

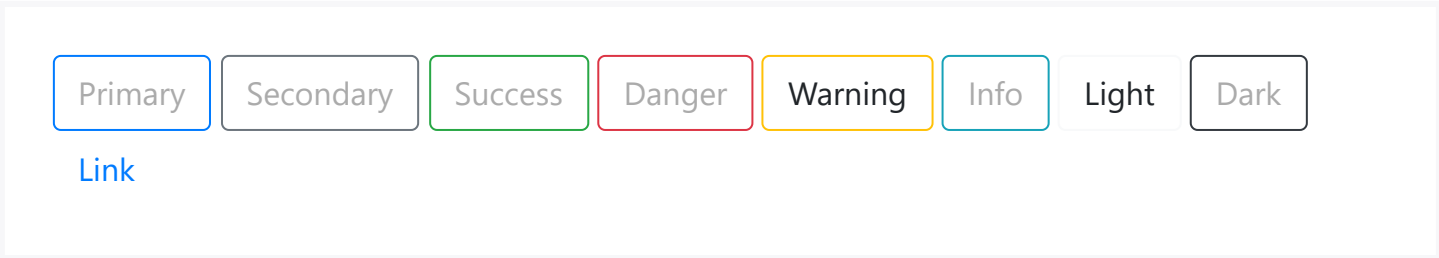
There's a newer version of Bootstrap 4!

Buttons

Use Bootstrap’s custom button styles for actions in forms, dialogs, and more with support for multiple sizes, states, and more.

Examples

Bootstrap includes several predefined button styles, each serving its own semantic purpose, with a few extras thrown in for more control.



```
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-dark">Dark</button>

<button type="button" class="btn btn-link">Link</button>
```

Copy

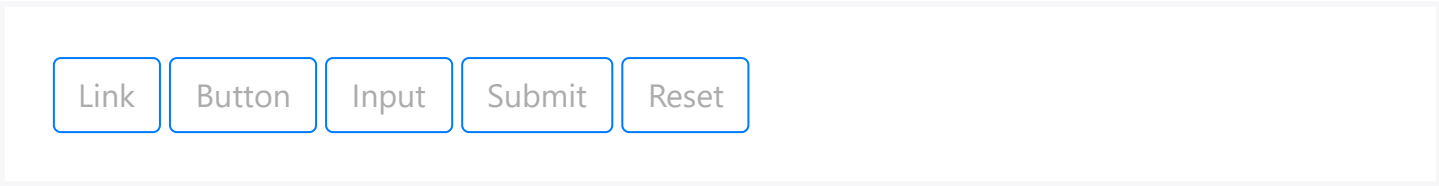
Conveying meaning to assistive technologies

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.

Button tags

The `.btn` classes are designed to be used with the `<button>` element. However, you can also use these classes on `<a>` or `<input>` elements (though some browsers may apply a slightly different rendering).

When using button classes on `<a>` elements that are used to trigger in-page functionality (like collapsing content), rather than linking to new pages or sections within the current page, these links should be given a `role="button"` to appropriately convey their purpose to assistive technologies such as screen readers.



```
<a class="btn btn-primary" href="#" role="button">Link</a>
<button class="btn btn-primary" type="submit">Button</button>
<input class="btn btn-primary" type="button" value="Input">
<input class="btn btn-primary" type="submit" value="Submit">
<input class="btn btn-primary" type="reset" value="Reset">
```

Copy

Outline buttons

In need of a button, but not the hefty background colors they bring? Replace the default modifier classes with the `.btn-outline-*` ones to remove all background images and colors on any button.

