

There's a newer version of Bootstrap 4!

Navs

Documentation and examples for how to use Bootstrap’s included navigation components.

Base nav

Navigation available in Bootstrap share general markup and styles, from the base `.nav` class to the active and disabled states. Swap modifier classes to switch between each style.

The base `.nav` component is built with flexbox and provide a strong foundation for building all types of navigation components. It includes some style overrides (for working with lists), some link padding for larger hit areas, and basic disabled styling.

The base `.nav` component does not include any `.active` state. The following examples include the class, mainly to demonstrate that this particular class does not trigger any special styling.

Active Link Link Disabled

```
<ul class="nav">
  <li class="nav-item">
    <a class="nav-link active" href="#">Active</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Link</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Link</a>
  </li>
  <li class="nav-item">
    <a class="nav-link disabled" href="#">Disabled</a>
  </li>
</ul>
```

Copy

Classes are used throughout, so your markup can be super flexible. Use ``s like above, or roll your own with say a `<nav>` element. Because the `.nav` uses `display: flex`, the nav links behave the same as nav items would, but without the extra markup.

Active Link Link Disabled

```
<nav class="nav">
  <a class="nav-link active" href="#">Active</a>
  <a class="nav-link" href="#">Link</a>
  <a class="nav-link" href="#">Link</a>
  <a class="nav-link disabled" href="#">Disabled</a>
</nav>
```

Copy

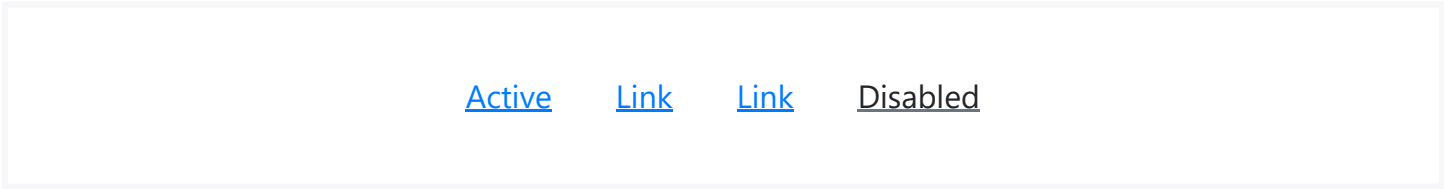
Available styles

Change the style of `.navs` component with modifiers and utilities. Mix and match as needed, or build your own.

Horizontal alignment

Change the horizontal alignment of your nav with [flexbox utilities](#). By default, navs are left-aligned, but you can easily change them to center or right aligned.

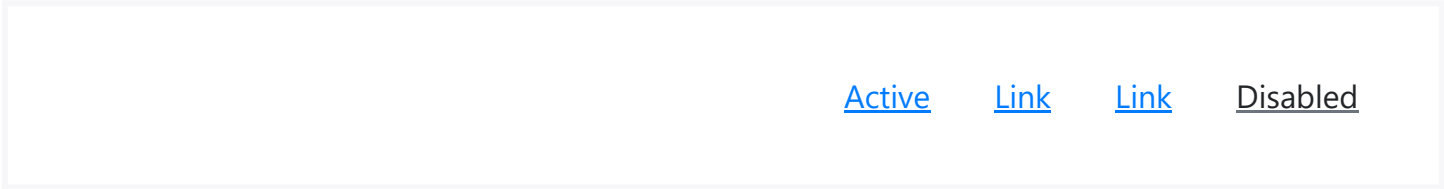
Centered with `.justify-content-center`:



Copy

```
<ul class="nav justify-content-center">
  <li class="nav-item">
    <a class="nav-link active" href="#">Active</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Link</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Link</a>
  </li>
  <li class="nav-item">
    <a class="nav-link disabled" href="#">Disabled</a>
  </li>
</ul>
```

Right-aligned with `.justify-content-end`:

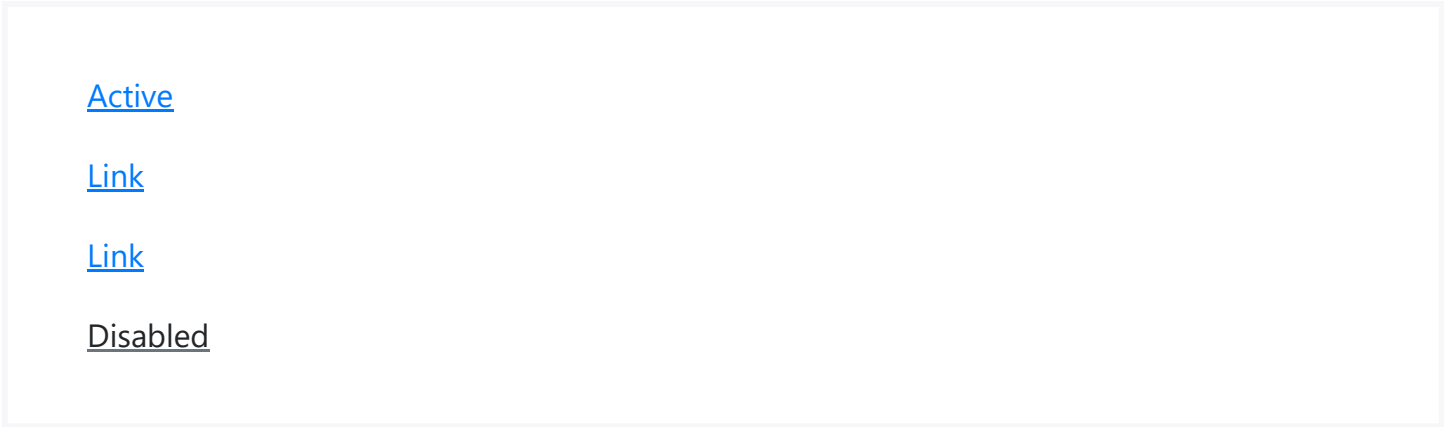


Copy

```
<ul class="nav justify-content-end">
  <li class="nav-item">
    <a class="nav-link active" href="#">Active</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Link</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Link</a>
  </li>
  <li class="nav-item">
    <a class="nav-link disabled" href="#">Disabled</a>
  </li>
</ul>
```

