=MÉMO 1 - jQUERY

Cours Javascript / jQuery

Frédéric LOSSIGNOL

Rappel

Nous avons vu précédemment comment bâtir de l'interaction avec Javascript :

- la sélection d'éléments HTML
- les modifications courantes du DOM
- les gestionnaires d'évènements

Maintenant nous allons aborder les librairies et plus particulièrement la librairie jQuery.

Pourquoi utiliser des librairies ?

Avant les librairies, il n'était pas rare devoir coder une solution pour chaque navigateurs (Internet Explorer, Firefox, Opera, Safari…) afin que notre code Javascript soit compatible.

jQuery, comme les autres librairies JS, *a été conçu initialement pour répondre principalement à la problématique :* **Gérer la compatibilité** entre TOUS les navigateurs. Avec une librairie, ***le développeur dispose ainsi de propriétés et méthodes compatibles sur toutes les plateformes***.

D'autres avantages découlent naturellement de l'utilisation des librairies :

- **L'écriture simplifiée du code (sélection, modification du DOM, évènements**
- **La gestion des animations, complexe à écrire en pure Javascript**
- **Les requêtes asynchrones de type AJAX (get, post, put, delete)**

SOMMAIRE

1. jQuery or not jQuery
    1. Pourquoi utiliser jQuery
    2. Charger jQuery comme un fichier JS
    3. Charger jQuery depuis un CDN

2. La sélection avec jQuery
    1. Sélection simple
    2. Sélection avancée

3. Les méthodes jQuery les plus courantes
    1. ***lire / modifier le contenu : .text(), .html()***
    2. ***afficher / masquer des éléments : .hide(), .show()***
    3. ***lire / modifier des attributs html : .attr(), .data()***
    4. ***ajouter/retirer des class : .addClass(), .removeClass, .toggleClass(), hasClass()***
    5. ***Lire / Modifier du CSS iniline : .css()***
    6. ***gérer les formulaires :***

4. Les animations
    1. Les animations simples

        .fadeIn(), .fadeOut(), slideDown(), .slideUp()

        A venir dans le cours suivant ...

    2. Les animations avec la méthode .animate()
    3. Une autre voie pour gérer les animations

        Animer en CSS en modifiant les class

1 jQuery or not jQuery

1.1 Pourquoi utiliser jQuery ?

Comme expliqué en introduction, l'opportunité d'utiliser les librairies a 3 raisons :

- ***Si on veut rendre le code compatible entre tous les navigateurs*** sans besoin d'ajouter des polyfills (des patchs de code pour répondre à la problématique de compatibilité sur un navigateur particulier).

- ***Si on souhaite écrire plus simplement*** et plus rapidement du code (sélection, modification du DOM, gestion des évènements).

- ***Si on a des animations à gérer,*** simples ou avancées.

Concrètement, qu'apporte jQuery ?
jQuery offre un un supplément de propriétés et de méthodes sur tous les objets de type String, Number, Array, Object.

Ces propriétés et méthodes supplémentaires au langage JAVASCRIPT vont permettre au développeur de coder plus intuitivement et plus simplement.

exemple en JS pour poser un écouteur d'événement sur plusieurs élément li :

```
// 1 On sélectionne tous les li
var itemsArray = document.getElementsByTagName('li');


// 2 On pose un écouteur d'événement sur chacun des li dans une boucle for
for(var i=0; i<itemsArray.length; i++) {
   itemsArray[i].addEventListener('click', uneFonction);
}
```

Même code avec jQuery :

```
// 1 On sélectionne tous les li
var items = $('li');
// 2 On pose un écouteur d'événement sur chacun des li dans une boucle for
items.on('click', uneFonction);
```

Pour charger jquery, rien de plus simple, vous devez l'inclure dans votre page html de la même façon que vous chargez un fichier .js externe. Vous disposez de 2 méthodes pour charger jQuery, soit en téléchargeant la dernière version depuis le site jQuery et en l'incluant dans votre page, soit depuis un CDN (Content Delivery Network).

> ***NB*** : Il faut toujours charger jQuery AVANT de l'utiliser, ou encapsuler votre code dans une fonction jQuery pour lui indiquer qu'il ne s'exécutera que lorsque jQery sera bien chargé.

