

Using Bootstrap

Using Bootstrap

About Bootstrap

Bootstrap is "a sleek, intuitive, and powerful mobile first front-end framework for faster and easier web development." This tools makes it a lot simpler to build responsive web sites – meaning websites which adapt their design to the size of the screen, from mobile phone to giant TV.

A Bootstrap design is made of:

- Conventions for structuring HTML code and naming CSS classes.
- A base CSS files (using the LESS or Sass format) built from a file of variables.
- A JavaScript for the the more usual functions.

Bootstrap uses a whole new way of working on web project that will really benefit all your forthcoming project – even non-PrestaShop ones! We strongly advise to dive into it:

- The [official Bootstrap website](#) has many informative section, including a [Getting started guide](#) (including code compilation, an [overview of how it uses CSS](#), and an explanation of [its use of JavaScript](#)).
- Sitepoint has several free articles/tutorials : [Understanding Twitter Bootstrap 3](#), [Building Responsive Websites Using Twitter Bootstrap](#), or [Twitter Bootstrap Tutorial – Handling Complex Designs](#).

General technical information

Bootstrap uses some specific HTML elements which make it a requirement to use HTML5. Be careful to use the proper HTML5 doctype for all your template files!

Likewise, CSS3 is used for many CSS properties.

```
<!DOCTYPE html>
<html lang="fr">
...
</html>
```

Bootstrap 3 is made to be mobile friendly from the start of your project: instead of adding optional mobile styles to your project, these styles are built into the core of the project, ensure that it displays well on all devices.

In order to ensure that your project renders well and that the touch zoom works as expected, you must add the `viewport` meta tag to your template's `head` element:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Bootstrap makes it easy to build a responsive design, and you can use a little trick to make your images responsive too: use Bootstrap's `img-responsive` class on the image. That class will make your image use `max-width: 100%;` and `height: auto;` in order to adapt to the parent element.

```

```

Fluid grid system

Bootstrap uses a 12-column grid system, which helps you build a fluid layout. You can use media queries to indicate breakpoints to the grid system, so that it can scale up or down its number of columns depending on the screen size. The default size is tailored for very small screen sizes (up to 480px width, for small phones), so no media query is required there. Bigger screen size are managed using the following queries:

```
/* Small devices (tablets, 768px and up) */
@media (min-width: @screen-sm) { ... }

/* Medium devices (desktops, 992px and up) */
@media (min-width: @screen-md) { ... }

/* Large devices (large desktops, 1200px and up) */
@media (min-width: @screen-lg) { ... }
```

Bootstrap uses class prefixes to differentiate devices by their screen sizes:

- `.col-xs-...`: extra small, for phones (< 768px)
- `.col-sm-...`: small, for tablets (768 px)
- `.col-md-...`: medium, for desktops (992 px)
- `.col-lg-...`: large, for large desktops and TVs (1200 px)

For instance:

- `col-xs-3`: 3 columns on phones.
- `col-md-4`: 4 columns on desktops.

For each device, Bootstrap also provides CSS classes, allowing you to change the column's position:

- `col-size-push-col`: move column to the left.
- `col-size-pull-col`: move column to the right.
- `col-size-offset-col` : position column according to other column.

For instance:

- `col-md-push-2`: On desktops, move this column by two columns to the left.

Bootstrap & Sass: Using variables and mixins

Variables

You can configure Bootstrap by editing the variables in the `_variables.scss` file. For instance:

```
@grid-columns: 12;
@grid-gutter-width: 30px;
@grid-float-breakpoint: 768px;
```

Mixins

A **mixin** lets you make groups of CSS declarations that you want to reuse throughout your site. For instance:

```
@mixin border-radius($radius) {
  -webkit-border-radius: $radius;
  -moz-border-radius: $radius;
  -ms-border-radius: $radius;
  border-radius: $radius;
}

.box { @include border-radius(10px); }
```

You can use mixins to define new styles. For instance:

```
.wrapper {
  .make-row();
}
.content-main {
  .make-lg-column(8);
}
.content-secondary {
  .make-lg-column(3);
  .make-lg-column-offset(1);
}
```

...for the following HTML code:

```
<div class="wrapper">
  <div class="content-main">...</div>
  <div class="content-secondary">...</div>
</div>
```

Bootstrap classes

Bootstrap has many default CSS classes, and PrestaShop defines several more, to help you build a consistent design.