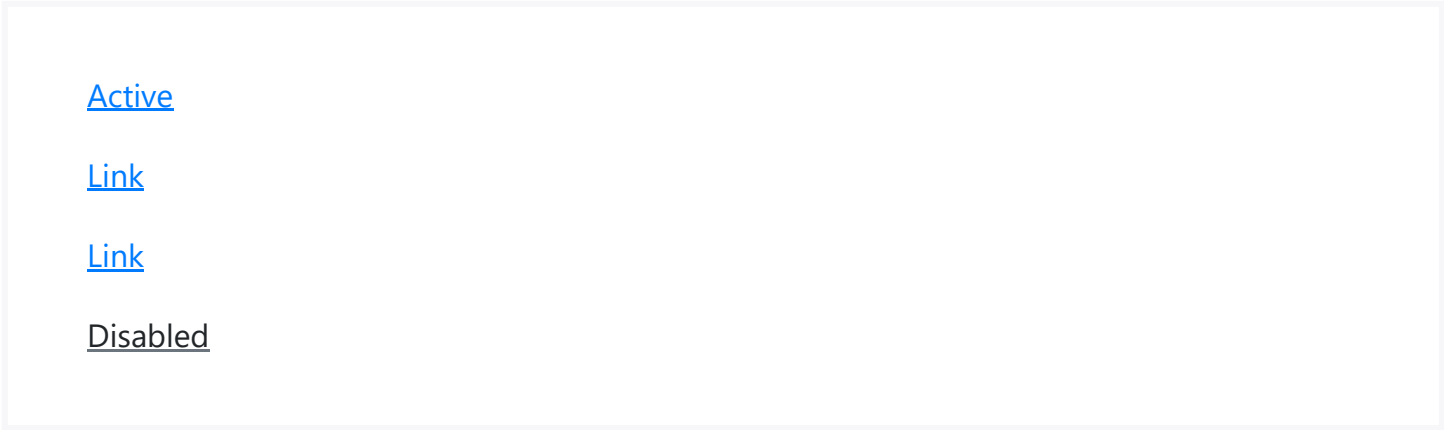
Vertical

Stack your navigation by changing the flex item direction with the `.flex-column` utility. Need to stack them on some viewports but not others? Use the responsive versions (e.g., `.flex-sm-column`).



```
<ul class="nav flex-column">
  <li class="nav-item">
    <a class="nav-link active" href="#">Active</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Link</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Link</a>
  </li>
  <li class="nav-item">
    <a class="nav-link disabled" href="#">Disabled</a>
  </li>
</ul>
```

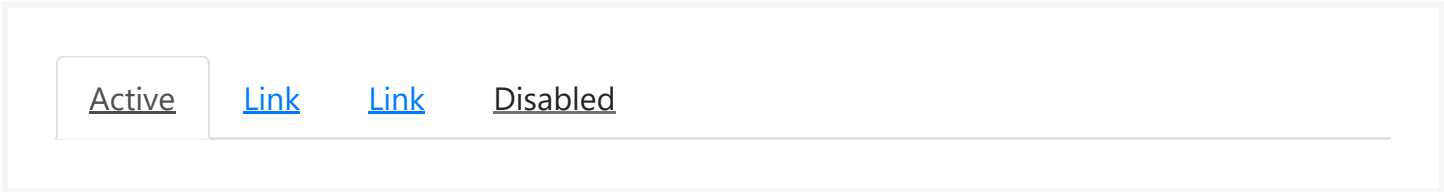
As always, vertical navigation is possible without ``s, too.



```
<nav class="nav flex-column">
  <a class="nav-link active" href="#">Active</a>
  <a class="nav-link" href="#">Link</a>
  <a class="nav-link" href="#">Link</a>
  <a class="nav-link disabled" href="#">Disabled</a>
</nav>
```

Tabs

Takes the basic nav from above and adds the `.nav-tabs` class to generate a tabbed interface. Use them to create tabbable regions with our [tab JavaScript plugin](#).



```
<ul class="nav nav-tabs">
  <li class="nav-item">
    <a class="nav-link active" href="#">Active</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Link</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Link</a>
  </li>
  <li class="nav-item">
    <a class="nav-link disabled" href="#">Disabled</a>
  </li>
</ul>
```

Pills

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



Copy

```
<ul class="nav nav-pills">
  <li class="nav-item">
    <a class="nav-link active" href="#">Active</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Link</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Link</a>
  </li>
  <li class="nav-item">
    <a class="nav-link disabled" href="#">Disabled</a>
  </li>
</ul>
```

Fill and justify

Force your `.nav`'s contents to extend the full available width one of two modifier classes. To proportionately fill all available space with your `.nav-items`, use `.nav-fill`. Notice that all horizontal space is occupied, but not every nav item has the same width.

ActiveLonger nav linkLinkDisabled

Copy

```
<ul class="nav nav-pills nav-fill">
  <li class="nav-item">
    <a class="nav-link active" href="#">Active</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Longer nav link</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Link</a>
  </li>
  <li class="nav-item">
    <a class="nav-link disabled" href="#">Disabled</a>
  </li>
</ul>
```

When using a `<nav>`-based navigation, be sure to include `.nav-item` on the anchors.

ActiveLinkLinkDisabled

Copy

```
<nav class="nav nav-pills nav-fill">
  <a class="nav-item nav-link active" href="#">Active</a>
  <a class="nav-item nav-link" href="#">Link</a>
  <a class="nav-item nav-link" href="#">Link</a>
  <a class="nav-item nav-link disabled" href="#">Disabled</a>
</nav>
```

