Search...

Getting started

Layout

Content

Components

Alerts

Badge

Breadcrumb

Buttons

Button group

Card

Carousel

Collapse

Dropdowns

Forms

Input group

Jumbotron

List group

Media object

Modal

Navs

Navbar

Navbar

View on GitHub

How it works
Supported content
- Brand
- Nav
- Forms
- Text

Color schemes
Containers
Placement
Scrolling
Responsive behaviors
- Toggler
- External content

Documentation and examples for Bootstrap’s powerful, responsive navigation header, the navbar. Includes support for branding, navigation, collapse plugin, and more.

How it works

Here’s what you need to know before getting started with the navbar:

- Navbars require a wrapping `.navbar` with `.navbar-expand{-sm|-md|-lg|-xl}` for responsive collapsing and color scheme classes.

- Navbars and their contents are fluid by default. Use optional containers to limit their horizontal width.
- Use our spacing and flex utility classes for controlling spacing and alignment within navbars.
- Navbars are responsive by default, but you can easily modify them to change that. Responsive behavior depends on our Collapse JavaScript plugin.
- Navbars are hidden by default when printing. Force them to be printed by adding `.d-print` to the `.navbar`. See the display utility class.
- Ensure accessibility by using a `<nav>` element or, if using a more generic element such as a `<div>`, add a `role="navigation"` to every navbar to explicitly identify it as a landmark region for users of assistive technologies.

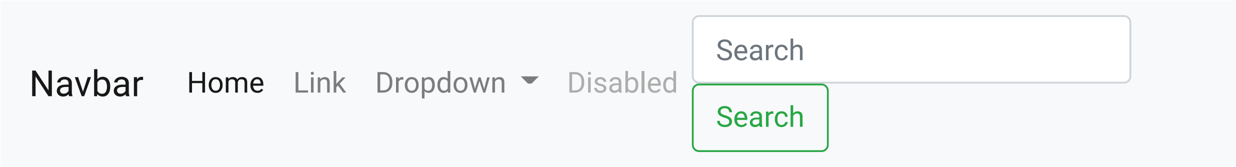
> The animation effect of this component is dependent on the `prefers-reduced-motion` media query. See the reduced motion section of our accessibility documentation.

Supported content

Navbars come with built-in support for a handful of sub-components. Choose from the following as needed:

- `.navbar-brand` for your company, product, or project name.
- `.navbar-nav` for a full-height and lightweight navigation (including support for dropdowns).
- `.navbar-toggler` for use with our collapse plugin and other navigation toggling behaviors.
- `.form-inline` for any form controls and actions.
- `.navbar-text` for adding vertically centered strings of text.
- `.collapse.navbar-collapse` for grouping and hiding navbar contents by a parent breakpoint.

Here’s an example of all the sub-components included in a responsive light-themed navbar that automatically collapses at the `lg` (large) breakpoint.



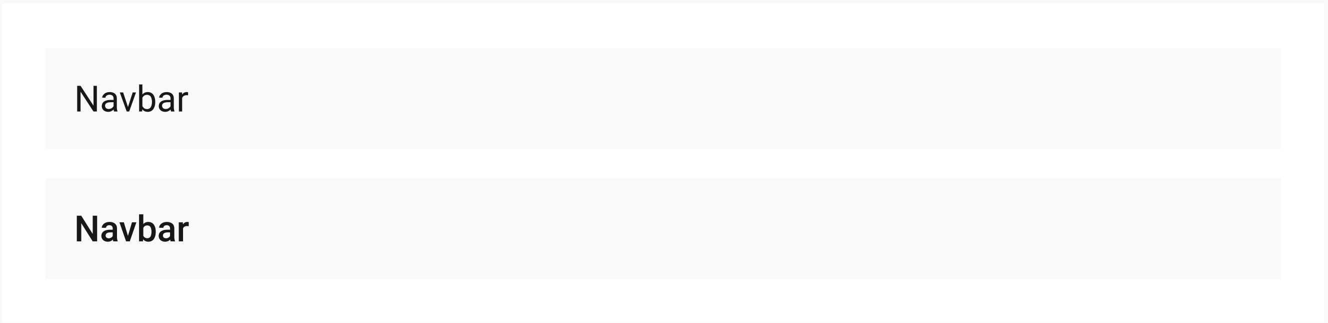
Copy

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="#">Navbar</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target='
    <span class="navbar-toggler-icon"></span>
  </button>

