

L3 informatique

2022-2023

Programmation Web

Partie 5 : jQuery

Étienne André

Université Sorbonne Paris Nord
Etienne.Andre@univ-paris13.fr



Version: 1^{er} février 2023

Outline

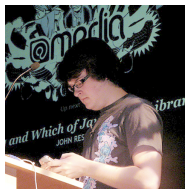
- 1 Motivation et syntaxe
- 2 Appel et syntaxe de base
- 3 Sélecteurs jQuery
- 4 Évènements jQuery

Objectif et historique

- Bibliothèque JavaScript libre et multiplateforme



- **Objectif** : faciliter l'écriture de scripts JavaScript (côté client) en liaison avec le code HTML
 - Scripts JavaScript **plus concis** et **plus lisibles**
- Créée à partir de 2006 par John Resig



Caractéristiques

- Fichier unique (écrit en JavaScript) de ≈ 274 kio
 - Peut être téléchargée : `https://code.jquery.com/jquery-3.6.1.js`
 - Version « minifiée » :
`https://code.jquery.com/jquery-3.6.1.min.js`
 - Voir toutes les versions : `https://releases.jquery.com/`
- Popularité : utilisée par 77,1 % des principaux sites Web
 - Source : `https://w3techs.com/technologies/overview/javascript_library` (décembre 2022)

Avantages

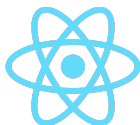
- Code plus concis
- Compatibilité avec tous les navigateurs
 - « Simplement » du code JavaScript supplémentaire
- Code de petite taille (poids)
- Licence libre (MIT)



Alternatives à jQuery

■ React

- Licence libre (MIT)
- Développée par Meta (Facebook, Instagram)
- Gère la partie « vue » du modèle MVC



■ Vue.js

- Licence libre (MIT)
- Initiée par Evan You
- Requiert **Node.js**
- Dédie aux applications Web monopages (*single-page application*)



Outline

- 1 Motivation et syntaxe
- 2 Appel et syntaxe de base**
- 3 Sélecteurs jQuery
- 4 Évènements jQuery

Appel à la bibliothèque jQuery

Appel depuis le code HTML

- Exécution **locale** (côté client) mais...
- ...chargement (généralement) **distant** (téléchargement depuis un serveur)
- Si la bibliothèque est présente en local (fichier téléchargé sur le serveur Web):

```
1 <script src="jquery-3.6.1.min.js"></script>
```

- Si la bibliothèque est présente sur un serveur tiers (**recommandé**):

```
1 <script  
  src="https://code.jquery.com/jquery-3.6.1.min.js"  
  integrity="sha256-o88AwQnZB+VDvE9tvIXrMQaPlFFSUTR+nldQm"  
  crossorigin="anonymous"></script>
```

- Les champs **integrity** et **crossorigin** permettent la vérification de l'intégrité dite SRI (*Subresource Integrity*)

Appel à la bibliothèque jQuery : ordre des imports

⚠ Appel à jQuery en premier

L'import de la bibliothèque jQuery doit évidemment être réalisé **avant** l'import de vos propres scripts

Sinon :

```
Uncaught ReferenceError: $ is not defined
```

Bien :

```
1 <!-- D'abord l'appel à jQuery -->
2 <script src="https://code.jquery.com/jquery-3.6.1.min.js"
   integrity="sha256-o88AwQnZB+VDvE9tvIXrMQaPlFFSUTR+nldQm1LuPU"
   crossorigin="anonymous"></script>
3
4 <!-- Puis l'appel à mon script utilisant jQuery -->
5 <script src="monscript.js"></script>
```



Démonstration

Syntaxe de base

```
1 $(selecteur).action()
```

- **\$** : l'appel à jQuery
- **selecteur** : un sélecteur pour les éléments HTML
- **action** : une action jQuery à effectuer

Exemples

```
1 $(a).hide()
```

- Cache tous les liens de la page HTML

```
1 $(#titre).toggle()
```

- Cache (si affiché) ou affiche (si masqué) l'élément HTML d'identifiant `titre`

```
1 $(.realisatrice).show()
```

