

# Basic Bootstrap Classes

`.container`

sets fixed width to an element (which changes depending on a screen size to other fixed values, so it's still responsive) on all screen sizes except xs - on xs, the width is calculated automatically (this behaviour can be changed)

a fixed width responsive website should be wrapped in `.container` (or just the content without menu, whatever - it's up to you)

`.container` can't be nested!

`.container-fluid`

sets 100% width, `margin-left` and `margin-right`: auto, `padding-left` and `padding-right`: 15px

a full screen website should be wrapped in `.container-fluid`

`.container-fluid` can't be nested!

`.row`

creates horizontal groups of columns (which usually have width classes, see below)

must be within `.container` to ensure correct padding and alignment

only columns (block elements with width classes) may be immediate children of rows

sets `margin-left` and `margin-right`: -15px

`.pull-right`

`.pull-left`

sets float to either left or right

`.center-block`

sets `margin-left` and `margin-right` to auto

Bootstrap Grid Classes

can be nested, the nested columns must be wrapped in `.row` as well

all columns have `float: left` and `position: relative` styles

`.col-xs-1`

`.col-xs-2`

`.col-xs-3`

`.col-xs-4`

.col-xs-5

.col-xs-6

.col-xs-7

.col-xs-8

.col-xs-9

.col-xs-10

.col-xs-11

.col-xs-12

sets width of a column (specified in %) on extra small screens (<768px)

.col-sm-1

.col-sm-2

.col-sm-3

.col-sm-4

.col-sm-5

.col-sm-6

.col-sm-7

.col-sm-8

.col-sm-9

.col-sm-10

.col-sm-11

.col-sm-12

sets width of a column (specified in %) on small screens (≥768px)

.col-md-1

.col-md-2

.col-md-3

.col-md-4

.col-md-5

.col-md-6

.col-md-7

.col-md-8

.col-md-9

.col-md-10

.col-md-11

.col-md-12

sets width of a column (specified in %) on medium size screens ( $\geq 992\text{px}$ )

.col-lg-1

.col-lg-2

.col-lg-3

.col-lg-4

.col-lg-5

.col-lg-6

.col-lg-7

.col-lg-8

.col-lg-9

.col-lg-10

.col-lg-11

.col-lg-12

sets width of a column (specified in %) on large screens ( $\geq 1200\text{px}$ )

.col-xs-offset-\* (.col-xs-offset-1, .col-xs-offset-2, .col-xs-offset-3 ... up to -12)

.col-sm-offset-\* (.col-sm-offset-1, .col-sm-offset-2, .col-sm-offset-3 ... up to -12)

.col-md-offset-\* (.col-md-offset-1, .col-md-offset-2, .col-md-offset-3 ... up to -12)

.col-lg-offset-\* (.col-lg-offset-1, .col-lg-offset-2, .col-lg-offset-3 ... up to -12)

sets offset (margin-left) by specified number of columns (eg. set .col-sm-offset-3 class to a column to move it by three columns on small screens)

.col-xs-pull-\* (.col-xs-pull-1, .col-xs-pull-2 ... up to -12)

.col-sm-pull-\* (.col-sm-pull-1, .col-sm-pull-2 ... up to -12)

.col-md-pull-\* (.col-md-pull-1, .col-md-pull-2 ... up to -12)

.col-lg-pull-\* (.col-lg-pull-1, .col-lg-pull-2 ... up to -12)

sets left position in % as it is with columns (e.g. left: 75% with -pull-9)

.col-xs-push-\* (.col-xs-push-1, .col-xs-push-2 ... up to -12)

.col-sm-push-\* (.col-sm-push-1, .col-sm-push-2 ... up to -12)

.col-md-push-\* (.col-md-push-1, .col-md-push-2 ... up to -12)

.col-lg-push-\* (.col-lg-push-1, .col-lg-push-2 ... up to -12)

sets right position in % as it is with columns (e.g. right: 25% with -push-3)