1.2 *Charger jQuery depuis le fichier jquery.js*

- Télécharger la version souhaitée de jQuery sur http://jquery.com/download/
- Inclure jQuery dans votre page
    - soit dans dans les balises **<head>**,
    - soit **juste avant la balise de fermeture </body>**

```
<head>
     ...
     <script src="jquery.js"></script>
     <script src="monscript.js"></script>
</head>
```

```
<body>
     ...
     <script src="jquery.js"></script>
     <script src="monscript.js"></script>
</body>
```

1.3 *Charger jQuery depuis un CDN*

L'avantage d'utiliser un CDN : la performance.

Comment ça marche ?

Beaucoup de sites web chargent jQuery depuis un CDN. Si votre site charge jQuery depuis un CDN courant, il est très probable que ce fichier soit déjà en cache dans le navigateur de vos visiteurs. Ainsi vous évitez une requête supplémentaire et jQuery sera servi depuis le cache du navigateur. Les CDN courants qui proposent jQuery

- Google CDN
- Microsoft CDN
- CDNJS CDN
- jsDelivr CDN

Exemple : Charger jQuery dans votre page depuis le CDN de Google

(dans le <head> ou avant la balise de fermeture </body>)

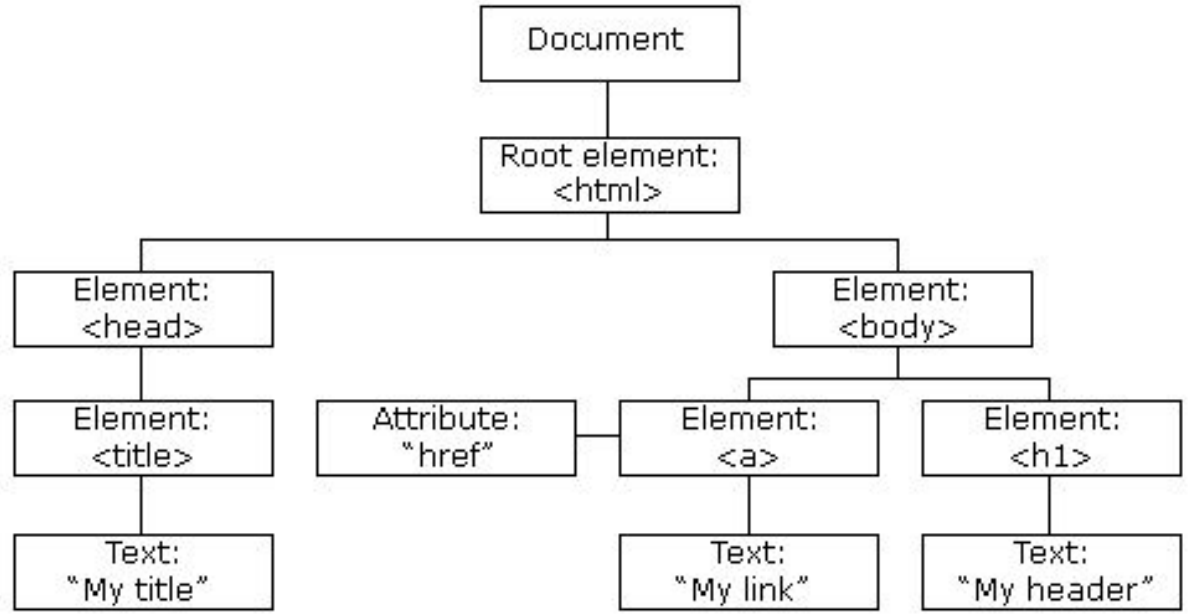
```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
```

Votre page web est maintenant prête à utiliser jQuery.

Vous pouvez coder en utilisant les propriétés et les méthodes proposées par jQuery.

2 La sélection avec jQuery

Rappel : le DOM est une représentation de votre page HTML sous forme d'arbre hiérarchique, stockée en mémoire du navigateur. La sélection des éléments HTML nativement en Javascript ou avec jQuery utilise le DOM.



A retenir :

> le DOM est un standard du W3C qui décrit une interface indépendante de tout langage de programmation et de toute plate-forme, permettant à des programmes informatiques d'accéder ou de mettre à jour le contenu, la structure ou le style de documents HTML et XML.

2.1 La sélection simple avec jQuery

jQuery va permettre de faire de la sélection très simplement par rapport à la sélection native native en Javascript.

