

## Développement Internet

### JavaScript - Librairie jQuery

*jQuery est une librairie JavaScript qui facilite grandement l'écriture de scripts.*

## 1 Présentation

JavaScript est parfois lourd à utiliser et impose de reprendre régulièrement le même code, de truffer le code de tests pour s'adapter au navigateur, ... Beaucoup de bibliothèques de scripts permettant d'alléger ou éliminer ces défauts : jQuery, Mootools, Dojo, ...

jQuery est certainement la plus répandue et un grand nombre de firmes majeures du domaine de l'IT l'utilise<sup>1</sup>). Ses fonctionnalités incluent :

1. manipulation du DOM
2. manipulation du CSS
3. gestion des événements
4. animations
5. Ajax (que nous reverrons au chapitre suivant)

Une référence de jQuery se trouve à l'adresse <https://jquery.com/>.

---

1. cf. <http://w3techs.com/technologies/details/js-jquery/all/all> pour une liste plus exhaustive



## 2 Utilisation

### 2.1 Inclusion de la librairie

La librairie elle-même doit être référencée dans votre page html, soit elle doit être fournie par votre site soit être référencée sur un site tiers.

Souvent on utilise un CDN, ou *Content Delivery Network* : une infrastructure réseau permettant de fournir vite et sans interruption de service des ressources stables dans le temps<sup>2</sup>, par exemple celui de :

1. [jQuery](#)
2. [Google](#)
3. [Microsoft](#)

D'autres CDN peuvent se trouver à l'adresse : <https://jquery.com/download/#other-cdns>. Ces CDN ont l'avantage d'être utilisés par tellement de sites que les clients de nos pages risquent d'avoir déjà la librairie en mémoire cache (et ne doivent donc pas la re-télécharger). En outre, même si la librairie n'est pas en cache, son téléchargement par l'utilisateur n'utilisera de toute façon pas la bande passante de notre site.

On inclut cette librairie comme tout autre script JavaScript, donc :

```
<script src="https://code.jquery.com/jquery-3.4.1.min.js"
  integrity="sha256-CSXorXvZcTkaix6Yvo6HppcZGetbYMGWSFlBw8HfCJo="
  crossorigin="anonymous"></script>
```

Ou plus simplement :

```
<script
  src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js">
</script>
```

Ces librairies sont condensées (*minified* en anglais, représenté dans le nom du fichier par un *.min*) pour minimiser le trafic réseau, donc quasiment illisibles (on parle de version de « production »). Nous pouvons accéder à la librairie « lisible » à l'adresse <https://code.jquery.com/jquery-3.4.1.js>. Cette version est généralement utilisée pour le développement, elle nous permettra de vérifier qu'il s'agit bien d'un « gros » fichier de fonctions JavaScript.

Notons bien sûr que, puisque les scripts JavaScript sont chargés dans leur ordre d'apparition dans le DOM, qu'il faudra s'assurer que jQuery est chargé **avant** nos scripts qui l'utilisent.

---

2. [https://fr.wikipedia.org/wiki/Réseau\\_de\\_diffusion\\_de\\_contenu](https://fr.wikipedia.org/wiki/Réseau_de_diffusion_de_contenu)).

## 2.2 Syntaxe de jQuery

Pour analyser quelques exemples d'utilisations de jQuery, nous utiliserons le fichier suivant :

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>Exemples d'utilisation de jQuery</title>
  <script src="https://code.jquery.com/jquery-3.4.1.min.js"
    integrity="sha256-CSXorXvZcTkaiX6Yvo6HppcZGetbYMGWSFlBw8HfCJo="
    crossorigin="anonymous"></script>
  <script src="script.js"></script>
</head>
<body>
  <div id="a">TODO write content</div>
  <p>Coucou</p>
  <p>Bonjour</p>
  <p>Hello</p>
</body>
</html>
```

Si le script `script.js` est vide, l'affichage de la page ressemble à ceci :

TODO write content

Coucou

Bonjour

Hello

### 2.2.1 La fonction jQuery

La librairie jQuery propose une seule fonction, `jQuery()`, qui peut aussi être appelée par son alias, `$()`. Cette fonction permet de rapidement sélectionner un élément, puis d'appeler d'autres fonctions sur cet élément, sous la forme suivante :