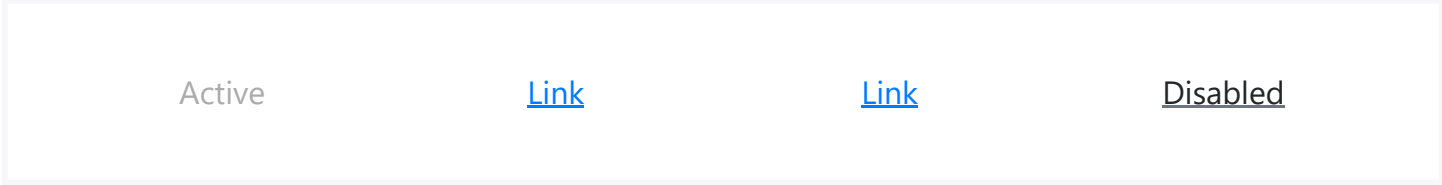
For equal-width elements, use `.nav-justified`. All horizontal space will be occupied by nav links, but unlike the `.nav-fill` above, every nav item will be the same width.

ActiveLonger nav linkLinkDisabled

Copy

```
<nav class="nav nav-pills nav-justified">
  <a class="nav-link active" href="#">Active</a>
  <a class="nav-link" href="#">Longer nav link</a>
  <a class="nav-link" href="#">Link</a>
  <a class="nav-link disabled" href="#">Disabled</a>
</nav>
```

Similar to the `.nav-fill` example using a `<nav>`-based navigation, be sure to include `.nav-item` on the anchors.

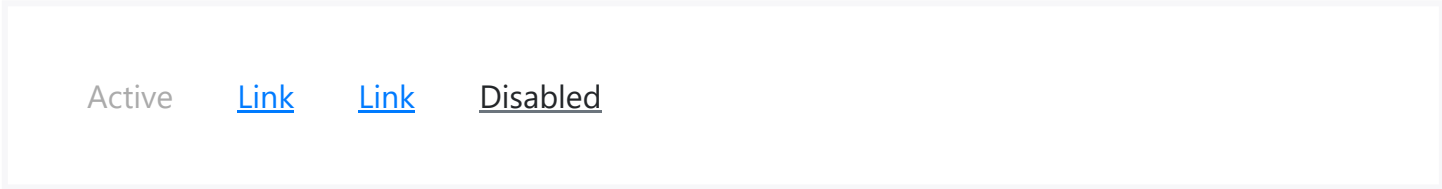


```
<nav class="nav nav-pills nav-justified">
  <a class="nav-item nav-link active" href="#">Active</a>
  <a class="nav-item nav-link" href="#">Link</a>
  <a class="nav-item nav-link" href="#">Link</a>
  <a class="nav-item nav-link disabled" href="#">Disabled</a>
</nav>
```

Copy

Working with flex utilities

If you need responsive nav variations, consider using a series of [flexbox utilities](#). While more verbose, these utilities offer greater customization across responsive breakpoints. In the example below, our nav will be stacked on the lowest breakpoint, then adapt to a horizontal layout that fills the available width starting from the small breakpoint.



```
<nav class="nav nav-pills flex-column flex-sm-row">
  <a class="flex-sm-fill text-sm-center nav-link active" href="#">Active</a>
  <a class="flex-sm-fill text-sm-center nav-link" href="#">Link</a>
  <a class="flex-sm-fill text-sm-center nav-link" href="#">Link</a>
  <a class="flex-sm-fill text-sm-center nav-link disabled" href="#">Disabled</a>
</nav>
```

Copy

Regarding accessibility

If you’re using navs to provide a navigation bar, be sure to add a `role="navigation"` to the most logical parent container of the ``, or wrap a `<nav>` element around the whole navigation. Do not add the role to the `` itself, as this would prevent it from being announced as an actual list by assistive technologies.

Note that navigation bars, even if visually styled as tabs with the `.nav-tabs` class, should **not** be given `role="tablist"`, `role="tab"` or `role="tabpanel"` attributes. These are only appropriate for dynamic tabbed interfaces, as described in the [WAI \(Web Accessibility Initiative\) ARIA \(Accessible Rich Internet Applications\) Authoring Practices](#). See [JavaScript behavior](#) for dynamic tabbed interfaces in this section for an example.

