



Version	Date	Auteur(s)	Action(s)
1.0	05/11/19	Lécu Régis	Création du document
1.1	11/11/19	Bost Vincent	Amendements

TABLE DES MATIERES

Table des matières	2
1. Introduction	5
2. Premiers pas en jQuery	6
2.1 Télécharger jQuery	6
2.2 Quelques éléments de syntaxe	7
3. Gérer les événements HTML en jQuery	9
3.1 Une première fonction événementielle en jQuery	9
3.2 Liste des événements	9
3.3 Séparer JavaScript et HTML	11
4. Gérer le DOM et les styles en jQuery	12
4.1 Manipuler le DOM en jQuery	12
4.2 Manipuler les styles CSS en jQuery	14
4.3 Parcourir l'arborescence du document avec jQuery	14
4.3.1 Rappels sur le DOM et les parcours d'arbre	14
4.3.2 Mise en pratique	15
4.3.3 Les méthodes de parcours d'arbre	15

Objectifs

A l'issue de cette unité d'apprentissage, vous devriez être capable de :

- Comprendre les avantages de jQuery par rapport à du code JavaScript sans Framework, en terme de facilité de codage et de compatibilité avec les navigateurs web
- D'intégrer jQuery pour simplifier vos développements web

Pré requis

Vous devez posséder les bases d'HTML, de CSS et de JavaScript.

Méthodologie

Ce document peut être utilisé en présentiel ou à distance.

Il précise la situation professionnelle visée par la séance, la situe dans la formation, et guide le stagiaire dans son apprentissage et ses recherches complémentaires.

Mode d'emploi

Symboles utilisés :



Renvoie à des supports de cours, des livres ou à la documentation en ligne.



Propose des exercices ou des mises en situation pratiques.



Point important qui mérite d'être souligné !

Ressources

- Site de référence de jQuery : <https://jquery.com/>
- Cours *w3schools* sur jQuery : <https://www.w3schools.com/jquery/default.asp>

1. INTRODUCTION

La devise de jQuery : « *write less, do more* »

jQuery est une bibliothèque JavaScript qui permet de simplifier énormément le codage du JavaScript dans les pages Web, en particulier les appels au **DOM** (*Document Object Model*) et l'utilisation d'AJAX pour le chargement asynchrone partiel. Nous travaillerons avec Ajax ultérieurement.

De nombreuses tâches complexes en JavaScript vont se réduire à une seule ligne : l'appel d'une méthode jQuery, qui aura le même comportement sur l'ensemble des navigateurs Web.

jQuery fournit une approche cross-plateform qui évite au développeur de se préoccuper des différences d'implémentation fonction de la famille du navigateur et la version.

jQuery simplifie les mises au point.

jQuery permet notamment de :

- manipuler les éléments et attributs HTML, d'une façon plus simple et intuitive qu'avec le DOM
- manipuler les styles CSS
- gérer les événements et leur associer des gestionnaires
- prendre en charge AJAX
- effectuer des effets graphiques et des animations (simples).

La bibliothèque jQuery fournit aussi des utilitaires, des composants d'interface (datepicker, table avec mécanismes de liaison et pagination, ...). Il existe une communauté importante de développeurs qui maintient cette bibliothèque.

2. PREMIERS PAS EN JQUERY

La bibliothèque jQuery est contenue dans un fichier **.js** que l'on peut au choix :

- Télécharger localement au niveau de son application Web
- Référencer en ligne, par un lien vers le site de jQuery

2.1 TELECHARGER JQUERY

<https://jquery.com/download/>

Téléchargez la dernière version non compressée (version développement).

jQuery est fournie sous deux versions : une version compressée qui sera plus rapide en production, et une version décompressée, conseillée pour la mise au point pendant la phase de développement.

Téléchargez la version décompressée ([jquery-3.4.1.js](#)), ou utilisez la version en ligne (souvent plus performant) à partir de jQuery ou depuis un autre site tel que Google par exemple `<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>`

Il est préférable de passer par une requête sur un site externe, le navigateur pouvant gérer plusieurs requêtes simultanées.