Helper classes

These classes are to be used when design an element that uses one of PrestaShop's [Helper classes](#).

- `pull-left`: For left float.
- `pull-right`: For right float.
- `text-muted`: For gray text.
- `clearfix`: To prevent excessive size of floating blocks.
- `close`: To create a close button (x).
- `caret`: To indicate a dropdown effect.
- `center-block`: To center an element.
- `show` and `hidden`: To show/hide an element.

Responsive classes

These classes are useful to show/hide an element depending on the device.

- `visible-xs / hidden-xs`
- `visible-sm / hidden-sm`
- `visible-md / hidden-md`

- visible-lg / hidden-lg
- visible-print / hidden-print

Navigation, tabs and menus

The main classes for these contexts are:

- navbar and navbar-inner: Container class for the navigation bar.
- navbar-fixed-top: To attach the navigation bar to the top of the page.
- brand: For the site's title/store name.
- nav, nav-tabs and nav-pills: For the navigation tabs.
- btn and btn-navba: For the buttons.
- nav-collapse, collapse, data-toggle, data-target: To automatically hide/show content.
- icon-th-list: To display an icon (on small screens only: th).

Here is an example of a navigation bar with a menu on the right:

```
<div class="navbar navbar-fixed-top">
  <div class="navbar-inner">
    <div class="container">
      <a class="btn btn-navbar" data-toggle="collapse" data-target=".nav-collapse">
        <span class="icon-th-list"></span></a>
      <a href="#" class="brand">SiteTitle</a>
      <div class="nav-collapse collapse">
        <ul class="nav pull-right">
          <li class="active"><a href="#">Home</a></li>
          <li><a href="#">About Us</a></li>
          <li><a href="#">Contact Us</a></li>
        </ul>
      </div>
    </div>
  </div>
</div>
```

Here is an example of menu tab:

```
<ul class="nav nav-tabs">
  <li class="active"><a href="#">Home</a></li>
  <li><a href="#">Profile</a></li>
  <li class="dropdown">
    <a href="#" data-toggle="dropdown" class="dropdown-toggle">Messages <b class="caret"></b></a>
    <ul class="dropdown-menu">
      <li><a href="#">Inbox</a></li>
      <li><a href="#">Drafts</a></li>
      <li class="divider"></li>
      <li><a class="disabled" href="#">Trash</a></li>
    </ul>
  </li>
</ul>
```

Tables

Main classes for tables:

- table, table-responsive: To create a table and activate the scrolling on smaller screens.
- table-striped: To add an alternating background in the rows.
- table-bordered: To add a border.
- table-hover: To add a background to the row when the mouse hovers it.

- `table-condensed`: To divide the cellpadding in two (makes the table more compact).
- `success`, `warning`, `danger`, etc.: To change the background color of a row or of a cell.

Sample usage::

```
<table class="table table-condensed table-hover">
<thead>
  <tr>
    <th>First Name</th>
    <th>Last Name</th>
    <th>Email</th>
  </tr>
</thead>
<tbody>
  <tr class="warning">
    <td>John</td>
    <td>Carter</td>
    <td>johncarter@mail.com</td>
  </tr>
  <tr>
    <td>Peter</td>
    <td>Parker</td>
    <td>peterparker@mail.com</td>
  </tr>
  <tr>
    <td>John</td>
    <td>Rambo</td>
    <td>johnrambo@mail.com</td>
  </tr>
</tbody>
</table>
```

Panels

Main classes for this context:

- `panel`, `panel-body`: To define a panel.
- `panel-default`, `panel-primary`, `panel-success`, `panel-info`, `panel-warning` and `panel-danger`: To adapt the layout to the panel.
- `panel-heading` and `panel-title`: To add a header and a title to the panel.
- `panel-footer`: To add a footer to the panel.

