List group

List groups are a flexible and powerful component for displaying a series of content. Modify and extend them to support just about any content within.

On this page

Basic example

The most basic list group is an unordered list with list items and the proper classes. Build upon it with the options that follow, or with your own CSS as needed.

An item

A second item

A third item

A fourth item

And a fifth one

```html
<ul class="list-group">
  <li class="list-group-item">An item</li>
  <li class="list-group-item">A second item</li>
  <li class="list-group-item">A third item</li>
  <li class="list-group-item">A fourth item</li>
  <li class="list-group-item">And a fifth one</li>
</ul>
```

Active items

Add `.active` to a `.list-group-item` to indicate the current active selection.

An active item

A second item

A third item

A fourth item

And a fifth one

```
<ul class="list-group">
  <li class="list-group-item active" aria-current="true">An active item</li>
  <li class="list-group-item">A second item</li>
  <li class="list-group-item">A third item</li>
  <li class="list-group-item">A fourth item</li>
  <li class="list-group-item">And a fifth one</li>
</ul>
```

Disabled items

Add `.disabled` to a `.list-group-item` to make it *appear* disabled. Note that some elements with `.disabled` will also require custom JavaScript to fully disable their click events (e.g., links).

A disabled item

A second item

A third item

A fourth item

And a fifth one

```
<ul class="list-group">
  <li class="list-group-item disabled" aria-disabled="true">A disabled item</li>
  <li class="list-group-item">A second item</li>
  <li class="list-group-item">A third item</li>
  <li class="list-group-item">A fourth item</li>
  <li class="list-group-item">And a fifth one</li>
</ul>
```

Links and buttons

Use `<a>`s or `<button>`s to create *actionable* list group items with hover, disabled, and active states by adding `.list-group-item-action`. We separate these pseudo-classes to ensure list groups made of non-interactive elements (like `<li>`s or `<div>`s) don't provide a click or tap affordance.

Be sure to **not use the standard `.btn` classes here**.

The current link item

A second link item

A third link item

A fourth link item

A disabled link item

```
<div class="list-group">
  <a href="#" class="list-group-item list-group-item-action active" aria-current="tr
    The current link item
  </a>
  <a href="#" class="list-group-item list-group-item-action">A second link item</a>
  <a href="#" class="list-group-item list-group-item-action">A third link item</a>
  <a href="#" class="list-group-item list-group-item-action">A fourth link item</a>
  <a href="#" class="list-group-item list-group-item-action disabled" tabindex="-1"
</div>
```

With `<button>`s, you can also make use of the `disabled` attribute instead of the `.disabled` class. Sadly, `<a>`s don't support the disabled attribute.

The current button

A second item

A third button item

A fourth button item

A disabled button item

```
<div class="list-group">
  <button type="button" class="list-group-item list-group-item-action active" aria-c
```

```
    The current button
  </button>
  <button type="button" class="list-group-item list-group-item-action">A second item
  <button type="button" class="list-group-item list-group-item-action">A third butto
  <button type="button" class="list-group-item list-group-item-action">A fourth butt
  <button type="button" class="list-group-item list-group-item-action" disabled>A di
```

Flush

Add `.list-group-flush` to remove some borders and rounded corners to render list group items edge-to-edge in a parent container (e.g., cards).

An item

A second item

A third item

A fourth item

And a fifth one

```
<ul class="list-group list-group-flush">
  <li class="list-group-item">An item</li>
  <li class="list-group-item">A second item</li>
  <li class="list-group-item">A third item</li>
  <li class="list-group-item">A fourth item</li>
  <li class="list-group-item">And a fifth one</li>
</ul>
```

Numbered

Add the `.list-group-numbered` modifier class (and optionally use an `<ol>` element) to opt into numbered list group items. Numbers are generated via CSS (as opposed to a `<ol>`s default browser styling) for better placement inside list group items and to allow for better customization.

