

[thinkadoo](#)  
[Home](#)  
[Get started](#)  
[Scaffolding](#)  
[Base CSS](#)  
[Components](#)  
[JavaScript](#)  
[Customize](#)

[Dropdown](#)  
[Button groups](#)  
[Button dropdowns](#)  
[Navs](#)  
[Navbar](#)  
[Breadcrumbs](#)  
[Pagination](#)  
[Labels and badges](#)  
[Typography](#)  
[Thumbnails](#)  
[Alerts](#)  
[Progress bars](#)  
[Media object](#)  
[Misc](#)

# Dropdown menus

## Example

Toggleable, contextual menu for displaying lists of links. Made interactive with the [dropdown JavaScript plugin](#).

[Action](#)  
[Another action](#)  
[Something else here](#)  
[Separated link](#)

```
<ul class="dropdown-menu" role="menu" aria-labelledby="dropdownMenu">
  <li><a tabindex="-1" href="#">Action</a></li>
  <li><a tabindex="-1" href="#">Another action</a></li>
  <li><a tabindex="-1" href="#">Something else here</a></li>
  <li class="divider"></li>
  <li><a tabindex="-1" href="#">Separated link</a></li>
</ul>
```

## Markup

Looking at just the dropdown menu, here's the required HTML. You need to wrap the dropdown's trigger and the dropdown menu within `.dropdown`, or another element that declares `position: relative`. Then just create the

menu.

```
<div class="dropdown">
  <!-- Link or button to toggle dropdown -->
  <ul class="dropdown-menu" role="menu" aria-labelledby="dLabel">
    <li><a tabindex="-1" href="#">Action</a></li>
    <li><a tabindex="-1" href="#">Another action</a></li>
    <li><a tabindex="-1" href="#">Something else here</a></li>
    <li class="divider"></li>
    <li><a tabindex="-1" href="#">Separated link</a></li>
  </ul>
</div>
```

## Options

Align menus to the right and add include additional levels of dropdowns.

### Aligning the menus

Add `.pull-right` to a `.dropdown-menu` to right align the dropdown menu.

```
<ul class="dropdown-menu pull-right" role="menu" aria-labelledby="dLabel">
  ...
</ul>
```

### Sub menus on dropdowns

Add an extra level of dropdown menus, appearing on hover like those of OS X, with some simple markup additions. Add `.dropdown-submenu` to any `li` in an existing dropdown menu for automatic styling.

Default

Action

Another action

Something else here

More options

Dropup

Action

Another action

Something else here

More options

Left submenu

Action

Another action

Something else here

More options

```
<ul class="dropdown-menu" role="menu" aria-labelledby="dLabel">
  ...
  <li class="dropdown-submenu">
    <a tabindex="-1" href="#">More options</a>
    <ul class="dropdown-menu">
      ...
    </ul>
  </li>
</ul>
```

# Button groups

## Examples

Two basic options, along with two more specific variations.

### Single button group

Wrap a series of buttons with `.btn` in `.btn-group`.

Left Middle Right

```
<div class="btn-group">
  <button class="btn">1</button>
  <button class="btn">2</button>
  <button class="btn">3</button>
</div>
```

### Multiple button groups

Combine sets of `<div class="btn-group">` into a `<div class="btn-toolbar">` for more complex components.

1 2 3 4  
5 6 7  
8

```
<div class="btn-toolbar">
  <div class="btn-group">
    ...
  </div>
</div>
```

## Vertical button groups

Make a set of buttons appear vertically stacked rather than horizontally.

```
<div class="btn-group btn-group-vertical">
  ...
</div>
```

## Checkbox and radio flavors

Button groups can also function as radios, where only one button may be active, or checkboxes, where any number of buttons may be active. View [the JavaScript docs](#) for that.

## Dropdowns in button groups

**Heads up!** Buttons with dropdowns must be individually wrapped in their own `.btn-group` within a `.btn-toolbar` for proper rendering.

# Button dropdown menus

## Overview and examples

Use any button to trigger a dropdown menu by placing it within a `.btn-group` and providing the proper menu markup.

- Action
- Action
- Danger
- Warning
- Success
- Info

Inverse

```
<div class="btn-group">
  <a class="btn dropdown-toggle" data-toggle="dropdown" href="#">
    Action
    <span class="caret"></span>
  </a>
  <ul class="dropdown-menu">
    <!-- dropdown menu links -->
  </ul>
</div>
```

## Works with all button sizes

Button dropdowns work at any size: `.btn-large`, `.btn-small`, or `.btn-mini`.

