

jQuery

Formation

Développeur Web et Web Mobile

Module : 03

Développer la partie front-end d'une application web


Séquence : 03

Développer une interface utilisateur web dynamique

Séance : 03

Développer des scripts clients dans une page web


Libellé réduit:	JQUERY
Code produit:	
Type de document:	Ressource
Version:	1
Référence	1-1-1
Date de création:	14/02/2015
Date de mise à jour:	07/11/2022

	Auteur	Centre de Créteil	Formation	Date Mise à jour	
	Didier Bonneau	GRN 164	DWWM	07/11/2022	support_de_cours-jquery.docx


Jquery

Sommaire

I. Introduction.....	1	VII. Les évènements	10
II. Installation de jQuery	2	A. La méthode « bind() ».....	10
III. Utilisation de jQuery.....	3	B. La méthode « on() ».....	10
IV. Les sélecteurs en jQuery	4	C. Les méthodes directes.....	11
A. Les sélecteurs de base.....	4	1. Relatifs à la souris	11
1. Sélecteur d'ID : « # »	4	2. Relatifs au clavier	12
2. Sélecteur de nom : « nom balise »	4	3. Relatifs au gestionnaire d'évènement.....	12
3. Sélecteur de classe : « . »	4	4. Relatifs aux formulaires	13
4. Sélecteur d'attribut : « [nomAttribut] »	4	5. Relatifs au navigateur	13
B. Les sélecteurs hiérarchiques	4	6. Relatifs au document.....	13
1. Sélecteur de descendants : « espace ».....	4	VIII. Les effets	14
2. Sélecteur d'enfants : « > »	5	A. Afficher et cacher des éléments.....	14
3. Sélecteur de frères : « ~ »	5	B. Jouer sur La hauteur verticalement.....	14
4. Sélecteur d'élément suivant : « + »	5	C. Jouer sur l'opacité.....	15
C. Les pseudo-sélecteurs.....	5	D. Différer les effets.....	15
V. Manipulation des attributs.....	6	E. Créer une animation	16
A. Connaître ou modifier la valeur d'un attribut	6	IX. Utiliser le DOM.....	17
B. Supprimer un attribut	7	A. Itérer sur une collection.....	17
C. Connaître ou modifier la valeur de l'attribut « value »	7	B. Filtrer une collection d'éléments	17
D. Connaître ou modifier la valeur de l'attribut « class »	7	1. Le premier ou le dernier.....	17
1. Ajouter une ou plusieurs classes.....	7	2. Ayant un certain index	17
2. Savoir si un élément est de classe déterminée	7	3. Correspondant à un certain type	18
VI. Manipulation du CSS.....	8	4. Ayant un descendant d'un certain type.....	18
A. Lecture ou modification de propriétés css	8	C. Chercher au même niveau.....	18
B. Accès direct aux propriétés de dimension	9	1. Le frère suivant ou précédent.....	18
1. La hauteur	9	2. Les frères suivants ou précédents.....	18
2. La largeur	9	3. Tous les frères.....	18
3. voir aussi	9		
C. Accès direct aux propriétés de position	9		

	Auteur	Centre de Créteil	Formation	Date Mise à jour	
	Didier Bonneau	GRN 164	DWWW	07/11/2022	support_de_cours-jquery.docx

D.	Chercher A un niveau inférieur	19
1.	Les enfants.....	19
2.	Les descendants	19
E.	Chercher A un niveau supérieur	19
1.	Le parent	19
2.	Les ascendants	19
X.	Manipulation du DOM.....	20
A.	Modifier le contenu d'un élément.....	20
B.	Insérer un élément.....	20
1.	Comme enfant.....	20
1.	Comme frère	21
2.	Comme père.....	21
C.	remplacer un élément.....	21
D.	copier un élément	21
XI.	Ajax avec jquery	22
XII.	Quelques Plug-Ins	23

	Auteur	Centre de Créteil	Formation	Date Mise à jour	
	Didier Bonneau	GRN 164	DWWM	07/11/2022	support_de_cours-jquery.docx

La bibliothèque jQuery

I. INTRODUCTION

jQuery est une bibliothèque JavaScript offrant une syntaxe concise et logique simplifiant énormément l'écriture de script.

Conçu par John Resig en Janvier 2006, il est actuellement développé par toute une communauté. La dernière version stable est la 1.10.1.

jQuery n'est pas la seule bibliothèque JS disponible, on peut citer : Scriptaculous, Prototype, Mootools, Yahoo UI, Mochikit, ExtJS, Dojo, ...

Attention : une bibliothèque JavaScript ne peut fonctionner que si le navigateur supporte le JavaScript.

La bibliothèque de base est déjà assez riche et permet de faire beaucoup de chose mais on peut ajouter des plug-ins pour faire encore plus.

jQuery est un simple fichier JavaScript qu'il faut intégrer à votre page web.

Deux versions existent :

- Production : version compressée, illisible mais moins lourde : 31ko
- Développement : version en code source lisible : 229ko

Il faut la télécharger à partir du site officiel : <http://jquery.com>

Exemple de code issu du site du zéro :


Récupération de la hauteur de la page web en pixel

JavaScript

```
function avoirDefilementVertical(){
    if(typeof( window.pageYOffset ) == 'number'){
        // Netscape
        return window.pageYOffset;
    } else if( document.body &&
(document.body.scrollTop) ||
navigator.appName=="Microsoft Internet Explorer") {
        // DOM
        return document.body.scrollTop;
    } else if( document.documentElement &&
(document.documentElement.scrollTop) ) {
        // Internet Explorer 6
        return document.documentElement.scrollTop;
    }
}
```

jQuery

```
function avoirDefilementVertical(){
    return $(document).scrollTop();
}
```

	Auteur	Centre Afpa de Créteil	Formation	Date de mise à jour	Page 1
	Didier Bonneau	GRN 164	DWWM	07/11/2022	support_de_cours-jquery.docx

II. INSTALLATION DE JQUERY

Plusieurs solutions sont possibles pour utiliser et mettre en œuvre jQuery.