### Bootstrap Visibility Classes

you can apply all as many of these classes on a single element as you wish and this mixing gives you even more possibilities for your layout

.clearfix

obvious, clear: both

often combined with classes below

.visible-xs-block and .visible-xs

.visible-sm-block and .visible-sm

.visible-md-block and .visible-md

.visible-lg-block and .visible-lg

displays element (sets display: block property) only on screens with specified size and hides it (sets display: none) on screens of other sizes automatically

often combined with .clearfix so that clearfix is effective only on the selected screen size

.visible-xs-inline-block

.visible-sm-inline-block

.visible-md-inline-block

.visible-lg-inline-block

displays element (sets display: inline-block property) only on screens with specified size and hides it (sets display: none) on screens of other sizes automatically

.visible-xs-inline

.visible-sm-inline

.visible-md-inline

.visible-lg-inline

displays element (sets display: block property) only on screens with specified size and hides it (sets display: none) on screens of other sizes automatically

.hidden-xs

.hidden-sm

.hidden-md

.hidden-lg

hides element (sets display: none property) only on screens with specified size and hides it (sets display: none) on screens of other sizes automatically

.visible-print-block

displays element (sets display: block) in print (pre)view

.visible-print-inline

displays element (sets display: inline) in print (pre)view

.visible-print-inline-block

displays element (sets display: inline-block) in print (pre)view

.hidden-print

hides element (sets display: none) in print (pre)view

### Bootstrap Typography Classes

default font-size of <body> a <p> elements is 14px, line-height 1.428; <p> also has margin-bottom of half of its line-height (20px with 14px font), therefore 10px by default

all elements like b, big, code, em, i, mark, small, strong, sub, sup, ins, del, s, u, abbr, address and blockquote (and small and cite inside) are properly styled

vast majority of these classes are most often set to either paragraph, strong or em elements

.page-header

class for a wrapping div of a heading

adds some margin, padding and border-bottom

.h1

.h2

.h3

.h4

.h5

.h6

makes an element look like heading of a chosen level

.lead

for highlighting an element

sets element's font-size to 21px, font-weight: 300, line-height: 1.4 and margin-bottom: 20px

.text-left

.text-center

.text-right

.text-justify

aligns texts, usually paragraphs, with text-align CSS property

.text-nowrap

keeps the whole text with this class on a single line, no wrapping occurs because of white-space: nowrap CSS property of this class

.text-lowercase

.text-uppercase

.text-capitalize

sets text-transform CSS property to make all characters inside the element with this class lowercase, uppercase or capitalized

.text-muted

makes text grey

.text-primary

makes text blue

.text-success

makes text green

.text-info

makes text dark blue

.text-warning

makes text dark yellow

.text-danger

makes text red

## Bootstrap Lists' Classes

.list-unstyled

removes list-style and margin-left on list items

doesn't apply for nested lists (if needed, the nested list has to have .list-unstyled class as well)

.list-inline

places list items on a single line (with display: inline-block and padding)

.dl-horizontal

class for a definition list (<dl>)

sets nice styles for <dt> and <dd> within the definition list (text align, font-weight)

.list-group

class for the parent element (usually <ul>, but can be <div> as well) of list items

list items have to have .list-group-item class

.list-group-item

class for .list-group children elements

for a nicer look of the list items

usually on <li>, but can be <a> as well

.list-group-item-heading

for a heading within .list-group-item, often <h4> or similar low-level heading

.list-group-item-text

for text within .list-group-item, often in the paragraph tag or just in a span

## Bootstrap Tables' Classes

.table

for an ordinary table

table heading has only border-bottom and <td>s have only border-top

.table-striped

every odd line will be darker for a nicer look (using :nth-child selector)

table heading <th> has only border-bottom and <td>s have only border-top

should be combined with .table class

.table-bordered

table with borders all around

should be combined with .table class

.table-hover

class for a table

adds hover/focus effect on the table's tbody rows

should be combined with .table class

.table-condensed

should be combined with .table class

for an ordinary table with half cell padding, so rows are not that high

.table-responsive

class for a wrapper (! mostly div) of a table to keep it responsive in the way that scrollbar appears on extra small devices