  <div class="collapse navbar-collapse" id="navbarSupportedContent">
    <ul class="navbar-nav mr-auto">
      <li class="nav-item active">
        <a class="nav-link" href="#">Home <span class="sr-only">(current)</span></a
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Link</a>
      </li>
      <li class="nav-item dropdown">
        <a class="nav-link dropdown-toggle" href="#" role="button" data-toggle="dro
          Dropdown
        </a>
        <div class="dropdown-menu">
          <a class="dropdown-item" href="#">Action</a>
          <a class="dropdown-item" href="#">Another action</a>
          <div class="dropdown-divider"></div>
          <a class="dropdown-item" href="#">Something else here</a>
        </div>
      </li>
      <li class="nav-item">
        <a class="nav-link disabled">Disabled</a>
      </li>
    </ul>
    <form class="form-inline my-2 my-lg-0">
      <input class="form-control mr-sm-2" type="search" placeholder="Search" aria-l
      <button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search</bu
    </form>
  </div>
</nav>
```

This example uses color (`bg-light`) and spacing (`my-2`, `my-lg-0`, `mr-sm-0`, `my-sm-0`) utility classes.

Brand

The `.navbar-brand` can be applied to most elements, but an anchor works best, as some elements might require utility classes or custom styles.



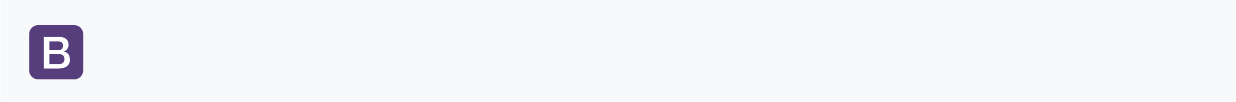
Navbar

Navbar

Copy

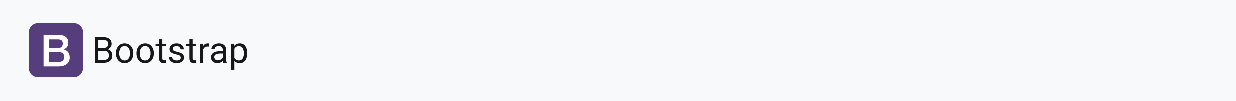
```
<!-- As a link -->
<nav class="navbar navbar-light bg-light">
  <a class="navbar-brand" href="#">Navbar</a>
</nav>

<!-- As a heading -->
<nav class="navbar navbar-light bg-light">
  <span class="navbar-brand mb-0 h1">Navbar</span>
</nav>
```

Adding images to the `.navbar-brand` will likely always require custom styles or utilities to properly size. Here are some examples to demonstrate.



Copy

```
<!-- Just an image -->
<nav class="navbar navbar-light bg-light">
  <a class="navbar-brand" href="#">
    <img src="/docs/4.6/assets/brand/bootstrap-solid.svg" width="30" height="30" a
  </a>
</nav>
```



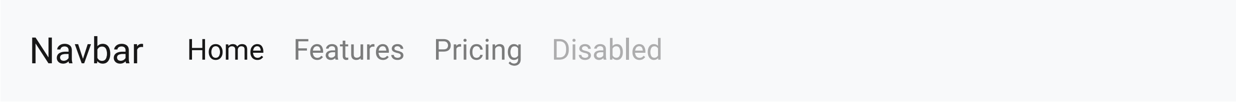
Copy

```
<!-- Image and text -->
<nav class="navbar navbar-light bg-light">
  <a class="navbar-brand" href="#">
    <img src="/docs/4.6/assets/brand/bootstrap-solid.svg" width="30" height="30" c
    Bootstrap
  </a>
</nav>
```

Nav

Navbar navigation links build on our `.nav` options with their own modifier class and require the use of toggler classes for proper responsive styling. **Navigation in navbars will also grow to occupy as much horizontal space as possible** to keep your navbar contents securely aligned.

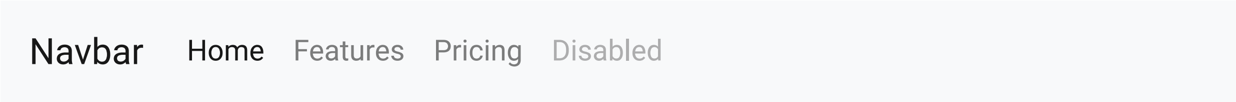
Active states—with `.active`—to indicate the current page can be applied directly to `.nav-link`s or their immediate parent `.nav-item`s.



Copy

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="#">Navbar</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target='
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarNav">
    <ul class="navbar-nav">
      <li class="nav-item active">
        <a class="nav-link" href="#">Home <span class="sr-only">(current)</span></
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Features</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Pricing</a>
      </li>
      <li class="nav-item">
        <a class="nav-link disabled">Disabled</a>
      </li>
    </ul>
  </div>
</nav>
```

