

# Bootstrap 4 : les plugins jQuery, introduction

jQuery est un framework Javascript devenu incontournable au fil des années. Les créateurs de Bootstrap l'ont choisi pour offrir des plugins esthétiques et faciles à mettre en œuvre, dans la plupart des cas sans même mettre les doigts dans du code Javascript ! Toutefois, si vous voulez obtenir le meilleur de ces plugins, il vous faudra un peu manipuler jQuery. Je vais donc dans ce chapitre rapidement présenter jQuery pour ensuite commencer à parler des librairies Javascript de Bootstrap.

## La librairie jQuery

### Présentation de jQuery

*Javascript, c'est quoi ?*

- C'est un langage de script qui permet de rendre les pages web interactives.
- C'est un langage qui fonctionne côté client et ne demande aucune ressource au niveau du serveur.
- C'est un langage interprété directement par le navigateur.

*jQuery, c'est quoi ?*

jQuery est une librairie Javascript, on dit aussi un framework, c'est à dire un ensemble de fonctionnalités bien pratiques pour simplifier la mise en œuvre de Javascript. On peut ainsi facilement :

- Modifier des éléments d'une page web : texte, style, animation...
- Gérer des événements tels : clic de souris, action sur une touche du clavier...
- Créer des effets graphiques...
- Utiliser Ajax...

# Installation de jQuery

Pour installer jQuery il faut faire un appel à la bibliothèque. Pour cela vous avez deux solutions :

1. Vous allez chercher la dernière version [sur le site](#) et vous copiez ce fichier dans un répertoire de votre serveur.
2. Vous référencez un CDN : [jQuery](#), [cdnjs](#), [Google CDN](#)...

Vous trouvez 2 possibilités au niveau du téléchargement :

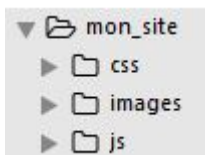
- **Uncompressed** : c'est une version lisible et commentée pour comprendre le code. Vous l'utilisez pendant le développement de vos sites si vous avez besoin de savoir ce qui se passe dans la bibliothèque.
- **Minified** : c'est une version compressée pour être la plus « légère » possible. C'est cette version que vous devez privilégier pour la mettre sur votre site en « production ».

Quand vous cliquez sur le lien, vous pouvez télécharger le fichier. Donnez-lui le nom **jquery.js** pour la version complète et **jquery.min.js** pour la version compressée.

Bootstrap se contente de la version slim de jQuery qui exclut ajax, les effets et du code déprécié. Mais évidemment ça fonctionne aussi avec la version complète.

## Placer jQuery

Il est bon d'organiser un peu les fichiers d'un site en dossiers thématiques, par exemple :



Si vous organisez ainsi l'architecture de votre site, déposez les fichiers jQuery dans le dossier **js**.

# Référencer jQuery

Si vous avez placé la librairie sur votre site, il suffit de référencer le fichier dans le code HTML, de préférence à la fin pour ne pas péjorer le chargement des éléments visuels :

```
<script src="js/jquery.slim.js"></script>
```

Si vous passez par un CDN, c'est la même chose, à part l'adresse qui change. Par exemple :

```
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCKRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous"></script>
```

*Pour mémoire le contrôle d'intégrité nous prémunit contre la falsification du fichier sur le serveur.*

# Utiliser jQuery

La principale fonction à connaître est **jQuery()**, mais on utilise le symbole **\$** pour faire plus court. On veut utiliser jQuery seulement lorsque le DOM de la page est totalement chargé pour éviter les erreurs d'exécution, ainsi on encapsule tout le code dans un bloc :

```
$(function(){
// ...
});
```

Si vous connaissez les sélecteurs CSS, vous savez déjà utiliser une grande partie de jQuery. En effet, cette librairie les utilise directement ! Par exemple :

```
var a = $('div');
```

Ici la variable « a » référence un objet jQuery qui sélectionne toutes les balises **div** du document.

jQuery propose aussi des méthodes, par exemple pour ajouter une

classe à un élément :

```
$('.img').addClass('ma_classe');
```

Cette action ajoute la classe **ma\_classe** à toutes les balises **<img>** du document. Il est possible de chaîner les méthodes. Par exemple :

```
$('#el').css('background-color', 'red').fadeIn('slow');
```

Ici on change la propriété **background-color** de l'élément avec l'identifiant **#el**, avec la valeur **red**, et on le fait apparaître progressivement (**slow**).

Une grande force de jQuery réside aussi dans sa gestion facilitée des événements, par exemple :

```
$('#action').click(function() {
```

```
... mon code ...
```

```
});
```

Ici, un clic sur l'élément d'identifiant **#action** va provoquer l'exécution du code.

C'est évidemment ici une présentation très rapide et succincte de jQuery, pour en apprendre plus référez-vous aux nombreux cours qui existent, en particulier sur OpenClassroom.