- Affiche tous les éléments HTML de classe `realisatrice`

```
1 $(this).hide()
```

- Cache l'élément HTML courant

Fonction `ready`

Fonction exécutée dès que le DOM est chargé et peut être **manipulé**

■ Utilisation :

```
1 $(document).ready(() => {  
2   // ici votre code  
3 });
```

■ Peut être vue comme l'équivalent du code JavaScript pur suivant :

```
1 window.addEventListener("load", () => {  
2   // ici votre code  
3 });
```

Fonction `ready` : exemple (HTML)

```
1 <!DOCTYPE html >
2 <html lang="fr">
3   <head>...</head>
4   <body>
5     <h1>Quiz jQuery</h1>
6
7     <button id="bouton">Cacher les réponses</button>
8
9     <h2>Quand jQuery a-t-elle été proposée ?</h2>
10    <p>En 2006</p>
11
12    <h2>Qu'est-ce que jQuery ?</h2>
13    <p>Une bibliothèque JavaScript</p>
14
15    <script
16      src="https://code.jquery.com/jquery-3.6.1.min.js"
17      integrity="sha256-o88AwQnZB+VDvE9tvIXrMQaPlFFSUTR+nldQm1LuP
18      crossorigin="anonymous"></script>
19    <script src="jQuery-minimal.js"></script>
20  </body>
21 </html>
```

Fonction `ready` : exemple (JavaScript + jQuery)

```
1 $(document).ready(() => {  
2     // Affichage dans la console  
3     console.log("DOM entièrement chargé");  
4     // Évènement sur les boutons  
5     $("button").click(() => {  
6         // Alterne la visibilité de tous les paragraphes  
7         $("p").toggle();  
8     });  
9 });
```

Démonstration

 Exemple téléchargeable depuis Moodle

Conflit d'alias et renommage

- Rappel : `$` est un nom valide de variable en JavaScript!
- Plusieurs bibliothèques JavaScript utilisent `$` pour leurs fonctions : risque de **conflit** entre bibliothèques
- La fonction `noConflict()` spécifie une variable pour remplacer le `$`

```
1 let JQ = jQuery.noConflict();
2 JQ(document).ready(() => {
3     // Affichage dans la console
4     console.log("DOM entièrement chargé");
5     // Évènement sur les boutons
6     JQ("button").click(() => {
7         // Alterne la visibilité de tous les paragraphes
8         JQ("p").toggle();
9     });
10 });
```

Outline

- 1 Motivation et syntaxe
- 2 Appel et syntaxe de base
- 3 Sélecteurs jQuery**
- 4 Évènements jQuery

Syntaxe des sélecteurs jQuery : éléments

- Syntaxe des éléments : syntaxe du langage CSS

- Exemples :

```
1 // Sélectionne tous les éléments p (paragraphes)
2 $("p")
3
4 // Sélectionne tous les éléments h1 appartenant à
   la classe "film"
5 $("h1.film")
6
7 // Sélectionne le span avec l'identifiant "titre"
8 $("span#titre")
```

Syntaxe des sélecteurs jQuery : attributs

■ Syntaxe des attributs : syntaxe de XPATH

```
1 // Sélectionne tous les éléments possédant un attribut
   "href" (car présence d'un "name") ayant pour valeur
   "par"
2 $("[name='par']")
3
4 // Sélectionne tous les éléments possédant un attribut
   "src" dont la valeur se termine par ".jpg"
5 $("[src$='.jpg']")
```

