

## [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](#)

### [Pagination](#)

### [Popovers](#)

### [Progress](#)

### [Scrollspy](#)

### [Spinners](#)

### [Toasts](#)

### [Tooltips](#)

## [Utilities](#)

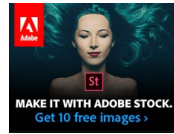
## [Extend](#)

## [Migration](#)

## [About](#)

# Scrollspy

Automatically update Bootstrap navigation or list group components based on scroll position to indicate which link is currently active in the viewport.



Limited time offer: Get 10 free Adobe Stock images.  
ads via Carbon

## How it works

Scrollspy has a few requirements to function properly:

- If you're building our JavaScript from source, it [requires util.js](#).
- It must be used on a Bootstrap [nav component](#) or [list group](#).
- Scrollspy requires `position: relative;` on the element you're spying on, usually the `<body>`.
- When spying on elements other than the `<body>`, be sure to have a `height` set and `overflow-y: scroll;` applied.
- Anchors (`<a>`) are required and must point to an element with that `id`.

When successfully implemented, your nav or list group will update accordingly, moving the `.active` class from one item to the next based on their associated targets.

## Example in navbar

Scroll the area below the navbar and watch the active class change. The dropdown items will be highlighted as well.

@fat

Ad leggings keytar, brunch id art party dolor labore. Pitchfork yr enim lo-fi before they sold out qui. Tumblr farm-to-table bicycle rights whatever. Anim keffiyeh carles cardigan. Velit seitan mcsweeney's photo booth 3 wolf moon irure. Cosby sweater lomo jean shorts, williamsburg hoodie minim qui you probably haven't heard of them et cardigan trust fund culpa biodiesel wes anderson aesthetic. Nihil tattooed accusamus, cred irony biodiesel keffiyeh artisan ullamco consequat.



Copy

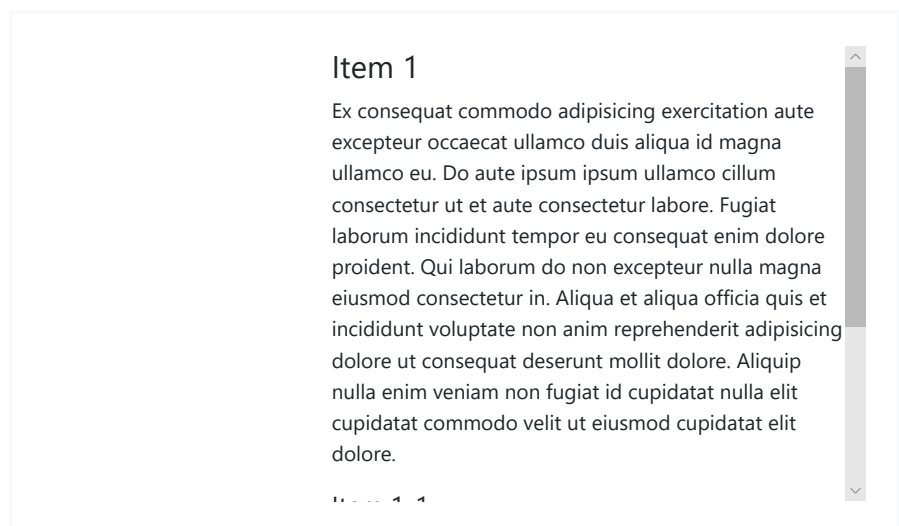
```

<nav id="navbar-example2" class="navbar navbar-light bg-light">
  <a class="navbar-brand" href="#">Navbar</a>
  <ul class="nav nav-pills">
    <li class="nav-item">
      <a class="nav-link" href="#fat">@fat</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#mdo">@mdo</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="#one">one</a>
        <a class="dropdown-item" href="#two">two</a>
        <div role="separator" class="dropdown-divider"></div>
        <a class="dropdown-item" href="#three">three</a>
      </div>
    </li>
  </ul>
</nav>
<div data-spy="scroll" data-target="#navbar-example2" data-offset="0">
  <h4 id="fat">@fat</h4>
  <p>...</p>
  <h4 id="mdo">@mdo</h4>
  <p>...</p>
  <h4 id="one">one</h4>
  <p>...</p>
  <h4 id="two">two</h4>
  <p>...</p>
  <h4 id="three">three</h4>
  <p>...</p>
</div>

```

## Example with nested nav

Scrollspy also works with nested `.navs`. If a nested `.nav` is `.active`, its parents will also be `.active`. Scroll the area next to the navbar and watch the active class change.