- Large button
- Small button
- Mini button

## Requires JavaScript

Button dropdowns require the [Bootstrap dropdown plugin](#) to function.

In some cases—like mobile—dropdown menus will extend outside the viewport. You need to resolve the alignment manually or with custom JavaScript.

## Split button dropdowns

Building on the button group styles and markup, we can easily create a split button. Split buttons feature a standard action on the left and a dropdown toggle on the right with contextual links.

- Action
- Action
- Danger
- Warning
- Success
- Info
- Inverse

```
<div class="btn-group">
  <button class="btn">Action</button>
  <button class="btn dropdown-toggle" data-toggle="dropdown">
    <span class="caret"></span>
  </button>
  <ul class="dropdown-menu">
    <!-- dropdown menu links -->
  </ul>
</div>
```

# Sizes

Utilize the extra button classes `.btn-mini`, `.btn-small`, or `.btn-large` for sizing.

- Large action
- Small action
- Mini action

```
<div class="btn-group">
  <button class="btn btn-mini">Action</button>
  <button class="btn btn-mini dropdown-toggle" data-toggle="dropdown">
    <span class="caret"></span>
  </button>
  <ul class="dropdown-menu">
    <!-- dropdown menu links -->
  </ul>
</div>
```

# Dropup menus

Dropdown menus can also be toggled from the bottom up by adding a single class to the immediate parent of `.dropdown-menu`. It will flip the direction of the `.caret` and reposition the menu itself to move from the bottom up instead of top down.

- Dropup
- Right dropup

```
<div class="btn-group dropup">
  <button class="btn">Dropup</button>
  <button class="btn dropdown-toggle" data-toggle="dropdown">
    <span class="caret"></span>
  </button>
  <ul class="dropdown-menu">
    <!-- dropdown menu links -->
  </ul>
</div>
```

# Nav: tabs, pills, and lists

## Lightweight defaults Same markup, different classes

All nav components here—tabs, pills, and lists—**share the same base markup and styles** through the `.nav` class.

### Basic tabs

Take a regular `<ul>` of links and add `.nav-tabs`:

- Home
- Profile
- Messages

```
<ul class="nav nav-tabs">
  <li class="active">
    <a href="#">Home</a>
  </li>
  <li><a href="#">...</a></li>
  <li><a href="#">...</a></li>
</ul>
```

### Basic pills

Take that same HTML, but use `.nav-pills` instead:

- Home
- Profile
- Messages

```
<ul class="nav nav-pills">
  <li class="active">
    <a href="#">Home</a>
  </li>
  <li><a href="#">...</a></li>
  <li><a href="#">...</a></li>
</ul>
```

## Disabled state

For any nav component (tabs, pills, or list), add `.disabled` for **gray links and no hover effects**. Links will remain clickable, however, unless you remove the `href` attribute. Alternatively, you could implement custom JavaScript to prevent those clicks.

Clickable link  
Clickable link  
Disabled link

```
<ul class="nav nav-pills">
  ...
  <li class="disabled"><a href="#">Home</a></li>
  ...
</ul>
```

## Component alignment

To align nav links, use the `.pull-left` or `.pull-right` utility classes. Both classes will add a CSS float in the specified direction.

## Stackable

As tabs and pills are horizontal by default, just add a second class, `.nav-stacked`, to make them appear vertically stacked.

## Stacked tabs

Home  
Profile



Messages

```
<ul class="nav nav-tabs nav-stacked">
  ...
</ul>
```

## Stacked pills

Home  
Profile  
Messages

```
<ul class="nav nav-pills nav-stacked">
  ...
</ul>
```

## Dropdowns

Add dropdown menus with a little extra HTML and the [dropdowns JavaScript plugin](#).

## Tabs with dropdowns

Home  
Help  
Dropdown

```
<ul class="nav nav-tabs">
  <li class="dropdown">
    <a class="dropdown-toggle"
      data-toggle="dropdown"
      href="#">
      Dropdown
      <b class="caret"></b>
    </a>
    <ul class="dropdown-menu">
      <!-- links -->
    </ul>
  </li>
</ul>
```

## Pills with dropdowns

Home  
Help  
Dropdown

```
<ul class="nav nav-pills">
  <li class="dropdown">
    <a class="dropdown-toggle"
      data-toggle="dropdown"
      href="#">
      Dropdown
      <b class="caret"></b>
    </a>
    <ul class="dropdown-menu">
      <!-- links -->
    </ul>
  </li>
</ul>
```

## Nav lists

A simple and easy way to build groups of nav links with optional headers. They're best used in sidebars like the Finder in OS X.