Numbers are generated by `counter-reset` on the `<ol>`, and then styled and placed with a `::before` pseudo-element on the `<li>` with `counter-increment` and `content`.

1. Cras justo odio
2. Cras justo odio
3. Cras justo odio

```
<ol class="list-group list-group-numbered">
  <li class="list-group-item">Cras justo odio</li>
  <li class="list-group-item">Cras justo odio</li>
  <li class="list-group-item">Cras justo odio</li>
</ol>
```

These work great with custom content as well.

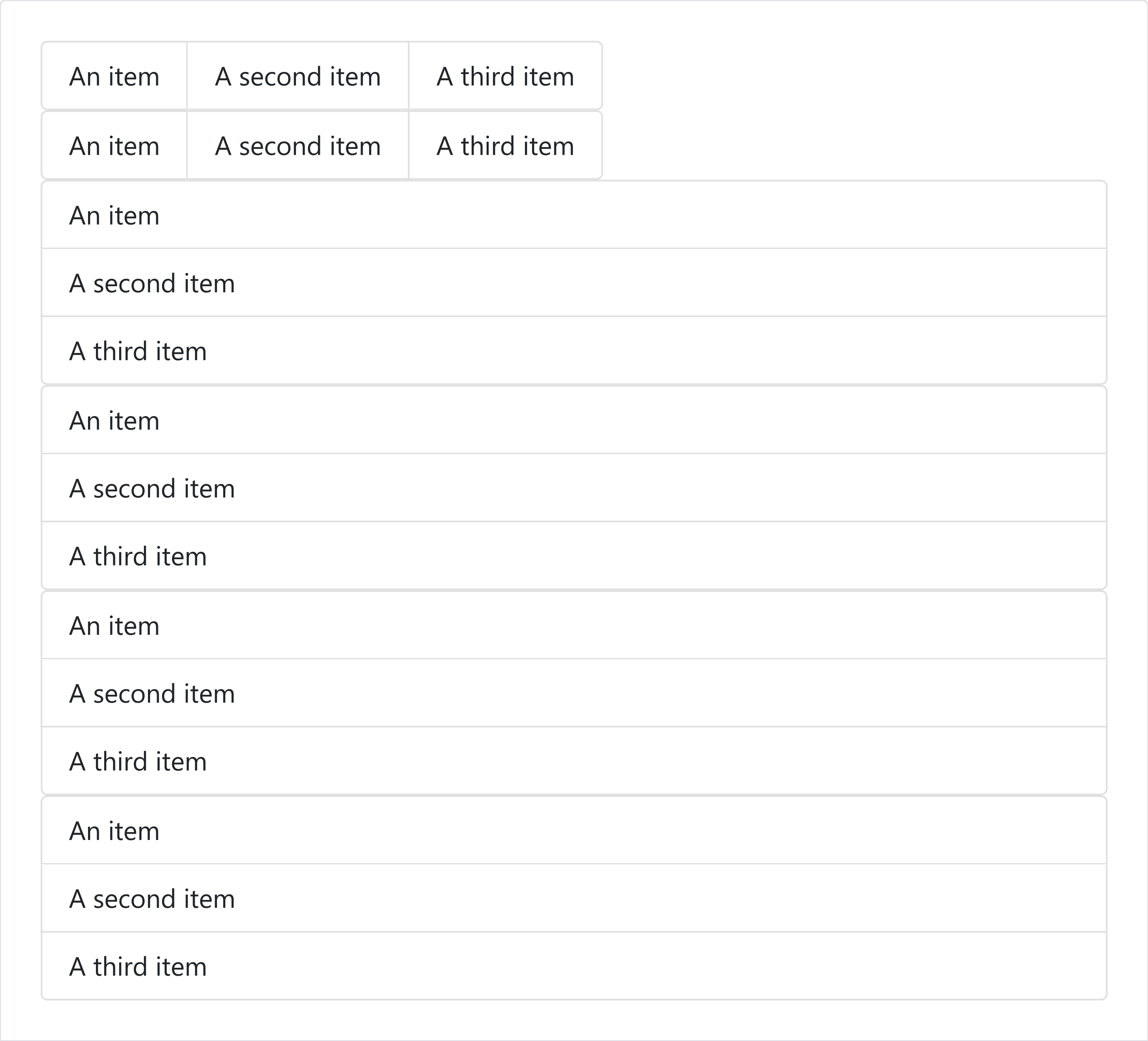
1. **Subheading** 14
   Cras justo odio
2. **Subheading** 14
   Cras justo odio
3. **Subheading** 14
   Cras justo odio

```
<ol class="list-group list-group-numbered">
  <li class="list-group-item d-flex justify-content-between align-items-start">
    <div class="ms-2 me-auto">
      <div class="fw-bold">Subheading</div>
      Cras justo odio
    </div>
    <span class="badge bg-primary rounded-pill">14</span>
  </li>
  <li class="list-group-item d-flex justify-content-between align-items-start">
    <div class="ms-2 me-auto">
      <div class="fw-bold">Subheading</div>
      Cras justo odio
    </div>
    <span class="badge bg-primary rounded-pill">14</span>
  </li>
  <li class="list-group-item d-flex justify-content-between align-items-start">
    <div class="ms-2 me-auto">
      <div class="fw-bold">Subheading</div>
      Cras justo odio
    </div>
    <span class="badge bg-primary rounded-pill">14</span>
  </li>
</ol>
```