Quelques sélecteurs jQuery : éléments

Sélecteur	Exemple	Description
*	<code>\$("*")</code>	Tous les éléments
Élément	<code>\$("p")</code>	Tous les éléments <code>p</code> (paragraphes)
#id	<code>\$("#id")</code>	L'élément d'identifiant <code>id</code>
.class	<code>\$(".MaClasse")</code>	Tous les éléments de classe <code>MaClasse</code>
div p	<code>\$("div p")</code>	Les <code>p</code> descendant de <code>div</code>
div>p	<code>\$("div>p")</code>	Les <code>p</code> descendant directement de <code>div</code>
:eq(index)	<code>\$("a:eq(2)")</code>	Le 2 ^e élément <code>a</code>
:gt(index)	<code>\$("p:gt(4)")</code>	Les <code>p</code> à partir du 5 ^e <code>p</code> inclus
:lt(index)	<code>\$("li:lt(4)")</code>	Les <code>p</code> jusqu'au 3 ^e <code>p</code> inclus
:even	<code>\$("div:even")</code>	Les <code>div</code> en position impaire
:odd	<code>\$("li:odd")</code>	Les puces (<code>li</code>) en position paire
not(selecteur)	<code>\$("not(p)")</code>	Tous les éléments sauf les <code>p</code>
:first	<code>\$("tr:first")</code>	La première ligne (<code>tr</code>)
:last	<code>\$("td:last")</code>	La dernière cellule (<code>td</code>)

Quelques sélecteurs jQuery : éléments (suite)

Sélecteur	Exemple	Description
:animated	<code>\$("#a:animated")</code>	Les liens (a) animés
:empty	<code>\$("#input:empty")</code>	Les champs (input) vides
:hidden	<code>\$("#h2:hidden")</code>	Les sous-titres (h2) non visibles
:visible	<code>\$("#img:visible")</code>	Les images (img) visibles
:contains(chaine)	<code>\$("#:contains(film)")</code>	Tous les éléments contenant la chaîne " film "
:input	<code>\$("#:input")</code>	Tous les éléments input
:text	<code>\$("#:text")</code>	Tous les champs text
:radio	<code>\$("#:radio")</code>	Tous les boutons radio
:selected	<code>\$("#:selected")</code>	Tous les éléments ayant selected à vrai
:checked	<code>\$("#:checked")</code>	Tous les éléments ayant checked à vrai
:enabled	<code>\$("#:enabled")</code>	Tous les éléments ayant enabled à vrai

Quelques sélecteurs jQuery : attributs

Sélecteur	Exemple	Description
<code>[attribut]</code>	<code>\$("[src]")</code>	Les éléments possédant un attribut <code>src</code>
<code>[attribut='valeur']</code>	<code>\$("[src='2046.jpg']")</code>	Les éléments possédant un attribut <code>src</code> ayant pour valeur <code>"2046.jpg"</code>
<code>[attribut!='valeur']</code>	<code>\$("[src!='2046.jpg']")</code>	Les éléments possédant un attribut <code>src</code> ayant une valeur différente de <code>"2046.jpg"</code>
<code>[attribut^='val']</code>	<code>\$("[src^='film-']")</code>	Les éléments possédant un attribut <code>src</code> ayant une valeur commençant par <code>"film-"</code>
<code>[attribut\$='val']</code>	<code>\$("[src\$='-thumb.jpg']")</code>	Les éléments possédant un attribut <code>src</code> ayant une valeur se terminant par <code>"-thumb.jpg"</code>
<code>[attribut*='val']</code>	<code>\$("[src*='thumb']")</code>	Les éléments possédant un attribut <code>src</code> ayant une valeur contenant la sous-chaîne <code>"thumb"</code>
<code>[attribut~='val']</code>	<code>\$("[class~='maClasse']")</code>	Les éléments possédant un attribut <code>class</code> ayant une ou plusieurs valeurs, dont une est <code>"maClasse"</code>

Outline

- 1 Motivation et syntaxe
- 2 Appel et syntaxe de base
- 3 Sélecteurs jQuery
- 4 Évènements jQuery**

Introduction aux évènements jQuery

- jQuery permet de simplifier l'affectation des évènements aux éléments et élimine tous les problèmes d'incompatibilité
- jQuery permet d'affecter un évènement de plusieurs façons et de le désaffecter
- jQuery peut même affecter des évènements à des éléments qui n'existent pas encore
- Pour plus de détails : jQuery Event Methods
 - <https://api.jquery.com/category/events/>
 - https://www.w3schools.com/jquery/jquery_ref_events.asp