1. Télécharger le fichier « .js » à partir du site officiel : <http://jquery.com> et l'enregistrer dans un dossier de votre site.
 - Permet de choisir la version « production » ou « développement »
 - Le fichier de bibliothèque se trouve sur votre site. Attention à la mise à jour.
2. Utiliser l'hébergement sur Media Temple.
3. Utiliser l'hébergement de Google qui maintient les différentes versions des fichiers de production de jQuery.
 - Le fichier est centralisé et le cache du navigateur est optimisé.
 - Pas de rechargement si plusieurs site l'utilise.
4. Intégrer les bibliothèques JavaScript et Ajax de Google.
 - Solution plus lourde qui permet d'intégrer les bibliothèques tel que : jQuery, jQuery-UI, Prototype, Scriptaculous, Mootools, Dojo, SWFObject, Ext-Core
 - Le script est toujours à la dernière version
 - On ne se soucie plus de l'url réelle du site

A partir du site Officiel : url du fichier jQuery

production : <http://code.jquery.com/jquery-1.10.1.min.js>
 développement : <http://code.jquery.com/jquery-1.10.1.js>

Une fois téléchargé, ajouter la ligne de code dans le « head » de la page web :

➡ `<script type="text/javascript" src="jquery.js"></script>`

L'hébergement

Ajouter la ligne de code dans le « head » de la page web :

➡ `<script type="text/javascript" src="http://code.jquery.com/jquery-latest.js"></script>`

➡ `<script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.1/jquery.min.js"></script>`


Intégration de bibliothèques

Ajouter la ligne de code dans le « head » de la page web :

➡ `<script type="text/javascript" src="http://www.google.com/jsapi"></script>`

Puis écrire du code JavaScript :

`// Charge la version 1.10.1.
google.load('jquery','1.10.1');`

	Auteur	Centre Afpa de Créteil	Formation	Date de mise à jour	Page 2
	Didier Bonneau	GRN 164	DWWM	07/11/2022	support_de_cours-jquery.docx

III. UTILISATION DE JQUERY

L'utilisation de jQuery repose sur une fonction « `jQuery()` » où l'on peut remplacer le mot « `jquery` » par son alias « `$` » ce qui donne une notation abrégée « `$()` ».

La fonction « `jquery()` » accepte un paramètre qui va servir à créer un objet jQuery. Ce paramètre peut être de différents types :

- Une chaîne de caractère : on parle de « sélecteur » (voir chapitre suivant).
- Un objet ou élément du DOM.
- Un objet JavaScript.
- Un objet jQuery : voir la méthode `clone()`.

Ce paramètre est utilisé par `jquery()` pour chercher l'objet ou les objets du DOM correspondants à ce paramètre. Elle retourne un objet renfermant une collection d'objets du DOM. Utiliser la propriété « `.length` » pour connaître le nombre d'éléments de l'objet jQuery.

Afin d'initialiser le code jQuery, il faut attendre la fin du chargement des objets du DOM constituant la page web. Ceci est réalisé en appelant la méthode « `ready()` » sur l'objet jQuery représentant l'objet « `document` » du DOM.

Attention, l'utilisation de jQuery avec d'autres bibliothèques JavaScript peut conduire à des conflits. En effet, Prototype utilise l'alias « `$` ». Il faut à ce moment-là appeler la méthode `jquery.noConflict()` et ne plus servir de l'alias « `$` » pour jQuery. Une autre solution est d'affecter le retour de cette méthode à une variable et de se servir de cette variable comme alias :

Utilisation

`$(paramètre).méthode()` ex: `$("div").size()`
`$(paramètre).propriété` ex: `$("div").length`

Remarque : beaucoup de méthodes renvoient l'objet jQuery sur lequel elles sont appliquées. Par conséquent les lignes de code suivantes sont équivalentes :

`$(paramètre).méthode1().méthode2();`
`$(paramètre).méthode1()`
 `.méthode2();`

Initialisation du code jQuery de la page web

```
<script type= "text/javascript">
    $(document).ready(function(){
        // code jQuery
    });
</script>
```

Écriture équivalente :

```
$(function(){
    // code jQuery
});
```

Attention aux conflits

```
jQuery.noConflict();
jQuery("div").hide();

var $jq = jQuery.noConflict();
$jq("div").hide();
```

IV. LES SELECTEURS EN JQUERY

Le paramètre à fournir à la fonction « `$()` » peut être une chaîne de caractère représentant un « sélecteur » et ressemble beaucoup aux sélecteurs CSS.

La recherche des éléments correspondant au sélecteur se fait à partir de l'objet « document ». Il est possible de passer un deuxième paramètre permettant de restreindre cette recherche à un certain contexte.

A. LES SELECTEURS DE BASE

Ce sont des sélecteurs simples qui sont équivalents à ceux du JavaScript.

1. Sélecteur d'ID : « # »

Sélectionne un élément unique repéré par un attribut « id ». C'est l'équivalent en JavaScript de `getElementById()`.

2. Sélecteur de nom : « nom balise »

Sélectionne un ou plusieurs éléments par le nom de la balise. C'est l'équivalent en JavaScript de `getElementsByName()`.

3. Sélecteur de classe : « . »

Sélectionne un ou plusieurs éléments par la valeur de son attribut « class ». C'est l'équivalent en JavaScript de `getElementsByClassName()`.

4. Sélecteur d'attribut : « [nomAttribut] »

Sélectionne un ou plusieurs éléments par le nom de son attribut

B. LES SELECTEURS HIERARCHIQUES

Ces sélecteurs donnent les éléments du DOM à partir d'un autre élément. On peut donc accéder aux éléments « enfant », « parent », « frère », ...

1. Sélecteur de descendants : « espace »

Sélectionne tous les descendants (enfants, petits-enfants et arrière-petits-enfants) d'une balise conteneur.

