

# Bootstrap 4 : les boutons

On a déjà vu qu'on peut avec quelques classes régler facilement l'aspect des boutons et leur disposition. Dans ce chapitre nous allons voir comment les rendre visuellement dépendants d'un état.

## Boutons bascule, « checkbox » et « radio »

### Bouton bascule

On a parfois besoin de boutons à 2 états stabilisés : repos et appuyé. Le plugin permet de réaliser cela facilement. Il suffit de créer le bouton en prévoyant **data-toggle= »button »** :

```
<a class="btn btn-success" data-toggle="button">Simple Bascule</a>
```

[Tester en ligne](#)



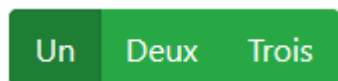
### « checkbox »

On peut grouper des boutons avec les classes **btn-group** et **btn-group-toggle** et les faire fonctionner comme des checkbox avec **data-toggle= »buttons »**. Les boutons restent indépendants dans leur fonctionnement mais ont l'avantage d'être groupés. Pour réaliser cela, il faut utiliser des contrôles **input** de type :

```
<div class="btn-group btn-group-toggle" data-toggle="buttons">
  <label class="btn btn-success active">
    <input type="checkbox" checked>Un
  </label>
  <label class="btn btn-success">
    <input type="checkbox">Deux
  </label>
</div>
```

```
    <input type="checkbox">Trois
  </label>
</div>
```

[Tester en ligne](#)



Notez que si vous voulez qu'un bouton soit actif au chargement il faut prévoir :

- la classe **active** dans la balise **<label>**,
- **checked** dans la balise **<input>**.

Et vous pouvez évidemment en activer plusieurs puisque ce sont de bouton « checkbox ».

## Boutons effet « radio »

On peut grouper des boutons avec les classes **btn-group** et **btn-group-toggle** et les faire fonctionner comme des boutons radio avec **data-toggle= »buttons »**. Les boutons sont maintenant alors liés dans leur fonctionnement, un seul peut être activé. Pour réaliser cela, il faut utiliser des contrôles **input** de type **radio** :

```
<div class="btn-group btn-group-toggle" data-toggle="buttons">
  <label class="btn btn-success active">
    <input type="radio" checked>Un
  </label>
  <label class="btn btn-success">
    <input type="radio">Deux
  </label>
  <label class="btn btn-success">
    <input type="radio">Trois
  </label>
</div>
```

[Tester en ligne](#)

C'est le même principe que vu ci-dessus pour l'activation d'un bouton au chargement (et évidemment un seul dans ce cas), mais cette fois ça s'impose parce que par principe dans un groupement « radio » il y a toujours un élément sélectionné.

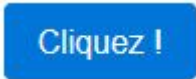
## Activation avec Javascript

Il n'y a pas trop d'intérêt d'activer les boutons avec Javascript. Par contre nous allons voir un cas assez courant d'utilisation dynamique d'un bouton qu'on peut facilement traiter avec jQuery.

Parfois on clique sur un bouton pour réaliser un processus qui demande un certain temps, comme par exemple un envoi de fichier.

D'abord on crée un bouton standard :

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



Ensuite on gère le clic :

```
$(function(){
  $('button').on('click', function() {
    var button = $(this)
    if(!button.hasClass('disabled')) {
      button.addClass('disabled').html('Chargement <span class="fa fa-spinner fa-pulse"></span>');
      setTimeout(function () {
        button.removeClass('disabled').text('Cliquez !')
      }, 4000)
    }
  })
})
```

Donc lorsqu'on clique sur le bouton on le rend inactif, on ajoute une icône animée et on change le texte :

[Tester en ligne](#)

Chargement 

Et au bout du délai créé ici artificiellement on remet le bouton dans son état initial :

Cliquez !

Maintenant voyons un cas d'utilisation un peu plus réaliste que notre minuterie. Nous avons une image lourde à charger, et nous voulons que le bouton change d'état le temps de chargement de l'image. Voilà le bouton et la balise prête à recevoir l'image :

```
<button type="button" class="btn btn-primary">Chargez l'image !</button>
<img id="mon_image">
```


[Tester en ligne](#)

Chargez l'image !

Et voici le code Javascript pour gérer l'effet :

```
$(function(){
  $('button').on('click', function() {
    var button = $(this)
    if(!button.hasClass('disabled')) {
      button.addClass('disabled').html('Chargement en cours <span
class="fa fa-spinner fa-pulse"></span>')
      var image = new Image()
      image.onload = function() {
        button.removeClass('disabled').text("Chargez l'image !")
        $("#mon_image").attr({ src:"images/legumes.jpg" })
      }
      image.src = "images/legumes.jpg"
    }
  })
})
```

Pendant le temps de chargement de l'image on a cet aspect :

Chargement en cours 

Et quand l'image a fini de charger on retrouve le bouton normal et évidemment l'image :

Chargez l'image !



## En résumé

- On peut créer des boutons « bascule », ou avec effet « radio » ou « checkbox ».
- On peut gérer l'aspect d'un bouton avec jQuery par exemple pour attendre la fin d'un processus.