Horizontal

Add `.list-group-horizontal` to change the layout of list group items from vertical to horizontal across all breakpoints. Alternatively, choose a responsive variant `.list-group-horizontal-{sm|md|lg|xl|xxl}` to make a list group horizontal starting at that breakpoint's `min-width`. Currently **horizontal list groups cannot be combined with flush list groups.**

ProTip: Want equal-width list group items when horizontal? Add `.flex-fill` to each list group item.



An item
A second item
A third item

An item
A second item
A third item

An item
A second item
A third item

An item
A second item
A third item

An item
A second item
A third item

An item
A second item
A third item

```
<ul class="list-group list-group-horizontal">
  <li class="list-group-item">An item</li>
  <li class="list-group-item">A second item</li>
  <li class="list-group-item">A third item</li>
</ul>
<ul class="list-group list-group-horizontal-sm">
  <li class="list-group-item">An item</li>
  <li class="list-group-item">A second item</li>
  <li class="list-group-item">A third item</li>
</ul>
<ul class="list-group list-group-horizontal-md">
  <li class="list-group-item">An item</li>
  <li class="list-group-item">A second item</li>
  <li class="list-group-item">A third item</li>
</ul>
<ul class="list-group list-group-horizontal-lg">
  <li class="list-group-item">An item</li>
  <li class="list-group-item">A second item</li>
  <li class="list-group-item">A third item</li>
</ul>
<ul class="list-group list-group-horizontal-xl">
  <li class="list-group-item">An item</li>
  <li class="list-group-item">A second item</li>
  <li class="list-group-item">A third item</li>
</ul>
<ul class="list-group list-group-horizontal-xxl">
  <li class="list-group-item">An item</li>
  <li class="list-group-item">A second item</li>
  <li class="list-group-item">A third item</li>
</ul>
```

Contextual classes

Use contextual classes to style list items with a stateful background and color.