Syntaxe générale

`$(selecteur [, contexte])`

Soit le bout de code html :

```
<div id="contenu">
  <div>
    <p> premier <span class="souligne">
      paragraphe</span></p>
    <p> premier <span class="souligne">
      paragraphe</span></p>
  </div>
</div>
```


Les accès aux divers éléments peuvent être :

Sélecteurs de base

`$("#contenu")` -> accès à l'objet div
`$("p")` -> accès aux 2 balises p
`$(".souligne")` accès aux balises de classe "souligne"
`$("span[class]")` -> accès aux 2 balises span

Sélecteurs hiérarchiques

`$("#contenu *")` accès aux balises 2ème div, p, span
`$("#contenu span")` accès aux 2 balises span
`$("div p")` accès aux 2 balises p
`$("#contenu>div")` accès à la 2ème balise div
`$("div~p")` accès à la 2ème balise p
`$("#contenu+div")` accès à la 2ème balise div

	Auteur	Centre Afpa de Créteil	Formation	Date de mise à jour	Page 4
	Didier Bonneau	GRN 164	DWWW	07/11/2022	support_de_cours-jquery.docx

2. Sélecteur d'enfants : « > »

Accès aux enfants directs d'une balise conteneur.

3. Sélecteur de frères : « ~ »

Accès aux frères directs d'une balise.

4. Sélecteur d'élément suivant : « + »

Accès à l'élément immédiatement suivant d'une autre balise

C. LES PSEUDO-SELECTEURS

Comme en CSS, on peut qualifier un sélecteur pour atteindre un ou plusieurs éléments particuliers.

- Le **premier** et le **dernier** : « **:first** » et « **:last** »
- Le **pair** et l'**impair** : « **:even** » et « **:odd** »
- Un **élément déterminé** : « **:eq(numéro)** »
- Exclusion d'éléments : « **:not(sélecteur1, sélecteur2, ...)** »
- Les éléments **supérieurs** (inférieurs) à un **indice** :
« **:gt(indice)** » et « **:lt(indice)** »
- Les éléments qui sont **parents** : « **:parent** »
- Les éléments qui n'ont pas d' **enfants** : « **:empty** »
- Le **premier** ou **dernier enfant** : « **:first-child** » et « **:last-child** »
- Le **nième enfant** : « **:nth-child(numéro)** »
- L'**enfant unique** : « **:only-child** »
- D'un **attribut « type » particulier** : « **:text, :radio, :checkbox, :password, :image, :reset, :submit, :hidden** »
- **Dans un état particulier** : « **:disabled, :enabled, :checked, :selected, :visible** »
- Des **éléments particulier** : « **:input** »

Les pseudos-sélecteurs

\$("#p:first") accès à la 1ère balise p

\$("#p:odd") accès à la 2ème balise p


\$("#span:eq(1)") accès à la 2ème balise span

\$("#p:not(:last)") accès à la 1ère balise p

En précisant le context

\$("#p", "#contenu") -> accès aux 2 balises p

La recherche est restreint aux descendants de la div portant l'attribut id="contenu"

	Auteur	Centre Afpa de Créteil	Formation	Date de mise à jour	Page 5
	Didier Bonneau	GRN 164	DWWM	07/11/2022	support_de_cours-jquery.docx

V. MANIPULATION DES ATTRIBUTS

jQuery permet bien sûr de manipuler les attributs des éléments sélectionnés par les différents sélecteurs vus précédemment.

Ceci se fait par l'appel à de nombreuses méthodes des objets jQuery.

A. CONNAÎTRE OU MODIFIER LA VALEUR D'UN ATTRIBUT

Ceci se fait par la méthode « **attr()** » qui admet un ou deux paramètres selon que l'on veut obtenir ou modifier la valeur de l'attribut.

La syntaxe pour une lecture est :

\$(selecteur).attr(nomAttribut)

« nomAttribut » peut être une chaîne de caractères constante ou une variable.

Pour une modification ou un ajout :

\$(selecteur).attr(nomAttribut, valeurAttribut)

« valeurAttribut » peut être une chaîne de caractères constante, une variable ou une fonction.

Il est aussi possible par cette méthode « attr » d'ajouter plusieurs attributs en même temps. Il faut lui fournir un paramètre dont la syntaxe ressemble fort à la déclaration des propriétés CSS.

\$(selecteur).attr({ propriétés })

« propriétés » est un ensemble de couples dont le premier est le nom de l'attribut suivi par le caractère « : » et le second sa valeur. Les couples sont eux séparés par le caractère « , ».

Exemple de code

Soit le bout de code HTML :

```
<span style="text-decoration: underline">
  lecture attribut style</span>

```


Script jQuery

```
// récupération de la valeur de l'attribut style
var css = $("span").attr("style");
```

```
// modification de la valeur de l'attribut style
$("span").attr("style", "text-decoration: none");
```

```
// modification des attributs src et alt
$("img").attr({ src: "image2.jpg",
               alt: "ceci est l'image 2"});
```

```
// suppression de l'attribut style
$("span").removeAttr("style")
```

	Auteur	Centre Afpas de Créteil	Formation	Date de mise à jour	Page 6
	Didier Bonneau	GRN 164	DWWM	07/11/2022	support_de_cours-jquery.docx

B. SUPPRIMER UN ATTRIBUT

La méthode est « **removeAttr()** » dont l'unique paramètre est le nom de cet attribut.

\$(selecteur).removeAttr(nomAttribut)

C. CONNAITRE OU MODIFIER LA VALEUR DE L'ATTRIBUT « VALUE »

Il faut faire appel à la méthode « **val()** ».

Sans paramètre, cette méthode permet de récupérer, sous forme d'une chaîne de caractères, la valeur de l'attribut « value » des champs d'un formulaire.

En fournissant une chaîne de caractère à cette méthode il est possible de modifier la valeur de l'attribut « value ».