.active

.success

.info

.warning

.danger

classes for either <tr> or just <td>

sets background color for all <td> in a table row (or just one <td>) to grey (active) / green (success) / blue (info) / yellow (warning) / red (danger)

Bootstrap Forms' Classes

don't forget that even form elements can be in the Bootstrap grid, so don't be afraid to use .row inside a form and then col-\*-\* wrappers

.form-group

when creating vertical forms, it's a class for its wrapper (mostly div) of each label and its input, adds some margin-bottom

.form-control

for every form element (input, textarea...), gives it 100% width

to make it narrower, use a narrower wrapper (row and col-\*-\* divs)

`.form-control-static`

for a paragraph that replaces a form element (input...)

`.form-horizontal`

for horizontal forms where label is on the same line as input

you have to use column classes (like `col-xs-10`) for each label and its input (or whatever it is)

still combined with `.form-group` wrapper and `.form-control` classes

`.form-inline`

for a single line form (but still vertical on extra small screens)

often used with the placeholder attribute to not need labels

labels of inline forms should get `.sr-only` class to be positioned so that they can't be seen (but they won't get `display: none` or `visibility: hidden` property)

`.input-lg`

`.input-sm`

makes a form element like input or select either bigger or smaller than default

`.input-group-lg`

`.input-group-sm`

classes for `.input-group` (see further)

always used together with `.input-group` class

`.input-group`

class for a wrapper of `.input-group-addon` or `.input-group-btn`

used instead of `form-group` when the classes above are used

`.input-group-addon`

class for a span that serves as a wrapper for either `.glyphicon` or a radio/checkbox input

adds your choice (glyphicon...) on the inner edge of an input field with grey background

the span with this class can be either before or after input, depends where you want the icon to show (the left or right side of the input)

must be within `.input-group`

`.input-group-btn`

class for a span that serves as a wrapper for a button that looks like it was within a form input

the span with this class can be either before or after input, depends where you want the button to show (left or right side of the input)

contains <button> (the button can also include .dropdown-toggle class, span with .caret class and data-toggle="dropdown" attribute for even more complicated buttons)

must be within .input-group

.caret

class for an empty span that creates a dropdown arrow

.help-block

class for a span that serves as a help text for users

has to be placed right after an input of a form

.has-success

.has-warning

.has-error

classes for wrapping divs with .form-group class

colours the label, input and .help-block span inside to green/yellow/red

the label within the wrapper with one of these classes has to have .control-label class to be coloured as well

.has-feedback

class for the wrapping divs with .form-group class when you also use glyphicon for the input within the wrapper

To disable input, use disabled="disabled" or readonly="readonly",

To disable whole fieldset, wrap form-groups to fieldset with disabled="disabled"

Bootstrap Image Classes

images should have .img-responsive or .thumbnail to be responsive and be wrapped in a column wrapper (like col-xs-3...)

.img-rounded

img class that provides little border-radius

.img-circle

img class that provides 50% border-radius making the img circle shaped

.img-thumbnail

img class that provides a 1px solid grey border, 4px padding, white background, display: inline-block property and fast CSS3 transition

.thumbnail

class for a wrapping div (or <a> in case of gallery) of an image, adds some styles (border, padding, background, transition) and set's the image max-width: 100%, height: auto; display: block; and margin-left and margin-right: auto

.img-responsive

img class that makes any image responsive (scaling to the parent element) as it has max-width: 100% and height: auto

may not work if placed in a floating element!