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Installation JQuery</title>
  <!-- <script src="bib/jquery-3.4.1.js" type="text/javascript"></script> -->
  <script src="https://code.jquery.com/jquery-3.4.1.js"></script>
</head>
<body>
  <script type="text/javascript">
    document.writeln("jquery, version : " + $.fn.jquery );
  </script>
</body>
</html>
```

Ecrire un script incorporé qui affiche la version de jQuery disponible. Dans le script, **\$** désigne l'objet jQuery

\$.fn.jquery affiche la version de jQuery

Ecrire un script JavaScript en utilisant les fonctions de base de jQuery

Afpa © 2019 – Section Tertiaire Informatique – Filière « Etude et développement »

2.2 QUELQUES ELEMENTS DE SYNTAXE

jQuery a une syntaxe intuitive :

`$(selector).action ()`

- **\$** désigne la bibliothèque jQuery
- **selector** : critère de recherche pour « sélectionner » les éléments HTML sur lesquels va porter la méthode **action**
- **action** : ce qu'il faut faire sur le/les éléments sélectionnés

Un sélecteur peut être :

- l'élément HTML courant désigné par le mot clé *this* :

`$(this).action() ;`

- un élément HTML unique désigné par son id : comme dans un fichier CSS, l'id sera précédé de **#**

Par exemple, pour l'élément *idtest* :

`$("#idtest").action() ;`

- tous les éléments HTML d'un certain type : par exemple, toutes les *div*

`$("div").action() ;`

- tous les éléments HTML qui possèdent une certaine classe CSS : comme dans un fichier CSS, la classe sera précédée de **.**

Par exemple, pour la classe *titre*

`$(".titre").action() ;`

Quelques fonctions pour commencer :

hide / show : cache ou affiche un/plusieurs éléments HTML

text : retourne le texte des éléments sélectionnés (sans les balises HTML)

html : retourne le contenu des éléments sélectionnés (y compris les balises HTML)

val : retourne le contenu des champs texte, dans un formulaire

append : ajoute un élément fils à un container (comme *div*, *table* etc.)

Ecrire un script JavaScript en utilisant les fonctions de base de jQuery

Afpa © 2019 – Section Tertiaire Informatique – Filière « Etude et développement »



Exercice 1 : Prise en main de jQuery

Dans cet exercice, les différentes opérations seront exécutées séquentiellement.

En partant de la page HTML **Exo1.html** mise à disposition.

- Cachez tous les éléments de type « paragraphe »
- Affichez uniquement le paragraphe « p1 ».
- Affichez le texte du paragraphe « p1 » puis changer son texte
- Affichez le contenu du paragraphe « p2 » (HTML compris)



Exercice 2 : Essayez d'autres « sélecteurs » jQuery

Testez les sélecteurs à partir du site *w3schools*. (« try it »)

https://www.w3schools.com/jquery/jquery_selectors.asp

More Examples of jQuery Selectors

Syntax	Description	Example
\$("#")	Selects all elements	Try it
\$(this)	Selects the current HTML element	Try it
\$("#p.intro")	Selects all <p> elements with class="intro"	Try it
\$("#p:first")	Selects the first <p> element	Try it
\$("#ul li:first")	Selects the first element of the first 	Try it
\$("#ul li:first-child")	Selects the first element of every 	Try it
\$("#[href]")	Selects all elements with an href attribute	Try it
\$("#a[target='_blank']")	Selects all <a> elements with a target attribute value equal to "_blank"	Try it
\$("#a[target!='_blank']")	Selects all <a> elements with a target attribute value NOT equal to "_blank"	Try it
\$("#:button")	Selects all <button> elements and <input> elements of type="button"	Try it
\$("#tr:even")	Selects all even <tr> elements	Try it
\$("#tr:odd")	Selects all odd <tr> elements	Try it

- Exécutez les exemples proposés
- Ne vous préoccupez pas pour le moment de l'aspect événementiel (chapitre suivant).
- N'hésitez pas à modifier le code en ligne, pour faire vos propres essais.

Réalisez les tests à la suite.

Ecrire un script JavaScript en utilisant les fonctions de base de jQuery

Afpa © 2019 – Section Tertiaire Informatique – Filière « Etude et développement »

3. GERER LES EVENEMENTS HTML EN JQUERY

Pour nos premiers pas en jQuery, nous avons utilisé un script séquentiel.

Nous allons découvrir maintenant comment gérer les événements avec jQuery.

3.1 UNE PREMIERE FONCTION EVENEMENTIELLE EN JQUERY

Exemple : gestion de l'événement « click » sur tous les paragraphes, en lançant une fonction événementielle (anonyme) :

