utilisant des technologies diverses pour communiquer avec le serveur (notamment envoyer et récupérer des données) de façon asynchrone, c’est-à-dire sans avoir à recharger la page.

Avant l’utilisation de code asynchrone et l’AJAX, la moindre action de l’utilisateur (envoi ou demande de données) résultait par le chargement d’une nouvelle page envoyée par le serveur. Ce processus était inefficace, lent, et peu agréable pour l’utilisateur.

Cette situation a amené à la création d’un objet nommé XMLHttpRequest permettant aux navigateurs de communiquer avec le serveur sans nécessiter de rechargement de page. Cet objet a été au coeur de l’AJAX pendant des années et est aujourd’hui encore très utilisé. A sa création, l’AJAX utilisait les technologies suivantes qui lui ont donné son nom :

* Le XML pour l’échange de données avec le serveur ;
* L’objet XMLHttpRequest pour la communication asynchrone ;
* Le JavaScript pour afficher les données de manière dynamique et permettre à l’utilisateur d’interagir avec les nouvelles informations ;
* Le HTML et le CSS pour la présentation des données.

Aujourd’hui, le XML a été largement délaissé au profit du JSON (JavaScript Object Notation) qui est une notation qui permet d’échanger des données relativement simplement tandis que l’objet XMLHttpRequest est lentement en train de laisser sa place à la nouvelle API Fetch.

“L’AJAX” ou plutôt “l’Ajax” est aujourd’hui un terme générique utilisé pour désigner toute technique côté client (côté navigateur) permettant d’envoyer et de récupérer des données depuis un serveur et de mettre à jour dynamiquement le DOM sans nécessiter l’actualisation complète de la page.

## Ajax et jQuery

Les demandes Ajax sont déclenchées par du code JavaScript : très schématiquement, on envoie une requête au serveur et on met en place des gestionnaires d’événements avec des fonctions de rappel pour traiter la réponse renvoyée.

Comme l’Ajax fait appel à différentes technologies, cependant, l’implémentation d’un navigateur à l’autre peut être différente et un même code peut donc provoquer des comportements différents.

jQuery fournit ici un support Ajax censé harmoniser ces différences et faire en sorte que nos requêtes s’exécutent de la même manière avec tous les navigateurs. jQuery met notamment à notre disposition une méthode globale $.ajax() et des méthodes comme $.get(), $.getScript(), $.getJSON() ou $.post() pour effectuer nos différentes opérations.

Le DOM (Document Object Model ou Modèle Objet de Document en Français) est une interface créée automatiquement par le navigateur lors du chargement d’une page HTML.

## La gestion des événements en JavaScript (rappels)

Un événement correspond à une action qu’on va pouvoir intercepter et à laquelle on va pouvoir réagir. Un événement est un signal qui indique que “quelque chose vient de se passer”.

Ces actions vont généralement provenir soit du navigateur lui même, soit des utilisateurs. Les événements load et DOMContentLoaded par exemple qui se déclenchent lorsque la page complète est chargée ou lorsque le DOM de la page est chargée tandis que l’événement click se déclenche lorsqu’un utilisateur clique sur un élément de la page.

Au moment où une telle action se produit, on dit que l’événement associée à l’action est déclenché (“fired” en anglais). Nous allons alors pouvoir “écouter” ou “intercepter” l’événement et y répondre en indiquant quoi faire dans une fonction gestionnaire d’événements.