Copy

```

<nav id="navbar-example3" class="navbar navbar-light bg-light">
  <a class="navbar-brand" href="#">Navbar</a>
  <nav class="nav nav-pills flex-column">
    <a class="nav-link" href="#item-1">Item 1</a>
    <nav class="nav nav-pills flex-column">
      <a class="nav-link ml-3 my-1" href="#item-1-1">Item 1-1</a>
      <a class="nav-link ml-3 my-1" href="#item-1-2">Item 1-2</a>
    </nav>
    <a class="nav-link" href="#item-2">Item 2</a>
    <a class="nav-link" href="#item-3">Item 3</a>
    <nav class="nav nav-pills flex-column">
      <a class="nav-link ml-3 my-1" href="#item-3-1">Item 3-1</a>
      <a class="nav-link ml-3 my-1" href="#item-3-2">Item 3-2</a>
    </nav>
  </nav>
</nav>

<div data-spy="scroll" data-target="#navbar-example3" data-offset="0">
  <h4 id="item-1">Item 1</h4>
  <p>...</p>
  <h5 id="item-1-1">Item 1-1</h5>
  <p>...</p>
  <h5 id="item-1-2">Item 1-2</h5>
  <p>...</p>
  <h4 id="item-2">Item 2</h4>
  <p>...</p>
  <h4 id="item-3">Item 3</h4>
  <p>...</p>
  <h5 id="item-3-1">Item 3-1</h5>
  <p>...</p>
  <h5 id="item-3-2">Item 3-2</h5>
  <p>...</p>
</div>

```

## Example with list-group

Scrollspy also works with `.list-groups`. Scroll the area next to the list group and watch the active class change.

<a href="#">Item 1</a>	<h3>Item 1</h3> <p>Ex consequat commodo adipisicing exercitation aute excepteur occaecat ullamco duis aliqua id magna ullamco eu. Do aute ipsum ipsum ullamco cillum consectetur ut et aute consectetur labore. Fugiat laborum incididunt tempor eu consequat enim dolore proident. Qui laborum do non excepteur nulla magna eiusmod consectetur in. Aliqua et aliqua officia quis et</p>
<a href="#">Item 2</a>	
<a href="#">Item 3</a>	
<a href="#">Item 4</a>	

Copy

```

<div id="list-example" class="list-group">
  <a class="list-group-item list-group-item-action" href="#list-item-1">Item 1</a>
  <a class="list-group-item list-group-item-action" href="#list-item-2">Item 2</a>
  <a class="list-group-item list-group-item-action" href="#list-item-3">Item 3</a>
  <a class="list-group-item list-group-item-action" href="#list-item-4">Item 4</a>
</div>
<div data-spy="scroll" data-target="#list-example" data-offset="0"
class="scrollspy-example">
  <h4 id="list-item-1">Item 1</h4>
  <p>...</p>
  <h4 id="list-item-2">Item 2</h4>
  <p>...</p>
  <h4 id="list-item-3">Item 3</h4>
  <p>...</p>
  <h4 id="list-item-4">Item 4</h4>
  <p>...</p>
</div>

```

## Usage

### Via data attributes

To easily add scrollspy behavior to your topbar navigation, add `data-spy="scroll"` to the element you want to spy on (most typically this would be the `<body>`). Then add the `data-target` attribute with the ID or class of the parent element of any Bootstrap `.nav` component.

```

body {
  position: relative;
}

```

Copy

```

<body data-spy="scroll" data-target="#navbar-example">
  ...
  <div id="navbar-example">
    <ul class="nav nav-tabs" role="tablist">
      ...
    </ul>
  </div>
  ...
</body>

```

Copy

### Via JavaScript

After adding `position: relative;` in your CSS, call the scrollspy via JavaScript:

```

$('body').scrollspy({ target: '#navbar-example' })

```

Copy

#### Resolvable ID targets required

Navbar links must have resolvable id targets. For example, a `<a href="#home">home</a>` must correspond to something in the DOM like `<div id="home"></div>`.

#### Non-:visible target elements ignored

Target elements that are not [:visible according to jQuery](#) will be ignored and their corresponding nav items will never be highlighted.

## Methods

### `.scrollspy('refresh')`

When using scrollspy in conjunction with adding or removing of elements from the DOM, you'll need to call the refresh method like so:

```
$( '[data-spy="scroll"]' ).each(function () {  
    var $spy = $(this).scrollspy('refresh')  
})
```

Copy

### `.scrollspy('dispose')`

Destroys an element's scrollspy.

## Options

Options can be passed via data attributes or JavaScript. For data attributes, append the option name to `data-`, as in `data-offset=""`.

Name	Type	Default	Description
offset	number	10	Pixels to offset from top when calculating position of scroll.
method	string	auto	Finds which section the spied element is in. <code>auto</code> will choose the best method to get scroll coordinates. <code>offset</code> will use jQuery offset method to get scroll coordinates. <code>position</code> will use jQuery position method to get scroll coordinates.
target	string		Specifies element to apply Scrollspy plugin.

## Events

Event Type	Description
activate.bs.scrollspy	This event fires on the scroll element whenever a new item becomes activated by the scrollspy.

```
$( '[data-spy="scroll"]' ).on('activate.bs.scrollspy', function () {  
    // do something...  
})
```

Copy