Sample usage:

```
<div class="panel panel-default">
  <div class="panel-heading">
    <h1 class="panel-title">Panel Title</h1>
  </div>
  <div class="panel-body">
    This is the content of the panel.
  </div>
  <div class="panel-footer clearfix">
    <div class="pull-right">
      <a href="#" class="btn btn-default">Go Back</a>
    </div>
  </div>
</div>
```

Images

- `img-rounded`: To display an image with rounded borders.
- `img-circle`: To display an image within a circle.

- `img-thumbnail`: To handle thumbnails.

Sample usage:

```
<div class="container">
  <div class="row">
    <div class="col-xs-6">
      <div class="thumbnail">
        
        <div class="caption">
          <h3>label</h3><p>description...</p>
        </div>
      </div>
    </div>
    <div class="col-xs-6">
      <div class="thumbnail">
        
        <div class="caption">
          <h3>label</h3><p>description...</p>
        </div>
      </div>
    </div>
  </div>
</div>
```

Lists

Main classes:

- `list-unstyled`: List without list-style and now padding-left.
- `list-inline`: Horizontal list.
- `breadcrumb`: Horizontal list for navigation path.
- `dl-horizontal`: List with two aligned elements.
- `list-group`: Panel-list-like display.

Breadcrumb example:

```
<ul class="breadcrumb">
  <li><a href="#">Home</a></li>
  <li><a href="#">Products</a></li>
  <li class="active">Accessories</li>
</ul>
```

Header and paragraph example:

```

<div class="list-group">
  <a href="#" class="list-group-item">
    <h4 class="list-group-item-heading">What is HTML?</h4>
    <p class="list-group-item-text">HTML stands for HyperText Markup Language.
      HTML is the main markup language for describing the structure of Web pages.</p>
  </a>
  <a href="#" class="list-group-item active">
    <h4 class="list-group-item-heading">What is Twitter Bootstrap?</h4>
    <p class="list-group-item-text">Twitter Bootstrap is a powerful front-end framework for faster
      and easier web development. It is a collection of HTML and CSS based design template.</p>
  </a>
  <a href="#" class="list-group-item">
    <h4 class="list-group-item-heading">What is CSS?</h4>
    <p class="list-group-item-text">CSS stands for Cascading Style Sheet. CSS allows you to specify
      various style properties for a given HTML element such as colors, backgrounds, fonts etc.</p>
  </a>
</div>

```

Forms

Main classes:

- `form-inline` and `form-horizontal` on the form element.
- `form-group` on the div that groups the label and the textarea. Enables automatic adjustment of spacing.
- `form-control` on the input, textarea and select elements– which by default have a width of 100%.

A container that has the `form-group` class always needs a label. To add a label and display it, you must use the `sr-only` class. For instance:

```

<div class="form-group">
  <label class="sr-only" for="exampleInput">Email address, label not visible</label>
  <input type="email" class="form-control" id="exampleInput" placeholder="Enter email">
</div>

```

Classes that change the appearance of zones depending on content validation:

- `has-warning`
- `has-error`
- `has-success`

Classes that enable the display of an icon in the textarea:

- `glyphicon` et `form-control-feedback`
- `glyphicon-ok`
- `glyphicon-warning-sign`
- `glyphicon-remove`

For instance:

```

<div class="form-group has-warning">
  <label class="control-label" for="input1">Label with warning</label>
  <input type="text" class="form-control" id="input1">
  <span class="glyphicon glyphicon-warning-sign form-control-feedback"></span>
</div>

```

Buttons and links

Main classes:

- **btn**: Base class for buttons
- **btn-default**, **btn-primary**, **btn-success**, **btn-info**, **btn-warning**, **btn-danger**, **btn-link**: To change the button's appearance
- **btn-lg**, **btn-sm** and **btn-xs**: To change the button's size.
- **btn-block**: To change the size of the parent element.
- **active** and **disabled**: to change the availability of a button.

```
<span class="button ajax_add_to_cart_button btn btn-default disabled">
  <span>{l s='Add to cart'}</span>
</span>
<a itemprop="url" class="button lnk_view btn btn-default"
  href="{ $product.link|escape:'html':'UTF-8'}" title="{l s='View'}">
  <span>{l s='More'}</span>
</a>
<button type="submit" name="submitMessage" id="submitMessage"
  class="button btn btn-default button-medium">
  <span>{l s='Send'}<i class="icon-chevron-right right"></i></span>
</button>
```