D. CONNAITRE OU MODIFIER LA VALEUR DE L'ATTRIBUT « CLASS »

1. Ajouter une ou plusieurs classes

La méthode « **addClass()** » permet d'ajouter une ou plusieurs classes CSS aux objets sélectionnés.

\$(selecteur).addClass(nomClass)

2. Savoir si un élément est de classe déterminée

La méthode « **hasClass()** » permet de tester si un élément est d'une certaine classe. Cette méthode renvoie un booléen « true » en cas de succès.

\$(selecteur).hasClass(nomClass)

Exemple de code

Soit le bout de code html :


```
<form action="...">
  <input type="radio" name="civilite" value="M."/>
  <input type="radio" name="civilite" value="Mme"/>
</form>
```

Script jQuery

```
// recuperation de la valeur de l'attribut value
// sur le bouton radio qui est sélectionné
var choix = $("input:radio:checked").val();
```

```
// exemple d'envoi du texte dans une div
$("div").text(choix);
```

```
// modification de l'attribut value
$("input:radio:checked").val("Monsieur");
```

	Auteur	Centre Afpas de Créteil	Formation	Date de mise à jour	Page 7
	Didier Bonneau	GRN 164	DWWM	07/11/2022	support_de_cours-jquery.docx

VI. MANIPULATION DU CSS

De même que l'on a pu accéder ou modifier des attributs d'un élément, il est possible d'accéder ou de modifier directement les propriétés de style d'un élément sélectionné.

A. LECTURE OU MODIFICATION DE PROPRIETES CSS

Pour cela il faut utiliser la méthode « `css()` » qui suit la même syntaxe que la méthode « `attr()` ».

La syntaxe pour une lecture est :

`$(selecteur).css(nomPropriétéCss)`

« `nomPropriétéCss` » peut être une chaîne de caractères constante ou une variable.

Pour une modification ou un ajout :

`$(selecteur).css(nomPropriétéCss, valeurPropriété)`

« `valeurPropriété` » peut être une chaîne de caractères constante, une variable ou une fonction. jQuery (1.6) accepte aussi les valeurs relatives d'incrément ou de décrémentation « `"+=200"` » et « `"-=100"` ».

Modification de plusieurs propriétés css.

`$(selecteur).css({ propriétés })`

« `propriétés` » suit la même syntaxe que « `attr()` ». Néanmoins les couples propriété/valeur peuvent suivre la syntaxe du CSS et du Dom. Les deux lignes suivantes sont équivalentes :

```
{ 'background-color' : '#ffe', 'border-left': '2px solid #ccc' }
{ backgroundColor : '#ffe', borderLeft: '2px solid #ccc' }
```

Exemple de code

Soit le bout de code html :

```
<style>
  #action {
    width: 300px ;
    float : left ;
  }
</style>

<div id="action">
  
</div>
```


Script jQuery

```
// récupération de la largeur de la div
var largeur = $("#action").css("width");
```

attention : `css("width")` ou `css("height")` renvoi une chaîne contenant la valeur et l'unité ex: `300px`. Pour être utilisé dans des calcul, utiliser la méthode « `height()` » (voir si après).

```
// modification de la largeur de la div
$("#action").css("width", 400);
```

```
// ajout de propriétés
$("#action").css({ margin: '10px',
                  border: '1px'
                });
```

	Auteur	Centre Afpa de Créteil	Formation	Date de mise à jour	Page 8
	Didier Bonneau	GRN 164	DWWM	07/11/2022	support_de_cours-jquery.docx

B. ACCES DIRECT AUX PROPRIETES DE DIMENSION

Par certaines méthodes il est possible d'accéder directement aux propriétés de dimension des éléments.

1. La hauteur

lecture : `$("sélecteur").height();`

modification : `$("sélecteur").height(250);`

2. La largeur

lecture : `$("sélecteur").width();`

modification : `$("sélecteur").width(250);`

3. voir aussi

`innerHeight(), innerWidth` -> `height() + padding`

`outerHeight(), outerWidth` -> `height() + padding + border`

C. ACCES DIRECT AUX PROPRIETES DE POSITION

Pour connaître la position d'un élément il existe deux méthodes qui permettent de récupérer la valeur « top » et « left » soit par rapport au document, soit par rapport à l'élément parent.

Attention ces méthodes renvoient toutes les deux un objet contenant deux propriétés « top » et « left ».

- Par rapport à l'élément parent, utiliser la méthode « **position()** ».
- Par rapport au document, utiliser la méthode « **offset()** ».

Exemple de code

Soit le bout de code html :

```
<style>
    #action {
        width: 300px ;
        height: 400px ;
    }
</style>

<div id="action">
    
</div>
```

Script jQuery


```
// récupération de la largeur de la div
var largeur = $("#action").width();
```

```
// modification de la largeur de la div
$("#action").width(largeur*2);
```

```
// récupération des positions top et left
```

```
// par rapport à l'élément parent
var posRelativeX = $("#action").position().left
var posRelativeY = $("#action").position().top
```

```
// par rapport au document
var posAbsolueX = $("#action").offset().left
var posAbsolueY = $("#action").offset().top
```

	Auteur	Centre Afpas de Créteil	Formation	Date de mise à jour	Page 9
	Didier Bonneau	GRN 164	DWWM	07/11/2022	support_de_cours-jquery.docx

VII. LES EVENEMENTS

jQuery reprend la gestion des évènements vue en JavaScript. Plusieurs manières existent pour attacher un gestionnaire d'évènement (event handler) à un élément du DOM :

- Utiliser la méthode « bind() » (depuis la version 1.0 de jQuery)
- Utiliser la méthode « on() » (depuis la version 1.7 de jQuery)
- Utiliser les méthodes portant les noms des évènements (voir ci-après)

Un gestionnaire d'évènement peut être une fonction jQuery déclarée explicitement ou une fonction anonyme (sans nom, déclarée par function()).