# Example nav list

Take a list of links and add `class="nav nav-list"`:

## LIST HEADER

[Home](#)

[Library](#)

[Applications](#)

## ANOTHER LIST HEADER

[Profile](#)

[Settings](#)

[Help](#)

```
<ul class="nav nav-list">
  <li class="nav-header">List header</li>
  <li class="active"><a href="#">Home</a></li>
  <li><a href="#">Library</a></li>
  ...
</ul>
```

**Note** For nesting within a nav list, include `class="nav nav-list"` on any nested `<ul>`.

## Horizontal dividers

Add a horizontal divider by creating an empty list item with the class `.divider`, like so:

```
<ul class="nav nav-list">
  ...
  <li class="divider"></li>
  ...
</ul>
```

## Tabbable nav

Bring your tabs to life with a simple plugin to toggle between content via tabs. Bootstrap integrates tabbable tabs in four styles: top (default), right, bottom, and left.

## Tabbable example

To make tabs tabbable, create a `.tab-pane` with unique ID for every tab and wrap them in `.tab-content`.

- Section 1
- Section 2
- Section 3

I'm in Section 1.

Howdy, I'm in Section 2.

What up girl, this is Section 3.

```
<div class="tabbable"> <!-- Only required for left/right tabs -->
  <ul class="nav nav-tabs">
    <li class="active"><a href="#tab1" data-toggle="tab">Section 1</a></li>
    <li><a href="#tab2" data-toggle="tab">Section 2</a></li>
  </ul>
  <div class="tab-content">
    <div class="tab-pane active" id="tab1">
      <p>I'm in Section 1.</p>
    </div>
    <div class="tab-pane" id="tab2">
      <p>Howdy, I'm in Section 2.</p>
    </div>
  </div>
</div>
```

## Fade in tabs

To make tabs fade in, add `.fade` to each `.tab-pane`.

## Requires jQuery plugin

All tabbable tabs are powered by our lightweight jQuery plugin. Read more about how to bring tabbable tabs to life [on the JavaScript docs page](#).

# Tabbable in any direction

## Tabs on the bottom

Flip the order of the HTML and add a class to put tabs on the bottom.

I'm in Section A.

Howdy, I'm in Section B.

What up girl, this is Section C.

- Section 1
- Section 2
- Section 3

```
<div class="tabbable tabs-below">
  <div class="tab-content">
    ...
  </div>
  <ul class="nav nav-tabs">
    ...
  </ul>
</div>
```

## Tabs on the left

Swap the class to put tabs on the left.

- Section 1
- Section 2
- Section 3

I'm in Section A.

Howdy, I'm in Section B.

What up girl, this is Section C.

```
<div class="tabbable tabs-left">
  <ul class="nav nav-tabs">
    ...
  </ul>
  <div class="tab-content">
    ...
  </div>
</div>
```

## Tabs on the right

Swap the class to put tabs on the right.

- Section 1
- Section 2
- Section 3

I'm in Section A.

Howdy, I'm in Section B.

What up girl, this is Section C.

```
<div class="tabbable tabs-right">
  <ul class="nav nav-tabs">
    ...
  </ul>
  <div class="tab-content">
    ...
  </div>
</div>
```

# Navbar

## Basic navbar

To start, navbars are static (not fixed to the top) and include support for a project name and basic navigation. Place one anywhere within a `.container`, which sets the width of your site and content.

```
Title
Home
Link
Link
```

```
<div class="navbar">
  <div class="navbar-inner">
    <a class="brand" href="#">Title</a>
    <ul class="nav">
      <li class="active"><a href="#">Home</a></li>
      <li><a href="#">Link</a></li>
      <li><a href="#">Link</a></li>
    </ul>
  </div>
</div>
```

## Navbar components

### Brand

A simple link to show your brand or project name only requires an anchor tag.

Title

```
<a class="brand" href="#">Project name</a>
```

## Nav links

Nav items are simple to add via unordered lists.

Home

Link

Link

```
<ul class="nav">
  <li class="active">
    <a href="#">Home</a>
  </li>
  <li><a href="#">Link</a></li>
  <li><a href="#">Link</a></li>
</ul>
```

You can easily add dividers to your nav links with an empty list item and a simple class. Just add this between links:

Home

Link

Link

```
<ul class="nav">
  ...
  <li class="divider-vertical"></li>
  ...
</ul>
```

## Forms

To properly style and position a form within the navbar, add the appropriate classes as shown below. For a default form, include `.navbar-form` and either `.pull-left` or `.pull-right` to properly align it.

