

Components

Over a dozen reusable components built to provide buttons, dropdowns, input groups, navigation, alerts, and much more.

Forms

Bootstrap provides several form control styles, layout options, and custom components for creating a wide variety of forms.

Contents

- Form controls
 - Textual inputs
- Form layouts
 - Form groups
 - Inline forms
 - Using the Grid
- Checkboxes and radios
 - Default (stacked)
 - Inline
 - Without labels
- Static controls
- Disabled states
- Readonly inputs
- Control sizing
- Column sizing
- Help text
 - Block level
 - Inline
- Validation
 - How it works
 - Defining states
 - Examples
- Custom forms
 - Checkboxes and radios
 - Checkboxes
 - Radios
 - Disabled
 - Validation states
 - Stacked
 - Select menu
 - File browser
 - Translating or customizing the strings

Form controls

Bootstrap's form controls expand on [our Rebooted form styles](#) with classes. Use these classes to opt into their customized displays for a more consistent rendering across browsers and devices. The example form below demonstrates common HTML form elements that receive updated styles from Bootstrap with additional classes.

Remember, since Bootstrap utilizes the HTML5 doctype, **all inputs must have a `type` attribute**.

Email address

We'll never share your email with anyone else.

Password

Example select

1

▼

Example multiple select

1, 2, 3

▼

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

- About
- Migration

1
2
3
4
5

Example textarea

File input

No file chosen

This is some placeholder block-level help text for the above input. It's a bit lighter and easily wraps to a new line.

Radio buttons

- Option one is this and that—be sure to include why it's great
 - Option two can be something else and selecting it will deselect option one
 - Option three is disabled
- Check me out

Copy

```
<form>
  <div class="form-group">
    <label for="exampleInputEmail1">Email address</label>
    <input type="email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp" />
    <small id="emailHelp" class="form-text text-muted">We'll never share your email with anyone else!</small>
  </div>
  <div class="form-group">
    <label for="exampleInputPassword1">Password</label>
    <input type="password" class="form-control" id="exampleInputPassword1" placeholder="Password">
  </div>
  <div class="form-group">
    <label for="exampleSelect1">Example select</label>
    <select class="form-control" id="exampleSelect1">
      <option>1</option>
      <option>2</option>
      <option>3</option>
      <option>4</option>
      <option>5</option>
    </select>
  </div>
  <div class="form-group">
    <label for="exampleSelect2">Example multiple select</label>
    <select multiple class="form-control" id="exampleSelect2">
      <option>1</option>
      <option>2</option>
      <option>3</option>
      <option>4</option>
      <option>5</option>
    </select>
  </div>
  <div class="form-group">
    <label for="exampleTextarea">Example textarea</label>
    <textarea class="form-control" id="exampleTextarea" rows="3"></textarea>
  </div>
  <div class="form-group">
    <label for="exampleInputFile">File input</label>
    <input type="file" class="form-control-file" id="exampleInputFile" aria-describedby="fileHelp" />
    <small id="fileHelp" class="form-text text-muted">This is some placeholder block-level help text for file inputs.</small>
  </div>
  <fieldset class="form-group">
    <legend>Radio buttons</legend>
    <div class="form-check">
      <label class="form-check-label">
        <input type="radio" class="form-check-input" name="optionsRadios" id="optionsRadios1" value="1" checked="" />
        Option one is this and that—be sure to include why it's great
      </label>
    </div>
    <div class="form-check">
      <label class="form-check-label">
        <input type="radio" class="form-check-input" name="optionsRadios" id="optionsRadios2" value="2" />
        Option two can be something else and selecting it will deselect option one
      </label>
    </div>
    <div class="form-check disabled">
      <label class="form-check-label">
        <input type="radio" class="form-check-input" name="optionsRadios" id="optionsRadios3" value="3" />
        Option three is disabled
      </label>
    </div>
  </fieldset>
  <div class="form-check">
    <label class="form-check-label">
      <input type="checkbox" class="form-check-input" checked="" />
      Check me out
    </label>
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>
```

Below is a complete list of the specific form controls supported by Bootstrap and the classes that customize them. Additional documentation is available for each group.