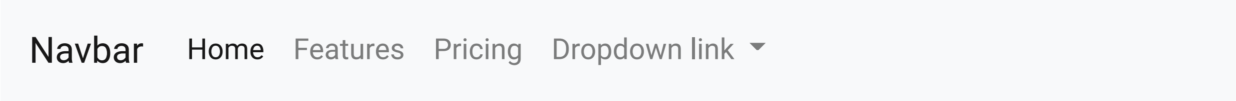
And because we use classes for our navs, you can avoid the list-based approach entirely if you like.



Copy

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="#">Navbar</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target='
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
    <div class="navbar-nav">
      <a class="nav-link active" href="#">Home <span class="sr-only">(current)</spa
      <a class="nav-link" href="#">Features</a>
      <a class="nav-link" href="#">Pricing</a>
      <a class="nav-link disabled">Disabled</a>
    </div>
  </div>
</nav>
```

You can also use dropdowns in your navbar. Dropdown menus require a wrapping element for positioning, so be sure to use separate and nested elements for `.nav-item` and `.nav-link` as shown below.



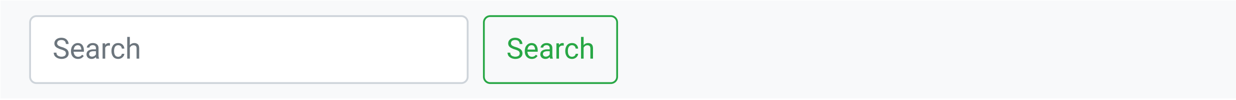
Copy

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="#">Navbar</a>
```

```
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-target='
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNavDropdown">
      <ul class="navbar-nav">
        <li class="nav-item active">
          <a class="nav-link" href="#">Home <span class="sr-only">(current)</span></a
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Features</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Pricing</a>
        </li>
        <li class="nav-item dropdown">
          <a class="nav-link dropdown-toggle" href="#" role="button" data-toggle="drc
            Dropdown link
          </a>
          <div class="dropdown-menu">
            <a class="dropdown-item" href="#">Action</a>
            <a class="dropdown-item" href="#">Another action</a>
            <a class="dropdown-item" href="#">Something else here</a>
          </div>
        </li>
      </ul>
    </div>
  </nav>
```

Forms

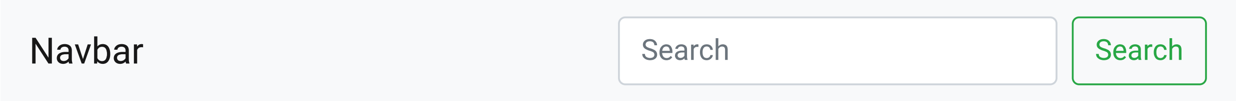
Place various form controls and components within a navbar with `.form-inline`.



Copy

```
<nav class="navbar navbar-light bg-light">
  <form class="form-inline">
    <input class="form-control mr-sm-2" type="search" placeholder="Search" aria-lab
    <button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search</butt
  </form>
</nav>
```

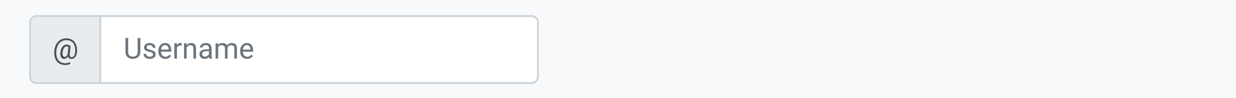
Immediate child elements of `.navbar` use flex layout and will default to `justify-content: space-between`. Use additional flex utilities as needed to adjust this behavior.



