



Bootstrap : le JavaScript (jQuery)



• Bootstrap : le JavaScript (jQuery)

0

Sommaire

- 1/ Introduction
- 2/ Les fenêtres modales
- 3/ Les menus déroulants
- 4/ Scrollspy
- 5/ Tab
- 6/ Tooltip
- 7/ Popover
- 8/ Message d'alerte
- 9/ Boutons
- 10/ Collapse
- 11/ Carousel

Introduction



1

Introduction

• Introduction

Bootstrap intègre des fonctionnalités "prêtes à l'emploi" qui utilisent du javascript-jquery et fait appel à différents plugins qui sont automatiquement insérés si vous utilisez le package complet.

L'ensemble de ces plugins se situe dans le fichier bootstrap.js ou bootstrap.min.js ; vous pouvez sinon les inclure individuellement si vous téléchargez un pack customisé (attention toutefois dans ce cas aux dépendances : certains plugins en utilisent d'autres).

N'oubliez pas la documentation officielle sur :

<https://getbootstrap.com/docs/4.3/getting-started/javascript/> .

Dans toutes les "fonctionnalités" présentées dans ce cours, vous pourrez utiliser 2 méthodes :

- la méthode automatique en ajoutant un data attribut,
- la méthode manuelle en ajoutant vous-même le code jQuery.

Fenêtres modales



2

Les fenêtres modales

- **Les fenêtres modales**

Les fenêtres modales sont des boîtes de dialogues (popups) s'ouvrant lors d'un événement. Vous pourrez y insérer le contenu de votre choix (même des rows et cols !).

Création de la popup

Dans un premier temps, il faut définir un élément comme étant la boîte de dialogue. Plusieurs classes sont possibles :

- modal : permet de créer la popup
- modal-dialog : applique les marges et largeur appropriées à une popup
- modal-title : définit le titre de la boîte
- modal-content : ajoute des styles tels qu'une bordure, arrière-plan etc. Cette classe est à utiliser lorsque vous ajoutez un header, un body ou un footer à votre popup.
- modal-header / modal-body / modal-footer : définissent le header, le body et le footer de la boîte
- modal-sm / modal-lg : définit une petite ou une grande boîte
- fade : ajoute une transition en fondu lors de l'ouverture ou la fermeture de la boîte



2

Les fenêtres modales

Interactions

Après avoir créé votre popup, vous devrez insérer les différentes interactions pour l'ouverture et la fermeture. C'est le rôle des data attributs :

- `data-toggle="modal"` : définit un bouton, un lien ou tout autre élément comme étant le "déclencheur" de la modale tant pour l'ouverture que la fermeture
- `data-target="#maBoite"` : la modale ayant `id="maBoite"` sera commandée par cet élément
- `data-dismiss="modal"` : définit l'élément comme étant un bouton de fermeture

Des options supplémentaires sont disponibles grâce à des attributs booléens; leur valeur seront donc `true` ou `false`.

- `data-backdrop` : Défaut : `true`. Active ou non la fermeture de la popup lors d'un clic en dehors de celle-ci. La valeur `"static"` est possible : elle désactive mais conserve le reste du document en plus foncé.
- `data-keyboard` : Défaut : `true`. Active ou non la fermeture de la popup lors de l'appui sur la touche Echap du clavier.
- `Data-show` : par défaut à `true`. Montre ou non la popup lors de l'appel.



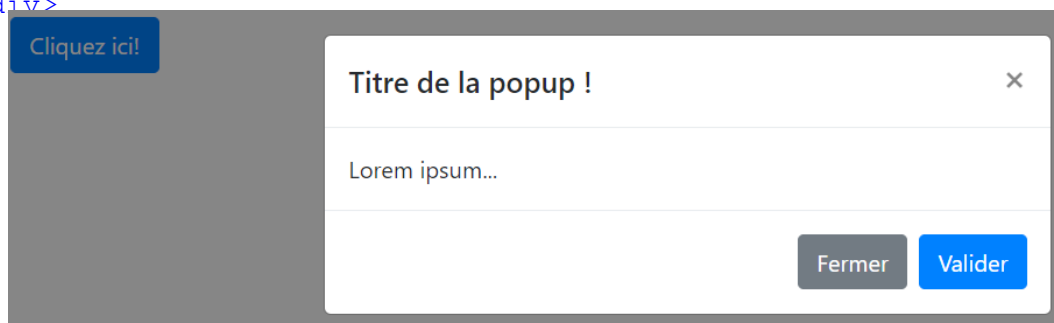
2

Les fenêtres modales

Exemple :

```
<!-- Bouton d'appel -->
<button type="button" class="btn btn-primary" data-toggle="modal" data-target="#maBoite">
  Cliquez ici!
</button>

<!-- Fenêtre modale -->
<div class="modal fade" id="maBoite" tabindex="-1" role="dialog" aria-
labelledby="maBoiteLabel" aria-hidden="true">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="maBoiteLabel">Titre de la popup !</h5>
        <button type="button" class="close" data-dismiss="modal" aria-
label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
      <div class="modal-body">
        Lorem ipsum...
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-
dismiss="modal">Fermer</button>
        <button type="button" class="btn btn-primary">Valider</button>
      </div>
    </div>
  </div>
</div>
```





2

Les fenêtres modales

JavaScript

Vous pouvez contrôler la popup manuellement en javascript, donc sans utiliser data-toggle et data-target. Pour cela, créez un script et exécutez la méthode modal() lors d'un événement.

