

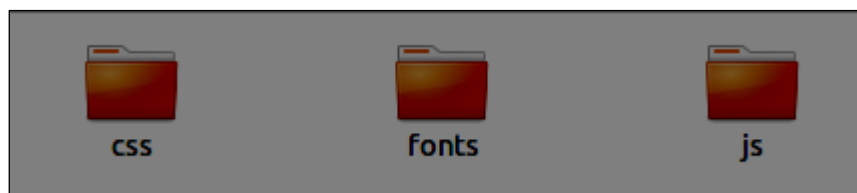


## Documentation Bootstrap

### I – Téléchargement & Installation

Pour télécharger Bootstrap, il suffit de vous rendre sur le site officiel de celui-ci : <http://getbootstrap.com/getting-started/>

Une fois cela fait vous obtiendrez plusieurs dossier et fichiers comme ci-dessous normalement :



En ce qui concerne l'intégration de ces fichiers dans nos pages web il suffit de procéder de la manière suivante :

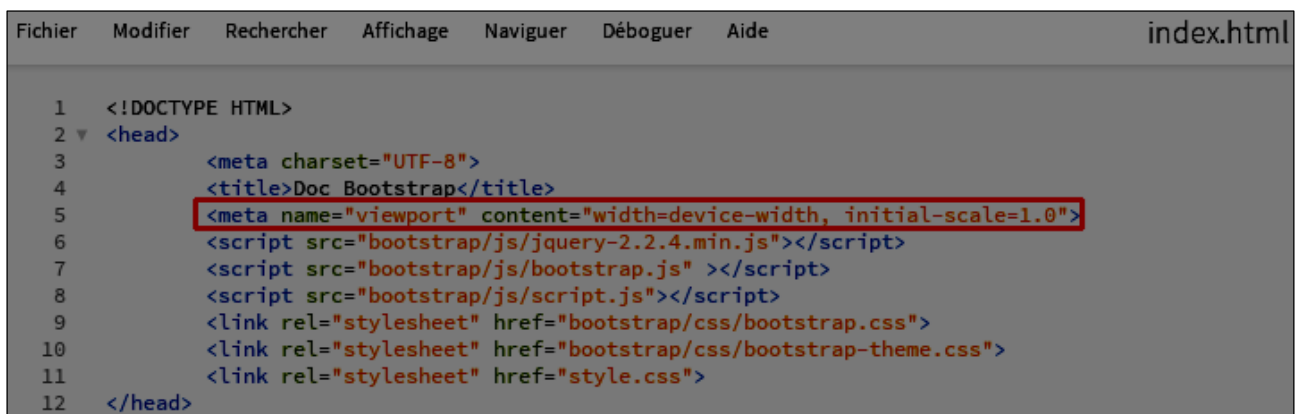
```
Fichier  Modifier  Rechercher  Affichage  Naviguer  Débuguer  Aide  index.html

1  <!DOCTYPE HTML>|
2  <head>
3      <meta charset="UTF-8">
4      <title>Doc Bootstrap</title>
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <script src="bootstrap/js/jquery-2.2.4.min.js"></script>
7      <script src="bootstrap/js/bootstrap.js"></script>
8      <script src="bootstrap/js/script.js"></script>
9      <link rel="stylesheet" href="bootstrap/css/bootstrap.css">
10     <link rel="stylesheet" href="bootstrap/css/bootstrap-theme.css">
11     <link rel="stylesheet" href="style.css">
12 </head>
```

RAPPEL : Les balises script permettent d'exécuter des scripts et par défaut le langage utilisé est le « JavaScript »

Il est également important de ne pas oublier la balise <Meta name= « viewport », celle-ci vous permettra de reconnaître la taille de l'appareil qui navigue sur votre page et ainsi adapter le design (width-device=width) et

d'initialiser le zoom à 1 (initial-scale=1,0),



```
Fichier  Modifier  Rechercher  Affichage  Naviguer  Débuguer  Aide  index.html

1  <!DOCTYPE HTML>
2  <head>
3      <meta charset="UTF-8">
4      <title>Doc Bootstrap</title>
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <script src="bootstrap/js/jquery-2.2.4.min.js"></script>
7      <script src="bootstrap/js/bootstrap.js" ></script>
8      <script src="bootstrap/js/script.js"></script>
9      <link rel="stylesheet" href="bootstrap/css/bootstrap.css">
10     <link rel="stylesheet" href="bootstrap/css/bootstrap-theme.css">
11     <link rel="stylesheet" href="style.css">
12 </head>
```