Copy

```
<nav class="navbar navbar-light bg-light">
  <a class="navbar-brand">Navbar</a>
  <form class="form-inline">
    <input class="form-control mr-sm-2" type="search" placeholder="Search" aria-lab
    <button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search</butt
  </form>
</nav>
```

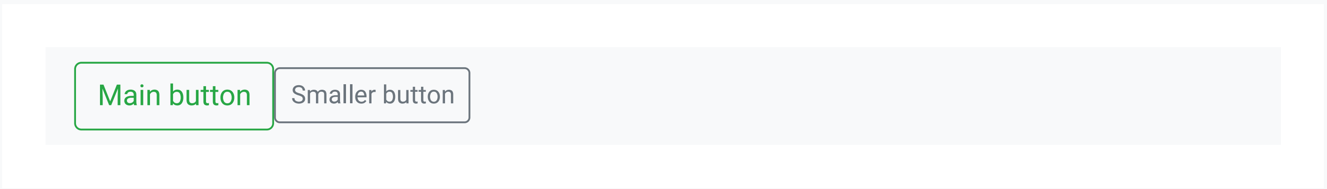
Input groups work, too:



Copy

```
<nav class="navbar navbar-light bg-light">
  <form class="form-inline">
    <div class="input-group">
      <div class="input-group-prepend">
        <span class="input-group-text" id="basic-addon1">@</span>
      </div>
      <input type="text" class="form-control" placeholder="Username" aria-label="Us
    </div>
  </form>
</nav>
```

Various buttons are supported as part of these navbar forms, too. This is also a great reminder that vertical alignment utilities can be used to align different sized elements.

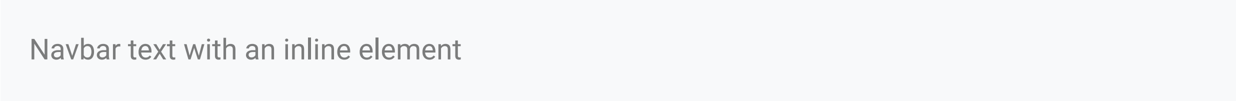


Copy

```
<nav class="navbar navbar-light bg-light">
  <form class="form-inline">
    <button class="btn btn-outline-success" type="button">Main button</button>
    <button class="btn btn-sm btn-outline-secondary" type="button">Smaller button</
  </form>
</nav>
```

Text

Navbars may contain bits of text with the help of `.navbar-text`. This class adjusts vertical alignment and horizontal spacing for strings of text.

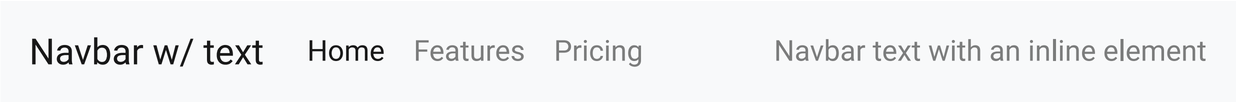


Navbar text with an inline element

Copy

```
<nav class="navbar navbar-light bg-light">
  <span class="navbar-text">
    Navbar text with an inline element
  </span>
</nav>
```

Mix and match with other components and utilities as needed.