```
$("#p").click ( function()  
    {  
        // traitement à effectuer  
        alert ("cliqué") ;  
        $(this).hide() ;  
    });
```

Le gestion d'événement est programmé sous forme d'une fonction anonyme passée en paramètre à la méthode *click*

Dans cet exemple, la fonction événementielle affiche « cliqué » puis cache le paragraphe qui a été cliqué, référencé par *\$(this)*, l'objet courant.



Exercice 3 :

Reprenez l'exercice 1 de base fourni et programmez les événements qui permettent de cacher les paragraphes cliqués.

3.2 LISTE DES EVENEMENTS

Principaux événements.

Souris	Clavier	Formulaire	Document/Fenêtre
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload
mousedown			
mouseup			
mousemove			

Chaque événement est géré par une méthode jQuery du même nom : `click()`, `dblclick()` etc.
La fonction événementielle reçoit l'objet événement par le paramètre nommé ici *event* :

```
$("*").mouseenter( function (event)  
{  
    $("#trace").text ("=> entre dans :" + event.relatedTarget );  
});
```

Dans cet exemple, la méthode événementielle affiche l'élément HTML dans lequel on entre (*event.relatedTarget*) dans un élément *"#trace"*



Exercice 4 :

A partir du fichier html mis à votre disposition, définissez des événements :

- de souris sur les éléments de type paragraphe
- de clavier sur le groupe de contrôles fieldset

Testez les événements et affichez les paramètres de l'objet *event*.

Affichez les informations des événements dans la Textarea en utilisant JQuery.

- Événements souris : *mousedown*, *mouseup*, *mousemove*

Afficher le type d'événement, la position de la souris, le bouton de la souris sélectionné

- Événements clavier : *keypress*, *keydown*, *keyup*

Afficher le type d'événement : *event.type* , les caractères et assurez-vous de bien distinguer *target* et *currentTarget*. Quelle conclusion en tirer ?

Vous pouvez vous appuyer sur le site *w3schools*, qui vous permettra d'essayer les événements avant de coder :

https://www.w3schools.com/jquery/jquery_ref_events.asp

ou sur la documentation officielle de jQuery :

<https://api.jquery.com/category/events/>

Ecrire un script JavaScript en utilisant les fonctions de base de jQuery

Afpa © 2019 – Section Tertiaire Informatique – Filière « Etude et développement »

3.3 SEPARER JAVASCRIPT ET HTML

Notre code est-il propre ? Respecte-t-il les normes de codages usuelles ?

Deux points faibles :

- les appels à jQuery sont placés dans le fichier HTML : la préconisation est de séparer la description des contenus (HTML) et les traitements événementiels (fichier .js)
- notre exemple est dépendant de la place du script dans la page HTML : il fonctionne car il est placé après la définition des éléments HTML qui sont manipulés. Si on le place en début de fichier (dans le header), il ne fonctionne plus.

jQuery fournit un moyen pratique pour résoudre ces problèmes : l'événement *ready* se déclenche lorsque le document courant (ou un autre élément au choix) est complètement chargé. Comme les autres événements, il est géré par la méthode du même nom :

```
$(document).ready ( function()
{
    // traitements à effectuer après le chargement complet du document :
    //  initialisation d'éléments HTML, création dynamique d'éléments par le DOM, association
    //  des événements et des méthodes événementielles
});
```

L'exemple précédent devient :