Reprenons l'exemple ci-dessus en modifiant la première ligne :

```
<button type="button" class="btn btn-primary" id="bouton">Cliquez ici !</button>
```

Puis ajoutez le script :

```
$("#bouton").click(function(){  
    $('#maBoite').modal();  
});
```

Il est possible de passer des paramètres à cette méthode :

- 'show' : Pour ouvrir la popup
- 'hide' : Cache la popup
- 'toggle' : Pour ouvrir ou fermer la popup
- objet : Pour ajouter les options identiques aux data attributs vus précédemment : backdrop, keyboard, show.

Exemple :

```
$("#bouton").click(function(){  
    $('#maBoite').modal({  
        backdrop: false,  
        keyboard: false  
    });  
});
```



2

Les fenêtres modales

Événements

Bootstrap a défini quelques événements personnalisés relatifs aux popups :

- show.bs.modal : juste avant l'ouverture
- shown.bs.modal : juste après l'ouverture
- hide.bs.modal : juste avant la fermeture
- hidden.bs.modal : juste après la fermeture

Exemple :

```
$("#bouton").on('hide.bs.modal', function(){  
    alert('Attention la boîte va se fermer !');  
});
```

Menu déroulant



3

Les menus déroulants

• Les menus déroulants

Ce plugin permet de créer des menus déroulants au clic.

Classes à appliquer

- dropdown : indique au bloc qu'il contient un menu déroulant
- dropdown-menu : à appliquer au bloc du sous-menu
- dropdown-menu-right : pour obtenir un sous-menu aligné à droite
- dropdown-header : pour créer une entête au sous-menu
- dropup : pour faire "dérouler" le sous-menu vers le haut (par défaut vers le bas)
- disabled : désactive un élément du sous-menu
- dropdown-divider : crée une ligne de séparation

Interactions

Un seul data attribut est à placer sur l'élément déclencheur (bouton, lien) pour activer le plugin : `data-toggle="dropdown"`.



9

Navigation

- **Barre de navigation**

Rappel du code pour afficher un menu :

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <div class="collapse navbar-collapse" id="menu">
    <!-- Contenu de la rangée -->
    <ul class="navbar-nav mr-auto">
      <li class="nav-item"><a class="nav-link" href='index.html'>Accueil</a></li>
      <li class="nav-item"><a class="nav-link" href='service.html'>Services</a></li>
      <li class="nav-item"><a class="nav-link" href='contact.html'>Contact</a></li>
    </ul>
  </div>
</nav>
```

Accueil Services Contact



9

Navigation

• *Barre de navigation*

Pour ajouter des sous-menus, il faut :

```
<div class="collapse navbar-collapse" id="menu">
  <ul class="navbar-nav mr-auto">
    <li class="nav-item"><a class="nav-link" href='home.html'>Accueil</a></li>
    <li class="nav-item dropdown">
      <a class="nav-link dropdown-toggle" href='#' id="navbarDropdown"
        role="button" data-toggle="dropdown">
        Services
      </a>
      <div class="dropdown-menu" aria-labelledby="navbarDropdown" >
        <a class="dropdown-item" href="services.html">Intégrateur</a>
        <a class="dropdown-item" href="services.html">Dév Back</a>
        <div class="dropdown-divider"></div>
        <a class="dropdown-item" href="services.html">Autres</a>
      </div>
    </li>
    <li class="nav-item"><a class="nav-link" href='contact.html'>Contact</a></li>
  </ul>
</div>
```

Nom du site Accueil Services ▼ Contact

Intégrateur

Dév Back

Autres



3

Les menus déroulants

JavaScript

Pour activer manuellement un menu déroulant, il suffit d'appliquer la méthode `dropdown()` au bouton ou tout autre élément déclencheur. Attention, celui-ci doit toujours comporter l'attribut `data-toggle="dropdown"` !

Ajoutez un id au bouton de l'exemple précédent et ajoutez le script :

```
$("#monBouton").dropdown();
```

Événements

De la même manière que pour les fenêtres modales, il existe des événements personnalisés relatifs aux menus déroulants :

- `show.bs.dropdown` : juste avant l'ouverture
- `shown.bs.dropdown` : juste après l'ouverture
- `hide.bs.dropdown` : juste avant la fermeture
- `hidden.bs.dropdown` : juste après la fermeture

Exemple :

```
$(".dropdown").on('hide.bs.dropdown', function(){  
    alert('Attention la menu va se fermer !');  
});
```

ScrollSpy



4

Scrollspy

- **Scrollspy**

Ce plugin permet de créer une navigation vers des zones d'une même page (site one page) ; l'utilisation du composant nav est obligatoire et le bloc comprenant les éléments "scrollables" doit être en position relative (CSS).

Attention si vous souhaitez voir un effet de défilement, vous devrez créer un script supplémentaire.

Data attributs à appliquer

- `data-spy="scroll"` : Doit être ajouté à l'élément parent des éléments "scrollables" (la plupart du temps le body)
- `data-target="XXX"` : Doit être ajouté à ce même élément où XXX sera la valeur de l'identifiant ou de la classe (précédé de # ou .), correspondant au parent de l'élément .nav
- `data-offset="XXX"` : Optionnel. Il doit être ajouté à ce même élément et correspond à un décalage vertical.



4

Scrollspy

Exemple :