Un gestionnaire d'évènement peut admettre un paramètre afin de récupérer l'objet évènement l'intérieur du gestionnaire. C'est le cas de l'évènement « mousemove » où il est intéressant de récupérer l'évènement afin d'accéder aux coordonnées de la souris (voir exemple). Attention, l'objet récupéré qui est à l'origine de l'évènement n'est pas un objet jQuery, il faut le traduire en objet jQuery en appelant la fonction \$() (voir exemple du click avec \$(event.target)) pour d'utiliser les méthodes jQuery sur cet objet.

A. LA METHODE « BIND() »

Cette méthode est remplacée par la méthode « on() » ci-dessous. Elle la même syntaxe de base.

B. LA METHODE « ON() »

Sa forme générale est un peu compliquée. Elle admet quatre paramètres dont deux sont optionnels.

\$(selecteur).on(évènements [, filtre] [, données], gestEvent(objetEvent))

Où :

- évènements : représentant un ou plusieurs évènements séparés par un espace
- filtre : c'est un sélecteur permettant de filtrer les descendants de l'objet sur lequel le gestionnaire d'évènement est appliqué et qui répondront à cet évènement. On parle de délégation.
- données : sont des données qui sont passées au gestionnaire d'évènement et récupérées par « objetEvent.data ».
- gestEvent : qui est le gestionnaire d'évènement avec son paramètre ObjetEvent recevant l'objet évènement lui-même.

Exemple

Avec une fonction explicite:


```
function cliquer() { alert("vous venez de cliquer !"); }
$("button").on("click", cliquer);
```

Avec une fonction anonyme:

```
$("button").on("click", function(){ alert("vous venez de cliquer !"); });
```

Avec délégation:

```
$("tbody").on("click", 'tr', function(){ alert("vous venez de cliquer !"); });
```

	Auteur	Centre Afpa de Créteil	Formation	Date de mise à jour	Page 10
	Didier Bonneau	GRN 164	DWWM	07/11/2022	support_de_cours-jquery.docx

C. LES METHODES DIRECTES

Ces méthodes portent le nom de l'évènement sans le préfixe « on » et attendent en paramètre un voir deux gestionnaires.

Ci-dessous la liste des méthodes les plus courants :

1. Relatifs à la souris

- `click()`, `dblclick()`

Déclenche un gestionnaire sur un click ou un double click de la souris
Syntaxe : **`$(sélecteur).click(gestionnaire)`**

- `hover()`

Admet deux gestionnaires en paramètre:

Déclenche le premier lorsque la souris passe sur élément

Déclenche le deuxième lorsque la souris quitte l'élément

Syntaxe : **`$(sélecteur).hover(gestionnaire1, gestionnaire2)`**

- `focusin()`, `focusout()`

Déclenche un gestionnaire lorsqu'un élément prend ou perd le focus.
Peuvent être déclenchés à partir de leurs parents.

Syntaxe : **`$(sélecteur).focusXXX(gestionnaire)`**

Exemple de code

Soit le bout de code html :

```
<div id="cible">
  <h2>Cliquez ici</h2>
</div>
```

Script jQuery

```
$("#cible").click(function() {
  alert("Vous venez de cliquer !.");
});
```

```
$("#cible").click(function(event) {
  var $elt = $(event.target) ;
  alert("Vous venez de cliquer sur" . $elt.text());
});
```

```
$("#cible").hover(
  function() {
    alert("Vous passez sur l'élément.");
  },
  function() {
    alert("Vous quittez l'élément.");
  }
);
```

- mousedown(), mouseup(), mouseenter(), mouseout()

Déclenche un gestionnaire sur :

Enfoncement du bouton de la souris

Relâchement du bouton de la souris

A l'entrée de la souris

A la sortie de la souris

Syntaxe : **\$(sélecteur).mouseXXX(gestionnaire)**

- mousemove()

Déclenche un gestionnaire sur un mouvement de la souris

Syntaxe : **\$(sélecteur).mousemove(gestionnaire([évènement]))**

2. Relatifs au clavier

- keydown(), keyup(), keyPress()

Déclenche un gestionnaire lorsqu'une touche clavier est enfoncée ou relâchée

Syntaxe : **\$(sélecteur).keyXXX(gestionnaire)**

3. Relatifs au gestionnaire d'évènement

- one()

Déclenche une et une seule fois le gestionnaire sur l'évènement considéré

Syntaxe : **\$(sélecteur).one(évènement, gestionnaire)**

Exemple de code

Soit le bout de code html :

```
<div id="cible">
  <h2>bougez la souris</h2>
</div>
```

Script jQuery

```
$("#cible").mousemove(function(event) {
  var msg = "Appel du gestionnaire mousemove() ";
  msg += "les coordonnées de la souris sont : ";
  msg += event.pageX + ", " + event.pageY;
});
```

Exemple de code de l'évènement one()

```
$("#cible").one("click", function() {
  alert("Ceci n'est affiché qu'une seule fois !.");
});
```


4. Relatifs aux formulaires

- `focus()`, `blur()`

Déclenche un gestionnaire lorsqu'un élément prend ou perd le focus sur des éléments de formulaire

Syntaxe : **`$(sélecteur).focus(gestionnaire)`**

- `change()`

Déclenche un gestionnaire lorsque la « valeur » d'un élément change

Syntaxe : **`$(sélecteur).change(gestionnaire)`**

- `select()`,

Déclenche un gestionnaire lorsqu'un texte est sélectionné dans un élément « `input type="text"` ou `textarea` »

