

Checks and radios

Contents

Approach	1
Checks	2
Indeterminate	2
Disabled	2
Radios	2
Disabled	2
Switches	3
Default (stacked)	3
Inline	3
Without labels	4
Toggle buttons	4
Checkbox toggle buttons	4
Radio toggle buttons	5
Outlined styles	5
Sass	5
Variables	5

Approach

Browser default checkboxes and radios are replaced with the help of `.form-check`, a series of classes for both input types that improves the layout and behavior of their HTML elements, that provide greater customization and cross browser consistency. Checkboxes are for selecting one or several options in a list, while radios are for selecting one option from many.

Structurally, our `<input>`s and `<label>`s are sibling elements as opposed to an `<input>` within a `<label>`. This is slightly more verbose as you must specify `id` and `for` attributes to relate the `<input>` and `<label>`. We use the sibling selector (`~`) for all our `<input>` states, like `:checked` or `:disabled`. When combined with the `.form-check-label` class, we can easily style the text for each item based on the `<input>`'s state.

Our checks use custom Bootstrap icons to indicate checked or indeterminate states.

Checks

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

Default checkbox

Checked checkbox

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

Indeterminate

Checkboxes can utilize the `:indeterminate` pseudo class when manually set via JavaScript (there is no available HTML attribute for specifying it).

```
{{< example class="bd-example-indeterminate">}}
```

Indeterminate checkbox

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

Disabled

Add the `disabled` attribute and the associated `<label>`s are automatically styled to match with a lighter color to help indicate the input's state.

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

Disabled checkbox

Disabled checked checkbox

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

Radios

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

Default radio

Default checked radio

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

Disabled

Add the `disabled` attribute and the associated `<label>`s are automatically styled to match with a lighter color to help indicate the input's state.

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

Disabled radio

Disabled checked radio

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

Switches

A switch has the markup of a custom checkbox but uses the `.form-switch` class to render a toggle switch. Consider using `role="switch"` to more accurately convey the nature of the control to assistive technologies that support this role. In older assistive technologies, it will simply be announced as a regular checkbox as a fallback. Switches also support the `disabled` attribute.

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

Default switch checkbox input

Checked switch checkbox input

Disabled switch checkbox input

Disabled checked switch checkbox input

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

Default (stacked)

By default, any number of checkboxes and radios that are immediate sibling will be vertically stacked and appropriately spaced with `.form-check`.

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

Default checkbox

Disabled checkbox

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

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

Default radio

Second default radio

Disabled radio

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

Inline

Group checkboxes or radios on the same horizontal row by adding `.form-check-inline` to any `.form-check`.

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

1

2

3 (disabled)

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

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

```
1
```

```
2
```

```
3 (disabled)
```

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

Without labels

Omit the wrapping `.form-check` for checkboxes and radios that have no label text. Remember to still provide some form of accessible name for assistive technologies (for instance, using `aria-label`). See the [forms overview accessibility]({{< docsref “/forms/overview#accessibility” >}}) section for details.

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

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

Toggle buttons

Create button-like checkboxes and radio buttons by using `.btn` styles rather than `.form-check-label` on the `<label>` elements. These toggle buttons can further be grouped in a [button group]({{< docsref “/components/button-group” >}}) if needed.

Checkbox toggle buttons

```
{{< example >}} Single toggle {{< /example >}}
```

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

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

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

Visually, these checkbox toggle buttons are identical to the [button plugin toggle buttons]({{< docsref “/components/buttons#button-plugin” >}}). 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 the button plugin 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 >}}

Radio toggle buttons

{{< example >}} Checked

Radio

Disabled

Radio {{< /example >}}

Outlined styles

Different variants of `.btn`, such as the various outlined styles, are supported.

{{< example >}} Single toggle

Checked

Checked success radio

Danger radio {{< /example >}}

Sass

Variables

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