```
$(sélecteur).action(...)
```

Cette sélection se fait via des sélecteurs, qui utilisent la même syntaxe que les sélecteurs CSS.

Par exemple, le script suivant

```
$(function() {  
    $("#a").html("Je modifie ceci avec jQuery");  
});
```

demande que l'élément donc l'id est "a" soit remplacé par l'argument de la **méthode** `html()`, qui est une des méthodes définies dans jQuery.

### 2.2.2 Les événements en jQuery

La plupart des événements du DOM disposent de leur équivalent en jQuery, généralement avec le même nom. Par exemple, la méthode `click()`<sup>3</sup> associe à l'élément HTML sélectionné un événement qui se produit lorsque l'utilisateur clique sur cet élément.

Ajoutons un événement de ce type à notre script :

```
$(function() {  
    $("#a").html('Je modifie ceci avec jQuery');  
    $("#a").click(function(){  
        $(this).hide();  
    });  
});
```

Au moment où l'élément d'identifiant "a" reçoit un clic de souris, il va maintenant disparaître via la méthode `hide()`, qui permet de "cacher" l'élément. Le mot-clé `this` désigne l'élément courant.

On notera que la méthode `on(event, handler)`<sup>4</sup> permet d'ajouter le gestionnaire d'un événement via son nom dans le DOM.

Enfin, notons que la méthode `jquery()` (et son alias `$()`), si elle reçoit un gestionnaire d'événement en paramètre au lieu d'un sélecteur CSS, sert de raccourci à l'expression :

```
$(document).ready(handler)
```

C'est-à-dire qu'il s'agit de l'événement qui se produit lorsque le DOM a terminé de se charger, qui comme nous l'avons vu permet de s'assurer que nos scripts JavaScript s'exécutent sur la page dans son ensemble.

### 2.2.3 Les méthodes de jQuery

Nous avons déjà vu quelques méthodes qui peuvent être appelées sur un élément une fois qu'il a été sélectionné via jQuery. Il ne sera pas possible d'en voir une liste exhaustive dans le cadre de ce cours : comme toujours, nous ferons référence à la documentation sur le sujet.

---

3. <http://api.jquery.com/click/>

4. <https://api.jquery.com/on/>

Notons toutefois quelques méthodes fréquentes :

1. `html(text)` <sup>5</sup> permet de modifier le contenu d'un élément HTML (similaire à `innerHTML` du DOM)
2. `css()` <sup>6</sup> permet d'accéder au style CSS de l'élément. Elle admet plusieurs syntaxes :
  - (a) `css("background")` retourne la valeur de la propriété CSS `background`.
  - (b) `css("color", "red")` modifie la valeur de la propriété CSS `color` en `red`.
  - (c) `css({ "color": "red", "font-style": "italic" })` modifie plusieurs propriétés d'un coup.

Par souci de séparation des rôles dans nos codes, il est généralement préférable de définir une classe CSS possédant ces propriétés, puis de donner cette classe aux éléments correspondants. De cette façon toutes les informations de style restent reléguées au CSS.

3. `attr()` <sup>7</sup> permet d'accéder à un attribut d'un élément HTML. Par exemple :
  - (a) `attr("class")` retourne la valeur de l'attribut `class` de l'élément.
  - (b) `attr("class", "myClass")` modifie la valeur de l'attribut `class` de l'élément en `myClass`.
4. `find()` <sup>8</sup> cherche et retourne les éléments descendants correspondants à certains critères, comme un sélecteur ou l'appel à la fonction `jQuery()`. Dans ce deuxième cas, les éléments retournés par cet appel à `jQuery()` sont filtrés pour ne conserver que ceux qui sont descendants de l'élément sur lequel `find()` est appelé.

Notons que les méthodes qui **modifient** un élément HTML (y compris par un ajout de gestionnaire d'événement) renvoient l'élément lui-même en résultat. Cela nous permet d'effectuer du **chaînage de méthodes**, où l'on appelle des méthodes successivement sans devoir rappeler la fonction `jQuery()`. Cela nous permet de gagner en efficacité, et d'avoir une écriture plus compacte.

Par exemple, nous pouvons modifier notre script précédent :