A simple default list group item

A simple primary list group item

A simple secondary list group item

A simple success list group item

A simple danger list group item

A simple warning list group item

A simple info list group item

A simple light list group item

A simple dark list group item

```
<ul class="list-group">
  <li class="list-group-item">A simple default list group item</li>

  <li class="list-group-item list-group-item-primary">A simple primary list group it
  <li class="list-group-item list-group-item-secondary">A simple secondary list grou
  <li class="list-group-item list-group-item-success">A simple success list group it
  <li class="list-group-item list-group-item-danger">A simple danger list group item
  <li class="list-group-item list-group-item-warning">A simple warning list group it
  <li class="list-group-item list-group-item-info">A simple info list group item</li
  <li class="list-group-item list-group-item-light">A simple light list group item</
  <li class="list-group-item list-group-item-dark">A simple dark list group item</li
</ul>
```

Contextual classes also work with `.list-group-item-action`. Note the addition of the hover styles here not present in the previous example. Also supported is the `.active` state; apply it to indicate an active selection on a contextual list group item.

A simple default list group item

A simple primary list group item

A simple secondary list group item

A simple success list group item

A simple danger list group item

A simple warning list group item

A simple info list group item

A simple light list group item

A simple dark list group item

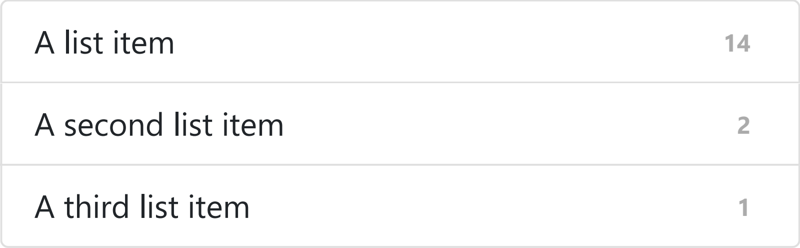
```
<div class="list-group">
  <a href="#" class="list-group-item list-group-item-action">A simple default list g

  <a href="#" class="list-group-item list-group-item-action list-group-item-primary"
  <a href="#" class="list-group-item list-group-item-action list-group-item-secondar
  <a href="#" class="list-group-item list-group-item-action list-group-item-success"
  <a href="#" class="list-group-item list-group-item-action list-group-item-danger">
  <a href="#" class="list-group-item list-group-item-action list-group-item-warning"
  <a href="#" class="list-group-item list-group-item-action list-group-item-info">A
  <a href="#" class="list-group-item list-group-item-action list-group-item-light">A
  <a href="#" class="list-group-item list-group-item-action list-group-item-dark">A
</div>
```

Conveying meaning to assistive technologies

Using color to add meaning only provides a visual indication, which will not be conveyed to users of assistive technologies – such as screen readers. Ensure that information denoted by the color is either obvious from the content itself (e.g. the visible text), or is included through alternative means, such as additional text hidden with the `.visually-hidden` class.

With badges

Add badges to any list group item to show unread counts, activity, and more with the help of some utilities.