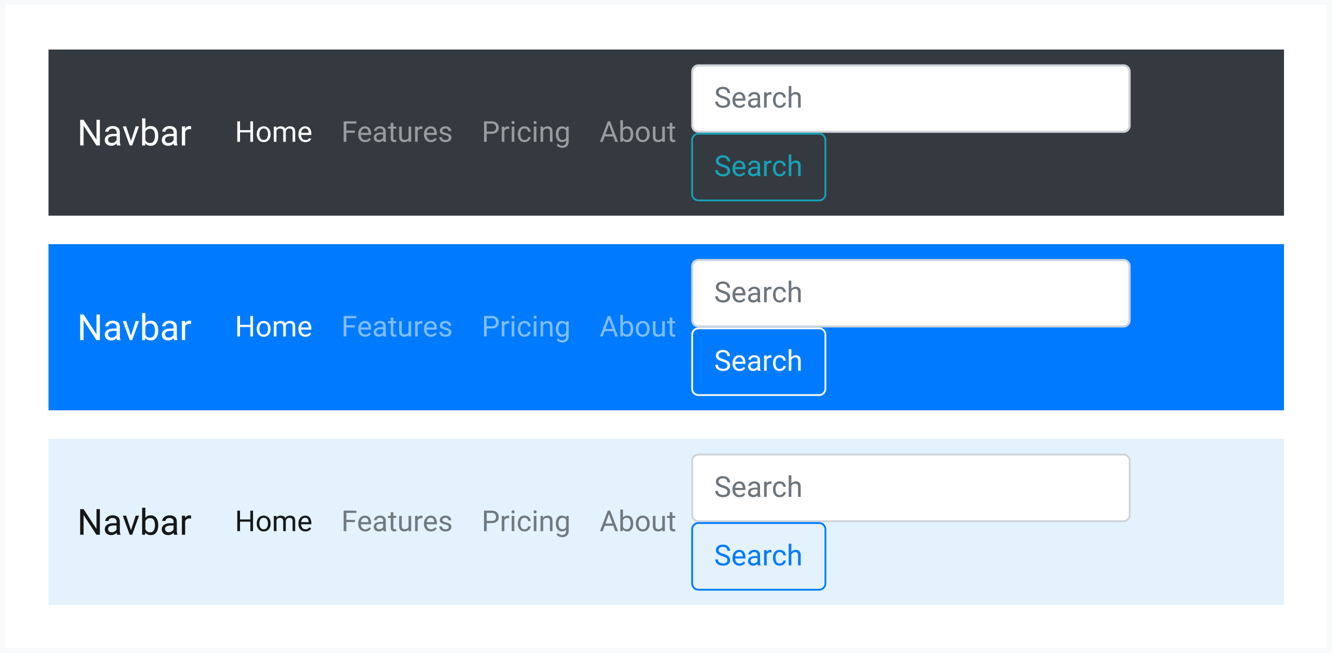
Navbar text with an inline element

Copy

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="#">Navbar w/ text</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target='
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarText">
    <ul class="navbar-nav mr-auto">
      <li class="nav-item active">
        <a class="nav-link" href="#">Home <span class="sr-only">(current)</span></a
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Features</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Pricing</a>
      </li>
  </ul>
  <span class="navbar-text">
   Navbar text with an inline element
  </span>
 </div>
</nav>
```

Color schemes

Theming the navbar has never been easier thanks to the combination of theming classes and `background-color` utilities. Choose from `.navbar-light` for use with light background colors, or `.navbar-dark` for dark background colors. Then, customize with `.bg-*` utilities.



Copy

```html
<nav class="navbar navbar-dark bg-dark">
  <!-- Navbar content -->
</nav>

<nav class="navbar navbar-dark bg-primary">
  <!-- Navbar content -->
</nav>

<nav class="navbar navbar-light" style="background-color: #e3f2fd;">
  <!-- Navbar content -->
</nav>
```

Containers

Although it’s not required, you can wrap a navbar in a `.container` to center it on a page. Or you can add a container inside the `.navbar` to only center the contents of a fixed or static top navbar.

Navbar

Copy

```html
<div class="container">
  <nav class="navbar navbar-expand-lg navbar-light bg-light">
    <a class="navbar-brand" href="#">Navbar</a>
  </nav>
</div>
```

When the container is within your navbar, its horizontal padding is removed at breakpoints lower than your specified `.navbar-expand{-sm|-md|-lg|-xl}` class. This ensures we’re not doubling up on padding unnecessarily on lower viewports when your navbar is collapsed.

Navbar

Copy

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <div class="container">
    <a class="navbar-brand" href="#">Navbar</a>
  </div>
</nav>
```

Placement

Use our position utilities to place navbars in non-static positions. Choose from fixed to the top, fixed to the bottom, or stickied to the top (scrolls with the page until it reaches the top, then stays there). Fixed navbars use `position: fixed`, meaning they’re pulled from the normal flow of the DOM and may require custom CSS (e.g., `padding-top` on the `<body>`) to prevent overlap with other elements.

Also note that **`.sticky-top` uses `position: sticky`, which isn’t fully supported in every browser**.