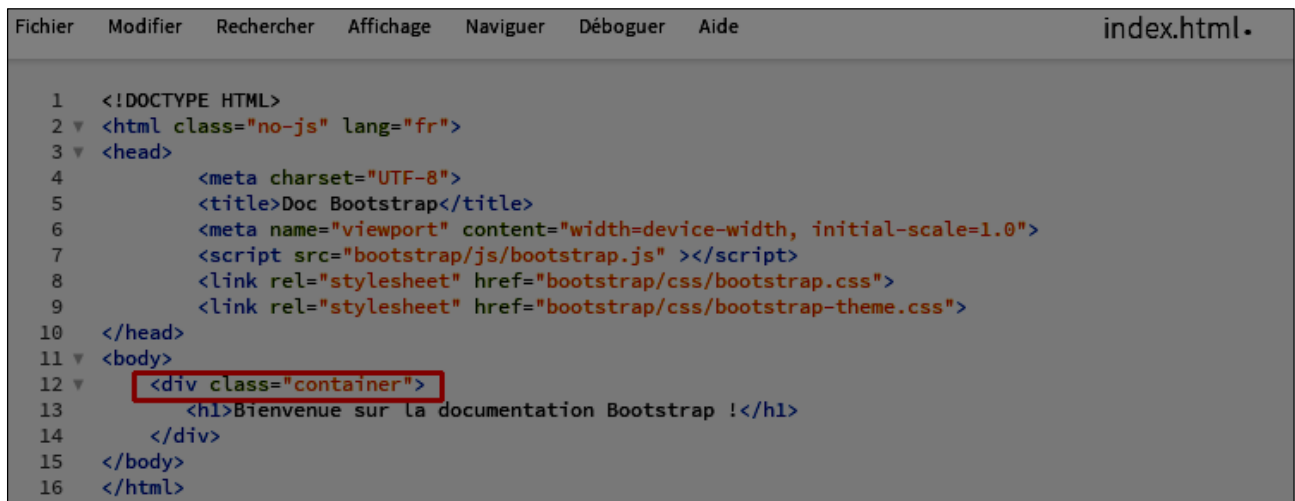
**ATTENTION :** Il est impératif d'avoir installé également jQuery afin de pouvoir utiliser les plugins JavaScript.

Pour télécharger jQuery, rendez vous sur la page officielle :

**<https://jquery.com/>**

## II – Utilisation

Afin que votre contenu soit responsive vous devriez placer votre code dans une div ayant pour attribut la classe « container », comme ci-dessous :



```
Fichier  Modifier  Rechercher  Affichage  Naviguer  Débuguer  Aide  index.html

1  <!DOCTYPE HTML>
2  <html class="no-js" lang="fr">
3  <head>
4      <meta charset="UTF-8">
5      <title>Doc Bootstrap</title>
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <script src="bootstrap/js/bootstrap.js" ></script>
8      <link rel="stylesheet" href="bootstrap/css/bootstrap.css">
9      <link rel="stylesheet" href="bootstrap/css/bootstrap-theme.css">
10 </head>
11 <body>
12     <div class="container">
13         <h1>Bienvenue sur la documentation Bootstrap !</h1>
14     </div>
15 </body>
16 </html>
```

**RAPPEL :** La balise <div></div> est un conteneur qui peut inclure des images, vidéos, tableaux, listes, et même d'autres div.

Les balises peuvent prendre en général deux attributs « class » et « id » qui permettent de les identifier dans le CSS par exemple, la différence étant qu'une même class pourra s'utiliser sur plusieurs éléments d'une même page alors qu'un id sera unique.