```
<body data-spy="scroll" data-target=".navbar" data-offset="50">
<nav class="navbar navbar-inverse navbar-fixed-top">
  <div class="container-fluid">
    <div class="navbar-header">
      <a class="navbar-brand" href="#">Mon site</a>
    </div>
    <div class="collapse navbar-collapse" id="myNavbar">
      <ul class="nav navbar-nav">
        <li><a href="#section1">Section 1</a></li>
        <li><a href="#section2">Section 2</a></li>
        <li><a href="#section3">Section 3</a></li>
        <li class="dropdown"><a class="dropdown-toggle" data-toggle="dropdown" href="#">
          Section 4 <span class="caret"></span></a>
          <ul class="dropdown-menu">
            <li><a href="#section41">Section 4-1</a></li>
            <li><a href="#section42">Section 4-2</a></li>
          </ul>
        </li>
      </ul>
    </div>
  </div>
</nav>
```

...



4

Scrollspy

...

```
<div id="section1" class="container-fluid">
  <h2>Section 1</h2>
  <p>Lorem ipsum...</p>
</div>
<div id="section2" class="container-fluid">
  <h2>Section 2</h2>
  <p>Lorem ipsum...</p>
</div>
<div id="section3" class="container-fluid">
  <h2>Section 3</h2>
  <p>Lorem ipsum...</p>
</div>
<div id="section41" class="container-fluid">
  <h2>Section 4 Submenu 1</h2>
  <p>Lorem ipsum...</p>
</div>
<div id="section42" class="container-fluid">
  <h2>Section 4 Submenu 2</h2>
  <p>Lorem ipsum...</p>
</div>
```



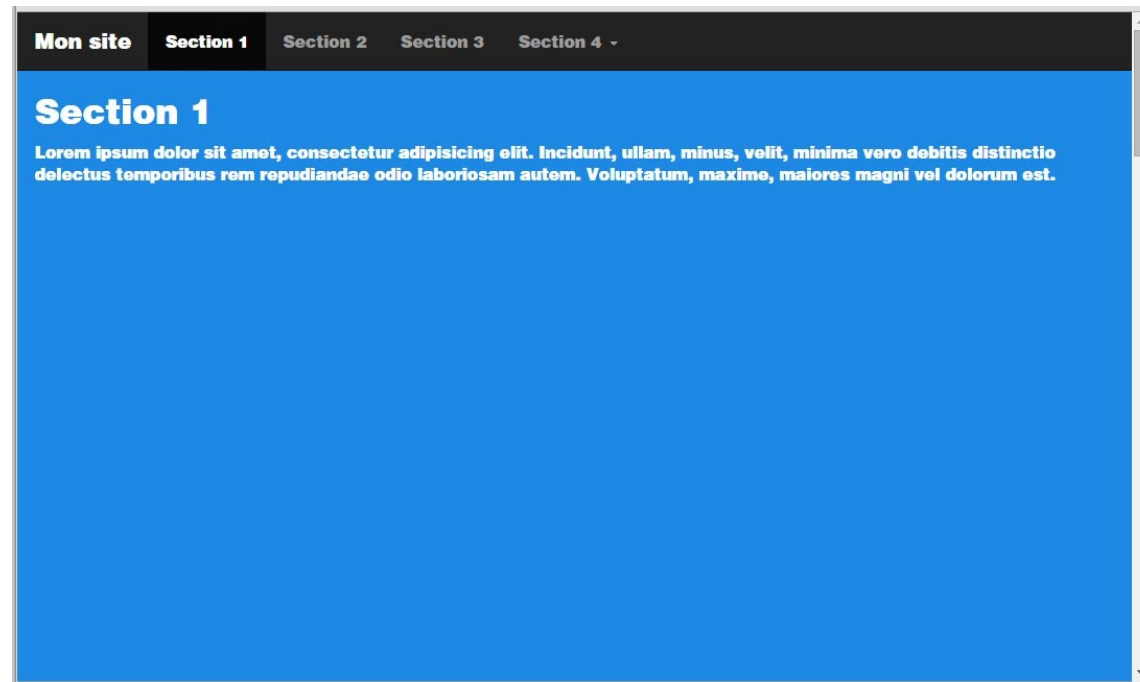
• Bootstrap : le JavaScript (jQuery)

4

Scrollspy

Avec ajout du CSS :

```
html,body{height:100%;}  
body {position: relative;}  
#section1 {padding-top:50px;height:100%;color: #fff; background-color: #1E88E5;}  
#section2 {padding-top:50px;height:100%;color: #fff; background-color: #673ab7;}  
#section3 {padding-top:50px;height:100%;color: #fff; background-color: #ff9800;}  
#section41 {padding-top:50px;height:100%;color: #fff; background-color: #00bcd4;}  
#section42 {padding-top:50px;height:100%;color: #fff; background-color: #009688;}
```





4

Scrollspy

JavaScript

Pour activer directement cette option via Javascript, il suffit d'appliquer la méthode `scrollspy()` au conteneur (ici le `body`). Un objet lui sera passé en paramètre afin de spécifier le `target` et éventuellement un `offset`.

Pour notre exemple, le JS sera donc le suivant :

```
$('body').scrollspy({  
    target: ".navbar",  
    offset: 50  
});
```

Événement

`activate.bs.scrollspy` :

Il s'agit de l'unique événement possible : occurrence lorsqu'un nouvel élément devient activé par scrollspy.



4

Scrollspy

Animation au scroll

Vous trouverez ci-dessous un exemple de code permettant d'ajouter une animation de scrolling.