Submit



```
<form class="navbar-form pull-left">
  <input type="text" class="span2">
  <button type="submit" class="btn">Submit</button>
</form>
```

## Search form

For a more customized search form, add `.navbar-search` to the `form` and `.search-query` to the input for specialized styles in the navbar.

```
<form class="navbar-search pull-left">
  <input type="text" class="search-query" placeholder="Search">
</form>
```

## Component alignment

Align nav links, search form, or text, use the `.pull-left` or `.pull-right` utility classes. Both classes will add a CSS float in the specified direction.

## Using dropdowns

Add dropdowns and dropups to the nav with a bit of markup and the [dropdowns JavaScript plugin](#).

```
<ul class="nav">
  <li class="dropdown">
    <a href="#" class="dropdown-toggle" data-toggle="dropdown">
      Account
      <b class="caret"></b>
    </a>
    <ul class="dropdown-menu">
      ...
    </ul>
  </li>
</ul>
```

Visit the [JavaScript dropdowns documentation](#) for more markup and information on calling dropdowns.

## Text

Wrap strings of text in an element with `.navbar-text`, usually on a `<p>` tag for proper leading and color.

---

# Optional display variations

Fix the navbar to the top or bottom of the viewport with an additional class on the outermost div, `.navbar`.

## Fixed to top

Add `.navbar-fixed-top` and remember to account for the hidden area underneath it by adding at least 40px `padding` to the `<body>`. Be sure to add this after the core Bootstrap CSS and before the optional responsive CSS.

Title  
Home  
Link  
Link

```
<div class="navbar navbar-fixed-top">
  ...
</div>
```

## Fixed to bottom

Add `.navbar-fixed-bottom` instead.

Title  
Home  
Link  
Link

```
<div class="navbar navbar-fixed-bottom">
  ...
</div>
```

## Static top navbar

Create a full-width navbar that scrolls away with the page by adding `.navbar-static-top`. Unlike the `.navbar-fixed-top` class, you do not need to change any padding on the `body`.

Title  
Home  
Link  
Link

```
<div class="navbar navbar-static-top">
  ...
</div>
```

# Responsive navbar

To implement a collapsing responsive navbar, wrap your navbar content in a containing div, `.nav-collapse.collapse`, and add the navbar toggle button, `.btn-navbar`.

Title

Home

Link

Link

Dropdown

Link

Dropdown

```
<div class="navbar">
  <div class="navbar-inner">
    <div class="container">

      <!-- .btn-navbar is used as the toggle for collapsed navbar content -->
      <a class="btn btn-navbar" data-toggle="collapse" data-target=".nav-collapse">
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </a>

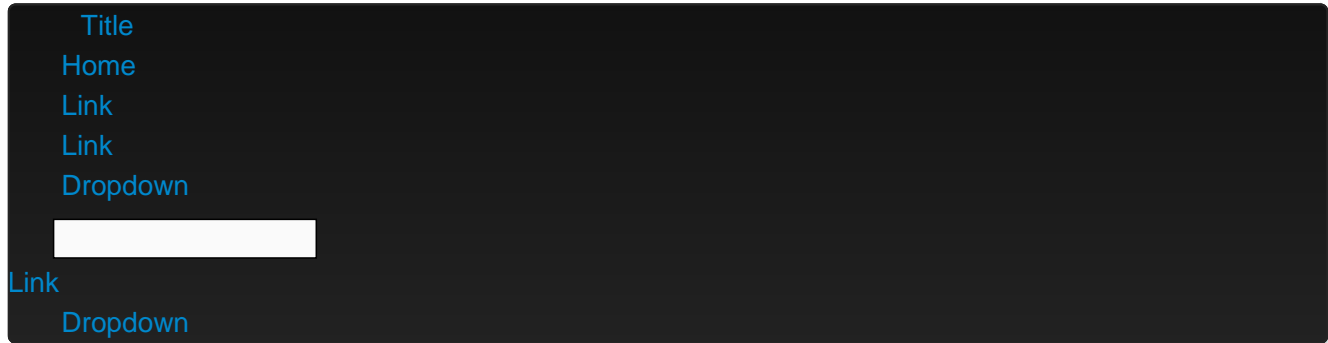
      <!-- Be sure to leave the brand out there if you want it shown -->
      <a class="brand" href="#">Project name</a>

      <!-- Everything you want hidden at 940px or less, place within here -->
      <div class="nav-collapse collapse">
        <!-- .nav, .navbar-search, .navbar-form, etc -->
        <div>

        </div>
      </div>
    </div>
  </div>
</div>
```

# Inverted variation

Modify the look of the navbar by adding `.navbar-inverse`.



```
<div class="navbar navbar-inverse">
  ...
</div>
```

# Breadcrumbs

## Examples

A single example shown as it might be displayed across multiple pages.