## Système de grille

Bootstrap est basé sur un système de grille (voir screen ci-dessous) qui est donc composé de lignes (row) et de colonnes (col) pouvant aller jusqu'à 12, ainsi vous pourrez définir la place que prendra vos éléments dans votre container en leur attribuant des valeurs qui doivent avoir pour somme 12. Il est important de se rappeler que 12 = à 100 % de la page, faites donc en sorte de respecter cela, il est également nécessaire de ne pas oublier que l'on doit commencer par une div de class row : « div class="row" » qui englobera nos éléments,

span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1
span 4				span 4				span 4			
span 4				span 8							
span 6						span 6					
span 12											

Pour illustrer mes propos je vais prendre un exemple, je souhaite créer un formulaire avec pour champs le « *Nom d'utilisateur* » ainsi que le « *Mot de passe* » d'une personne, on veut donc mettre ces éléments sur une même ligne, si on se reporte à l'image ci-dessus on pourrait utiliser deux div ayant pour attribut "col-md-6" et ainsi faire 12 au total mais afin d'avoir quelque-chose de plus « aéré » nous pouvons utiliser l'attribut "col-offset-md-X" sautant ainsi X colonnes sur notre ligne cependant ce ne sera pas interprété par notre page si nous ne l'indiquons pas, il faut donc ajouter avant notre « div class= "col-md-4" » un "col-md-offset-2", ce qui donnera « div class="col-md-offset-2 col-md-4 " ».

Ce sera plus clair avec le code et le résultat avant après que vous pourrez trouver ci-dessous :

Avec deux col-md-6 :

```
<div class="row">
  <div class="col-md-6">
    <label for="username">Nom d'utilisateur</label>
    <input type="text" class="form-control" id="username" placeholder="Nom d'utilisateur">
  </div>

  <div class="col-md-6">
    <label for="password">Mot de passe</label>
    <input type="password" class="form-control" id="password" placeholder="Mot de passe">
  </div>
</div>
```

## Bienvenue sur la documentation Bootstrap !

**Nom d'utilisateur**

**Mot de passe**

Avec trois col-md-4 :

Voilà ce qu'il se passe lorsque nous diminuons le col-md-X, les champs sont désormais plus petit et nous avons plus de place pour mettre un autre élément. Mais il se peut que vous ne vouliez pas forcément rajouter un champ, seulement réduire ceux déjà présent, si c'est le cas regardez l'exemple ci-dessous.

```

<div class="row">
  <div class="col-md-4">
    <label for="username">Nom d'utilisateur</label>
    <input type="text" class="form-control" id="username" placeholder="Nom d'utilisateur">
  </div>

  <div class="col-md-4">
    <label for="password">Mot de passe</label>
    <input type="password" class="form-control" id="password" placeholder="Mot de passe">
  </div>

  <div class="col-md-4" id="demo">
    <p>Ceci prends 4 colonnes</p>
  </div>
</div>

```

## Bienvenue sur la documentation Bootstrap !

**Nom d'utilisateur**

**Mot de passe**

Ceci prends 4 colonnes

Avec deux col-md-4 et en utilisant les col-md-offset-2

Les champs font 4 colonnes chacun, la propriété col-md-offset-2 fait en sorte que l'on saute deux colonnes avant le champ, voilà donc ce qu'il se passe :  
 saut de deux colonnes → champ de 4 colonnes → saut de deux colonnes → champ de 4 colonnes, 12 colonnes au total.

```

<div class="row">

  <div class="col-md-offset-2 col-md-4">
    <label for="username">Nom d'utilisateur</label>
    <input type="text" class="form-control" id="username" placeholder="Nom d'utilisateur">
  </div>

  <div class=" col-md-offset-2 col-md-4">
    <label for="password">Mot de passe</label>
    <input type="password" class="form-control" id="password" placeholder="Mot de passe">
  </div>

</div>

```

## Bienvenue sur la documentation Bootstrap !

Nom d'utilisateur

Mot de passe

Désormais il serait plus esthétique et plus simple pour l'utilisateur si nous mettions des icônes devant les champs afin de savoir directement de quoi l'on parle et Bootstrap nous permet cela grâce à « Glyphicon » qui s'utilise de la sorte suivante :

```

<div class="row">

  <div class="col-md-offset-2 col-md-4">
    <div class="input-group">
      <span class="input-group-addon"><span class="glyphicon glyphicon-user"></span></span>
      <input type="text" class="form-control" id="username" placeholder="Nom d'utilisateur">
    </div>
  </div>


  <div class=" col-md-offset-2 col-md-4">
    <div class="input-group">
      <span class="input-group-addon"><span class="glyphicon glyphicon-flag"></span></span>
      <input type="password" class="form-control" id="password" placeholder="Mot de passe">
    </div>
  </div>


</div>

```

Le "input-group-addon" indique que l'on désire mettre notre icône à l'intérieur de notre champ input, tandis que le glyphicon-flag définit l'icône que l'on souhaite mettre en place, ce qui donne en l'occurrence le résultat suivant :

## Bienvenue sur la documentation Bootstrap !

 Nom d'utilisateur

 Mot de passe

Il est possible de choisir ses icônes via ce lien :  
<http://getbootstrap.com/components/>

**ATTENTION** : Il faut retirer les label sinon vos icônes seront plus grandes que vos champs et le rendu sera nettement moins esthétique.