```
$(document).ready(function(){
    //Appel
    $('body').scrollspy({target: ".navbar", offset: 50});
    // Ajout de l'effet de scrolling
    $("#myNavbar a").on('click', function(e){
        // désactive les liens a
        e.preventDefault();
        // Enregistrement du hash (le # des href)
        var hash = this.hash;
        // Utilisation de la méthode jQuery animate()
        $('html, body').animate({
            scrollTop: $(hash).offset().top
        }, 800, function(){
            // Ajout du hash (#) dans l'URL après le scroll
            window.location.hash = hash;
        });
    });
});
```

Tabs



5 Tab

• Tab

Ce plugin permet d'ajouter des effets de transition lors d'une navigation à onglets.

L'effet de fade entre chaque contenu sera activé par l'ajout de la classe fade sur chaque panneau (classe tab-pane). Le premier panneau se verra aussi ajouté la classe in.

L'attribut data-toggle="tab" doit être spécifié sur chaque élément du menu.

Exemple :

```
<div class="col-12">
  <h2>Navigation Tabs</h2>
  <ul class="nav nav-tabs">
    <li class="active"><a data-toggle="tab" href="#un">Onglet 1</a></li>
    <li><a data-toggle="tab" href="#deux">Onglet 2</a></li>
    <li><a data-toggle="tab" href="#trois">Onglet 3</a></li>
  </ul>
  <div class="tab-content text-center">
    <div id="un" class="tab-pane fade in active">Contenu 1</div>
    <div id="deux" class="tab-pane fade">Contenu 2</div>
    <div id="trois" class="tab-pane fade">Contenu 3</div>
  </div>
</div>
```

Navigation Tabs

Onglet 1

Onglet 2

Onglet 3

Contenu 1



5 Tab

JavaScript

Pour activer manuellement ce plugin sans utiliser le data-attribut, vous devez appliquer la méthode `tab('show')` à l'élément cliqué. Exemple :

```
$(document).ready(function(){  
    $(".nav-tabs a").click(function(){  
        $(this).tab('show');  
    });  
});
```

Evénements

Bootstrap a défini quelques événements personnalisés relatifs aux tabs :

- `show.bs.tab` : juste avant l'ouverture de la cible
- `shown.bs.tab` : juste après l'ouverture de la cible
- `hide.bs.tab` : juste avant la fermeture de la cible
- `hidden.bs.tab` : juste après la fermeture de la cible

Exemple :

```
$('.nav-tabs a').on('show.bs.tab', function(){  
    alert("Un nouvel onglet va s'ouvrir.");  
});
```

Tooltip



6

Tooltip

- **Tooltip**

Un Tooltip est une petite bulle qui apparaît au survol d'un élément. Le contenu de celui-ci sera le texte spécifié dans l'attribut title. Il viendra donc en remplacement de la bulle par défaut des navigateurs. Pour les éléments HTML ne supportant pas l'attribut title, il sera possible dans les options d'en créer un.

Le plugin va créer en fait un élément qui sera le suivant :

```
<div class="tooltip bs-tooltip-top" role="tooltip">  
  <div class="tooltip-arrow"></div>  
  <div class="tooltip-inner">  
    Valeur de l'attribut title.  
  </div>  
</div>
```

Vous pourrez donc le customiser selon votre souhait dans votre fichier CSS personnalisé en faisant référence aux classes ci-dessus.



6

Tooltip

Construction

Afin de définir l'élément qui aura le tooltip, vous devez simplement lui spécifier l'attribut `data-toggle="tooltip"`.

Ensuite, ce plugin nécessite obligatoirement l'activation via JavaScript en appliquant la méthode `tooltip()` à l'élément.

Exemple :

HTML :

```
<a href="#" data-toggle="tooltip" title="Informations au survol">Survolez-moi !</a>
```

CSS :

```
.tooltip-inner{background:red;}
```

JS :