Syntaxe de base des évènements

Syntaxe (rappel) :

```
$(selecteur).action()
```

Exemple :

```
1 $(document).ready(() => {  
2   // Affichage dans la console  
3   console.log("DOM entièrement chargé");  
4   // Évènement sur les boutons  
5   $("button").click(() => {  
6     // Alterne la visibilité de tous les paragraphes  
7     $("p").toggle();  
8   });  
9 });
```

- Auditeur défini sur l'évènement `ready`
- Auditeur défini sur l'évènement `click`

Fonctions de gestion des évènements

Fonction	Description
<code>.on(evt, [selecteur,] auditeur)</code>	attache un gestionnaire aux éléments sélectionnés (s'applique aussi aux éléments créés dynamiquement)
<code>.off(evt, [selecteur])</code>	supprime l'auditeur des éléments
<code>.one(evt, [selecteur,] auditeur)</code>	associe un gestionnaire à un évènement, qui ne s'exécutera qu'une seule fois
<code>.click(auditeur)</code>	clic de souris (raccourci pour <code>.on("click", auditeur)</code>)
<code>.hover(auditeur)</code>	survol par la souris (exécuté à l'entrée et à la sortie)
<code>.hover(entree, sortie)</code>	survol par la souris (<code>entree</code> étant le gestionnaire à l'entrée de la zone, <code>sortie</code> en sortant)
<code>.trigger(evt)</code>	déclenche manuellement un évènement et exécute les gestionnaires associés

Autres fonctions : `.mouseenter()`, `.mouseleave()`, etc.

Voir la liste complète sur <https://api.jquery.com/category/events/>

Manipulation du HTML (1/3)

- `.append(codeHTML)` Ajoute du code HTML aux éléments

```
1 $( "#bouton1" ).on( "click", function () {  
2     $( "#log" ).append( "<p>Clic bouton 1</p>" );  
3 });
```

- `.empty()` Vide les enfants des éléments

```
1 $( "#effacer" ).click( function () {  
2     $( "#log" ).empty ();  
3 });
```

- `.remove()` Supprimer les éléments et tous ses descendants (fonctionne aussi avec un argument spécifiant *quels* éléments supprimer)

```
1 $( "#effacer" ).click( function () {  
2     $( "#log p" ).remove ();  
3 });
```

Manipulation du HTML (2/3)

- `.prop(propriete)` Récupère la valeur de la propriété *du premier élément*, ou modifie la valeur de la propriété de *tous les éléments*

```
1 // Changer l'opacité avec jQuery et CSS
2 $('#plus').prop('disabled', true);
3 $("#moins").click(function (){
4     $("#log p").fadeOut("slow");    console.log("Bouton +");
5
6     $('#moins').prop('disabled', true);
7     $('#plus').prop('disabled', false);
8 });
9 $("#plus").click(function (){
10    $("#log p").fadeIn("fast");
11    $('#moins').prop('disabled', false);
12    $('#plus').prop('disabled', true);
13 });
```

Manipulation du HTML (3/3)

- `.html()` Récupère le code HTML de l'élément
 - (probablement pas utile en TP, hors avec `console.log` pour le débogage)

```
1 $("#bouton2").click(function () {  
2   // Affiche dans la console le contenu (code HTML) du  
   #log  
3   console.log($("#log").html());  
4   // Exemple d'affichage : "<p>Clic bouton 2</p><p>Clic  
   bouton 1</p>"  
5 });
```

- Autres fonctions utiles : `addClass()`, `appendTo()`, `detach()`, `insertAfter()`, `insertBefore()`, `prepend()`, `replace()`, etc.
 - Voir liste sur <https://api.jquery.com/category/manipulation/>