- Home
- Home / Library
- Home / Library / Data

```
<ul class="breadcrumb">
  <li><a href="#">Home</a> <span class="divider"></span></li>
  <li><a href="#">Library</a> <span class="divider"></span></li>
  <li class="active">Data</li>
</ul>
```

# Pagination

Two options for paging through content

## Standard pagination

Simple pagination inspired by Rdio, great for apps and search results. The large block is hard to miss, easily scalable, and provides large click areas.

- «
- 1
- 2
- 3
- 4
- 5
- »

```
<div class="pagination">
  <ul>
    <li><a href="#">Prev</a></li>
    <li><a href="#">1</a></li>
    <li><a href="#">2</a></li>
    <li><a href="#">3</a></li>
    <li><a href="#">4</a></li>
    <li><a href="#">Next</a></li>
  </ul>
</div>
```

## Options

### Disabled and active states

Links are customizable for different circumstances. Use `.disabled` for unclickable links and `.active` to indicate the current page.

- «
- 1
- 2
- 3
- 4
- 5
- »

```
<div class="pagination">
  <ul>
    <li class="disabled"><a href="#">Prev</a></li>
    <li class="active"><a href="#">1</a></li>
    ...
  </ul>
</div>
```

You can optionally swap out active or disabled anchors for spans to remove click functionality while retaining intended styles.

```
<div class="pagination">
  <ul>
    <li class="disabled"><span>Prev</span></li>
    <li class="active"><span>1</span></li>
    ...
  </ul>
</div>
```

## Sizes

Fancy larger or smaller pagination? Add `.pagination-large`, `.pagination-small`, or `.pagination-mini` for additional sizes.

- «
  - 1
  - 2
  - 3
  - 4
  - 5
  - »
- 
- «
  - 1
  - 2
  - 3
  - 4
  - 5
  - »

- «
- 1
- 2
- 3
- 4
- 5
- »

- «
- 1
- 2
- 3
- 4
- 5
- »

```
<div class="pagination pagination-large">
  <ul>
    ...
  </ul>
</div>

<div class="pagination">
  <ul>
    ...
  </ul>
</div>

<div class="pagination pagination-small">
  <ul>
    ...
  </ul>
</div>

<div class="pagination pagination-mini">
  <ul>
    ...
  </ul>
</div>
```

# Alignment

Add one of two optional classes to change the alignment of pagination links: `.pagination-centered` and `.pagination-right`.

- «
- 1
- 2
- 3
- 4
- 5
- »

```
<div class="pagination pagination-centered">
  ...
</div>
```

- «
- 1
- 2
- 3
- 4
- 5
- »

```
<div class="pagination pagination-right">
  ...
</div>
```

# Pager

Quick previous and next links for simple pagination implementations with light markup and styles. It's great for simple sites like blogs or magazines.

## Default example

By default, the pager centers links.

Previous  
Next

```
<ul class="pager">
  <li><a href="#">Previous</a></li>
  <li><a href="#">Next</a></li>
</ul>
```



# Aligned links

Alternatively, you can align each link to the sides:

[← Older](#)  
[Newer →](#)

```
<ul class="pager">  
  <li class="previous">  
    <a href="#">&larr; Older</a>  
  </li>  
  <li class="next">  
    <a href="#">Newer &rarr;</a>  
  </li>  
</ul>
```

# Optional disabled state

Pager links also use the general `.disabled` utility class from the pagination.

[← Older](#)  
[Newer →](#)

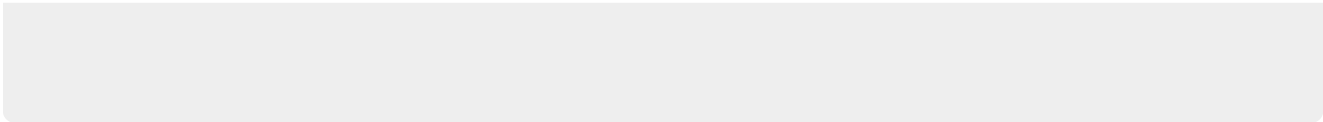
```
<ul class="pager">  
  <li class="previous disabled">  
    <a href="#">&larr; Older</a>  
  </li>  
  ...  
</ul>
```