Default

Copy

```
<nav class="navbar navbar-light bg-light">
  <a class="navbar-brand" href="#">Default</a>
</nav>
```

Fixed top

Copy

```
<nav class="navbar fixed-top navbar-light bg-light">
  <a class="navbar-brand" href="#">Fixed top</a>
</nav>
```

Fixed bottom

Copy

```
<nav class="navbar fixed-bottom navbar-light bg-light">
  <a class="navbar-brand" href="#">Fixed bottom</a>
</nav>
```

Sticky top

Copy

```
<nav class="navbar sticky-top navbar-light bg-light">
  <a class="navbar-brand" href="#">Sticky top</a>
```

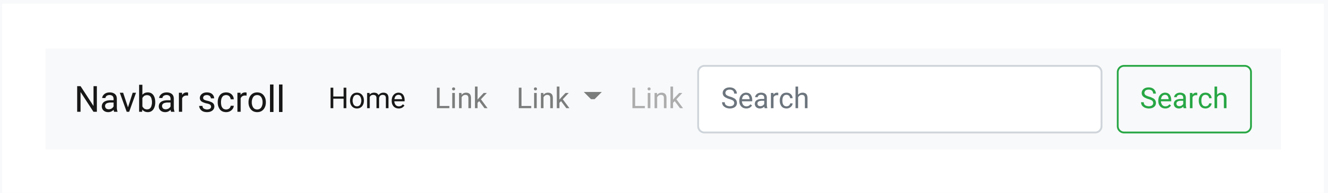
```
</nav>
```

Scrolling

Add `.navbar-nav-scroll` to a `.navbar-collapse` (or other navbar sub-component) to enable vertical scrolling within the toggleable contents of a collapsed navbar. By default, scrolling kicks in at `75vh` (or 75% of the viewport height), but you can override that with inline or custom styles. At larger viewports when the navbar is expanded, content will appear as it does in a default navbar.

Please note that this behavior comes with a potential drawback of `overflow`—when setting `overflow-y: auto` (required to scroll the content here), `overflow-x` is the equivalent of `auto`, which will crop some horizontal content.

Here's an example navbar using `.navbar-nav-scroll` with `style="max-height: 100px;"`, with some extra margin utilities for optimum spacing.



Copy

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="#">Navbar scroll</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target='
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarScroll">
    <ul class="navbar-nav mr-auto my-2 my-lg-0 navbar-nav-scroll" style="max-heigh
      <li class="nav-item active">
        <a class="nav-link" href="#">Home <span class="sr-only">(current)</span></a
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Link</a>
      </li>
      <li class="nav-item dropdown">
        <a class="nav-link dropdown-toggle" href="#" role="button" data-toggle="dr
          Link
        </a>
        <ul class="dropdown-menu">
          <li><a class="dropdown-item" href="#">Action</a></li>
          <li><a class="dropdown-item" href="#">Another action</a></li>
          <li><hr class="dropdown-divider"></li>
          <li><a class="dropdown-item" href="#">Something else here</a></li>
        </ul>
      </li>
      <li class="nav-item">
        <a class="nav-link disabled">Link</a>
      </li>
    </ul>
    <form class="d-flex">
      <input class="form-control mr-2" type="search" placeholder="Search" aria-labe
      <button class="btn btn-outline-success" type="submit">Search</button>
    </form>
  </div>
</nav>
```

Responsive behaviors

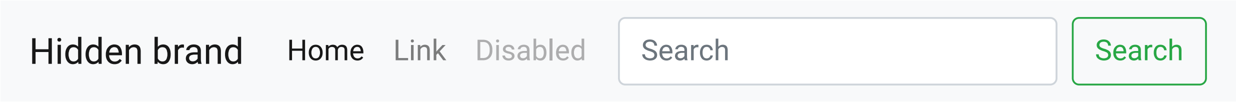
Navbars can use `.navbar-toggler`, `.navbar-collapse`, and `.navbar-expand{-sm|-md|-lg|-xl}` classes to determine when their content collapses behind a button. In combination with other utilities, you can easily choose when to show or hide particular elements.

For navbars that never collapse, add the `.navbar-expand` class on the navbar. For navbars that always collapse, don’t add any `.navbar-expand` class.

Toggler

Navbar togglers are left-aligned by default, but should they follow a sibling element like a `.navbar-brand`, they'll automatically be aligned to the far right. Reversing your markup will reverse the placement of the toggler. Below are examples of different toggle styles.

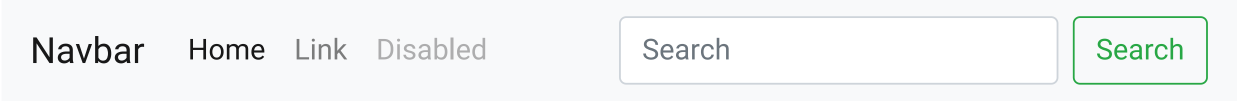
With no `.navbar-brand` shown at the smallest breakpoint:



Copy

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target='
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarTogglerDemo01">
    <a class="navbar-brand" href="#">Hidden brand</a>
    <ul class="navbar-nav mr-auto mt-2 mt-lg-0">
      <li class="nav-item active">
        <a class="nav-link" href="#">Home <span class="sr-only">(current)</span></a
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Link</a>
      </li>
      <li class="nav-item">
        <a class="nav-link disabled">Disabled</a>
      </li>
    </ul>
    <form class="form-inline my-2 my-lg-0">
      <input class="form-control mr-sm-2" type="search" placeholder="Search" aria-l
      <button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search</bu
    </form>
  </div>
</nav>
```