Using dropdowns

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

Tabs with dropdowns

ActiveDropdown▼LinkDisabled

Copy

```
<ul class="nav nav-tabs">
  <li class="nav-item">
    <a class="nav-link active" href="#">Active</a>
  </li>
  <li class="nav-item dropdown">
    <a class="nav-link dropdown-toggle" data-toggle="dropdown" href="#" role="button"
aria-haspopup="true" aria-expanded="false">Dropdown</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 class="dropdown-divider"></div>
      <a class="dropdown-item" href="#">Separated link</a>
    </div>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Link</a>
  </li>
  <li class="nav-item">
    <a class="nav-link disabled" href="#">Disabled</a>
  </li>
</ul>
```

Pills with dropdowns

ActiveDropdown▼LinkDisabled

Copy

```
<ul class="nav nav-pills">
  <li class="nav-item">
    <a class="nav-link active" href="#">Active</a>
  </li>
  <li class="nav-item dropdown">
    <a class="nav-link dropdown-toggle" data-toggle="dropdown" href="#" role="button"
aria-haspopup="true" aria-expanded="false">Dropdown</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 class="dropdown-divider"></div>
      <a class="dropdown-item" href="#">Separated link</a>
    </div>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Link</a>
  </li>
  <li class="nav-item">
    <a class="nav-link disabled" href="#">Disabled</a>
  </li>
</ul>
```

JavaScript behavior

Use the tab JavaScript plugin—include it individually or through the compiled `bootstrap.js` file—to extend our navigational tabs and pills to create tabbable panes of local content, even via dropdown menus.

If you’re building our JavaScript from source, it [requires util.js](#).

Dynamic tabbed interfaces, as described in the [WAI \(Web Accessibility Initiative\) ARIA \(Accessible Rich Internet Applications\) Authoring Practices](#), require `role="tablist"`, `role="tab"`, `role="tabpanel"`, and additional `aria-` attributes in order to convey their structure, functionality and current state to users of assistive technologies (such as screen readers).

Note that dynamic tabbed interfaces should *not* contain dropdown menus, as this causes both usability and accessibility issues. From a usability perspective, the fact that the currently displayed tab’s trigger element is not immediately visible (as it’s inside the closed dropdown menu) can cause confusion. From an accessibility point of view, there is currently no sensible way to map this sort of construct to a standard WAI ARIA pattern, meaning that it cannot be easily made understandable to users of assistive technologies.

HomeProfileContact

Raw denim you probably haven't heard of them jean shorts Austin. Nesciunt tofu stumptown aliqua, retro synth master cleanse. Mustache cliche tempor, williamsburg carles vegan helvetica. Reprehenderit butcher retro keffiyeh dreamcatcher synth. Cosby sweater eu banh mi, qui irure terry richardson ex squid. Aliquip placeat salvia cillum iphone. Seitan aliquip quis cardigan american apparel, butcher voluptate nisi qui.

Copy

```
<ul class="nav nav-tabs" id="myTab" role="tablist">
  <li class="nav-item">
    <a class="nav-link active" id="home-tab" data-toggle="tab" href="#home" role="tab"
aria-controls="home" aria-selected="true">Home</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" id="profile-tab" data-toggle="tab" href="#profile" role="tab"
aria-controls="profile" aria-selected="false">Profile</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" id="contact-tab" data-toggle="tab" href="#contact" role="tab"
aria-controls="contact" aria-selected="false">Contact</a>
  </li>
</ul>
<div class="tab-content" id="myTabContent">
  <div class="tab-pane fade show active" id="home" role="tabpanel" aria-
labelledby="home-tab">...</div>
  <div class="tab-pane fade" id="profile" role="tabpanel" aria-labelledby="profile-
tab">...</div>
  <div class="tab-pane fade" id="contact" role="tabpanel" aria-labelledby="contact-
tab">...</div>
</div>
```

To help fit your needs, this works with ``-based markup, as shown above, or with any arbitrary “roll your own” markup. Note that if you’re using `<nav>`, you shouldn’t add `role="tablist"` directly to it, as this would override the element’s native role as a navigation landmark. Instead, switch to an alternative element (in the exammple below, a simple `<div>`) and wrap the `<nav>` around it.