Souvenez-vous, pour sélectionner des éléments HTML **avec Javascript** :

.getElementById('idCible')

.getElementsByClassName('classCible')

.getElementsByTagName('ul')

.querySelector() ou .querySelectorAll()

Avec jQuery, on utilise une seule syntaxe :

$(selector CSS)

4 exemples

```
$('h1');  // retourne tous les Elts h1 de la page ou l'Elt h1 s'il n'y en a qu'un
$('.maSuperClass');
// retourne tous les Elts qui ont la class maSuperClass ou un seul s'il n'y en a qu'un
$('#monSuperId');  // retourne l' Elt qui a l'id monSuperId
$("input[name='message']");
 // retourne l'élément input qui a un attribut name='message'
```

ATTENTION : un élément sélectionné avec jQuery, dispose des propriétés et des méthodes jQuery. Cela signifie que vous ne pouvez pas appeler des propriétés et méthodes Javascript sur cet objet.

PS : A quelques exceptions près comme .length qui est une propriété commune aux objets ***String et Array*** de JS et jQuery).

+

2.2 La sélection avancée

La sélection simple fonctionne parfaitement. Cependant, nous pouvons avoir besoin de sélectionner des éléments de façon avancée.

14 méthodes de sélection avancée :

Toutes les méthodes suivantes ne sont applicables QUE sur des éléments (ou noeuds HTML), préalablement sélectionnés AVEC jQuery.

.children(); *// renvoie les descendants directs, optionnellement filtrés par un sélecteur en param*

.parent(); *// renvoie le parent*

.siblings() *// Renvoie tous les frères*

.find(); *// parcourt tous les descendants pour trouver l'élément fourni en paramètre*

next(); *// renvoie le voisin (frère) suivant*

prev(); *// renvoie le voisin (frère) précédent*

.first(); *// Renvoie le premier élément d'une collection*

.last(); *// Renvoie le dernier élément d'une collection*

.***eq();*** *// renvoie, parmi une liste d'Elts (collection), celui qui se trouve à l'index fourni en paramètre*

.visible() ou .hidden() *// Renvoie les éléments visibles ou cachés*

.***not();*** *// retire les éléments sélectionnés en paramètre d'une collection d'éléments*

3 Les méthodes jQuery les plus courantes

Maintenant que vous savez sélectionner des éléments (ou noeuds HTML) avec jQuery, vous pouvez utiliser des méthodes jQuery pour modifier le DOM.

3.1 Lire / Modifier le contenu de texte ou HTML

.text()

// permet de lire le contenu texte d'un élément

.text('Texte à insérer dans la balise');

// permet d'écrire un contenu texte dans un élément cible

.html()

// permet de lire le contenu html d'un élément

.html('<h1>Super titre</h1>');

// permet d'écrire un contenu html dans un élément cible

***exemple** :*

```
/*
 * Cibler l'elt div qui a la class .header et injecter du HTML à l'intérieur
*/
$('div.header').html('<h1>Comment insérer du contenu avec jQuery ?</h1>');
```

Astuce : pour ajouter du texte ou html sans remplacer tout le contenu de l'élément cible, utiliser la méthode .append()

3.2 Afficher / Masquer un élément HTML

elt.show()

// permet de montrer un élément

Elt.hide()

// permet de cacher un élément

```
/*
 * Masquer un élément div
*/
$('div').hide();


/*
 * Montrer un élément div
*/
$('div').show();
```

Comment ça marche :

.hide() ajoute un style='display:none' sur l'élément (style inline)

.show() ajoute un style='display:block' sur l'élément (style inline)

3.3 Lire / Modifier des attributs HTML

.attr("src")

// permet de lire la valeur de l'attribut src de l'élément sélectionné

.attr("alt", "super image")

// permet de modifier la valeur de l'attribut alt de l'élément sélectionné

.data("firstname")

// permet de lire la valeur d'un l'attribut qui aurait pour nom data-id

.data("firstname", "Fred")

// permet de modifier la valeur d'un l'attribut qui aurait pour nom data-id

Lien : comprendre l'intérêt des attributs data et leur utilisation

https://www.alsacreations.com/article/lire/1397-html5-attribut-data-dataset.html

3 exemples de manipulation des attributs *:*

```
// Modifier à la volée la valeur d'un attribut src d'une balise image
$('img').attr('src', 'bateau.jpg');

// Récupérer la valeur d'un attribut id d'une balise h1
$('h1').attr('id');

// Récupérer la valeur d'un attribut data-gps d'une balise div qui a la class address
$('div.address').data('gps');
/*
 * Même code en JAVASCRIPT
 * document.querySelector('div.address').dataset.gps;
*/
```

3.4 Ajouter / Retirer des Class / Tester si une class est présente

.addClass(nomDeLaClass)

// Ajouter une class à un élément HTML.