```
$(document).ready(function(){  
    $('[data-toggle="tooltip"]').tooltip();  
});
```





6

Tooltip

Options

Les options sont applicables :

- via un l'attribut data-XXX="YYY" où XXX représente le nom d'une option ci-dessous et YYY la valeur de celle-ci.
- via javascript en passant les options dans un objet en paramètre de la méthode tooltip() : tooltip({option1:"valeur",option2:"valeur",etc});

Les options principales disponibles sont (liste complète sur la documentation) :

- animation : permet de créer un effet de fade lors de l'affichage de la bulle. La valeur est un booléen (par défaut : true).
- container : permet d'ajouter le tooltip à un élément HTML spécifique. Exemple : container : 'body' (valeur par défaut : false).
- delay : permet de spécifier un temps avant que le tooltip apparaisse / disparaisse. La valeur correspond à un nombre en millisecondes. Pour donner un delay différent pour l'apparition et la disparition, la valeur sera un objet de la forme {"show:500",hide:"100"}
- html : autorise l'ajout de contenu HTML dans le tooltip. La valeur est un booléen (par défaut : false).



6

Tooltip

- placement : permet de positionner le tooltip par rapport à l'élément. La valeur sera de type string : top / bottom / right / left / auto. La valeur auto permet de laisser le navigateur gérer lui-même. Si par exemple le placement est spécifié à "auto top", la bulle sera placée au-dessus si elle a de la place mais se positionnera au-dessous dans le cas contraire. (Défaut : "top").
- selector : permet d'ajouter le tooltip à un sélecteur (élément) spécifique. La valeur sera un sélecteur CSS, comme par exemple ".btn" (par défaut : false). Cela sera utile lors de la délégation d'événement.
Exemple : `$('#body').tooltip({selector:".btn"});`
- template : permet de spécifier une chaîne HTML spécifique pour la création du tooltip. Par défaut, la chaîne utilisée est celle présentée au début de ce chapitre. Vous devrez toutefois utiliser les mêmes classes.
- title : permet de spécifier le texte affiché à l'intérieur du tooltip. Il sera aussi possible de passer une fonction JS à cette option. Le `this` fera référence à l'élément auquel le tooltip est attaché.
- trigger : permet de spécifier lors de quel événement le tooltip sera déclenché. La valeur pourra être hover (par défaut), click ou focus.



6

Tooltip

Exemple en utilisant les data-attributs :

```
<button class="btn" data-title="contenu de la tooltip" data-placement="bottom" data-trigger="click" >
    Cliquez-moi
</button>
```

Exemple en utilisant javascript :

```
$(document).ready(function(){
    $('button').tooltip({
        title : "contenu de la tooltip",
        placement : "bottom",
        trigger:"click",
    });
});
```



6

Tooltip

Méthodes

Il sera possible d'attacher le tooltip lors d'un événement sur un élément quelconque.

Ces méthodes seront `tooltip("show")` , `tooltip("hide")`, `tooltip("toggle")` ou `tooltip("destroy")` permettant respectivement de faire apparaître, cacher, montrer ou cacher, détruire le tooltip. Exemple :

Code HTML :

```
<a href="#" data-toggle="tooltip" title="Texte de la tooltip.">Exemple</a>  
<button type="button" class="btn btn-primary">Montrer / cacher la bulle !</button>
```

Code JS :

```
$(document).ready(function(){  
    $('button').click(function(){  
        $('[data-toggle='tooltip']").tooltip('toggle');  
    });  
});
```

La bulle s'affiche alors lors du clic sur le bouton.



• Bootstrap : le JavaScript (jQuery)

6

Tooltip

Événements

Comme tous les plugins Bootstrap, des événements personnalisés sont disponibles :

- `show.bs.tooltip` : juste avant l'ouverture de la bulle
- `shown.bs.tooltip` : juste après l'ouverture de la bulle
- `hide.bs.tooltip` : juste avant la fermeture de la bulle
- `hidden.bs.tooltip` : juste après la fermeture de la bulle
- `inserted.bs.tooltip` : juste après l'événement `show.bs.tooltip`

Popovers



7

Popover

• Popover

Les popovers sont très similaires aux tooltips. Une popover pourra simplement avoir plus de contenu.

(Data-)attributs

Ces attributs sont à spécifier sur l'élément qui sera cliqué (bouton, lien...).

- `data-toggle="popover"` : Permet d'activer le plugin.
- `title="XXX"` : Permet de spécifier le titre, le header de la popover.
- `data-content="XXX"` : Permet de spécifier le contenu de la popover.

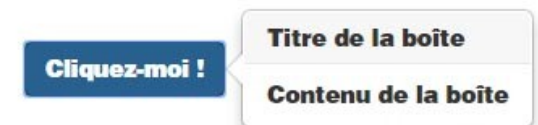
Ensuite le plugin sera activé grâce à la méthode `popover()`. Exemple :

HTML :

```
<button class="btn btn-primary" data-toggle="popover" title="Titre de la boîte" data-content="Contenu de la boîte"> Cliquez-moi !</button>
```

JS :

```
$(document).ready(function(){
    $('[data-toggle="popover"]').popover();
});
```



Options / Javascript / Méthodes / Événements

Ils fonctionnent de la même façon que pour le plugin tooltip : via les data-attributs ou en paramètre de la méthode `popover()`. Ils sont également les mêmes avec en plus l'option `content` qui permet de spécifier le contenu de la boîte.

Alertes



8

Message d'alerte

- **Message d'alerte**

Les messages d'alertes sont utilisés pour spécifier des messages nécessitant l'attention des internautes comme des avertissements, des erreurs ou des confirmations. Bootstrap permet de créer facilement ces messages avec la possibilité d'y ajouter un bouton.

Ces messages seront à afficher lors d'événements à définir.

Création de la boîte

Afin de définir un bloc comme étant une boîte d'alerte, vous devez lui ajouter la classe "alert" puis une autre classe afin de lui spécifier une couleur : "alert-info" (ou success / warning / danger).

Animation

Il est possible d'obtenir un effet de fondu lors de l'ouverture ou la fermeture de l'alerte. Pour cela, vous devez lui ajouter les classes "fade" et "in".

Fermeture

Pour donner la possibilité de fermer le message, ajoutez l'attribut data-dismiss="alert" au lien ou bouton de fermeture. Pour rappel, si vous lui ajoutez la classe "close", cela aura pour effet de le positionner en haut à droite de la boîte.



8

Message d'alerte

Exemple :