# Labels and badges

## Labels

Labels	Markup
Default	<code>&lt;span class="label"&gt;Default&lt;/span&gt;</code>

Labels	Markup
Success	<code>&lt;span class="label label-success"&gt;Success&lt;/span&gt;</code>
Warning	<code>&lt;span class="label label-warning"&gt;Warning&lt;/span&gt;</code>
Important	<code>&lt;span class="label label-important"&gt;Important&lt;/span&gt;</code>
Info	<code>&lt;span class="label label-info"&gt;Info&lt;/span&gt;</code>
Inverse	<code>&lt;span class="label label-inverse"&gt;Inverse&lt;/span&gt;</code>



```
<div class="hero-unit">
  <h1>Heading</h1>
  <p>Tagline</p>
  <p>
    <a class="btn btn-primary btn-large">
      Learn more
    </a>
  </p>
</div>
```

# Page header

A simple shell for an `h1` to appropriately space out and segment sections of content on a page. It can utilize the `h1`'s default `small`, element as well most other components (with additional styles).

## Example page header Subtext for header

```
<div class="page-header">
  <h1>Example page header <small>Subtext for header</small></h1>
</div>
```

# Thumbnails Grids of images, videos, text, and more

## Default thumbnails

By default, Bootstrap's thumbnails are designed to showcase linked images with minimal required markup.

260 x 180

260 x 180

260 x 180

260 x 180

## Highly customizable

With a bit of extra markup, it's possible to add any kind of HTML content like headings, paragraphs, or buttons into thumbnails.

300 x 200

Thumbnail labelCras justo odio, dapibus ac facilisis in, egestas

eget quam. Donec id elit non mi porta gravida at eget metus. Nullam id dolor id nibh ultricies vehicula ut id elit.

Action Action

A large gray square placeholder for a thumbnail image, with the text "300 x 200" centered in a light gray font.

Thumbnail labelCras justo odio, dapibus ac facilisis in, egestas

eget quam. Donec id elit non mi porta gravida at eget metus. Nullam id dolor id nibh ultricies vehicula ut id elit.

Action Action

A large gray square placeholder for a thumbnail image, with the text "300 x 200" centered in a light gray font.

Thumbnail labelCras justo odio, dapibus ac facilisis in, egestas

eget quam. Donec id elit non mi porta gravida at eget metus. Nullam id dolor id nibh ultricies vehicula ut id elit.

Action Action

## Why use thumbnails

Thumbnails (previously `.media-grid` up until v1.4) are great for grids of photos or videos, image search results, retail products, portfolios, and much more. They can be links or static content.

## Simple, flexible markup

Thumbnail markup is simple—a `ul` with any number of `li` elements is all that is required. It's also super flexible, allowing for any type of content with just a bit more markup to wrap your contents.

## Uses grid column sizes

Lastly, the thumbnails component uses existing grid system classes—like `.span2` or `.span3`—for control of thumbnail dimensions.

## Markup

As mentioned previously, the required markup for thumbnails is light and straightforward. Here's a look at the default setup **for linked images**:

```
<ul class="thumbnails">
  <li class="span4">
    <a href="#" class="thumbnail">
      
    </a>
  </li>
  ...
</ul>
```

For custom HTML content in thumbnails, the markup changes slightly. To allow block level content anywhere, we swap the `<a>` for a `<div>` like so:

```
<ul class="thumbnails">
  <li class="span4">
    <div class="thumbnail">
      
      <h3>Thumbnail label</h3>
      <p>Thumbnail caption...</p>
    </div>
  </li>
  ...
</ul>
```

## More examples

Explore all your options with the various grid classes available to you. You can also mix and match different sizes.



360 x 270



260 x 120

160 x 120

260 x 120

160 x 120

# Alerts

Styles for success, warning, and error messages

## Default alert

Wrap any text and an optional dismiss button in `.alert` for a basic warning alert message.

× **Warning!** Best check yo self, you're not looking too good.

```
<div class="alert">
  <button type="button" class="close" data-dismiss="alert">×</button>
  <strong>Warning!</strong> Best check yo self, you're not looking too good.
</div>
```

## Dismiss buttons

Mobile Safari and Mobile Opera browsers, in addition to the `data-dismiss="alert"` attribute, require an `href="#"` for the dismissal of alerts when using an `<a>` tag.

```
<a href="#" class="close" data-dismiss="alert">×</a>
```

Alternatively, you may use a `<button>` element with the data attribute, which we have opted to do for our docs. When using `<button>`, you must include `type="button"` or your forms may not submit.

```
<button type="button" class="close" data-dismiss="alert">×</button>
```

# Dismiss alerts via JavaScript

Use the [alerts jQuery plugin](#) for quick and easy dismissal of alerts.