```
$(document).ready ( function()
{
    $("p").click ( function()
    {
        // traitement à effectuer
        alert ("cliqué") ;
        $(this).hide() ;
    } ) ;
});
```

C'est la façon la plus répandue d'écrire le JavaScript.



Exercice 5 :

A partir du résultat obtenu lors de l'exercice 4, externalisez le script, placez le dans les entêtes du fichier HTML et assurez-vous qu'il fonctionne correctement en s'assurant que les éléments du DOM aient été chargés.

4. GERER LE DOM ET LES STYLES EN JQUERY

jQuery permet de manipuler HTML et les styles CSS en :

- récupérant et modifiant des attributs d'éléments HTML existants (vu précédemment)
- créant ou supprimant des éléments (à voir)
- récupérant, modifiant, créant ou supprimant des styles CSS (à voir)

4.1 MANIPULER LE DOM EN JQUERY

jQuery possède 4 méthodes pour insérer de nouveaux éléments :

<i>append</i>	insère du contenu à la fin des éléments sélectionnés
<i>prepend</i>	insère du contenu au début des éléments sélectionnés
<i>after</i>	insère du contenu après les éléments sélectionnés
<i>before</i>	insère du contenu avant les éléments sélectionnés

et deux méthodes pour supprimer des éléments :

<i>remove</i>	supprime l'élément sélectionné et ses nœuds enfant
<i>empty</i>	« vide » l'élément sélectionné : supprime uniquement les nœuds enfant en conservant l'élément sélectionné

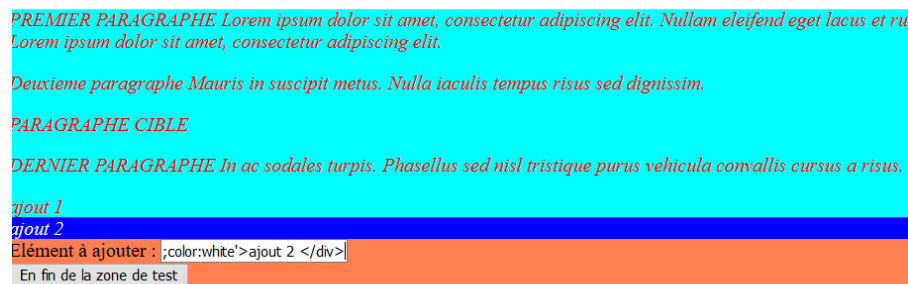
Attention :

- ces fonctions reçoivent en paramètre du HTML et non du texte : donc éventuellement aussi du JavaScript etc. Pensez à filtrer les données en entrée, pour la sécurité de l'application
- *append* et *prepend* s'appliquent à un ou plusieurs container (*p*, *div*, *span* etc.) qui sont les nœuds parent dans le DOM. *append* ajoute un nouvel élément après le dernier nœud enfant, *prepend* ajoute un nouvel élément avant le premier nœud enfant
- *after* et *before* s'appliquent à un des nœuds enfant : *after* insère le nouvel élément après le nœud enfant, *before* avant le nœud enfant

Lancez la démonstration : **demoDOM.html** :

ajoutez des nœuds avec du texte, mais aussi avec du HTML et du JavaScript

Jquery : utilisation du DOM



Vous pouvez aussi lancer les démonstrations sur le site [w3schools](https://www.w3schools.com/jquery/jquery_dom_add.asp) :

https://www.w3schools.com/jquery/jquery_dom_add.asp

et

https://www.w3schools.com/jquery/jquery_dom_remove.asp



Exercice 6 :

Le code source est mis à votre disposition [exercice6.zip](#)

Complétez le code fourni **demoDOM.html**, en ajoutant les fonctionnalités :

- ajouter un nouvel élément en début de liste
- ... avant un élément de la liste
- ... après un élément de la liste
- supprimer un élément et son arborescence
- vider un élément

Jquery : utilisation du DOM

ajouté au début

PREMIER PARAGRAPHE Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam eleifend eget lacus et rutrum. Phasellus vel faucibus sem, a pellentesque metus. sit amet, consectetur adipiscing elit.

Deuxieme paragraphe Mauris in suscipit metus. Nulla iaculis tempus risus sed dignissim.

ajouté AVANT

PARAGRAPHE CIBLE

ajouté APRES

DERNIER PARAGRAPHE In ac sodales turpis. Phasellus sed nisl tristique purus vehicula convallis cursus a risus. Nunc ornare augue et pharetra faucibus.

ajouté à la fin

Elément à ajouter :

Ecrire un script JavaScript en utilisant les fonctions de base de jQuery

Afpa © 2019 – Section Tertiaire Informatique – Filière « Etude et développement »

4.2 MANIPULER LES STYLES CSS EN JQUERY

<https://api.jquery.com/category/manipulation/class-attribute/>

jQuery permet de manipuler facilement les « classes » CSS :

<i>addClass</i>	ajoute une ou plusieurs classes CSS aux éléments sélectionnés https://api.jquery.com/addClass/
<i>removeClass</i>	supprime une ou plusieurs classes CSS des éléments sélectionnés https://api.jquery.com/removeClass/
<i>toggleClass</i>	bascule entre ajout/suppression de classes CSS dans les éléments sélectionnés
<i>css</i>	modifie ou retourne un attribut de style https://api.jquery.com/css/

Vous pouvez aussi utiliser la démonstration de *w3schools* sur l'utilisation des css en jQuery:

https://www.w3schools.com/jquery/jquery_css.asp



Exercice 7 :

Vous allez consulter la documentation en ligne de jQuery, sur les classes et les styles.

En adaptant et combinant à votre goût les exemples de la documentation, effectuez :

- un ajout de classe simple sur un ou plusieurs éléments
- un ajout de plusieurs classes sur un ou plusieurs éléments
- une suppression de classe sur un ou plusieurs éléments
- une modification de style (par exemple la taille des fontes) sur tous les éléments.

4.3 PARCOURIR L'ARBORESCENCE DU DOCUMENT AVEC JQUERY

4.3.1 Rappels sur le DOM et les parcours d'arbre

jQuery permet de parcourir l'arborescence d'un document (*Traversing*), de manière plus pratique que l'API du DOM (*document.getElementById*, *document.getElementsByTagName* etc.).

Avant de présenter les méthodes de jQuery, il faut rappeler la structure d'un document HTML.

Ecrire un script JavaScript en utilisant les fonctions de base de jQuery

Afpa © 2019 – Section Tertiaire Informatique – Filière « Etude et développement »

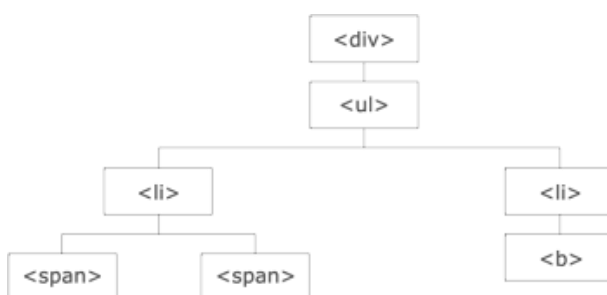
Chaque élément (ou balise, *tag*) HTML est un « *nœud* » du document courant (objet racine *document* dans la bibliothèque DOM en JavaScript).

jQuery permet de :

- remonter d'un nœud vers son parent (ou vers l'un de ses ancêtres)
- lister les enfants d'un nœud donné (ou ses descendants)
- chercher le nœud avant/après un élément, au même niveau de l'arbre (un membre de la même fratrie, en anglais « *siblings* »)

Regardons cet exemple de document HTML (fourni par *w3schools*) :

https://www.w3schools.com/jquery/jquery_traversing.asp



<div> est le **parent** de **** et l'ancêtre de tous les autres nœuds

**** est le **parent** de deux **** et l'enfant (**child**) de **<div>**

Le **** de gauche est le **parent** de deux ****, l'enfant de **** et le **descendant** de **<div>**

Les deux **** sous **** sont « frères » (**siblings**) : ils ont le même parent.

etc.

4.3.2 Mise en pratique

Pour mettre en pratique les méthodes de parcours d'arbre fournies par jQuery, vous allez vous appuyer sur *w3schools* qui est très visuel : *jQuery Traversing*

https://www.w3schools.com/jquery/jquery_traversing.asp

et suivants, jusqu'à *jQuery Filtering*.

4.3.3 Les méthodes de parcours d'arbre

Quelques méthodes jQuery pour parcourir l'arbre HTML et agir sur lui (non exhaustif) :

children retourne la liste des enfants d'un élément HTML (uniquement les descendants directs)

`$("#div1").children();` // tous les enfants de *div1*

Ecrire un script JavaScript en utilisant les fonctions de base de jQuery