.caption

class for a div inside .thumbnail element (a or div) that may include h3, p, a (also buttons) for a proper margin/padding

.media

class for a div (or li, whatever) that may contain an image, a heading, some <small> text and paragraphs ensures proper image margin/padding

heading inside should also have media-heading attribute and should be (together with paragraphs) wrapped in a .media-body wrapper

any image within should have .pull-left or .pull-right class to float

.media-list

class for a list (ul, ol) that contains .media inside

ensures better margin/padding

.glyphicon

base class for glyphs (font-based icons), sets font, its styles and positioning

should be used on a span

.glyphicon-\* (replace the star with any name of a [glyphicon from this list](#))

sets picture via font and :before CSS3 selector with content of the font

must be used together with .glyphicon class to show proper image

should be used on a span

Bootstrap Embed Classes

.embed-responsive

general class for responsive embedding wrapper

`.embed-responsive-4by3`

`.embed-responsive-16by9`

specific class for the embed wrapper with ratio already set

used together with `.embed-responsive` class

`.embed-responsive-item`

used on the `iframe` or whatever it is inside the wrapper

Bootstrap Navigation Classes

`.nav`

the basic class for every navigation structure (`<ul>`), removes any list-style, left padding and bottom margin

also sets display: block and position relative for its `<li>` children

must be set on every navigation

`.active`

class for a list item (nav item, `li`) that makes it look active / current

`.nav-tabs`

class for navigation structure that gives it a tab style (proper borders + float: left)

`.nav-pills`

class for navigation structure that gives it a pills style (background on active nav item)

`.nav-stacked`

makes vertical pills navigation

used together with `.nav-pills` (and `.nav` of course)

`.dropdown`

class for an

nav item to make it dropdown

needs additional classes elsewhere to work (see below)

`.dropdown-toggle`

class for an `<a>` which is within `<li class="dropdown">` to make a dropdown menu

needs additional classes elsewhere to work (see below)

.dropdown-menu

class for an <ul> which is within <a class="dropdown-toggle"> to make a dropdown menu

provides proper displaying and hiding

the most important dropdown menu class

needs .dropdown and .dropdown-toggle classes on parent elements (see above)

.nav-justified

class for an <ul>, together with .nav-tabs or .nav-pills creates a navigation with centered text and links' parent elements of the same width

.disabled

class for <li> in a navigation

disables the link inside VISUALLY and also adds a special mouse cursor, but the link remains active - you need to remove the href attribute of the <a> to completely disable the link

Bootstrap Navbar Classes

the difference between nav and navbar classes is the fact that navbar can collapse the whole nav (the nav can't do it on its own)

.navbar

basic class for a navigation wrapper (usually <nav>), sets border, bottom margin, min-height and position: relative

if it's not supposed to be fixed, can be placed anywhere in .container or .container-fluid

.navbar-default

sets other basic styles for a navigation wrapper (background gradient and box-shadow) and also its items (including .active class on <li>s)

for dark colours, see .navbar-inverse below

.navbar-inverse

variation of navbar-default but with dark colours

never use navbar-default and navbar-inverse on one element as it doesn't make any sense

.navbar-brand

sets styles for special hyperlinks and other "simple text" in a menu (float: left, bigger font size, some padding, text-shadow...)

.navbar-fixed-top

.navbar-fixed-bottom

additional class to `.navbar` that makes the navigation fixed at the top (or bottom) of the page as you scroll  
used together with `.navbar` and also `.navbar default` (just to keep the nice look)

you should add some top or bottom padding to your `<body>` to not let the navbar overflow your content

`.navbar-static-top`

removes top, right and left border and also border-radius of your navigation

`.breadcrumb`

class for `<ul>` that serves as breadcrumbs

gives breadcrumbs a cool look

`.pagination`

class for `<ul>` that serves as pagination

properly styles `<li>` within this pagination also with `.disabled` and `.active` classes (keep in mind that `.disabled` doesn't remove link's functionality)

`.pagination-lg`

`.pagination-sm`

used on `<ul>` together with `.pagination`, makes pagination either bigger or smaller than default

`.pager`

used on `<ul>`, styles "Previous" and "Next" links to look better with a grey border

its list items can have `.previous`, `.next` and also `disabled` classes for even better styling

`.next`

`.previous`

used on `<li>` within `<ul class="pager">`, adds arrows to the links