```
<div class="alert alert-info fade in" id="message">  
  <a href="#" class="close" data-dismiss="alert">&times;</a>  
  <strong>Information : </strong> Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
</div>
```

Information : Lorem ipsum dolor sit amet, consectetur adipiscing elit.



JavaScript

Si vous souhaitez appliquer la fermeture manuellement (sans l'attribut data-dismiss), vous pouvez utiliser la méthode jQuery alert("close"). Exemple :

```
$(document).ready(function(){  
  $(".close").click(function(){  
    $("#message").alert("close");  
  });  
});
```

A noter que si vous ne passez pas "close" en paramètre, vous pourrez fermer toutes les alertes affichées.

Événements

2 événements personnalisés peuvent être appelés grâce à ce plug in :

- close.bs.alert : Juste avant la fermeture du message.
- closed.bs.alert : Juste après la fermeture du message.

Boutons



9

Boutons

• Boutons

Ce plug in permet d'étendre les possibilités sur les boutons.

Bouton de statut

Cela permet de changer le statut visuel du bouton si celui-ci a déjà été cliqué. Pour éviter un bug de Mozilla, on utilise un "hack" en ajoutant l'attribut autocomplete="off".

Exemple :

HTML :

```
<button type="button" id="monBouton" data-toggle="button" class="btn btn-primary" autocomplete="off"> Cliquez-moi ! </button>
```

JS :

```
$(document).ready(function(){  
  $('#monBouton').on('click', function () {  
    $(this).button('toggle');  
  });  
});
```



Single toggle

Note : un "toggle button" peut être simplement inséré via l'attribut data-toggle="button"



9

Boutons

Checkbox / Radio

En CSS habituel, il n'est pas possible de styliser les boutons radio ou checkboxes. Bootstrap nous le permet !

Pour cela vous devez ajouter l'attribut `data-toggle="buttons"` au bloc parent des input type checkbox ou radio, qui possède la classe `.btn-group`

Exemple :

```
<div class="btn-group" data-toggle="buttons">
  <label class="btn btn-primary active">
<input type="checkbox" autocomplete="off" checked> Case 1
  </label>
  <label class="btn btn-primary">
    <input type="checkbox" autocomplete="off"> Case 2
  </label>
  <label class="btn btn-primary">
    <input type="checkbox" autocomplete="off"> Case 3
  </label>
</div>
```

Collapse



10

Collapse

• Collapse

Ce plugin permet d'afficher / cacher du contenu lors d'un clic sur un élément (afin de créer par exemple un "menu accordéon").

Attributs / classes à appliquer

- `data-toggle="collapse"` : Doit être appliqué à l'élément qui déclenchera l'apparition du contenu supplémentaire.
- `data-target="#XXX"` : Doit être appliqué à ce même élément. XXX correspondra à l'id de l'élément ciblé. Cet attribut peut-être remplacé par l'attribut `href` si vous utilisez un lien.
- `class="collapse"` : Permet de masquer le contenu ciblé au chargement de la page.
- `class="collapse show"` : Permet d'afficher le contenu ciblé au chargement de la page.
- `class="collapsing"` : Cette classe est ajoutée automatiquement au contenu ciblé uniquement lors de l'effet d'apparition ou de disparition. Vous pourrez donc définir cette classe en CSS.



10

Collapse

Exemple simple collapse :

CSS :

```
.collapsing{  
    background-color:#000;  
    color:#fff;  
}
```

HTML :

```
<button class="btn btn-primary" type="button" data-toggle="collapse" data-target="#contenu" >  
    Clic me !  
</button>  
<div id="contenu" class="collapse">  
    <p>Lorem ipsum dolor sit amet...</p>  
</div>
```

Clic me !

Lorem ipsum dolor sit amet...



10

Collapse

Menu accordéon

Un menu accordéon est en fait une extension du plugin collapse appliqué aux cards.

L'attribut `data-parent="#xxx"` doit être ajouté à chaque contenu afin de refermer tous les autres contenus de l'accordéon lors du clic. "xxx" représente l'id du conteneur parent.

Un exemple est toujours plus parlant :

Contenu 1

Lorem ipsum dolor sit amet...

Contenu 2

Contenu 3



• Bootstrap : le JavaScript (jQuery)

10

Collapse

```
<div class="accordion" id="accordeon">
  <div class="card">
    <div class="card-header">
      <h4 class="mb-0">
        <button class="btn btn-link" type="button" data-toggle="collapse" data-target="#contenu1">Contenu 1</button>
      </h4>
    </div>
    <div id="contenu1" class="collapse show" data-parent="accordeon">
      <div class="card-body">Lorem ipsum dolor sit amet...</div>
    </div>
  </div>
  <div class="card">
    <div class="card-header">
      <h4 class="mb-0">
        <button class="btn btn-link" type="button" data-toggle="collapse" data-target="#contenu2">Contenu 2</button>
      </h4>
    </div>
    <div id="contenu2" class="collapse show" data-parent="accordeon">
      <div class="card-body">Lorem ipsum dolor sit amet...</div>
    </div>
  </div>
  <div class="card">
    <div class="card-header">
      <h4 class="mb-0">
        <button class="btn btn-link" type="button" data-toggle="collapse" data-target="#contenu3">Contenu 3</button>
      </h4>
    </div>
    <div id="contenu3" class="collapse show" data-parent="accordeon">
      <div class="card-body">Lorem ipsum dolor sit amet...</div>
    </div>
  </div>
</div>
```



10

Collapse

JavaScript

Si vous souhaitez activer manuellement ce plugin, vous devez alors utiliser la méthode `collapse()`.

Options :

- `collapse('show')` : Pour montrer l'élément.
- `collapse('hide')` : Pour cacher l'élément.
- `collapse('toggle')` : Pour montrer/cacher l'élément.

Exemple :

HTML :

