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.

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

Copy

```
<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

Sidebar navigation

```
<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).

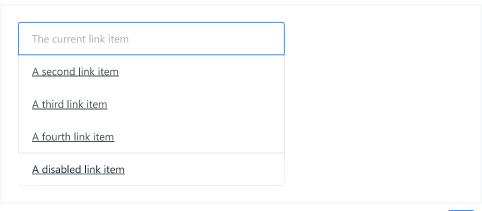| A disabled item |
| A second item |
| A third item |
| A fourth item |
| And a fifth one |

```
<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 |

```html
<div class="list-group">
  <a href="#" class="list-group-item list-group-item-action active" aria-current="true">
    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 class="list-group-item list-group-item-action disabled">A disabled link item</a>
</div>
```
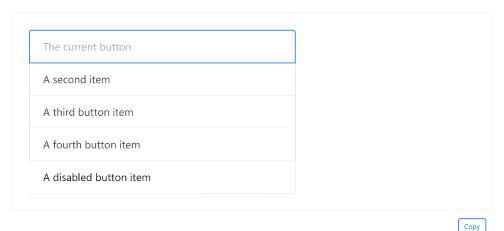
With `<button>`s, you can also make use of the `disabled` attribute instead of the `.disabled` class. Sadly, `<a>`s don't support the disabled attribute.

The current button

A second item

A third button item

A fourth button item

A disabled button item

Copy

```html
<div class="list-group">
  <button type="button" class="list-group-item list-group-item-action active" aria-current="true">
    The current button
  </button>
  <button type="button" class="list-group-item list-group-item-action">A second item</button>
  <button type="button" class="list-group-item list-group-item-action">A third button item</button>
  <button type="button" class="list-group-item list-group-item-action">A fourth button item</button>
  <button type="button" class="list-group-item list-group-item-action" disabled>A disabled button item</button>
</div>
```

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

Copy

```
<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>
```

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}` 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

Copy

```
<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>
```

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

<div style="text-align: right"><button>Copy</button></div>

```html
<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
item</li>
  <li class="list-group-item list-group-item-secondary">A simple secondary list group
item</li>
  <li class="list-group-item list-group-item-success">A simple success list group
item</li>
  <li class="list-group-item list-group-item-danger">A simple danger list group
item</li>
  <li class="list-group-item list-group-item-warning">A simple warning list group
item</li>
  <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>
  <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

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

  <a href="#" class="list-group-item list-group-item-action list-group-item-primary">A
simple primary list group item</a>
  <a href="#" class="list-group-item list-group-item-action list-group-item-
secondary">A simple secondary list group item</a>
  <a href="#" class="list-group-item list-group-item-action list-group-item-success">A
simple success list group item</a>
  <a href="#" class="list-group-item list-group-item-action list-group-item-danger">A
simple danger list group item</a>
  <a href="#" class="list-group-item list-group-item-action list-group-item-warning">A
simple warning list group item</a>
  <a href="#" class="list-group-item list-group-item-action list-group-item-info">A
simple info list group item</a>
  <a href="#" class="list-group-item list-group-item-action list-group-item-light">A
simple light list group item</a>
  <a href="#" class="list-group-item list-group-item-action list-group-item-dark">A
simple dark list group item</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 `.sr-only` 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

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

Copy

```html
<div class="list-group">
  <a href="#" class="list-group-item list-group-item-action active">
    <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>
</div>
```

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.

Copy

```
<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"
data-toggle="list" href="#list-home" role="tab" aria-controls="home">Home</a>
      <a class="list-group-item list-group-item-action" id="list-profile-list" data-
toggle="list" href="#list-profile" role="tab" aria-controls="profile">Profile</a>
      <a class="list-group-item list-group-item-action" id="list-messages-list" data-
toggle="list" href="#list-messages" role="tab" aria-controls="messages">Messages</a>
      <a class="list-group-item list-group-item-action" id="list-settings-list" data-
toggle="list" href="#list-settings" role="tab" aria-controls="settings">Settings</a>
    </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-
labelledby="list-home-list">...</div>
      <div class="tab-pane fade" id="list-profile" role="tabpanel" aria-
labelledby="list-profile-list">...</div>
      <div class="tab-pane fade" id="list-messages" role="tabpanel" aria-
labelledby="list-messages-list">...</div>
      <div class="tab-pane fade" id="list-settings" role="tabpanel" aria-
labelledby="list-settings-list">...</div>
    </div>
  </div>
</div>
```

Using data attributes

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

Copy

```html
<div role="tabpanel">
  <!-- List group -->
  <div class="list-group" id="myList" role="tablist">
    <a class="list-group-item list-group-item-action active" data-toggle="list"
href="#home" role="tab">Home</a>
    <a class="list-group-item list-group-item-action" data-toggle="list"
href="#profile" role="tab">Profile</a>
    <a class="list-group-item list-group-item-action" data-toggle="list"
href="#messages" role="tab">Messages</a>
    <a class="list-group-item list-group-item-action" data-toggle="list"
href="#settings" role="tab">Settings</a>
  </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):

Copy

```javascript
$('#myList a').on('click', function (event) {
  event.preventDefault()
  $(this).tab('show')
})
```

You can activate individual list item in several ways:

Copy

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

Copy

```html
<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

$().tab

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

Copy

```
<div class="list-group" id="myList" role="tablist">
  <a class="list-group-item list-group-item-action active" data-toggle="list"
href="#home" role="tab">Home</a>
    <a class="list-group-item list-group-item-action" data-toggle="list" href="#profile"
role="tab">Profile</a>
    <a class="list-group-item list-group-item-action" data-toggle="list" href="#messages"
role="tab">Messages</a>
    <a class="list-group-item list-group-item-action" data-toggle="list" href="#settings"
role="tab">Settings</a>
</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>
  $(function () {
    $('#myList a:last-child').tab('show')
  })
</script>
```

.tab('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).

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

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.

Event type	Description
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.

<button>Copy</button>

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