.removeClass(nomDeLaClass)

// Supprimer une class d'un élément HTML

.toggleClass(nomDeLaClass)

// Aajouter et supprimer alternativement une class à un élément HTML

.hasClass(nomDeLaClass)

// Tester si un élément a une class spécifiée en paramètre (retourne TRUE ou FALSE)

***exemples** :*

```
/*
 * Ajouter la class selected aux éléments li
*/
$('li').addClass('selected');
```

```
/*
 * retirer la class selected aux éléments li
*/
$('li').removeClass('selected');
```

3.5 Lire / Modifier du CSS

.css(nomDeLaProprieteCss)

// retourne la valeur.

.css(nomDeLaProprieteCss, valeur)

// Set une propriété CSS inline sur l'élément sélectionné

***exemples** :*

```
/*
 * Lire la propriété color d'un Elt <p>*/
$('p').css('color');
```

```
/*
 * Modifier la propriété CSS font-weight d'un Elt H1*/
$('h1').removeClass('font-weight', 'bold');
```

3.6 Gérer les formulaires

Pour gérer les formulaires avec jQuery, vous devez :

- Savoir détecter l'évènement quand l'utilisateur soumet le formulaire
- Savoir récupérer les valeurs des inputs, textarea, select, …

<u>Comment détecter l'évènement quand l'utilisateur soumet le formulaire</u> ?

Grâce à l'événement submit

Exemple :

```
$('form').on('submit', function(e) {
    // empêcher la soumission du formulaire de recharger la page courante
    e.preventDefault();
} );
```

> ***Attention : quand on détecte la soumission du formulaire, on doit stopper le comportement naturel d'un formulaire HTML avec la méthode .preventDefault() disponible sur l'objet event.***

<u>Comment récupérer la valeur d'un input</u> ?

Grâce à la méthode .val()

Exemple :

```
$('form').on('submit', function(e) {
    // empêcher la soumission du formulaire de recharger la page courante
    e.preventDefault();
    // Récupérer la valeur d'un input
    var saisieUtilisateur = $('input').val();
} );
```

4 Les animations

4.1 Les animations simples

.fadeIn()

// fait apparaître l'Elt sélectionné de manière progressive.

.fadeOut()

// fait disparaître l'Elt sélectionné de manière progressive.

.fadeToggle)

// fait apparaître et disparaître alternativement l'Elt sélectionné de manière progressive.

.slideDown()

// fait apparaîtrre l'Elt sélectionné en glissé

.slideUp()

// fait disparaîtrre l'Elt sélectionné en glissé

.slideToogle)

// fait apparaître et disparaître alternativement l'Elt sélectionné en glissé

Le Pdf Suivant traitera des autres manières de gérer les animations :

- ***avec la fonction animate()***
- ***via la modification des class JS + les animations en CSS***

Les outils du développeur / liens utiles

Les outils du développeur :

- Un éditeur de texte ou un IDE (Sublime Text, Visual Studio Code,...)
- L'outil développement sur les navigateurs (Firebug sur FireFox et l'outil de développement sur Chrome, tous deux accessibles avec le raccourci F12)
- DevDocs (www.devdocs.io) [Le dictionnaire du développeur]
- **Documentation jQuery** : http://api.jquery.com/

Conseil : révisez les notions apprises en relisant ce document et le précédent, relire le code des exemples et des exercices, et pratiquez en implémentant jQuery dans vos propres projets.

Si vous avez des questions ou des remarques, je suis joignable par Email : frederic.lossignol@gmail.com ou https://opinionway-dev.slack.com

Frédéric Lossignol, Formateur développement front-end - back-end,
JS / jQuery / Angular 5 / Ionic / Cordova
PHP7 / NodeJs