```
<button type="button" class="btn btn-primary" id="bouton">clic me !</button>
<div class="collapse">
  Lorem ipsum dolor sit amet...
</div>
```

JS :

```
$(document).ready(function(){
  $("#bouton").click(function(){
    $(".collapse").collapse('toggle');
  });
});
```



10

Collapse

Événements

Les événements personnalisés disponibles pour ce plugin sont les suivants :

- show.bs.collapse : Juste avant l'ouverture du contenu ciblé.
- shown.bs.collapse : Juste après l'ouverture du contenu ciblé.
- hide.bs.collapse : Juste avant la fermeture du contenu ciblé.
- hidden.bs.collapse : Juste après la fermeture du contenu ciblé.

Exemple :

HTML :

```
<button type="button" class="btn btn-primary" id="bouton">
  <i class="fas fa-arrow-down"></i> Ouvrir
</button>
<div class="collapse" id="contenu">
  Lorem ipsum dolor sit amet...
</div>
```

JS :

```
$("#bouton").click(function(){
  $(".collapse").collapse('toggle');
});
$("#contenu").on("hide.bs.collapse", function(){
  $("#bouton").html('<i class="fas fa-arrow-down"></i> Ouvrir');
});
$("#contenu").on("show.bs.collapse", function(){
  $("#bouton").html('<i class="fas fa-arrow-up"></i> Fermer');
});
```


Carousel



11

Carousel

• Carousel

Ce plugin nous permet de construire un slideshow (un diaporama).

Classes à appliquer :

- carousel : Doit être appliquée au conteneur principal du carousel. Si plusieurs slideshows sont présents sur une même page, vous ne devez pas oublier de leur ajouter un id spécifique pour le bon fonctionnement.
- slide : Doit être appliquée au conteneur principal du carousel. Cette classe permet d'activer l'effet de "glissé" lors du passage à une autre image.
- carousel-indicators : Doit être appliquée à une liste enfant du carousel. Cela permet d'ajouter un nombre de puces sous chaque slide indiquant le nombre d'images total. La puce correspondant à l'image en cours sera pleine et les autres vides. Attention des data-attributs doivent être ajoutés à chaque li (voir plus bas).
- carousel-inner : Doit être appliquée au conteneur principal des slides (en dehors des "indicators" et des flèches).
- carousel-item : Doit être appliquée au conteneur de chacun des slides. Il contiendra une balise img et un éventuel bloc de texte.



• Bootstrap : le JavaScript (jQuery)

11

Carousel

- `carousel-control-prev` / `carousel-control-next`: Doit être appliquée au lien "précédent" / "suivant". L'attribut `href` de ces liens doit avoir comme valeur `"#xxx"` où `"xxx"` correspond à l'id du carousel.
- `carousel-caption`: Doit être appliquée au bloc de texte de l'image qui s'affichera par-dessus celle-ci.

Data-attributs :

- `data-ride="carousel"`: Doit être appliqué au conteneur principal du carousel. Il permet d'activer le plugin.
- `data-slide="prev"` / `data-slide="next"`: Doivent être appliqués aux liens "précédent" / "suivant".
- `data-slide-to="N"`: Doit être appliqué à chaque "puce".
Il spécifie quel slide atteindre lors du clic sur celle-ci (N représente l'index du slide, le premier étant 0).





11

Carousel

Exemple :

HTML :

```
<div class="container">
  <div id="monSlide" class="carousel slide" data-ride="carousel">
    <ol class="carousel-indicators">
      <li data-target="#monSlide" data-slide-to="0" class="active"></li>
      <li data-target="#monSlide" data-slide-to="1"></li>
      <li data-target="#monSlide" data-slide-to="2"></li>
    </ol>
    <div class="carousel-inner">
      <div class="carousel-item active">
        
        <div class="carousel-caption">
          <h3>Titre image 1</h3>
          <p>Texte pour l'image.</p>
        </div>
      </div>
      <div class="carousel-item">
        
        <div class="carousel-caption">
          <h3>Titre image 2</h3>
          <p>Texte pour l'image.</p>
        </div>
      </div>
    </div>
  </div>
</div>
```



11

Carousel