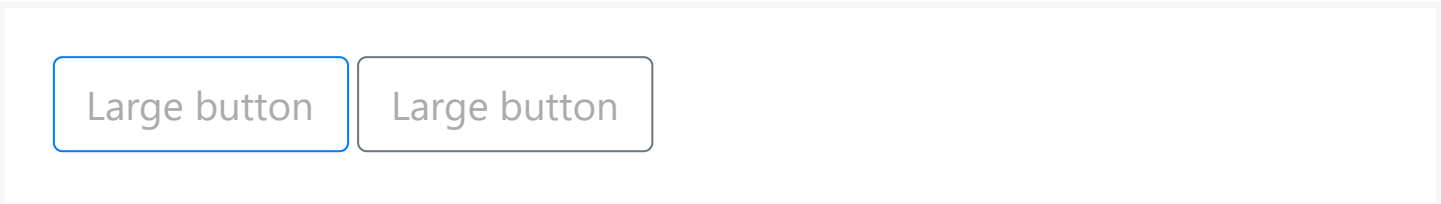


```
<button type="button" class="btn btn-outline-primary">Primary</button>
<button type="button" class="btn btn-outline-secondary">Secondary</button>
<button type="button" class="btn btn-outline-success">Success</button>
<button type="button" class="btn btn-outline-danger">Danger</button>
<button type="button" class="btn btn-outline-warning">Warning</button>
<button type="button" class="btn btn-outline-info">Info</button>
<button type="button" class="btn btn-outline-light">Light</button>
<button type="button" class="btn btn-outline-dark">Dark</button>
```

Copy

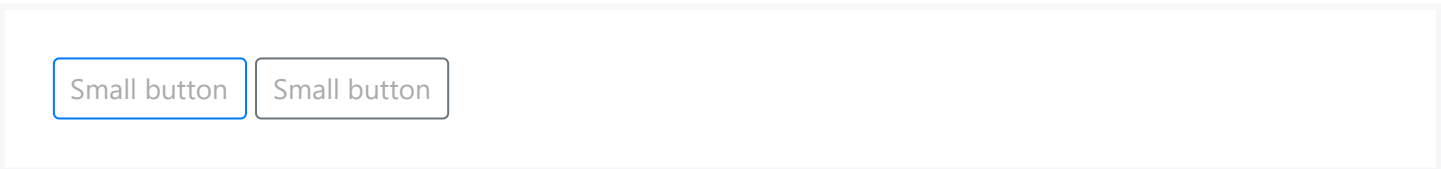
Sizes

Fancy larger or smaller buttons? Add `.btn-lg` or `.btn-sm` for additional sizes.



```
<button type="button" class="btn btn-primary btn-lg">Large button</button>
<button type="button" class="btn btn-secondary btn-lg">Large button</button>
```

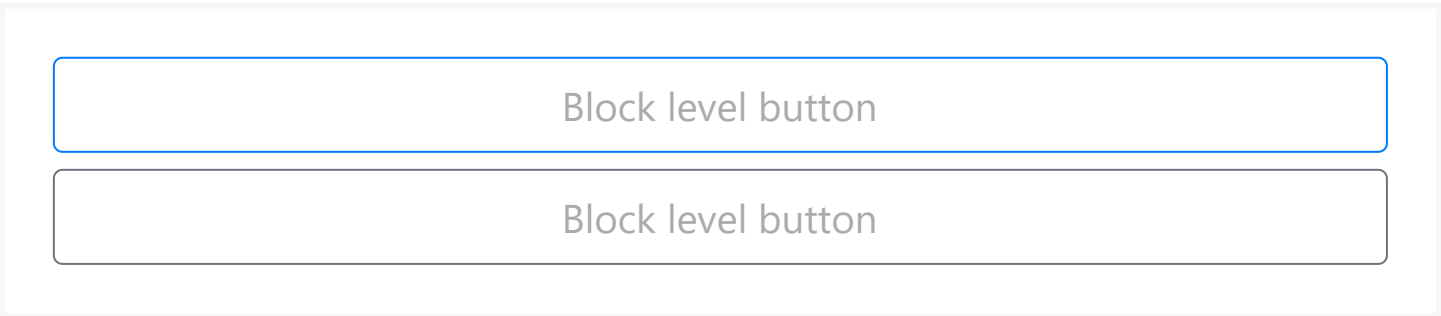
Copy



```
<button type="button" class="btn btn-primary btn-sm">Small button</button>
<button type="button" class="btn btn-secondary btn-sm">Small button</button>
```

Copy

Create block level buttons—those that span the full width of a parent—by adding `.btn-block`.



```
<button type="button" class="btn btn-primary btn-lg btn-block">Block level
button</button>
<button type="button" class="btn btn-secondary btn-lg btn-block">Block level
button</button>
```