A list item 14

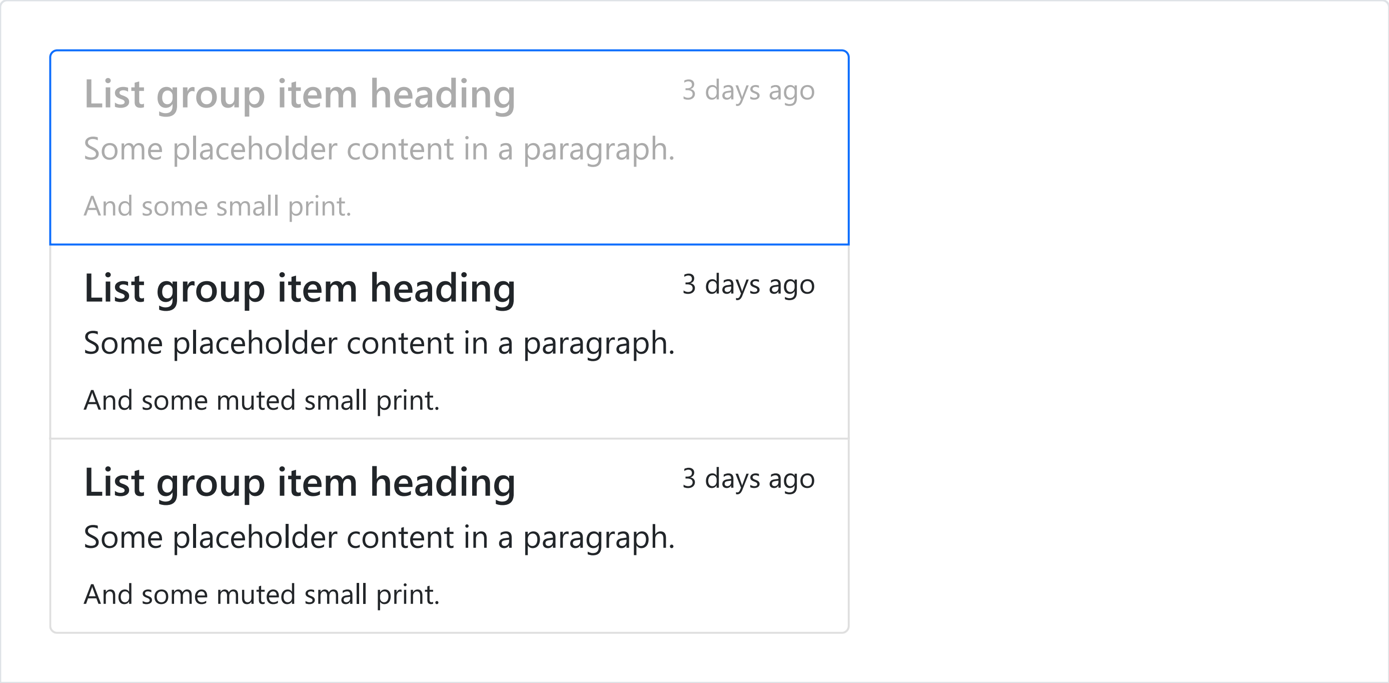
A second list item 2

A third list item 1

```
<ul class="list-group">
  <li class="list-group-item d-flex justify-content-between align-items-center">
    A list item
    <span class="badge bg-primary rounded-pill">14</span>
  </li>
  <li class="list-group-item d-flex justify-content-between align-items-center">
    A second list item
    <span class="badge bg-primary rounded-pill">2</span>
  </li>
  <li class="list-group-item d-flex justify-content-between align-items-center">
    A third list item
    <span class="badge bg-primary rounded-pill">1</span>
  </li>
</ul>
```

Custom content

Add nearly any HTML within, even for linked list groups like the one below, with the help of [flexbox utilities](#).



List group item heading
3 days ago
Some placeholder content in a paragraph.
And some small print.

List group item heading
3 days ago
Some placeholder content in a paragraph.
And some muted small print.

List group item heading
3 days ago
Some placeholder content in a paragraph.
And some muted small print.

```
  <div class="list-group">
    <a href="#" class="list-group-item list-group-item-action active" aria-current="tr
      <div class="d-flex w-100 justify-content-between">
        <h5 class="mb-1">List group item heading</h5>
        <small>3 days ago</small>
      </div>
      <p class="mb-1">Some placeholder content in a paragraph.</p>
      <small>And some small print.</small>
    </a>
    <a href="#" class="list-group-item list-group-item-action">
      <div class="d-flex w-100 justify-content-between">
        <h5 class="mb-1">List group item heading</h5>
        <small class="text-muted">3 days ago</small>
      </div>
      <p class="mb-1">Some placeholder content in a paragraph.</p>
      <small class="text-muted">And some muted small print.</small>
    </a>
    <a href="#" class="list-group-item list-group-item-action">
      <div class="d-flex w-100 justify-content-between">
        <h5 class="mb-1">List group item heading</h5>
        <small class="text-muted">3 days ago</small>
      </div>
      <p class="mb-1">Some placeholder content in a paragraph.</p>
      <small class="text-muted">And some muted small print.</small>
    </a>
```

Checkboxes and radios

Place Bootstrap's checkboxes and radios within list group items and customize as needed. You can use them without `<label>`s, but please remember to include an `aria-label` attribute and value for accessibility.