Syntaxe : **`$(sélecteur).select(gestionnaire)`**

- `submit()`

Déclenche un gestionnaire lorsqu'un formulaire est envoyé

Syntaxe : **`$(sélecteur).submit(gestionnaire)`**

5. Relatifs au navigateur

- `resize()`

Déclenche un gestionnaire lorsque la fenêtre du navigateur est retaillée

Syntaxe : **`$(sélecteur).resize(gestionnaire)`**

- `scroll()`

Déclenche un gestionnaire lorsque l'on scroll à l'intérieur d'un élément

Syntaxe : **`$(sélecteur).scroll(gestionnaire)`**

6. Relatifs au document

- `Load()`, `unload()`

Déclenche un gestionnaire lorsqu'un élément associé à une URL est totalement chargé

Syntaxe : **`$(sélecteur).load(gestionnaire)`**

- `ready()`

Déclenche un gestionnaire lorsqu'un le document html (le DOM) est complètement chargé

Syntaxe : **`$(document).ready(gestionnaire)`**

Exemple de code du gestionnaire « `submit()` »

Soit le bout de code html :

```
<form id="cible" action="destination.html">
  <input type="text" value="Bienvenu" />
  <input type="submit" value="Aller" />
</form>
```

Script jQuery

```
$('#cible').submit(function() {
  reponse=confirm('Confirmez l'envoi ?');
  if( !reponse ){
    return false; // arrêt de l'envoi
  }
});
```

Si le gestionnaire reçoit l'évènement, il est possible de l'annuler en appelant la méthode « `.preventDefault()` »

VIII. LES EFFETS

A. AFFICHER ET CACHER DES ELEMENTS

Les premiers effets que l'on peut mettre en œuvre sont les possibilités de montrer ou de cacher des éléments.

La syntaxe la plus simple est :

\$(sélecteur).show() ; ou \$(sélecteur).hide() ;

Une autre syntaxe plus complexe permet de passer un ou deux paramètres à ces méthodes. Le premier sera un temps de durée en milliseconde et le deuxième est un gestionnaire appelé lorsque l'effet est complètement terminé.

\$(sélecteur).show(durée [, gestionnaire]) ;

\$(sélecteur).hide(durée [, gestionnaire]) ;

- Le paramètre « durée » peut prendre les valeurs prédéfinies « slow », « normal » et « fast ».

Une méthode « toggle() » permet à chaque click de cacher et de montrer un élément. Elle peut accepter deux gestionnaires en paramètre qui seront appelés au click paire et impaire.

B. JOUER SUR LA HAUTEUR VERTICALEMENT

Les méthodes « slideDown() » et « slideUp() » permettent de jouer sur la hauteur d'un élément.

\$(sélecteur).slideXXX(vitesse [, gestionnaire]) ;

- Le paramètre « vitesse » peut prendre les valeurs « slow », « normal » et « fast ».

Il existe aussi une méthode « slideToggle(vitesse, gestionnaire) » qui passe de « slideDown() » à « slideUp() » à chaque click.

Exemple de code de la méthode « hide » avec un gestionnaire


Soit le bout de code html :

```
<button>Cacher le paragraphe</button>

<p style="display: block">
    Un paragraphe qui disparaît
</p>
```

Script jQuery

```
$("#button").click(function () {
    $("#p").hide("slow", function(){
        alert("disparu !");
    });
});
```

	Auteur	Centre Afpas de Créteil	Formation	Date de mise à jour	Page 14
	Didier Bonneau	GRN 164	DWWM	07/11/2022	support_de_cours-jquery.docx

C. JOUER SUR L'OPACITE

Les méthodes « `fadeIn()` » et « `fadeOut()` » permettent de jouer sur l'opacité d'un élément, le faisant ainsi apparaître ou disparaître progressivement.

`$(sélécteur).fadeXXX(vitesse [, gestionnaire]) ;`

Le paramètre « vitesse » peut prendre les valeurs « slow », « normal » et « fast ».

Une autre méthode existe permettant de régler l'opacité maximale.

`$(sélécteur).fadeTo(vitesse, opacitéMax, [, gestionnaire]) ;`

- Le paramètre « vitesse » peut prendre les valeurs « slow », « normal » et « fast ».
- Le paramètre « opacitéMax » prend une valeur entre 0 et 1.
- Avec le paramètre « vitesse » à 0, cette méthode permet de jouer sur l'opacité.

Là encore, la méthode « `fadeToggle(vitesse, gestionnaire)` » permet de passer de `fadeIn` à `fadeOut`.

D. DIFFERER LES EFFETS

Pour différer un effet, il faut appeler la méthode « `delay()` » qui prend en paramètre une durée en milliseconde.

Exemple de code de « `fadeTo()` »

Soit le bout de code html :

```
<div id="cible">
    Cliquez ici
</div>

```

Script jquery

```
$('#cible').click(function() {
    $('#idImg').fadeTo('slow', 0.5, function() {
        Alert("Animation terminée");
    });
});
```

Exemple de code de « `fadeTo()` »

```
<p><button>Démarrer</button></p>
<div class="une"></div>
<div class="deux"></div>
```

```
$(".button").click(function() {
    $(".div.une").slideUp(300)
        .delay(800)
        .fadeIn(400);
    $(".div.deux").slideUp(300).fadeIn(400);
});
```

E. CREER UNE ANIMATION

Par la méthode « animate() » on est capable de créer une petite animation en réglant ses propres paramètres.

La syntaxe est la suivante :

\$(sélecteur).animate(paramètres [, vitesse][, easing][, gestionnaire]) ;

« paramètres » est un ensemble de couples « propriété/valeur » sur lequel est basée l'animation. La propriété peut être la largeur, la hauteur, l'opacité,... et la valeur fixe cette propriété.