### Éléments graphique

Bootstrap permet également l'utilisation de ressources graphiques préconçue tel que des boutons comme ci-dessous :



Pour notre début de formulaire je décide d'utiliser le premier bouton en tant que submit voici son code basique donné sur le site de Bootstrap : <http://v4-alpha.getbootstrap.com/components/buttons/>, vous retrouverez également sur celui-ci d'autres design de boutons.


```
<!-- Provides extra visual weight and identifies the primary action in a set of buttons -->  
<button type="button" class="btn btn-primary">Primary</button>
```


Cependant la taille ne me convient pas et j'aimerais qu'elle prenne quatre colonnes de la grille je dois procéder donc de la manière suivante :

```
<div class="row">  
  <div class="col-md-offset-5 col-md-4">  
    <button type="button" class="btn btn-primary btn-block">Connexion</button>  
  </div>  
</div>
```

Je lui ai donc attribué le col-md-4 et un col-md-offset-5 pour qu'il soit centré, pour les boutons si l'on veut qu'ils prennent 100 % de la div où ils sont il est nécessaire de rajouter la propriété souligné : btn-block.

## Bienvenue sur la documentation Bootstrap !

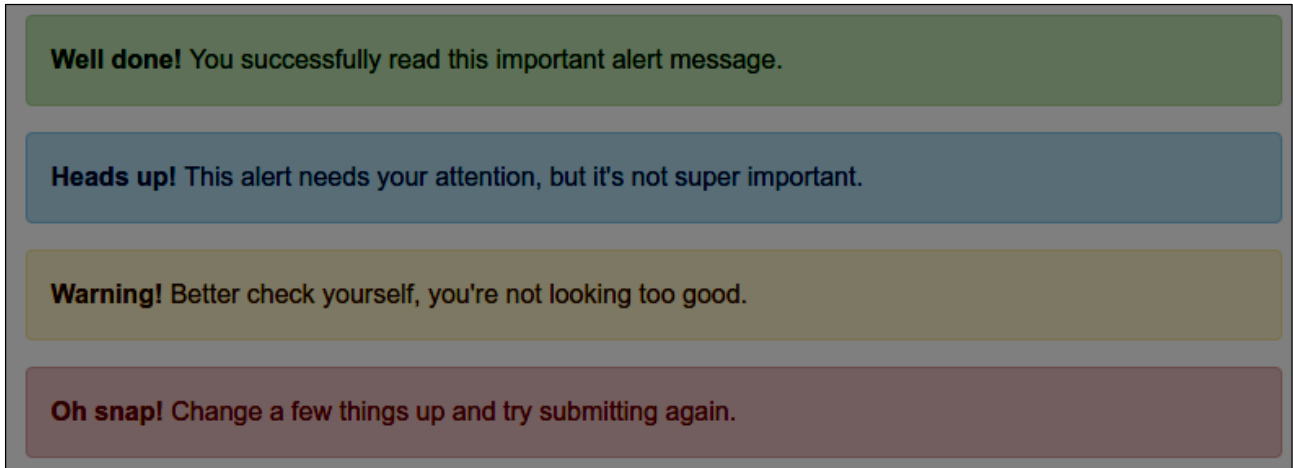
 Nom d'utilisateur

 Mot de passe

Connexion

## Les alertes

Bootstrap nous permet également d'afficher des messages de ce type(disponible ici : <http://v4-alpha.getbootstrap.com/components/alerts/>) là :



Ce qui peut-être assez utile pour la confirmation d'envoi de mail lors d'un formulaire de contact par exemple.  
Dans l'exemple ci-dessous j'ai utilisé du JavaScript afin d'afficher l'alerte lors du clic sur le bouton « Connexion ».