Copy

Active state

Buttons will appear pressed (with a darker background, darker border, and inset shadow) when active. **There’s no need to add a class to `<button>s` as they use a pseudo-class.** However, you can still force the same active appearance with `.active` (and include the `aria-pressed="true"` attribute) should you need to replicate the state programmatically.



```
<a href="#" class="btn btn-primary btn-lg active" role="button" aria-pressed="true">Primary link</a>
<a href="#" class="btn btn-secondary btn-lg active" role="button" aria-pressed="true">Link</a>
```

Copy

Disabled state

Make buttons look inactive by adding the `disabled` boolean attribute to any `<button>` element.



```
<button type="button" class="btn btn-lg btn-primary disabled">Primary button</button>
<button type="button" class="btn btn-secondary btn-lg disabled">Button</button>
```

Copy

Disabled buttons using the `<a>` element behave a bit different:

- `<a>`s don't support the `disabled` attribute, so you must add the `.disabled` class to make it visually appear disabled.
- Some future-friendly styles are included to disable all `pointer-events` on anchor buttons. In browsers which support that property, you won't see the disabled cursor at all.
- Disabled buttons should include the `aria-disabled="true"` attribute to indicate the state of the element to assistive technologies.



```
<a href="#" class="btn btn-primary btn-lg disabled" role="button" aria-disabled="true">Primary link</a>
<a href="#" class="btn btn-secondary btn-lg disabled" role="button" aria-disabled="true">Link</a>
```

Copy

Link functionality caveat

The `.disabled` class uses `pointer-events: none` to try to disable the link functionality of `<a>`s, but that CSS property is not yet standardized. In addition, even in browsers that do support `pointer-events: none`, keyboard navigation remains unaffected, meaning that sighted keyboard users and users of assistive technologies will still be able to activate these links. So to be safe, add a `tabindex="-1"` attribute on these links (to prevent them from receiving keyboard focus) and use custom JavaScript to disable their functionality.

Button plugin

Do more with buttons. Control button states or create groups of buttons for more components like toolbars.

Toggle states

Add `data-toggle="button"` to toggle a button’s `active` state. If you’re pre-toggling a button, you must manually add the `.active` class **and** `aria-pressed="true"` to the `<button>`.

Search...

Getting started

Layout

Content

Components

- Alerts
- Badge
- Breadcrumb
- Buttons
- Button group
- Card
- Carousel
- Collapse
- Dropdowns
- Forms
- Input group
- Jumbotron
- List group
- Modal
- Navs
- Navbar
- Pagination
- Popovers
- Progress
- Scrollspy
- Tooltips

Utilities

Extend

Migration

About

Single toggle

```
<button type="button" class="btn btn-primary" data-toggle="button" aria-pressed="false"
autocomplete="off">
  Single toggle
</button>
```

Checkbox and radio buttons

Bootstrap’s `.button` styles can be applied to other elements, such as `<label>`s, to provide checkbox or radio style button toggling. Add `data-toggle="buttons"` to a `.btn-group` containing those modified buttons to enable their toggling behavior via JavaScript and add `.btn-group-toggle` to style the `<input>`s within your buttons. **Note that you can create single input-powered buttons or groups of them.**

The checked state for these buttons is **only updated via click event** on the button. If you use another method to update the input—e.g., with `<input type="reset">` or by manually applying the input’s `checked` property—you’ll need to toggle `.active` on the `<label>` manually.

Note that pre-checked buttons require you to manually add the `.active` class to the input’s `<label>`.

Checked

```
<div class="btn-group-toggle" data-toggle="buttons">
  <label class="btn btn-secondary active">
    <input type="checkbox" checked autocomplete="off"> Checked
  </label>
</div>
```

ActiveRadioRadio

```
<div class="btn-group btn-group-toggle" data-toggle="buttons">
  <label class="btn btn-secondary active">
    <input type="radio" name="options" id="option1" autocomplete="off" checked> Active
  </label>
  <label class="btn btn-secondary">
    <input type="radio" name="options" id="option2" autocomplete="off"> Radio
  </label>
  <label class="btn btn-secondary">
    <input type="radio" name="options" id="option3" autocomplete="off"> Radio
  </label>
</div>
```

Methods

Method	Description
<code>\$(<i>element</i>).button('toggle')</code>	Toggles push state. Gives the button the appearance that it has been activated.
<code>\$(<i>element</i>).button('dispose')</code>	Destroys an element’s button.

