

How it works

The collapse JavaScript plugin is used to show and hide content. Buttons or anchors are used as triggers that are mapped to specific elements you toggle. Collapsing an element will animate the `height` from its current value to `0`. Given how CSS handles animations, you cannot use `padding` on a `.collapse` element. Instead, use the class as an independent wrapping element.

```
{{< callout info >}} {{< partial "callout-info-prefersreducedmotion.md" >}} {{< /callout >}}
```

Example

Click the buttons below to show and hide another element via class changes:

- `.collapse` hides content
- `.collapsing` is applied during transitions
- `.collapse.show` shows content

Generally, we recommend using a button with the `data-bs-target` attribute. While not recommended from a semantic point of view, you can also use a link with the `href` attribute (and a `role="button"`). In both cases, the `data-bs-toggle="collapse"` is required.

```
{{< example >}}
```

[Link with href](#) Button with data-bs-target

Some placeholder content for the collapse component. This panel is hidden by default but revealed when the user activates the relevant trigger.

```
{{< /example >}}
```

Horizontal

The collapse plugin also supports horizontal collapsing. Add the `.collapse-horizontal` modifier class to transition the `width` instead of `height` and set a `width` on the immediate child element. Feel free to write your own custom Sass, use inline styles, or use our [width utilities]({{< docsref "/utilities/sizing" >}}).

```
{{< callout info >}}
```

 Please note that while the example below has a `min-height` set to avoid excessive repaints in our docs, this is not explicitly required. **Only the `width` on the child element is required.**

```
{{< /callout >}}
```

```
{{< example >}}
```

Toggle width collapse

This is some placeholder content for a horizontal collapse. It's hidden by default and shown when triggered.

```
{{< /example >}}
```

Multiple targets

A `<button>` or `<a>` can show and hide multiple elements by referencing them with a selector in its `href` or `data-bs-target` attribute. Multiple `<button>` or `<a>` can show and hide an element if they each reference it with their `href` or `data-bs-target` attribute

{{< example >}}

[Toggle first element](#) [Toggle second element](#) [Toggle both elements](#)

Some placeholder content for the first collapse component of this multi-collapse example. This panel is hidden by default but revealed when the user activates the relevant trigger.

Some placeholder content for the second collapse component of this multi-collapse example. This panel is hidden by default but revealed when the user activates the relevant trigger.

{{< /example >}}

Accessibility

Be sure to add `aria-expanded` to the control element. This attribute explicitly conveys the current state of the collapsible element tied to the control to screen readers and similar assistive technologies. If the collapsible element is closed by default, the attribute on the control element should have a value of `aria-expanded="false"`. If you've set the collapsible element to be open by default using the `show` class, set `aria-expanded="true"` on the control instead. The plugin will automatically toggle this attribute on the control based on whether or not the collapsible element has been opened or closed (via JavaScript, or because the user triggered another control element also tied to the same collapsible element). If the control element's HTML element is not a button (e.g., an `<a>` or `<div>`), the attribute `role="button"` should be added to the element.

If your control element is targeting a single collapsible element – i.e. the `data-bs-target` attribute is pointing to an `id` selector – you should add the `aria-controls` attribute to the control element, containing the `id` of the collapsible element. Modern screen readers and similar assistive technologies make use of this attribute to provide users with additional shortcuts to navigate directly to the collapsible element itself.

Note that Bootstrap's current implementation does not cover the various *optional* keyboard interactions described in the [WAI-ARIA Authoring Practices 1.1 accordion pattern](#) - you will need to include these yourself with custom JavaScript.

Sass

Variables

{{< scss-docs name="collapse-transition" file="scss/_variables.scss" >}}

Classes

Collapse transition classes can be found in `scss/_transitions.scss` as these are shared across multiple components (collapse and accordion).

{{< scss-docs name="collapse-classes" file="scss/_transitions.scss" >}}

Usage

The collapse plugin utilizes a few classes to handle the heavy lifting:

- `.collapse` hides the content
- `.collapse.show` shows the content

- `.collapsing` is added when the transition starts, and removed when it finishes

These classes can be found in `_transitions.scss`.

Via data attributes

Just add `data-bs-toggle="collapse"` and a `data-bs-target` to the element to automatically assign control of one or more collapsible elements. The `data-bs-target` attribute accepts a CSS selector to apply the collapse to. Be sure to add the class `collapse` to the collapsible element. If you'd like it to default open, add the additional class `show`.

To add accordion-like group management to a collapsible area, add the data attribute `data-bs-parent="#selector"`. Refer to the [accordion page]({{< docsref "/components/accordion" >}}) for more information.

Via JavaScript

Enable manually with:

```
var collapseElementList = [].slice.call(document.querySelectorAll('.collapse'))
var collapseList = collapseElementList.map(function (collapseEl) {
  return new bootstrap.Collapse(collapseEl)
})
```

Options

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

Name	Type	Default	Description
parent	selector jQuery object DOM element	false	If parent is provided, then all collapsible elements under the specified parent will be closed when this collapsible item is shown. (similar to traditional accordion behavior - this is dependent on the <code>card</code> class). The attribute has to be set on the target collapsible area.
toggle	boolean	true	Toggles the collapsible element on invocation

Methods

{{< callout danger >}} {{< partial "callout-danger-async-methods.md" >}} {{< /callout >}}

Activates your content as a collapsible element. Accepts an optional options `object`.

You can create a collapse instance with the constructor, for example:

```
var myCollapse = document.getElementById('myCollapse')
var bsCollapse = new bootstrap.Collapse(myCollapse, {
  toggle: false
})
```

Method	Description
<code>toggle</code>	Toggles a collapsible element to shown or hidden. Returns to the caller before the collapsible element has actually been shown or hidden (i.e. before the <code>shown.bs.collapse</code> OR <code>hidden.bs.collapse</code> event occurs).
<code>show</code>	Shows a collapsible element. Returns to the caller before the collapsible element has actually been shown (e.g., before the <code>shown.bs.collapse</code> event occurs).
<code>hide</code>	Hides a collapsible element. Returns to the caller before the collapsible element has actually been hidden (e.g., before the <code>hidden.bs.collapse</code> event occurs).
<code>dispose</code>	Destroys an element's collapse. (Removes stored data on the DOM element)
<code>getInstance</code>	Static method which allows you to get the collapse instance associated to a DOM element, you can use it like this: <code>bootstrap.Collapse.getInstance(element)</code>
<code>getOrCreateInstance</code>	Static method which returns a collapse instance associated to a DOM element or create a new one in case it wasn't initialized. You can use it like this: <code>bootstrap.Collapse.getOrCreateInstance(element)</code>

Events

Bootstrap's collapse class exposes a few events for hooking into collapse functionality.

Event type	Description
<code>show.bs.collapse</code>	This event fires immediately when the <code>show</code> instance method is called.
<code>shown.bs.collapse</code>	This event is fired when a collapse element has been made visible to the user (will wait for CSS transitions to complete).
<code>hide.bs.collapse</code>	This event is fired immediately when the <code>hide</code> method has been called.
<code>hidden.bs.collapse</code>	This event is fired when a collapse element has been hidden from the user (will wait for CSS transitions to complete).

```
var myCollapsible = document.getElementById('myCollapsible')
myCollapsible.addEventListener('hidden.bs.collapse', function () {
  // do something...
})
```