HomeProfileContact

Et et consectetur ipsum labore excepteur est proident excepteur ad velit occaecat qui minim occaecat veniam. Fugiat veniam incididunt anim aliqua enim pariatur veniam sunt est aute sit dolor anim. Velit non irure adipisicing aliqua ullamco irure incididunt irure non esse consectetur nostrud minim non minim occaecat. Amet duis do nisi duis veniam non est eiusmod tempor incididunt tempor dolor ipsum in qui sit. Exercitation mollit sit culpa nisi culpa non adipisicing reprehenderit do dolore. Duis reprehenderit occaecat anim ullamco ad duis occaecat ex.

Copy


```
<nav>
  <div class="nav nav-tabs" id="nav-tab" role="tablist">
    <a class="nav-item nav-link active" id="nav-home-tab" data-toggle="tab" href="#nav-home" role="tab" aria-controls="nav-home" aria-selected="true">Home</a>
    <a class="nav-item nav-link" id="nav-profile-tab" data-toggle="tab" href="#nav-profile" role="tab" aria-controls="nav-profile" aria-selected="false">Profile</a>
    <a class="nav-item nav-link" id="nav-contact-tab" data-toggle="tab" href="#nav-contact" role="tab" aria-controls="nav-contact" aria-selected="false">Contact</a>
  </div>
</nav>
<div class="tab-content" id="nav-tabContent">
  <div class="tab-pane fade show active" id="nav-home" role="tabpanel" aria-labelledby="nav-home-tab">...</div>
  <div class="tab-pane fade" id="nav-profile" role="tabpanel" aria-labelledby="nav-profile-tab">...</div>
  <div class="tab-pane fade" id="nav-contact" role="tabpanel" aria-labelledby="nav-contact-tab">...</div>
</div>
```

The tabs plugin also works with pills.

Home Profile Contact

Consequat occaecat ullamco amet non eiusmod nostrud dolore irure incididunt est duis anim sunt officia. Fugiat velit proident aliquip nisi incididunt nostrud exercitation proident est nisi. Irure magna elit commodo anim ex veniam culpa eiusmod id nostrud sit cupidatat in veniam ad. Eiusmod consequat eu adipisicing minim anim aliquip cupidatat culpa excepteur quis. Occaecat sit eu exercitation irure Lorem incididunt nostrud.

Copy

```
<ul class="nav nav-pills mb-3" id="pills-tab" role="tablist">
  <li class="nav-item">
    <a class="nav-link active" id="pills-home-tab" data-toggle="pill" href="#pills-home" role="tab" aria-controls="pills-home" aria-selected="true">Home</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" id="pills-profile-tab" data-toggle="pill" href="#pills-profile" role="tab" aria-controls="pills-profile" aria-selected="false">Profile</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" id="pills-contact-tab" data-toggle="pill" href="#pills-contact" role="tab" aria-controls="pills-contact" aria-selected="false">Contact</a>
  </li>
</ul>
<div class="tab-content" id="pills-tabContent">
  <div class="tab-pane fade show active" id="pills-home" role="tabpanel" aria-labelledby="pills-home-tab">...</div>
  <div class="tab-pane fade" id="pills-profile" role="tabpanel" aria-labelledby="pills-profile-tab">...</div>
  <div class="tab-pane fade" id="pills-contact" role="tabpanel" aria-labelledby="pills-contact-tab">...</div>
</div>
```

And with vertical pills.

