

How it works

The accordion uses `[collapse]` [\({{< docsref "/components/collapse" >}}\)](/components/collapse) internally to make it collapsible. To render an accordion that's expanded, add the `.open` class on the `.accordion`.

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

Example

Click the accordions below to expand/collapse the accordion content.

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

Accordion Item #1

This is the first item's accordion body. It is shown by default, until the collapse plugin adds the appropriate classes that we use to style each element. These classes control the overall appearance, as well as the showing and hiding via CSS transitions. You can modify any of this with custom CSS or overriding our default variables. It's also worth noting that just about any HTML can go within the `.accordion-body`, though the transition does limit overflow.

Accordion Item #2

This is the second item's accordion body. It is hidden by default, until the collapse plugin adds the appropriate classes that we use to style each element. These classes control the overall appearance, as well as the showing and hiding via CSS transitions. You can modify any of this with custom CSS or overriding our default variables. It's also worth noting that just about any HTML can go within the `.accordion-body`, though the transition does limit overflow.

Accordion Item #3

This is the third item's accordion body. It is hidden by default, until the collapse plugin adds the appropriate classes that we use to style each element. These classes control the overall appearance, as well as the showing and hiding via CSS transitions. You can modify any of this with custom CSS or overriding our default variables. It's also worth noting that just about any HTML can go within the `.accordion-body`, though the transition does limit overflow.

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

Flush

Add `.accordion-flush` to remove the default `background-color`, some borders, and some rounded corners to render accordions edge-to-edge with their parent container.

```
{{< example class="bg-light" >}}
```

Accordion Item #1

Placeholder content for this accordion, which is intended to demonstrate the `.accordion-flush` class. This is the first item's accordion body.

Accordion Item #2

Placeholder content for this accordion, which is intended to demonstrate the `.accordion-flush` class. This is the second item's accordion body. Let's imagine this being filled with some actual content.

Accordion Item #3

Placeholder content for this accordion, which is intended to demonstrate the `.accordion-flush` class. This is the third item's accordion body. Nothing more exciting happening here in terms of content, but just filling up the space to

make it look, at least at first glance, a bit more representative of how this would look in a real-world application.

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

Always open

Omit the `data-bs-parent` attribute on each `.accordion-collapse` to make accordion items stay open when another item is opened.

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

Accordion Item #1

This is the first item's accordion body. It is shown by default, until the collapse plugin adds the appropriate classes that we use to style each element. These classes control the overall appearance, as well as the showing and hiding via CSS transitions. You can modify any of this with custom CSS or overriding our default variables. It's also worth noting that just about any HTML can go within the `.accordion-body`, though the transition does limit overflow.

Accordion Item #2

This is the second item's accordion body. It is hidden by default, until the collapse plugin adds the appropriate classes that we use to style each element. These classes control the overall appearance, as well as the showing and hiding via CSS transitions. You can modify any of this with custom CSS or overriding our default variables. It's also worth noting that just about any HTML can go within the `.accordion-body`, though the transition does limit overflow.

Accordion Item #3

This is the third item's accordion body. It is hidden by default, until the collapse plugin adds the appropriate classes that we use to style each element. These classes control the overall appearance, as well as the showing and hiding via CSS transitions. You can modify any of this with custom CSS or overriding our default variables. It's also worth noting that just about any HTML can go within the `.accordion-body`, though the transition does limit overflow.

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

Accessibility

Please read the [collapse accessibility section]({{< docsref "/components/collapse#accessibility" >}}) for more information.

Sass

Variables

```
{{< scss-docs name="accordion-variables" file="scss/_variables.scss" >}}
```