## Options

For longer messages, increase the padding on the top and bottom of the alert wrapper by adding `.alert-block`.

×

### Warning!

Best check yo self, you're not looking too good. Nulla vitae elit libero, a pharetra augue. Praesent commodo cursus magna, vel scelerisque nisl consectetur et.

```
<div class="alert alert-block">
  <button type="button" class="close" data-dismiss="alert">×</button>
  <h4>Warning!</h4>
  Best check yo self, you're not...
</div>
```

## Contextual alternatives

Add optional classes to change an alert's connotation.

### Error or danger

× Oh snap! Change a few things up and try submitting again.

```
<div class="alert alert-error">
  ...
</div>
```



# Success

× **Well done!** You successfully read this important alert message.

```
<div class="alert alert-success">
  ...
</div>
```

# Information

× **Heads up!** This alert needs your attention, but it's not super important.

```
<div class="alert alert-info">
  ...
</div>
```

# Progress bars

For loading, redirecting, or action status

## Examples and markup

### Basic

Default progress bar with a vertical gradient.



```
<div class="progress">
  <div class="bar" style="width: 60%;"></div>
</div>
```

# Striped

Uses a gradient to create a striped effect. Not available in IE7-8.



```
<div class="progress progress-striped">
  <div class="bar" style="width: 20%;"></div>
</div>
```

# Animated

Add `.active` to `.progress-striped` to animate the stripes right to left. Not available in all versions of IE.



```
<div class="progress progress-striped active">
  <div class="bar" style="width: 40%;"></div>
</div>
```

# Stacked

Place multiple bars into the same `.progress` to stack them.



```
<div class="progress">
  <div class="bar bar-success" style="width: 35%;"></div>
  <div class="bar bar-warning" style="width: 20%;"></div>
  <div class="bar bar-danger" style="width: 10%;"></div>
</div>
```

# Options

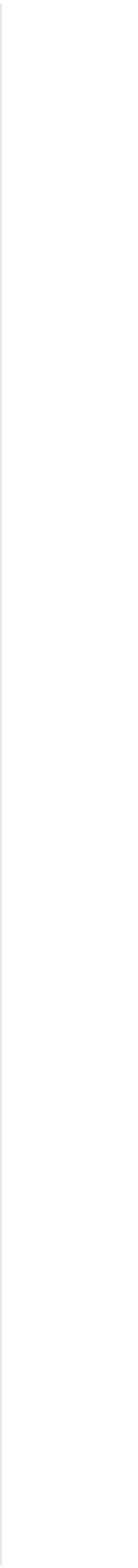
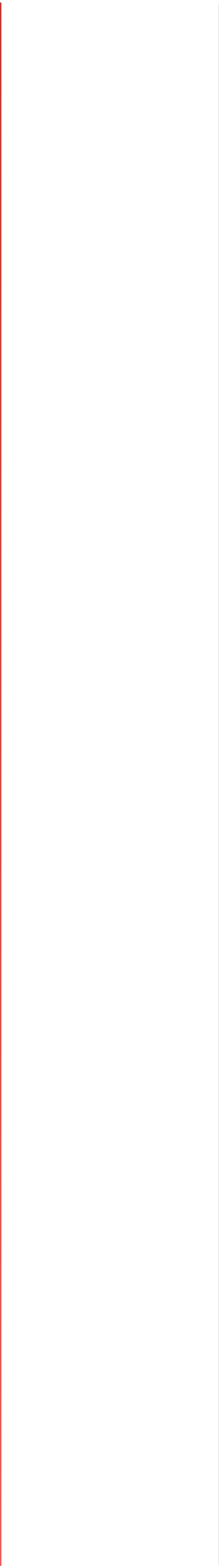
## Additional colors

Progress bars use some of the same button and alert classes for consistent styles.











```
<div class="progress progress-info">
  <div class="bar" style="width: 20%"></div>
</div>

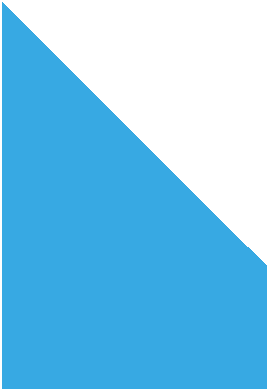
<div class="progress progress-success">
  <div class="bar" style="width: 40%"></div>
</div>

<div class="progress progress-warning">
  <div class="bar" style="width: 60%"></div>
</div>

<div class="progress progress-danger">
  <div class="bar" style="width: 80%"></div>
</div>
```