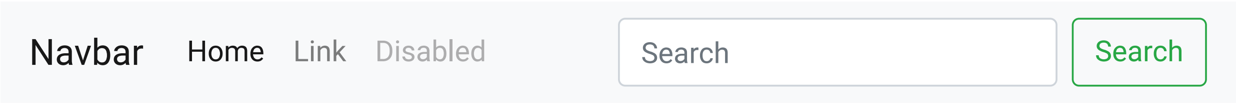
With a brand name shown on the left and toggler on the right:



Copy

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="#">Navbar</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target='
    <span class="navbar-toggler-icon"></span>
  </button>

  <div class="collapse navbar-collapse" id="navbarTogglerDemo02">
    <ul class="navbar-nav mr-auto mt-2 mt-lg-0">
      <li class="nav-item active">
        <a class="nav-link" href="#">Home <span class="sr-only">(current)</span></
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Link</a>
      </li>
      <li class="nav-item">
        <a class="nav-link disabled">Disabled</a>
      </li>
    </ul>
    <form class="form-inline my-2 my-lg-0">
      <input class="form-control mr-sm-2" type="search" placeholder="Search">
      <button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search</bu
    </form>
  </div>
</nav>
```

With a toggler on the left and brand name on the right:



Copy

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target='
    <span class="navbar-toggler-icon"></span>
  </button>
  <a class="navbar-brand" href="#">Navbar</a>

  <div class="collapse navbar-collapse" id="navbarTogglerDemo03">
    <ul class="navbar-nav mr-auto mt-2 mt-lg-0">
      <li class="nav-item active">
        <a class="nav-link" href="#">Home <span class="sr-only">(current)</span></a
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Link</a>
      </li>
      <li class="nav-item">
        <a class="nav-link disabled">Disabled</a>
      </li>
    </ul>
    <form class="form-inline my-2 my-lg-0">
      <input class="form-control mr-sm-2" type="search" placeholder="Search" aria-l
      <button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search</bu
    </form>
  </div>
</nav>
```

External content

Sometimes you want to use the collapse plugin to trigger a container element for content that structurally sits outside of the `.navbar` . Because our plugin works on the `id` and `data-target` matching, that's easily done!

Copy

```
<div class="fixed-top">
  <div class="collapse" id="navbarToggleExternalContent">
    <div class="bg-dark p-4">
      <h5 class="text-white h4">Collapsed content</h5>
      <span class="text-muted">Toggleable via the navbar brand.</span>
    </div>
  </div>
  <nav class="navbar navbar-dark bg-dark">
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-targe
      <span class="navbar-toggler-icon"></span>
    </button>
  </nav>
</div>
```

When you do this, we recommend including additional JavaScript to move the focus programmatically to the container when it is opened. Otherwise, keyboard users and users of assistive technologies will likely have a hard time finding the newly revealed content - particularly if the container that was opened comes *before* the toggler in the document's structure. We also recommend making sure that the toggler has the `aria-controls` attribute, pointing to the `id` of the content container. In theory, this allows assistive technology users to jump directly from the toggler to the container it controls–but support for this is currently quite patchy.