□ First checkbox

□ Second checkbox

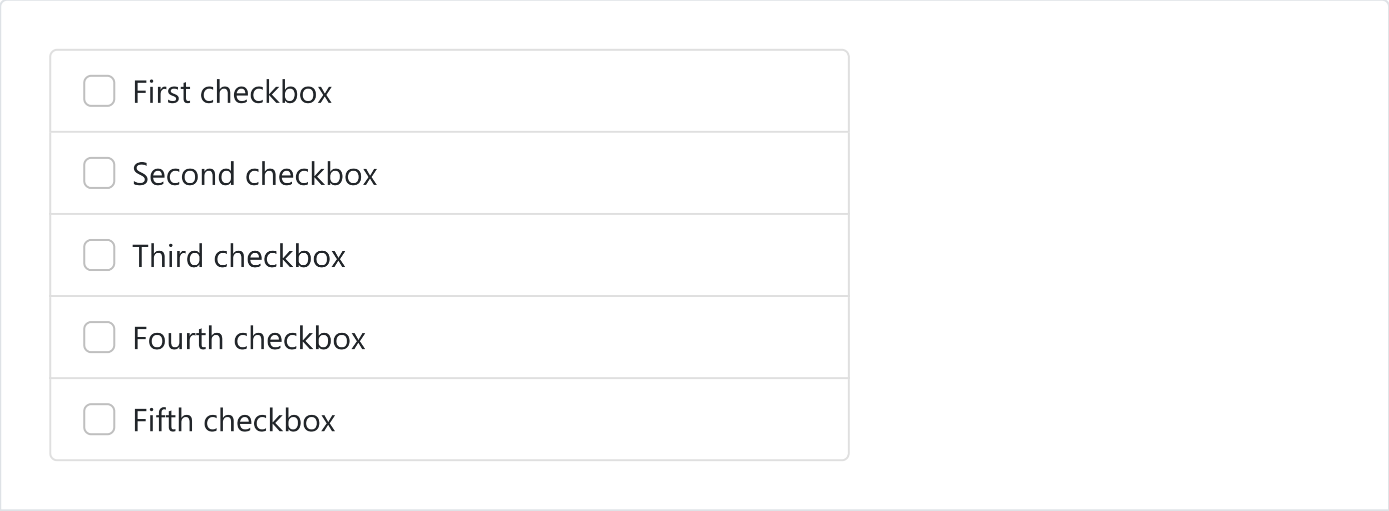
□ Third checkbox

□ Fourth checkbox

□ Fifth checkbox

```
<ul class="list-group">
  <li class="list-group-item">
    <input class="form-check-input me-1" type="checkbox" value="" aria-label="...">
    First checkbox
  </li>
  <li class="list-group-item">
    <input class="form-check-input me-1" type="checkbox" value="" aria-label="...">
    Second checkbox
  </li>
  <li class="list-group-item">
    <input class="form-check-input me-1" type="checkbox" value="" aria-label="...">
    Third checkbox
  </li>
  <li class="list-group-item">
    <input class="form-check-input me-1" type="checkbox" value="" aria-label="...">
    Fourth checkbox
  </li>
  <li class="list-group-item">
    <input class="form-check-input me-1" type="checkbox" value="" aria-label="...">
    Fifth checkbox
  </li>
</ul>
```

And if you want `<label>`s as the `.list-group-item` for large hit areas, you can do that, too.