```
...  
<div class="carousel-item">  
    
  <div class="carousel-caption">  
    <h3>Titre image 3</h3>  
    <p>Texte pour l'image.</p>  
  </div>  
</div>  
</div>  
</div>  
<a class="carousel-control-prev" href="#monSlide" data-slide="prev">  
  <span class="carousel-control-prev-icon"></span>  
  <span class="sr-only">Previous</span>  
</a>  
<a class="carousel-control-next" href="#monSlide" data-slide="next">  
  <span class="carousel-control-next-icon"></span>  
  <span class="sr-only">Next</span>  
</a>  
</div>  
</div>
```



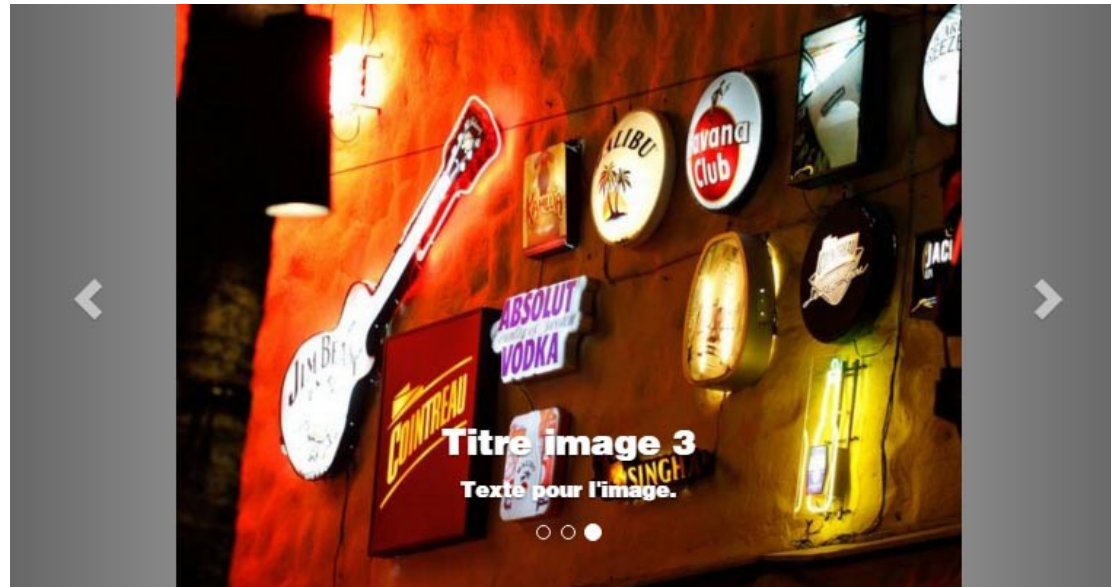
• Bootstrap : le JavaScript (jQuery)

11

Carousel

CSS :

```
.container{
  width:50%;
  margin:50px auto;
}
.carousel{
  background:#888;
  margin-top: 20px;
}
.carousel-inner > .carousel-item > img,
.carousel-inner > .carousel-item > a > img {
  width: 70%;
  margin: auto;
}
```





11

Carousel

JavaScript

- Pour activer le plugin manuellement (sans les data-attributs), vous devez utiliser la méthode `carousel()`.
- Différents paramètres sont à spécifier : (toutes ces méthodes sont à appliquer au conteneur général)
- `carousel()` : Déclenche le slider au chargement de la page.
- `carousel("cycle")` : Déclenche le slider de manière cyclique. A appliquer lors d'un événement.
- `carousel("pause")` : Arrête le cycle du slider. A appliquer lors d'un événement.
- `carousel(n)` : Affiche un item spécifique où `n` représente l'index de celui-ci. A appliquer par exemple lors du clic sur une "puce".
- `carousel("prev")` / `carousel("next")` : Affiche l'item précédent / suivant. A appliquer par exemple lors du clic sur les flèches.

Exemple :

En retirant tous les data-attributs et ajoutant une classe à chaque li dans le HTML précédent, on obtient :



11

Carousel

```
$(document).ready(function(){  
    $("#monSlide").carousel();  
  
    $(".puce1").click(function(){  
        $("#monSlide").carousel(0);  
    });  
    $(".puce2").click(function(){  
        $("#monSlide").carousel(1);  
    });  
    $(".puce3").click(function(){  
        $("#monSlide").carousel(2);  
    });  
  
    $(".left").click(function(){  
        $("#monSlide").carousel("prev");  
    });  
    $(".right").click(function(){  
        $("#monSlide").carousel("next");  
    });  
});
```




11

Carousel

Options

Quelques options peuvent être ajoutées au carousel en passant un objet à la méthode `carousel({...})` ou en utilisant les data-attributs qui seront de la forme `data-option="valeur"` où "option" sera remplacé par le nom de l'une décrite ci-dessous :

- `interval` : Permet de spécifier un délai (en ms) entre l'affichage de chaque slide. Par défaut la valeur est de 5000.
- `pause` : Permet de stopper le slide lors de son survol. La valeur par défaut est "hover". Pour désactiver cette option, vous devez la passer à false.
- `wrap` : Indique si le slider repart au premier slide après la dernière. Les valeurs sont true (par défaut) ou false.
- `keyboard` : Indique si le slider peut être commandé par le clavier. Les valeurs sont true (par défaut) ou false.

Evénements

Les événements personnalisés disponibles sont :

- `slide.bs.carousel` : Juste avant l'apparition d'un slide.
- `slid.bs.carousel` : Juste après l'apparition d'un slide.



• Bootstrap : le JavaScript (jQuery)

Conclusion

- **Conclusion**

Bootstrap est un framework très complet qui nécessite donc de la pratique afin de bien le maîtriser mais après cela, vous pourrez alors mettre en forme très rapidement vos applications web.

N'oubliez pas que la documentation est très bien expliquée. N'hésitez donc pas à vous y référer.



Fin du support Bootstrap : le
JavaScript (jQuery)