Home
Profile
Messages
Settings

Cillum ad ut irure tempor velit nostrud occaecat ullamco aliqua anim Lorem sint. Veniam sint duis incididunt do esse magna mollit excepteur laborum qui. Id id reprehenderit sit est eu aliqua occaecat quis et velit excepteur laborum mollit dolore eiusmod. Ipsum dolor in occaecat commodo et voluptate minim reprehenderit mollit pariatur. Deserunt non laborum enim et cillum eu deserunt excepteur ea incididunt minim occaecat.


```
<div class="nav flex-column nav-pills" id="v-pills-tab" role="tablist" aria-orientation="vertical">
  <a class="nav-link active" id="v-pills-home-tab" data-toggle="pill" href="#v-pills-home" role="tab" aria-controls="v-pills-home" aria-selected="true">Home</a>
  <a class="nav-link" id="v-pills-profile-tab" data-toggle="pill" href="#v-pills-profile" role="tab" aria-controls="v-pills-profile" aria-selected="false">Profile</a>
  <a class="nav-link" id="v-pills-messages-tab" data-toggle="pill" href="#v-pills-messages" role="tab" aria-controls="v-pills-messages" aria-selected="false">Messages</a>
  <a class="nav-link" id="v-pills-settings-tab" data-toggle="pill" href="#v-pills-settings" role="tab" aria-controls="v-pills-settings" aria-selected="false">Settings</a>
</div>
<div class="tab-content" id="v-pills-tabContent">
  <div class="tab-pane fade show active" id="v-pills-home" role="tabpanel" aria-labelledby="v-pills-home-tab">...</div>
  <div class="tab-pane fade" id="v-pills-profile" role="tabpanel" aria-labelledby="v-pills-profile-tab">...</div>
  <div class="tab-pane fade" id="v-pills-messages" role="tabpanel" aria-labelledby="v-pills-messages-tab">...</div>
  <div class="tab-pane fade" id="v-pills-settings" role="tabpanel" aria-labelledby="v-pills-settings-tab">...</div>
</div>
```

Using data attributes

You can activate a tab or pill navigation without writing any JavaScript by simply specifying `data-toggle="tab"` or `data-toggle="pill"` on an element. Use these data attributes on `.nav-tabs` or `.nav-pills`.

```
<!-- Nav tabs -->
<ul class="nav nav-tabs" id="myTab" role="tablist">
  <li class="nav-item">
    <a class="nav-link active" id="home-tab" data-toggle="tab" href="#home" role="tab" aria-controls="home" aria-selected="true">Home</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" id="profile-tab" data-toggle="tab" href="#profile" role="tab" aria-controls="profile" aria-selected="false">Profile</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" id="messages-tab" data-toggle="tab" href="#messages" role="tab" aria-controls="messages" aria-selected="false">Messages</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" id="settings-tab" data-toggle="tab" href="#settings" role="tab" aria-controls="settings" aria-selected="false">Settings</a>
  </li>
</ul>

<!-- Tab panes -->
<div class="tab-content">
  <div class="tab-pane active" id="home" role="tabpanel" aria-labelledby="home-tab">...
</div>
  <div class="tab-pane" id="profile" role="tabpanel" aria-labelledby="profile-tab">...
</div>
  <div class="tab-pane" id="messages" role="tabpanel" aria-labelledby="messages-tab">...</div>
  <div class="tab-pane" id="settings" role="tabpanel" aria-labelledby="settings-tab">...</div>
</div>
```

Via JavaScript

Enable tabbable tabs via JavaScript (each tab needs to be activated individually):

Copy

```
$('#myTab a').on('click', function (e) {
  e.preventDefault()
  $(this).tab('show')
})
```

You can activate individual tabs in several ways:

Copy

```
$('#myTab a[href="#profile"]').tab('show') // Select tab by name
$('#myTab li:first-child a').tab('show') // Select first tab
$('#myTab li:last-child a').tab('show') // Select last tab
$('#myTab li:nth-child(3) a').tab('show') // Select third tab
```

Fade effect

To make tabs fade in, add `.fade` to each `.tab-pane`. The first tab pane must also have `.show` to make the initial content visible.

Copy

```
<div class="tab-content">
  <div class="tab-pane fade show active" id="home" role="tabpanel" aria-
labelledby="home-tab">...</div>
  <div class="tab-pane fade" id="profile" role="tabpanel" aria-labelledby="profile-
tab">...</div>
  <div class="tab-pane fade" id="messages" role="tabpanel" aria-labelledby="messages-
tab">...</div>
  <div class="tab-pane fade" id="settings" role="tabpanel" aria-labelledby="settings-
tab">...</div>
</div>
```

Methods

Asynchronous methods and transitions

All API methods are **asynchronous** and start a **transition**. They return to the caller as soon as the transition is started but **before it ends**. In addition, a method call on a **transitioning component will be ignored**.