```
$("#div1").children("p"); // les paragraphes, enfants de div1  
$("#div1").find("p.titre"); // les paragraphes, enfants de div1, qui possèdent la classe titre
```

find retourne la liste de tous les descendants (enfants, petits-enfants etc.) d'un élément, pour le type demandé (ou * pour tous les types)

```
$("#div1").find("*"); // tous les descendants de div1  
$("#div1").find("p"); // tous les paragraphes qui descendent de div1
```

first retourne le premier élément d'une liste

```
$("p").first(); // le premier paragraphe
```

last retourne le dernier élément d'une liste

```
$("p").last(); // le dernier paragraphe
```

eq(i) retourne l'élément d'indice i dans une liste (à partir de 0)

```
$("p").eq(1); // le deuxième paragraphe
```

filter(critere) retourne les éléments correspondant au critère

```
$("div").filter(".entete") // toutes les div qui ont la classe entete
```

parent retourne le parent direct du/des élément

```
$("p").parent() // les éléments parent auxquels appartiennent les paragraphes
```

parents retourne tous les ancêtres (parent, grands-parents etc.) de/des éléments

```
$("p").parents() // tous les éléments ancêtres des paragraphes
```

etc.

Vous trouverez le détail de ces méthodes, avec des démonstrations visuelles sur le site de w3schools :

https://www.w3schools.com/jquery/jquery_ref_traversing.asp