HTML :

```
<div class="row">
  <div class="col-md-offset-5 col-md-4">
    <button type="submit" class="btn btn-primary btn-block" id="alertMe">Connexion</button>
  </div>
</div>
<br>
<div class="row">
  <div class="col-md-offset-3 col-md-6">
    <div class="alert alert-success" role="alert" id="successAlert">
      <strong>Félicitations !</strong> Vous vous êtes bien connecté.
    </div>
  </div>
</div>
</div>
```

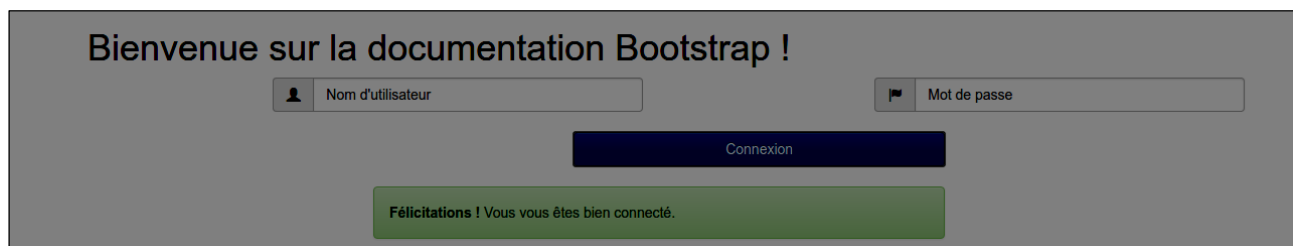
CSS : .alert{  
display:none}

JS :

```
$(function() {
  $('#alertMe').click(function(e){
    e.preventDefault();

    $('#successAlert').slideDown();
  });
});
```

Résultat :

A screenshot of a login form titled "Bienvenue sur la documentation Bootstrap !". It features two input fields: "Nom d'utilisateur" with a person icon and "Mot de passe" with a key icon. Below these is a dark blue "Connexion" button. At the bottom, a green message box says "Félicitations ! Vous vous êtes bien connecté."

## Le système responsive de la grille

Bootstrap nous permet de faire du responsive et notamment par rapport à la taille que nous donnons à nos éléments en fonction de la grille

	Extra small devices Phones (<768px)	Small devices Tablets (≥768px)	Medium devices Desktops (≥992px)	Large devices Desktops (≥1200px)
Grid behavior	Horizontal at all times	Collapsed to start, horizontal above breakpoints		
Container width	None (auto)	750px	970px	1170px
Class prefix	.col-xs-	.col-sm-	.col-md-	.col-lg-
# of columns	12			
Column width	Auto	~62px	~81px	~97px
Gutter width	30px (15px on each side of a column)			
Nestable	Yes			
Offsets	Yes			
Column ordering	Yes			

Comme vous pouvez le voir ci-dessus, il y'a principalement quatre classes à utiliser **col-xs** pour tout ce qui est en général des **téléphones** d'une taille inférieur à **768px**, **col-sm** pour ce qui est des **tablettes** avec une taille supérieur ou égale à **768px**, **col-md** pour ce qui est des **ordinateurs** avec un écran de taille moyenne supérieur ou égale à **992px** et pour finir les **ordinateurs** avec de larges-écran ayant une taille supérieur ou égale à **1200px**.

Remarque : Il est important de garder en mémoire que peu importe la taille de notre écran le système de grille est ici basé sur 12 colonnes, que ce soit un portable ou un ordinateur.

## Système de validation des champs

Prérequis : Il est nécessaire d'avoir le fichier validator.js afin de pouvoir vérifier le statut des champs, celui-ci peut être trouvé sur le github suivant : <https://github.com/1000hz/bootstrap-validator>



La documentation quant à elle est disponible ici :  
<http://1000hz.github.io/bootstrap-validator/>

Il est possible à l'aide de Bootstrap de vérifier le statut des champs en temps réel que le champ soit validé ou qu'il contienne une erreur comme le montre le screen ci-dessous :

