Buttons

Contents

Examples

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

{{< example >}} {{< buttons.inline >}} {{- range (index $.Site.Data “theme-colors”) }} {{ .name | title }} {{- end -}} {{< /buttons.inline >}}

Link

{{< /example >}}

{{< callout info >}} {{< partial “callout-warning-color-assistive-technologies.md” >}} {{< /callout >}}

Disable text wrapping

If you don’t want the button text to wrap, you can add the `.text-nowrap` class to the button. In Sass, you can set `$btn-white-space: nowrap` to disable text wrapping for each button.

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.

{{< example >}} Link Button {{< /example >}}

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.

{{< example >}} {{< buttons.inline >}} {{- range (index $.Site.Data “theme-colors”) }} {{ .name | title }} {{- end -}} {{< /buttons.inline >}} {{< /example >}}

{{< callout info >}} Some of the button styles use a relatively light foreground color, and should only be used on a dark background in order to have sufficient contrast. {{< /callout >}}

Sizes

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

{{< example >}} Large button Large button {{< /example >}}

{{< example >}} Small button Small button {{< /example >}}

Disabled state

Make buttons look inactive by adding the `disabled` boolean attribute to any `<button>` element. Disabled buttons have `pointer-events: none` applied to, preventing hover and active states from triggering.

{{< example >}} Primary button Button {{< /example >}}

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.

- Disabled buttons using `<a>` should include the `aria-disabled="true"` attribute to indicate the state of the element to assistive technologies.
- Disabled buttons using `<a>` *should not* include the `href` attribute.

{{< example >}} Primary link Link {{< /example >}}

Link functionality caveat

To cover cases where you have to keep the `href` attribute on a disabled link, the `.disabled` class uses `pointer-events: none` to try to disable the link functionality of `<a>`s. Note that this CSS property is not yet standardized for HTML, but all modern browsers support it. 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, in addition to `aria-disabled="true"`, also include a `tabindex="-1"` attribute on these links to prevent them from receiving keyboard focus, and use custom JavaScript to disable their functionality altogether.

{{< example >}} Primary link Link {{< /example >}}

Block buttons

Create responsive stacks of full-width, "block buttons" like those in Bootstrap 4 with a mix of our display and gap utilities. By using utilities instead of button specific classes, we have much greater control over spacing, alignment, and responsive behaviors.

{{< example >}}

Button Button

{{< /example >}}

Here we create a responsive variation, starting with vertically stacked buttons until the `md` breakpoint, where `.d-md-block` replaces the `.d-grid` class, thus nullifying the `gap-2` utility. Resize your browser to see them change.

{{< example >}}

Button Button

{{< /example >}}

You can adjust the width of your block buttons with grid column width classes. For example, for a half-width "block button", use `.col-6`. Center it horizontally with `.mx-auto`, too.

{{< example >}}

Button Button

{{< /example >}}

Additional utilities can be used to adjust the alignment of buttons when horizontal. Here we've taken our previous responsive example and added some flex utilities and a margin utility on the button to right align the buttons when they're no longer stacked.

{{< example >}}

Button Button

{{< /example >}}

Button plugin

The button plugin allows you to create simple on/off toggle buttons.

{{< callout info >}} Visually, these toggle buttons are identical to the [checkbox toggle buttons]({{< docsref "/forms/checks-radios#checkbox-toggle-buttons" >}}). However, they are conveyed differently by assistive technologies: the checkbox toggles will be announced by screen readers as "checked"/"not checked" (since, despite their appearance, they are fundamentally still checkboxes), whereas these toggle buttons will be announced as "button"/"button pressed". The choice between these two approaches will depend on the type of toggle you are creating, and whether or not the toggle will make sense to users when announced as a checkbox or as an actual button. {{< /callout >}}

Toggle states

Add `data-bs-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 ensure that it is conveyed appropriately to assistive technologies.

{{< example >}} Toggle button Active toggle button Disabled toggle button {{< /example >}}

{{< example >}} Toggle link Active toggle link Disabled toggle link {{< /example >}}

Methods

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

```
var button = document.getElementById('myButton')
var bsButton = new bootstrap.Button(button)
```

Method

Description

toggle

Toggles push state. Gives the button the appearance that it has been activated.

dispose

Destroys an element's button. (Removes stored data on the DOM element)

getInstance

Static method which allows you to get the button instance associated to a DOM element, you can use it like this: bootstrap.Button.getInstance(element)

getOrCreateInstance

Static method which returns a button instance associated to a DOM element or create a new one in case it wasn't initialized. You can use it like this: bootstrap.Button.getOrCreateInstance(element)

For example, to toggle all buttons

```
var buttons = document.querySelectorAll('.btn')
buttons.forEach(function (button) {
  var button = new bootstrap.Button(button)
  button.toggle()
})
```

Sass

Variables

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

Mixins

There are three mixins for buttons: button and button outline variant mixins (both based on `$theme-colors`), plus a button size mixin.

{{< scss-docs name="btn-variant-mixin" file="scss/mixins/_buttons.scss" >}}

{{< scss-docs name="btn-outline-variant-mixin" file="scss/mixins/_buttons.scss" >}}

{{< scss-docs name="btn-size-mixin" file="scss/mixins/_buttons.scss" >}}

Loops

Button variants (for regular and outline buttons) use their respective mixins with our `$theme-colors` map to generate the modifier classes in `scss/_buttons.scss`.

{{< scss-docs name="btn-variant-loops" file="scss/_buttons.scss" >}}