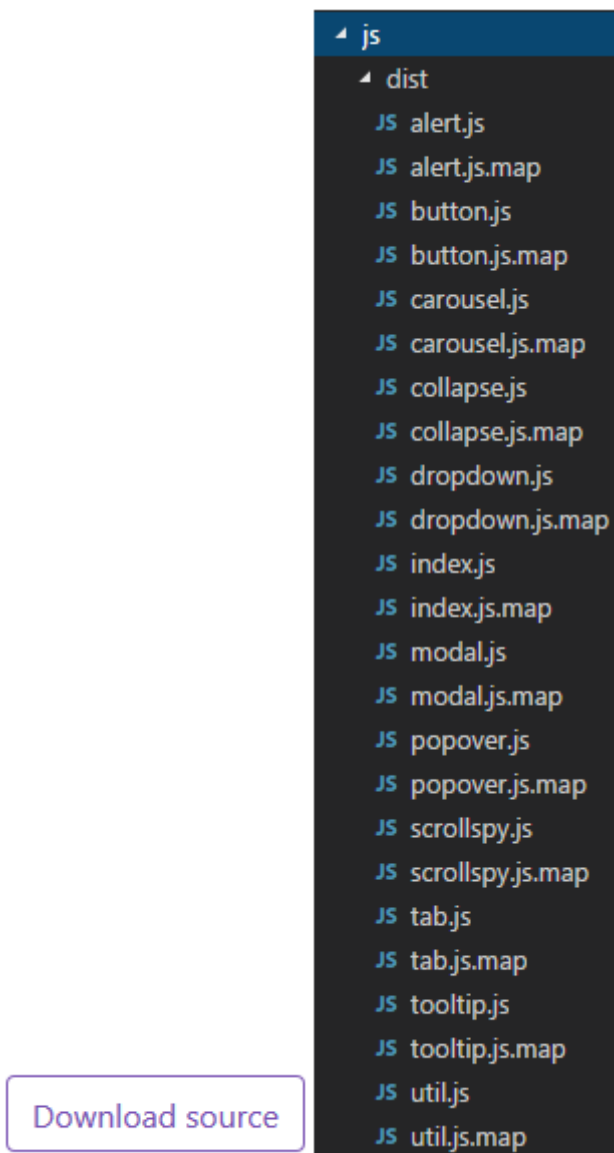
# Les bibliothèques Javascript de Bootstrap

## Référencement

Bootstrap propose un certain nombre de plugins pour améliorer les pages web, ils sont tous fondés sur jQuery. Il y a deux façons de faire appel à ces plugins :

- soit vous référez juste ce qui vous est utile en prenant

la librairie correspondante dans le pack de Bootstrap (que vous récupérez en cliquant sur « Download source » sur la page de téléchargement).



- soit vous référencez la bibliothèque complète.

Pour référencer les plugins dans l'hypothèse de l'utilisation individuelle, par exemple pour

```
<script src="js/jquery.slim.js"></script>
<script src="js/popper.js"></script>
<script src="js/tab.js "></script>
```

On déclare d'abord jQuery, la librairie **popper.js** qui est nécessaire pour ce plugin, et ensuite le plugin.

Dans l'hypothèse de la librairie globale :

```
<script src="js/jquery.slim.min.js"></script>
<script src="js/bootstrap.bundle.min.js "></script>
```

*La version **bundle** contient **popper.js**.*

Vous pouvez utiliser les versions commentées pendant la phase de développement :

```
<script src="js/jquery.js"></script>
<script src="js/bootstrap.bundle.js"></script>
```

## Mise en œuvre

Il y a 2 façons d'utiliser un plugin de Bootstrap :

- à partir des attributs **data**,
- à partir d'objets jQuery.

Prenons un exemple avec l'effet accordéon que nous verrons en détail ultérieurement :

```
<a href="#item" data-toggle="collapse">Changer</a>
<div id="item" class="collapse">
```

Contenu

```
</div>
<script src="assets/js/jquery.slim.min.js"></script>
<script src="assets/js/bootstrap.bundle.min.js"></script>
```

Ne vous inquiétez pas pour la syntaxe globale que nous verrons plus tard. Ce plugin est destiné à faire apparaître ou disparaître un contenu lors d'une action de l'utilisateur, ici un simple clic. Le contenu concerné dans cet exemple est le mot « Contenu ». Le plugin est déclenché par l'attribut **data-toggle= »collapse »**. Dans cette hypothèse d'utilisation, vous n'avez pas à manipuler de Javascript.

On peut obtenir exactement le même résultat en utilisant jQuery :

```
<a>Changer</a>
<div id="item" class="collapse">
  Contenu
```

```

</div>
<script src="assets/js/jquery.slim.min.js"></script>
<script src="assets/js/bootstrap.bundle.min.js"></script>
<script>
  $(function() {
    $('a').click(function() {
      $('#item').collapse('toggle')
    })
  })
</script>

```

Cette fois je n'ai pas utilisé l'attribut mais un objet jQuery avec le sélecteur `a` pour pointer le lien et la méthode **collapse** avec le paramètre **toggle**. Nous verrons plusieurs cas d'utilisation de cette façon de procéder qui est plus laborieuse que la précédente, mais qui s'avère incontournable selon les actions que nous voulons effectuer.

## Les événements

La plupart des plugins exposent des événements. En poursuivant l'exemple précédent, on peut faire apparaître un message lorsque le contenu est visible :

```

<a>Changer</a>
<div id="item" class="collapse">
  Contenu
</div>
<script src="assets/js/jquery.slim.min.js"></script>
<script src="assets/js/bootstrap.bundle.min.js"></script>
<script>
  $(function() {
    $('a').click(function() {
      $('#item').collapse('toggle')
    })
    $('#item').on('shown.bs.collapse', function () {
      alert('On me voit !')
    })
  })
</script>

```

Nous verrons plusieurs exemples d'événements dans les plugins de

Bootstrap.

## En résumé

- Les plugins de Bootstrap fonctionnent avec jQuery, il est donc souhaitable de connaître cette librairie pour utiliser avec efficacité les plugins.
- La plupart des plugins de Bootstrap fonctionnent avec des attributs data pour éviter d'utiliser du code Javascript, mais ça devient vite nécessaire pour certaines actions.