□ First checkbox

□ Second checkbox

□ Third checkbox

□ Fourth checkbox

□ Fifth checkbox

```
<div class="list-group">
  <label class="list-group-item">
    <input class="form-check-input me-1" type="checkbox" value="">
    First checkbox
  </label>
  <label class="list-group-item">
    <input class="form-check-input me-1" type="checkbox" value="">
    Second checkbox
  </label>
  <label class="list-group-item">
    <input class="form-check-input me-1" type="checkbox" value="">
    Third checkbox
  </label>
  <label class="list-group-item">
    <input class="form-check-input me-1" type="checkbox" value="">
    Fourth checkbox
  </label>
  <label class="list-group-item">
    <input class="form-check-input me-1" type="checkbox" value="">
    Fifth checkbox
  </label>
</div>
```

Sass

Variables

```
$list-group-color:                  $gray-900;
$list-group-bg:                     $white;
$list-group-border-color:           rgba($black, .125);
$list-group-border-width:           $border-width;
$list-group-border-radius:          $border-radius;

$list-group-item-padding-y:         $spacer * .5;
$list-group-item-padding-x:         $spacer;
$list-group-item-bg-scale:          -80%;
$list-group-item-color-scale:       40%;

$list-group-hover-bg:               $gray-100;
$list-group-active-color:           $component-active-color;
$list-group-active-bg:              $component-active-bg;
$list-group-active-border-color:    $list-group-active-bg;

$list-group-disabled-color:         $gray-600;
$list-group-disabled-bg:            $list-group-bg;

$list-group-action-color:           $gray-700;
$list-group-action-hover-color:     $list-group-action-color;

$list-group-action-active-color:    $body-color;
$list-group-action-active-bg:       $gray-200;
```

Mixins

Used in combination with `$theme-colors` to generate the contextual variant classes for `.list-group-item`s.

```
@mixin list-group-item-variant($state, $background, $color) {
  .list-group-item-#{$state} {
    color: $color;
    background-color: $background;

    &.list-group-item-action {
      &:hover,
      &:focus {
        color: $color;
        background-color: shade-color($background, 10%);
      }

      &.active {
        color: $white;
        background-color: $color;
        border-color: $color;
      }
    }
  }
}
```

Loop

Loop that generates the modifier classes with the `list-group-item-variant()` mixin.

```
// List group contextual variants
//
// Add modifier classes to change text and background color on individual items.
// Organizationally, this must come after the `:hover` states.

@each $state, $value in $theme-colors {
  $list-group-variant-bg: shift-color($value, $list-group-item-bg-scale);
  $list-group-variant-color: shift-color($value, $list-group-item-color-scale);
  @if (contrast-ratio($list-group-variant-bg, $list-group-variant-color) < $min-cont
    $list-group-variant-color: mix($value, color-contrast($list-group-variant-bg), a
  }

  @include list-group-item-variant($state, $list-group-variant-bg, $list-group-varia
}
```

JavaScript behavior

Use the tab JavaScript plugin—include it individually or through the compiled `bootstrap.js` file—to extend our list group to create tabbable panes of local content.

Home

Profile

Messages

Settings

Some placeholder content in a paragraph relating to "Home". And some more content, used here just to pad out and fill this tab panel. In production, you would obviously have more real content here. And not just text. It could be anything, really. Text, images, forms.

```
<div class="row">
  <div class="col-4">
    <div class="list-group" id="list-tab" role="tablist">
      <a class="list-group-item list-group-item-action active" id="list-home-list" d
      <a class="list-group-item list-group-item-action" id="list-profile-list" data-
      <a class="list-group-item list-group-item-action" id="list-messages-list" data
      <a class="list-group-item list-group-item-action" id="list-settings-list" data
    </div>
  </div>
  <div class="col-8">
    <div class="tab-content" id="nav-tabContent">
      <div class="tab-pane fade show active" id="list-home" role="tabpanel" aria-lab
      <div class="tab-pane fade" id="list-profile" role="tabpanel" aria-labelledby="
      <div class="tab-pane fade" id="list-messages" role="tabpanel" aria-labelledby=
      <div class="tab-pane fade" id="list-settings" role="tabpanel" aria-labelledby=
    </div>
  </div>
</div>
```

