

Components

<https://getbootstrap.com/docs/5.2/components/accordion/>

<pre><!DOCTYPE html> <html lang="en"> <head> <title>Bootstrap Example</title> <meta charset="utf-8"> <meta name="viewport" content="width=device-width, initial-scale=1"> <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"> <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script> </head> <body> <div class="container"></pre>
<pre></div> </body> </html></pre>

Accordion

Using fieldset tag to apply disable effect to all sub elements
<pre><!DOCTYPE html> <html lang="en"> <head> <title>Bootstrap Example</title> <meta charset="utf-8"> <meta name="viewport" content="width=device-width, initial-scale=1"> <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"</pre>

```
rel="stylesheet">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class="container">
```

```
<div class="accordion" id="accordionExample">
  <div class="accordion-item">
    <h2 class="accordion-header" id="headingOne">
      <button class="accordion-button" type="button" data-bs-toggle="collapse"
data-bs-target="#collapseOne" aria-expanded="true" aria-controls="collapseOne">
        Accordion Item #1
      </button>
    </h2>
    <div id="collapseOne" class="accordion-collapse collapse show"
aria-labelledby="headingOne" data-bs-parent="#accordionExample">
      <div class="accordion-body">
        <strong>This is the first item's accordion body.</strong> 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 <code>.accordion-body</code>, though the
transition does limit overflow.
      </div>
    </div>
  </div>
  <div class="accordion-item">
    <h2 class="accordion-header" id="headingTwo">
      <button class="accordion-button collapsed" type="button" data-bs-toggle="collapse"
data-bs-target="#collapseTwo" aria-expanded="false" aria-controls="collapseTwo">
        Accordion Item #2
      </button>
    </h2>
    <div id="collapseTwo" class="accordion-collapse collapse" aria-labelledby="headingTwo"
data-bs-parent="#accordionExample">
      <div class="accordion-body">
        <strong>This is the second item's accordion body.</strong> 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 <code>.accordion-body</code>, though the
transition does limit overflow.
      </div>
    </div>
  </div>
  <div class="accordion-item">
    <h2 class="accordion-header" id="headingThree">
      <button class="accordion-button collapsed" type="button" data-bs-toggle="collapse"
```



```
<h2 class="accordion-header" id="flush-headingOne">
  <button class="accordion-button collapsed" type="button" data-bs-toggle="collapse"
data-bs-target="#flush-collapseOne" aria-expanded="false"
aria-controls="flush-collapseOne">
    Accordion Item #1
  </button>
</h2>
<div id="flush-collapseOne" class="accordion-collapse collapse"
aria-labelledby="flush-headingOne" data-bs-parent="#accordionFlushExample">
  <div class="accordion-body">Placeholder content for this accordion, which is intended to
demonstrate the <code>.accordion-flush</code> class. This is the first item's accordion
body.</div>
</div>
</div>
<div class="accordion-item">
  <h2 class="accordion-header" id="flush-headingTwo">
    <button class="accordion-button collapsed" type="button" data-bs-toggle="collapse"
data-bs-target="#flush-collapseTwo" aria-expanded="false"
aria-controls="flush-collapseTwo">
      Accordion Item #2
    </button>
  </h2>
  <div id="flush-collapseTwo" class="accordion-collapse collapse"
aria-labelledby="flush-headingTwo" data-bs-parent="#accordionFlushExample">
    <div class="accordion-body">Placeholder content for this accordion, which is intended to
demonstrate the <code>.accordion-flush</code> class. This is the second item's accordion
body. Let's imagine this being filled with some actual content.</div>
  </div>
</div>
<div class="accordion-item">
  <h2 class="accordion-header" id="flush-headingThree">
    <button class="accordion-button collapsed" type="button" data-bs-toggle="collapse"
data-bs-target="#flush-collapseThree" aria-expanded="false"
aria-controls="flush-collapseThree">
      Accordion Item #3
    </button>
  </h2>
  <div id="flush-collapseThree" class="accordion-collapse collapse"
aria-labelledby="flush-headingThree" data-bs-parent="#accordionFlushExample">
    <div class="accordion-body">Placeholder content for this accordion, which is intended to
demonstrate the <code>.accordion-flush</code> 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.</div>
  </div>
</div>
</div>
```

```
</div>

</body>
</html>
```

Accordion

Using fieldset tag to apply disable effect to all sub elements

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet">
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class="container">
```

```
<div class="accordion" id="accordionPanelsStayOpenExample">
  <div class="accordion-item">
    <h2 class="accordion-header" id="panelsStayOpen-headingOne">
      <button class="accordion-button" type="button" data-bs-toggle="collapse"
data-bs-target="#panelsStayOpen-collapseOne" aria-expanded="true"
aria-controls="panelsStayOpen-collapseOne">
        Accordion Item #1
      </button>
    </h2>
    <div id="panelsStayOpen-collapseOne" class="accordion-collapse collapse show"
aria-labelledby="panelsStayOpen-headingOne">
      <div class="accordion-body">
        <strong>This is the first item's accordion body.</strong> 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
<code>.accordion-body</code>, though the transition does limit overflow.
      </div>
    </div>
  </div>
  <div class="accordion-item">
    <h2 class="accordion-header" id="panelsStayOpen-headingTwo">
      <button class="accordion-button collapsed" type="button" data-bs-toggle="collapse"
data-bs-target="#panelsStayOpen-collapseTwo" aria-expanded="false"
aria-controls="panelsStayOpen-collapseTwo">
```