Exemple de code de « fadeTo() »

Soit le bout de code html :

```
<div id="cible">
    Cliquez ici
</div>

```

Script jQuery

```
$('#cible').click(function() {
    $('#idImg').animate({
        opacity: 0.25,
        left: '+=50',
        height: 'toggle'
    }, 5000);
});
```

Sur cette animation :

1) au premier click

l'opacité part de 1 à 0.25

la position gauche est incrémentée de 50


la hauteur part de 123 à 0

2) // puis au 2ème click de 0 à 123

l'opacité reste à 0.25

la position gauche est incrémentée de 50

la hauteur part de 0 à 123

	Auteur	Centre Afpa de Créteil	Formation	Date de mise à jour	Page 16
	Didier Bonneau	GRN 164	DWWM	07/11/2022	support_de_cours-jquery.docx

IX. UTILISER LE DOM

A partir d'un l'élément ou d'une collection d'éléments sélectionné par un sélecteur quelconque, il est possible d'atteindre les autres éléments en parcourant l'arbre DOM. Ces éléments peuvent se trouver :

- Au même niveau dans l'arbre -> frères
- A un niveau inférieur -> enfants et descendants
- A un niveau supérieur -> parent et ascendants

A. ITERER SUR UNE COLLECTION

Lorsque l'on a une collection d'éléments, il est parfois intéressant de traiter tous ces éléments l'un après l'autre. La méthode « each() » permet ceci.

`$(sélecteur).each(fonction())`

Le paramètre « fonction() » sera appliqué à tous les éléments qui répondent au sélecteur.

B. FILTRER UNE COLLECTION D'ELEMENTS

Sur une collection d'éléments retournée par un sélecteur, il est souvent utile d'atteindre un certain élément de cette liste. Plusieurs méthodes existent.

1. Le premier ou le dernier

Réduit la sélection au premier ou au dernier élément.

`$(sélecteur).first() ou $(sélecteur).last()`

2. Ayant un certain index

Choix d'un élément à partir de l'élément sélectionné.

`$(sélecteur).eq(index)`

- « index » peut avoir une valeur positive ou négative

Exemple de code

Soit le bout de code html :

```
<ul>
  <li id="first">premier item</li>
  <li>deuxième item</li>
  <li>troisième item</li>
</ul>
```


Script jQuery

```
$('.li').each( function(index){
  alert( index + ' : ' + $(this).text() ;
});
```

Affiche successivement 3 boîtes de dialogue avec les textes :

```
0 : premier item
1 : deuxième item
2 : troisième item
```

`$("#first").eq(2)` équivalent à `$("li").last()`

	Auteur	Centre Afpa de Créteil	Formation	Date de mise à jour	Page 17
	Didier Bonneau	GRN 164	DWWM	07/11/2022	support_de_cours-jquery.docx

3. Correspondant à un certain type

\$(sélecteur).is([filtre])

- « filtre » est le type d'un élément
- Cette méthode renvoie une valeur booléenne

4. Ayant un descendant d'un certain type

\$(sélecteur).has([filtre])

- « filtre » est le type d'un élément
- Cette méthode renvoie un objet jQuery

C. CHERCHER AU MEME NIVEAU

1. Le frère suivant ou précédent

Retourne le premier frère suivant ou précédent de chaque élément de la sélection.

\$(sélecteur).next([filtre]) ou **\$(sélecteur).prev([filtre])**

Le paramètre « filtre » est un sélecteur permettant de retourner le frère qui correspond à ce sélecteur.

2. Les frères suivants ou précédents

Il est possible de rechercher tous les frères.

\$(sélecteur).nextAll([filtre]) ou **\$(sélecteur).prevAll([filtre])**

3. Tous les frères

Recherche de tous les frères précédents et suivants :

\$(sélecteur).siblings([filtre])


Exemple de code

Soit le bout de code html :

```
<ul>
  <li id="first">premier item</li>
  <li>deuxième item</li>
  <li>troisième item</li>
</ul>
```

Script jQuery

```
$('#ul').click( function(event){
  var $cible = $(event.target);
  if ( $cible.is('li') {
    $cible.css('text-color': 'red');
  }
});
```

	Auteur	Centre Afpas de Créteil	Formation	Date de mise à jour	Page 18
	Didier Bonneau	GRN 164	DWWM	07/11/2022	support_de_cours-jquery.docx

D. CHERCHER A UN NIVEAU INFERIEUR

1. Les enfants

Afin de trouver tous les éléments enfants (1 niveau en dessous), il faut utiliser la méthode « children() ».

\$(sélecteur).children();

2. Les descendants

Pour les descendants (plusieurs sous niveaux dans l'arbre dom) utiliser la méthode « find() ».

\$(sélecteur).find();

Il est possible de filtrer parmi tous les descendants un certain type d'élément en fournissant un sélecteur à cette méthode.

\$(sélecteur).find([filtre]);

E. CHERCHER A UN NIVEAU SUPERIEUR

1. Le parent

Afin de trouver l'élément parent (1 niveau au-dessus), il faut utiliser la méthode « parent() ».

\$(sélecteur).parent();

2. Les ascendants

Si on veut atteindre tous les ascendants utiliser « parents() ».

\$(sélecteur).parents();

Pour ces deux méthodes, on peut filtrer certains ascendants en donnant un paramètre « filtre ».

\$(sélecteur).parent([filtre]);

Exemple de code

Soit le bout de code html :

```
<ul id="first">
  <li>premier item
    <ul>
      <li>premier sous item</li>
      <li>deuxième sous item</li>
      <li>troisième sous item</li>
    </ul>
  </li>
  <li>deuxième item</li>
  <li>troisième item</li>
</ul>
```

Script jQuery

```
$('#ul#first').find( 'li').css('font-weight', 'bold');
```

X. MANIPULATION DU DOM

Il est aussi possible par quelques méthodes de modifier la structure de l'arbre DOM.

A. MODIFIER LE CONTENU D'UN ELEMENT

La méthode « `html()` » renvoi ou modifie la totalité du contenu d'un élément.

- Pour lire le contenu :

`$(sélecteur).html()`

- Pour remplacer la totalité du contenu :

`$(sélecteur).html('morceau de html')`

- Pour supprimer la totalité du contenu :

`$(sélecteur).empty()`

- Pour lire ou modifier la valeur du nœud texte d'une balise :

`$(sélecteur).text()`

B. INSERER UN ELEMENT

1. Comme enfant

Les méthodes « `prepend()` » et « `append()` » ajoutent du contenu en tant qu'enfant (respectivement premier et dernier) aux éléments sur lesquels s'applique la méthode.