**Bienvenue sur la documentation Bootstrap !**

**Votre nom \***  
Ce champ ne peut uniquement contenir des lettres.

**Email \***  
Veuillez inclure "@" dans l'adresse e-mail. Il manque un symbole "@" dans "4".

**Objet \***  
Veuillez allonger ce texte pour qu'il comporte au moins 14 caractères. Il en compte actuellement 1.

**Message: \***

Envoyer

On peut voir ici que quatre champs ne sont pas validés et la raison est bien indiquée, de plus les champs qui ne sont pas validés passent en rouge et ceux qui le sont passent en vert, tant que les champs requis ne seront pas validés, le bouton « Submit » ne pourra pas être activé.

```
<div class="form-group has-feedback row4 col-md-offset-2 col-md-4 col-xs-offset-2 col-xs-6 col-sm-offset-2 col-sm-5 col-lg-offset-2 col-lg-3 col-xl-offset-2 col-xl-2">
  <label for="name" class="control-label">Votre nom <span class="asterisk">*</span></label>
  <div class="input-group">
    <span class="input-group-addon">
      <i class="glyphicon glyphicon-user"></i>
    </span>
    <input pattern="[a-zA-Z\s]+" type="text" class="form-control" id="name" name="name" placeholder="Ex: Paul Dupont" data-feedback="success" maxlength="50"
      required data-error="Ce champ ne peut uniquement contenir des lettres.">
    </div>
    <span class="glyphicon form-control-feedback" aria-hidden="true"></span>
  </div>
  <div class="help-block with-errors"></div>
</div>
```

Petite explication des attributs utilisés ci-dessus :

Les class : form-group, form-control permettent de gérer la validation des champs.

La class : has-feedback permet d'ajouter des icônes à la fin des champs que ce soit un « check » qui est là pour montrer que le champ est bien validé ou une croix rouge qui montre le contraire.

La class : control-label permet de répercuter le statut du champ sur le label, ainsi si le champ est invalide le label deviendra rouge tout comme le champ.

L'attribut `pattern`, permet de définir quels sont les caractères pouvant être utilisés dans un champ, on peut voir ci-dessus que dans notre cas ça se limite à l'utilisation de majuscule et minuscule car un nom ne peut pas être composé de caractères spéciaux tel que des « @ » ni de chiffre.

L'attribut `required`, permet de définir un champ comme étant nécessaire au formulaire, de ce fait si il n'est pas ou mal rempli le bouton submit sera bloqué.

L'attribut `maxlength`, définit la taille maximum de notre champ, un nom fait rarement plus de 50 caractères, cela nous servira de sécurité contre tout spam possible.

L'attribut `data-error`, nous permet de définir le message d'erreur lorsque le champ est mal rempli, pour mon exemple celui qui était inséré par défaut était « Merci de respecter le format », mais j'ai mis le mien « Ce champ ne peut uniquement contenir des lettres », ainsi je donne une information plus précise sur ce que j'attends dans mon champ.

La balise `div` contient les class `help-block` & `with-errors` permet d'afficher par défaut les erreurs rencontrés lorsqu'un champ est mal rempli.

Pour finir notre balise `<span class="Glyphicon form-control-feedback" aria-hidden="true"></span>`, permet elle aussi d'avoir les icône du `has-feedback` et de les cachés lorsque le champ n'a pas encore été rempli.

## Plug-in supplémentaire :

Il peut-être intéressant que lorsqu'on demande dans un champ un numéro de téléphone plutôt que d'avoir tout les chiffres collés de cette façon : 0659876545 nous les ayons de celle-ci : 06-59-87-65-45, que ce soit d'un point de vue de lisibilité ou de praticité et bien c'est possible et sans que la personne ne mette les tirets, exactement lorsqu'on entre une licence dans un logiciel ou un numéro de carte.

Pour cela il suffit de télécharger les fichiers correspondant

```
<div class="form-group has-feedback row4 col-md-offset-2 col-md-4 col-xs-offset-2 col-xs-6 col-sm-offset-2 col-sm-5 col-lg-offset-2 col-lg-3 col-xl-offset-2 col-xl-2">
  <label for="tel" class="control-label">Numéro de téléphone</label>
  <div class="input-group">
    <span class="input-group-addon">
      <i class="glyphicon glyphicon-earphone"></i>
    </span>
    <input type="tel" class="form-control bfh-phone" id="tel" name="tel" placeholder="Ex: 06-45-68-98-56" size="14" maxlength="14" minlength="14" data-format="dd-dd-dd-dd">
  </div>
  <span class="glyphicon form-control-feedback" aria-hidden="true"></span>
  <div class="help-block with-errors"></div>
</div>
```

Sources :

<http://getbootstrap.com/>  
<http://1000hz.github.io/bootstrap-validator/>  
<http://bootstrapformhelpers.com/>