Using data attributes

You can activate a list group navigation without writing any JavaScript by simply specifying `data-bs-toggle="list"` or on an element. Use these data attributes on `.list-group-item`.

```
<div role="tabpanel">
  <!-- List group -->
  <div class="list-group" id="myList" role="tablist">
    <a class="list-group-item list-group-item-action active" data-bs-toggle="list" h
    <a class="list-group-item list-group-item-action" data-bs-toggle="list" href="#p
    <a class="list-group-item list-group-item-action" data-bs-toggle="list" href="#m
    <a class="list-group-item list-group-item-action" data-bs-toggle="list" href="#s
  </div>

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

Via JavaScript

Enable tabbable list item via JavaScript (each list item needs to be activated individually):

```
var triggerTabList = [].slice.call(document.querySelectorAll('#myTab a'))
triggerTabList.forEach(function (triggerEl) {
  var tabTrigger = new bootstrap.Tab(triggerEl)

  triggerEl.addEventListener('click', function (event) {
    event.preventDefault()
    tabTrigger.show()
  })
})
```

You can activate individual list item in several ways:

```
var triggerEl = document.querySelector('#myTab a[href="#profile"]')
bootstrap.Tab.getInstance(triggerEl).show() // Select tab by name

var triggerFirstTabEl = document.querySelector('#myTab li:first-child a')
bootstrap.Tab.getInstance(triggerFirstTabEl).show() // Select first tab
```

Fade effect

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

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

Methods

constructor

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

```
<div class="list-group" id="myList" role="tablist">
  <a class="list-group-item list-group-item-action active" data-bs-toggle="list" hre
  <a class="list-group-item list-group-item-action" data-bs-toggle="list" href="#pro
  <a class="list-group-item list-group-item-action" data-bs-toggle="list" href="#mes
  <a class="list-group-item list-group-item-action" data-bs-toggle="list" href="#set
</div>

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

<script>
  var firstTabEl = document.querySelector('#myTab a:last-child')
  var firstTab = new bootstrap.Tab(firstTabEl)

  firstTab.show()
</script>
```

show

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

```
var someListItemEl = document.querySelector('#someListItem')
var tab = new bootstrap.Tab(someListItemEl)

tab.show()
```

dispose

Destroys an element’s tab.

getInstance

Static method which allows you to get the tab instance associated with a DOM element

```
var triggerEl = document.querySelector('#trigger')
var tab = bootstrap.Tab.getInstance(triggerEl) // Returns a Bootstrap tab instance
```

getOrCreateInstance

Static method which allows you to get the tab instance associated with a DOM element, or create a new one in case it wasn't initialised

```
var triggerEl = document.querySelector('#trigger')
var tab = bootstrap.Tab.getOrCreateInstance(triggerEl) // Returns a Bootstrap tab in
```

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, 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 `event.target` and `event.relatedTarget` to target the active tab and the previous active tab (if available) respectively.
`shown.bs.tab`	This event fires on tab show after a tab has been shown. Use `event.target` and `event.relatedTarget` 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 `event.target` and `event.relatedTarget` 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 `event.target` and `event.relatedTarget` to target the previous active tab and the new active tab, respectively.

```
var tabElms = document.querySelectorAll('a[data-bs-toggle="list"]')
tabElms.forEach(function(tabElm) {
  tabElm.addEventListener('shown.bs.tab', function (event) {
    event.target // newly activated tab
    event.relatedTarget // previous active tab
  })
}
```

Bootstrap

Designed and built with all the love in the world by the Bootstrap team with the help of our contributors.

Currently v5.0.2.

Analytics by Fathom.

Links

- Home
- Docs
- Examples
- Themes
- Blog
- Swag Store

Guides

- Getting started
- Starter template
- Webpack
- Parcel

Projects

- Bootstrap 5
- Bootstrap 4
- Icons
- RFS
- npm starter

Community

- Issues
- Discussions
- Corporate sponsors
- Open Collective
- Stack Overflow