[See our JavaScript documentation for more information.](#)

\$.tab

Activates a tab element and content container. Tab should have either a `data-target` or an `href` targeting a container node in the DOM.

Copy

Search...

[Getting started](#)

[Layout](#)

[Content](#)

[Components](#)

[Alerts](#)

[Badge](#)

[Breadcrumb](#)

[Buttons](#)

[Button group](#)

[Card](#)

[Carousel](#)

[Collapse](#)

[Dropdowns](#)

[Forms](#)

[Input group](#)

[Jumbotron](#)

[List group](#)

[Modal](#)

[Navs](#)

[Navbar](#)

[Pagination](#)

[Popovers](#)

[Progress](#)

[Scrollspy](#)

[Tooltips](#)

[Utilities](#)

[Extend](#)

```
<ul class="nav nav-tabs" id="myTab" role="tablist">
  <li class="nav-item">
    <a class="nav-link active" id="home-tab" data-toggle="tab" href="#home" role="tab"
aria-controls="home" aria-selected="true">Home</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" id="profile-tab" data-toggle="tab" href="#profile" role="tab"
aria-controls="profile" aria-selected="false">Profile</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" id="messages-tab" data-toggle="tab" href="#messages" role="tab"
aria-controls="messages" aria-selected="false">Messages</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" id="settings-tab" data-toggle="tab" href="#settings" role="tab"
aria-controls="settings" aria-selected="false">Settings</a>
  </li>
</ul>

<div class="tab-content">
  <div class="tab-pane active" id="home" role="tabpanel" aria-labelledby="home-tab">...
</div>
  <div class="tab-pane" id="profile" role="tabpanel" aria-labelledby="profile-tab">...
</div>
  <div class="tab-pane" id="messages" role="tabpanel" aria-labelledby="messages-
tab">...</div>
  <div class="tab-pane" id="settings" role="tabpanel" aria-labelledby="settings-
tab">...</div>
</div>

<script>
  $(function () {
    $('#myTab li:last-child a').tab('show')
  })
</script>
```

.tab(‘show’)

Selects the given tab and shows its associated pane. Any other tab that was previously selected becomes unselected and its associated pane is hidden. **Returns to the caller before the tab pane has actually been shown** (i.e. before the `shown.bs.tab` event occurs).

```
$( '#someTab' ).tab( 'show' )
```

Copy

.tab(‘dispose’)

Destroys an element’s tab.

Events

When showing a new tab, the events fire in the following order:

- 1. `hide.bs.tab` (on the current active tab)
- 2. `show.bs.tab` (on the to-be-shown tab)
- 3. `hidden.bs.tab` (on the previous active tab, the same one as for the `hide.bs.tab` event)
- 4. `shown.bs.tab` (on the newly-active just-shown tab, the same one as for the `show.bs.tab` event)

If no tab was already active, then the `hide.bs.tab` and `hidden.bs.tab` events will not be fired.

Event Type	Description
show.bs.tab	This event fires on tab show, but before the new tab has been shown. Use <code>event.target</code> and <code>event.relatedTarget</code> to target the active tab and the previous active tab (if available) respectively.

[Migration](#)

[About](#)

Event Type	Description
shown.bs.tab	This event fires on tab show after a tab has been shown. Use <code>event.target</code> and <code>event.relatedTarget</code> to target the active tab and the previous active tab (if available) respectively.
hide.bs.tab	This event fires when a new tab is to be shown (and thus the previous active tab is to be hidden). Use <code>event.target</code> and <code>event.relatedTarget</code> to target the current active tab and the new soon-to-be-active tab, respectively.
hidden.bs.tab	This event fires after a new tab is shown (and thus the previous active tab is hidden). Use <code>event.target</code> and <code>event.relatedTarget</code> to target the previous active tab and the new active tab, respectively.

```
$('a[data-toggle="tab"]').on('shown.bs.tab', function (e) {  
  e.target // newly activated tab  
  e.relatedTarget // previous active tab  
})
```

Copy