```
$(function() {  
    $("#a").html('Je modifie ceci avec jQuery').click(function(){  
        $(this).hide();  
    });  
});
```

---

5. <https://api.jquery.com/html/#html2>

6. <https://api.jquery.com/css/>

7. <https://api.jquery.com/attr/>

8. <https://api.jquery.com/find/>

## 2.3 Validation de formulaires

Rien n'est prévu à la base de jQuery mais le plugin validate<sup>9</sup> est accessible pour subvenir à nos besoins. Nous devons bien sûr le télécharger :

```
<script  
src="https://cdn.jsdelivr.net/npm/jquery-validation@1.19.1/dist/jquery.validate.min.js"  
</script>
```

La validation se fait via la création d'un objet JavaScript dans lequel nous définirons diverses règles que la validation prendra en compte. Nous reviendrons sur la syntaxe des objets JavaScript<sup>10</sup> lors du chapitre dédié à JSON, mais prenons un exemple :

```
$("#formId").validate({  
  rules: { //gestion des règles de validation  
    nom: "required", // l'élément de nom nom est obligatoire  
           // et ne reprend pas d'autre règle  
    mail: {  
      email: true // l'élément de nom mail est un email  
           // mais pas obligatoire  
    },  
    numero: { // numero obligatoire et compris entre 30000  
              // et 999999  
      required: true,  
      range: [30000, 99999]  
    },  
    //...  
  },  
  messages: { // redéfinition des messages  
    nom: "Le nom est obligatoire",  
    mail: {  
      email: "Entrez une adresse mail valide"  
    },  
    numero: {  
      required: "Le numéro est obligatoire",  
      range: "Le numéro doit être compris entre {0} et {1}"  
    },  
    //...  
  }  
});
```

---

9. <https://jqueryvalidation.org/>

10. [https://www.w3schools.com/js/js\\_objects.asp](https://www.w3schools.com/js/js_objects.asp)

## 2.4 Animations

jQuery permet de réaliser aisément un grand nombre d'animations<sup>11</sup>... dont il ne faut bien sûr pas abuser.

Prenez le temps de parcourir les quelques exemples présentés par W3School à partir de [http://www.w3schools.com/jquery/jquery\\_hide\\_show.asp](http://www.w3schools.com/jquery/jquery_hide_show.asp)

Un élément à bien comprendre est la notion de **callback function**.

## 3 Exercices

### 3.1 Gestion de sélections

Le fichier `boxes.html` contient deux boîtes, l'une contenant les dix premières lettres de l'alphabet chacune dans une `div` séparée, l'autre vide, qui contiendra les lettres sélectionnées par l'utilisateur.

En utilisant jQuery, assurez-vous que lorsque l'utilisateur clique sur une lettre dans la boîte non sélectionnée, celle-ci est sélectionnée et donc déplacée vers l'autre boîte. De même, cliquer sur une lettre sélectionnée permet de la dé-sélectionner, et de la replacer dans la boîte "non sélectionnées".

### 3.2 Galerie

À partir du document HTML `gallery.html`, prévoir le CSS et le jQuery nécessaires afin de ne présenter qu'une photo à la fois et de les faire défiler soit par clic, soit automatiquement par animation (glissement), soit automatiquement par fading.

### 3.3 Validation de formulaire

Reprenez la validation de formulaire écrite lors du TD sur JavaScript, et adaptez celle-ci en utilisant la validation par jQuery. Appréciez la facilité! :)

---

11. <https://api.jquery.com/category/effects/>