Quelques effets jQuery

Fonction	Description
<code>.show()</code>	affiche les éléments sélectionnés
<code>.hide()</code>	masque les éléments sélectionnés
<code>.toggle()</code>	affiche ou masque les éléments sélectionnés
<code>.fadeTo()</code>	ajuste l'opacité des éléments sélectionnés
<code>.fadeIn()</code>	affiche les éléments sélectionnés en fondu
<code>.fadeOut()</code>	masque les éléments sélectionnés en fondu
<code>.fadeToggle()</code>	affiche ou masque les éléments sélectionnés en fondu
<code>.slideDown()</code>	affiche les éléments sélectionnés avec effet de glisse
<code>.slideUp()</code>	masque les éléments sélectionnés avec effet de glisse
<code>.slideToggle()</code>	affiche ou masque les éléments sélectionnés avec effet de glisse
<code>.animate()</code>	effectue une animation des éléments sélectionnés à l'aide de propriétés CSS

Voir liste complète sur <https://api.jquery.com/category/effects/>

Manipulation du CSS (1/2)

Fonction `.css()`

- `.css(name, value)` : une propriété
- `.css({ codeCSS })` : un ensemble de propriétés CSS

```
1 $("h2").click(function() {  
2   // On change la couleur du log (technique moche !)  
3   let couleur = "#" +  
4     Math.floor(Math.random()*16777215).toString(16);  
5   $(this).css({  
6     "background-color": couleur,  
7     "font-weight"      : "-=50",  
8     "font-size"       : "+=10",  
9   });  
10  
11   // On lit la couleur obtenue  
12   console.log($(this).css("background-color"));  
});
```

Manipulation du CSS (2/2)


Autres fonctions dédiées à la manipulation du CSS :

- `.addClass`
- `.hasClass`
- `.height`
- `.removeClass`
- `.width`
- etc.

Voir liste complète sur <https://api.jquery.com/category/css/>

Exemple complet : jQuery + manipulation HTML et CSS

 **Démonstration**

 Exemple téléchargeable depuis Moodle

Sources et références

Sources et références

Références

- L'API officielle jQuery : <https://api.jquery.com>
- Frédéric Delobel : [JavaScript et jQuery – La programmation Web par la pratique](#)
- Robin Nixon : [Développer un site web en PHP, MySQL JavaScript jQuery, CSS3 et HTML5 : Un guide étape par étape pour créer des sites Web dynamiques \(O'Reilly\)](#)
- [Wikipedia: jQuery](#)


Sources

- Ce cours est partiellement inspiré des supports de cours de Thamer MECHARNIA (Université Sorbonne Paris Nord)

Crédits

Source des images utilisées I




Titre : warning sign
Auteur : ?
Source : https://commons.wikimedia.org/wiki/File:Warning_sign.png
Licence : 



Titre : Check Mark CSS Green
Auteur : Pigeon43
Source : https://commons.wikimedia.org/wiki/File:Check_Mark_CSS_Green.svg
Licence : 



Titre : small-n-flat
Auteur : paomedia
Source : <https://commons.wikimedia.org/wiki/File:Device-computer-icon.png>
Licence : 




Titre : Nuvola filesystems folder download
Auteur : David Vignoni
Source : https://commons.wikimedia.org/wiki/File:Nuvola_filesystems_folder_download.png
Licence : 




Titre : Logo de jQuery
Auteur : jQuery

Source des images utilisées II

Source : <https://fr.wikipedia.org/wiki/Fichier:Jquery-logo.png>
Licence : 




Titre : John Resig presenting an overview of JavaScript libraries
Auteur : Martin Kliehm
Source : <https://commons.wikimedia.org/wiki/File:Jresig.png>
Licence : 



Titre : The logo for React
Auteur : Facebook
Source : <https://commons.wikimedia.org/w/index.php?title=File:React-icon.svg>
Licence : 



Titre : Vue.js logo
Auteur : Evan You
Source : https://commons.wikimedia.org/w/index.php?title=File:Vue.js_Logo_2.svg
Licence : 

Licence de ce document

Ce support de cours peut être réutilisé, modifié et republié selon les termes de la licence Creative Commons **Attribution-NonCommercial-ShareAlike 4.0 Unported (CC BY-NC-SA 4.0)**



<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Auteur : Étienne André

(Source \LaTeX disponible aux enseignant·e·s sur demande)

UNIVERSITÉ
SORBONNE
PARIS NORD

Institut
Galilée

Version : 1^{er} février 2023