Les méthodes « `prependTo()` » et « `appendTo()` » s'appliquent au contenu qui est ajouté aux éléments cibles (voir exemples).

- Premier enfant : **`$(selecteur).prepend(contenuHtml)`**
`$(contenuHtml).prependTo(cible)`
- Dernier enfant : **`$(selecteur).append(contenuHtml)`**
`$(selecteur).appendTo(cible)`

Exemple de code

Soit le bout de code html :

```
<ul id="first">
  <li>premier item</li>
  <li>deuxième item</li>
</ul>
```

Script jquery

```
$( 'ul#first' ).append( '<li>troisième item</li>' );
$( '<li>quatrième item</li>' ).appendTo( 'ul#first' );
```

```
<ul id="first">
  <li>premier item</li>
  <li>deuxième item</li>
  <li>troisième item</li>
</ul>
```

```
$( 'ul#first' ).prepend( '<li>super prems item</li>' );
$( '<li>quatrième item</li>' ).prependTo( 'ul#first' );
```

```
<ul id="first">
  <li>super prems item </li>
  <li>premier item</li>
  <li>deuxième item</li>
</ul>
```

```
$( 'ul#first' ).before( '<p>voici la liste</p>' );
```

```
<p>voici la liste</p>
<ul id="first">
  <li>premier item</li>
  <li>deuxième item</li>
  <li>troisième item</li>
</ul>
```


1. Comme frère

Les méthodes « before() » et « after() » ajoutent du contenu en tant que frère (respectivement avant et après) aux éléments sur lesquels s'applique la méthode.

Les méthodes « insertBefore() » et « insertAfter() » s'appliquent au contenu qui est ajouté aux éléments cibles (voir exemples).

- Avant : `$(selecteur).before(contenu)`
`$(contenu).insertBefore(cible)`
- Après : `$(selecteur).after(contenu)`
`$(contenu).insertAfter(cible)`

2. Comme père

Il est possible d'entourer chaque élément ou une collection d'éléments par un certain contenu (qui devient père). C'est le rôle des méthodes « wrap() » et « wrapAll() ».

- chaque élément : `$(selecteur).wrap(elementPere)`
- la collection : `$(selecteur).wrapAll(elementPere)`

C. REMPLACER UN ELEMENT

La méthode « replaceWith () » remplace l'élément sélectionné par le contenu donné en paramètre. La méthode « replaceAll () » s'appliquent au contenu qui va remplacer l'élément cible (voir exemples).

- syntaxe `$(selecteur).replaceWith(element)`
`$(element).replaceAll(cible)`

D. COPIER UN ELEMENT

Il est possible de faire une copie (un clone) en profondeur d'un élément.

`$(selecteur).clone()`

```
$('li').wrap( '<div class="monLi"></div>' );
```

```
<ul id="first">
  <div class="monLi">
    <li>premier item</li>
  </div>
  <div class="monLi">
    <li>deuxième item</li>
  </div>
</ul>
```

```
$('li').wrapAll( '<div class="monLi"></div>' );
```

```
<ul id="first">
  <div class="monLi">
    <li>premier item</li>
    <li>deuxième item</li>
  </div>
</ul>
```

```
$('li').replaceWith('<h2>un item</h2>');
$('<h2>un item</h2>').replaceAll('li');
```

```
<ul id="first">
  <h2>un item</h2>
  <h2>un item</h2>
  <h2>un item</h2>
</ul>
```

Cloner des éléments :

```
Var monLI = $('li :first').clone()
```

XI. AJAX AVEC JQUERY

jQuery possède une méthode spécifique à la mise en œuvre d'applications Ajax :

\$.ajax() ;

Cette méthode ajax() comporte plusieurs options (type, url, data, dataType, success) qui permettent de configurer la requête Ajax ainsi créée.

type : chaîne de caractères valant GET ou POST

url : l'URL du serveur

data : string ou objet javascript représentant la queryString

dataType : chaîne de caractères précisant le type des données :

- xml
- html
- json
- script
- text

success : fonction de traitement de la réponse

error : fonction de traitement des erreurs éventuelles

Beaucoup d'autres méthodes existent pour faire de l'ajax :

Voir \$.get(), \$.post(), \$.ajaxStart(), \$.ajaxStop(), \$.ajaxSuccess(), \$.getJSON(), ...

Pour plus de détails, voir la doc <http://api.jquery.com/category/ajax/>


Exemple d'une requête POST :

```
$.ajax({
    type: 'POST',
    url: 'serveur.php',
    data: "nom=Dejour&prenom=Adan",
    success: fonctionRappel
});

function fonctionRappel(reponse) {
    alert( "La réponse est : " + reponse );
}
```

Qui peut aussi s'écrire en utilisant une fonction anonyme :

```
$.ajax({
    type: 'POST',
    url: 'serveur.php',
    data: {nom:'Dejour', prenom:'Adan'},
    success: function (reponse) {
        alert( "La réponse est : " + reponse );
    }
});
```

	Auteur	Centre Afpa de Créteil	Formation	Date de mise à jour	Page 22
	Didier Bonneau	GRN 164	DWWM	07/11/2022	support_de_cours-jquery.docx

XII. QUELQUES PLUG-INS

Pour manipuler les tableaux :

<http://datatables.net/>


<http://tipsntools.pasapix.com/integration/plugin-jquery-tableau/>

Le site de jQuery :

<http://archive.plugins.jquery.com/most-popular>

Le site JQuery-UI

<http://jqueryui.com/>

	Auteur	Centre Afpa de Créteil	Formation	Date de mise à jour	Page 23
	Didier Bonneau	GRN 164	DWWM	07/11/2022	support_de_cours-jquery.docx