# Striped bars

Similar to the solid colors, we have varied striped progress bars.









```
<div class="progress progress-info progress-striped">
  <div class="bar" style="width: 20%"></div>
</div>

<div class="progress progress-success progress-striped">
  <div class="bar" style="width: 40%"></div>
</div>

<div class="progress progress-warning progress-striped">
  <div class="bar" style="width: 60%"></div>
</div>

<div class="progress progress-danger progress-striped">
  <div class="bar" style="width: 80%"></div>
</div>
```

# Browser support

Progress bars use CSS3 gradients, transitions, and animations to achieve all their effects. These features are not supported in IE7-9 or older versions of Firefox.

Versions earlier than Internet Explorer 10 and Opera 12 do not support animations.

# Media object

Abstract object styles for building various types of components (like blog comments, Tweets, etc) that feature a left- or right-aligned image alongside textual content.

# Default example

The default media allow to float a media object (images, video, audio) to the left or right of a content block.



## Media heading

Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin commodo. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis. Fusce condimentum nunc ac nisi vulputate fringilla. Donec lacinia congue felis in faucibus.



## Media heading

Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin commodo. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis. Fusce condimentum nunc ac nisi vulputate fringilla. Donec lacinia congue felis in faucibus.



## Media heading

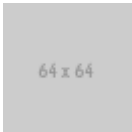
Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin commodo. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis. Fusce condimentum nunc ac nisi vulputate fringilla. Donec lacinia congue felis in faucibus.

```
<div class="media">
  <a class="pull-left" href="#">
    
  </a>
  <div class="media-body">
    <h4 class="media-heading">Media heading</h4>
    ...

    <!-- Nested media object -->
    <div class="media">
      ...
    </div>
  </div>
</div>
```

# Media list

With a bit of extra markup, you can use media inside list (useful for comment threads or articles lists).



Media headingCras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante

sollicitudin commodo. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis.



Nested media headingCras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante

sollicitudin commodo. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis.

64 x 64

Nested media headingCras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante

sollicitudin commodo. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis.

64 x 64

Nested media headingCras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante  
sollicitudin commodo. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis.

64 x 64

Media headingCras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante

sollicitudin commodo. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis.

```
<ul class="media-list">
  <li class="media">
    <a class="pull-left" href="#">
      
    </a>
    <div class="media-body">
      <h4 class="media-heading">Media heading</h4>
      ...

      <!-- Nested media object -->
      <div class="media">
        ...
      </div>
    </div>
  </li>
</ul>
```

## Miscellaneous

Lightweight utility components

## Wells

Use the well as a simple effect on an element to give it an inset effect.

Look, I'm in a well!

```
<div class="well">
  ...
</div>
```

## Optional classes

Control padding and rounded corners with two optional modifier classes.

Look, I'm in a well!

```
<div class="well well-large">
  ...
</div>
```

Look, I'm in a well!

```
<div class="well well-small">
  ...
</div>
```

## Close icon

Use the generic close icon for dismissing content like modals and alerts.

×

```
<button class="close">&times;</button>
```

iOS devices require an href="#" for click events if you would rather use an anchor.

```
<a class="close" href="#">&times;</a>
```

## Helper classes

Simple, focused classes for small display or behavior tweaks.



## .pull-left

Float an element left

```
class="pull-left"
```

```
.pull-left {  
  float: left;  
}
```

## .pull-right

Float an element right

```
class="pull-right"
```

```
.pull-right {  
  float: right;  
}
```

## .muted

Change an element's color to `#999`

```
class="muted"
```

```
.muted {  
  color: #999;  
}
```

## .clearfix

Clear the `float` on any element

```
class="clearfix"
```

```
.clearfix {
  *zoom: 1;
  &:before,
  &:after {
    display: table;
    content: "";
  }
  &:after {
    clear: both;
  }
}
```

[Back to top](#)

Designed and built with all the love in the world by [@mdo](#) and [@fat](#).

Code licensed under [Apache License v2.0](#), documentation under [CC BY 3.0](#).

[Glyphicons Free](#) licensed under [CC BY 3.0](#).

- [Blog](#)
- .
- [Issues](#)
- .
- [Roadmap and changelog](#)