```

        Accordion Item #2
    </button>
</h2>
<div id="panelsStayOpen-collapseTwo" class="accordion-collapse collapse"
aria-labelledby="panelsStayOpen-headingTwo">
    <div class="accordion-body">
        <strong>This is the second item's accordion body.</strong> 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
<code>.accordion-body</code>, though the transition does limit overflow.
    </div>
</div>
</div>
<div class="accordion-item">
    <h2 class="accordion-header" id="panelsStayOpen-headingThree">
        <button class="accordion-button collapsed" type="button" data-bs-toggle="collapse"
data-bs-target="#panelsStayOpen-collapseThree" aria-expanded="false"
aria-controls="panelsStayOpen-collapseThree">
            Accordion Item #3
        </button>
    </h2>
    <div id="panelsStayOpen-collapseThree" class="accordion-collapse collapse"
aria-labelledby="panelsStayOpen-headingThree">
        <div class="accordion-body">
            <strong>This is the third item's accordion body.</strong> 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
<code>.accordion-body</code>, though the transition does limit overflow.
        </div>
    </div>
</div>
</div>
</div>
</body>
</html>

```

Alerts

Provide contextual feedback messages for typical user actions with the handful of available and flexible alert messages.

```

<!DOCTYPE html>
<html lang="en">
<head>

```

```

<title>Bootstrap Example</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class="container">

<div class="alert alert-primary" role="alert">
  A simple primary alert—check it out!
</div>
<div class="alert alert-secondary" role="alert">
  A simple secondary alert—check it out!
</div>
<div class="alert alert-success" role="alert">
  A simple success alert—check it out!
</div>
<div class="alert alert-danger" role="alert">
  A simple danger alert—check it out!
</div>
<div class="alert alert-warning" role="alert">
  A simple warning alert—check it out!
</div>
<div class="alert alert-info" role="alert">
  A simple info alert—check it out!
</div>
<div class="alert alert-light" role="alert">
  A simple light alert—check it out!
</div>
<div class="alert alert-dark" role="alert">
  A simple dark alert—check it out!
</div>

</div>

</body>
</html>

```

Live example

Click the button below to show an alert (hidden with inline styles to start), then dismiss (and destroy) it with the built-in close button.

<https://getbootstrap.com/docs/5.2/components/alerts/>

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet">
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class="container">
```

```
<div id="liveAlertPlaceholder"></div>
<button type="button" class="btn btn-primary" id="liveAlertBtn">Show live alert</button>
```

```
</div>
```

```
<script>
  const alertPlaceholder = document.getElementById('liveAlertPlaceholder')

  const alert = (message, type) => {
    const wrapper = document.createElement('div')
    wrapper.innerHTML = [
      `<div class="alert alert-${type} alert-dismissible" role="alert">`,
      `  <div>${message}</div>`,
      `  <button type="button" class="btn-close" data-bs-dismiss="alert"
aria-label="Close"></button>`,
      '</div>'
    ].join("")

    alertPlaceholder.append(wrapper)
  }

  const alertTrigger = document.getElementById('liveAlertBtn')
  if (alertTrigger) {
    alertTrigger.addEventListener('click', () => {
      alert('Nice, you triggered this alert message!', 'success')
    })
  }
</script>
```

```
</body>
```

```
</html>
```

Additional content

Alerts can also contain additional HTML elements like headings, paragraphs and dividers.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet">
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class="container">

<div class="alert alert-success" role="alert">
  <h4 class="alert-heading">Well done!</h4>
  <p>Aww yeah, you successfully read this important alert message. This example text is going to run a bit
longer so that you can see how spacing within an alert works with this kind of content.</p>
  <hr>
  <p class="mb-0">Whenever you need to, be sure to use margin utilities to keep things nice and tidy.</p>
</div>

</div>

</body>
</html>
```

Breadcrumb

Indicate the current page's location within a navigational hierarchy that automatically adds separators via CSS.

<https://getbootstrap.com/docs/5.2/components/breadcrumb/>

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet">
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class="container">

<nav aria-label="breadcrumb">
  <ol class="breadcrumb">
    <li class="breadcrumb-item active" aria-current="page">Home</li>
  </ol>
</nav>

<nav aria-label="breadcrumb">
  <ol class="breadcrumb">
    <li class="breadcrumb-item"><a href="#">Home</a></li>
    <li class="breadcrumb-item active" aria-current="page">Library</li>
  </ol>
</nav>

<nav aria-label="breadcrumb">
  <ol class="breadcrumb">
    <li class="breadcrumb-item"><a href="#">Home</a></li>
    <li class="breadcrumb-item"><a href="#">Library</a></li>
    <li class="breadcrumb-item active" aria-current="page">Data</li>
  </ol>
</nav>

</div>

</body>
</html>

```

Cards

Bootstrap's cards provide a flexible and extensible content container with multiple variants and options.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet">
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
<body>
<div class="container">
```

```
<div class="card" style="width: 18rem;">
  
  <div class="card-body">
    <h5 class="card-title">Card title</h5>
    <p class="card-text">Some quick example text to build on the card title and make up the bulk of the card's
content.</p>
    <a href="#" class="btn btn-primary">Go somewhere</a>
  </div>
</div>
```

```
</div>
```

```
</body>
</html>
```

<https://monaca.io/>

<https://drive.google.com/file/d/1LVsM7DsZyATqMbDiMAOWPNbKjFhD8CGt/view?usp=sharing>