Classes	Used for	Supported variations
.form-group	Any group of form controls	Use with any block-level element like <code><fieldset></code> or <code><div></code>
.form-control	Textual inputs	<code>text</code> , <code>password</code> , <code>datetime-local</code> , <code>date</code> , <code>month</code> , <code>time</code> , <code>week</code> , <code>number</code> , <code>email</code> , <code>url</code> , <code>search</code> , <code>tel</code> , <code>color</code>
	Select menus	<code>multiple</code> , <code>size</code>
	Textareas	N/A
.form-control-file	File inputs	<code>file</code>
.form-check	Checkboxes and radios	N/A

Textual inputs

Here are examples of `.form-control` applied to each textual HTML5 `<input>` `type`.

Text	<input class="form-control" type="text" value="Artisanal kale"/>
Search	<input class="form-control" type="search" value="How do I shoot web"/>
Email	<input class="form-control" type="email" value="bootstrap@example.com"/>
URL	<input class="form-control" type="url" value="https://getbootstrap.com"/>
Telephone	<input class="form-control" type="tel" value="1-(555)-555-5555"/>
Password	<input class="form-control" type="password" value="*****"/>
Number	<input class="form-control" type="number" value="42"/>
Date and time	<input class="form-control" type="datetime" value="19/08/2011 13:45"/> (x)
Date	<input class="form-control" type="date" value="19/08/2011"/> (x)
Month	<input class="form-control" type="month" value="August 2011"/> (x)
Week	<input class="form-control" type="week" value="Week 33, 2011"/> (x)
Time	<input class="form-control" type="time" value="13:45"/> (x)
Color	<input class="form-control" type="color"/>

```

<div class="form-group row">
  <label for="example-text-input" class="col-2 col-form-label">Text</label>
  <div class="col-10">
    <input class="form-control" type="text" value="Artisanal kale" id="example-text-input">
  </div>
</div>
<div class="form-group row">
  <label for="example-search-input" class="col-2 col-form-label">Search</label>
  <div class="col-10">
    <input class="form-control" type="search" value="How do I shoot web" id="example-search-input">
  </div>
</div>
<div class="form-group row">
  <label for="example-email-input" class="col-2 col-form-label">Email</label>
  <div class="col-10">
    <input class="form-control" type="email" value="bootstrap@example.com" id="example-email-input">
  </div>
</div>
<div class="form-group row">
  <label for="example-url-input" class="col-2 col-form-label">URL</label>
  <div class="col-10">
    <input class="form-control" type="url" value="https://getbootstrap.com" id="example-url-input">
  </div>
</div>
<div class="form-group row">
  <label for="example-tel-input" class="col-2 col-form-label">Telephone</label>
  <div class="col-10">
    <input class="form-control" type="tel" value="1-(555)-555-5555" id="example-tel-input">
  </div>
</div>
<div class="form-group row">
  <label for="example-password-input" class="col-2 col-form-label">Password</label>
  <div class="col-10">
    <input class="form-control" type="password" value="hunter2" id="example-password-input">
  </div>
</div>

```

```

<!-- ... -->
<div class="form-group row">
  <label for="example-number-input" class="col-2 col-form-label">Number</label>
  <div class="col-10">
    <input class="form-control" type="number" value="42" id="example-number-input">
  </div>
</div>
<div class="form-group row">
  <label for="example-datetime-local-input" class="col-2 col-form-label">Date and time</label>
  <div class="col-10">
    <input class="form-control" type="datetime-local" value="2011-08-19T13:45:00" id="example-date">
  </div>
</div>
<div class="form-group row">
  <label for="example-date-input" class="col-2 col-form-label">Date</label>
  <div class="col-10">
    <input class="form-control" type="date" value="2011-08-19" id="example-date-input">
  </div>
</div>
<div class="form-group row">
  <label for="example-month-input" class="col-2 col-form-label">Month</label>
  <div class="col-10">
    <input class="form-control" type="month" value="2011-08" id="example-month-input">
  </div>
</div>
<div class="form-group row">
  <label for="example-week-input" class="col-2 col-form-label">Week</label>
  <div class="col-10">
    <input class="form-control" type="week" value="2011-W33" id="example-week-input">
  </div>
</div>
<div class="form-group row">
  <label for="example-time-input" class="col-2 col-form-label">Time</label>
  <div class="col-10">
    <input class="form-control" type="time" value="13:45:00" id="example-time-input">
  </div>
</div>
<div class="form-group row">
  <label for="example-color-input" class="col-2 col-form-label">Color</label>
  <div class="col-10">
    <input class="form-control" type="color" value="#563d7c" id="example-color-input">
  </div>
</div>

```

Form layouts

Since Bootstrap applies `display: block` and `width: 100%` to almost all our form controls, forms will by default stack vertically. Additional classes can be used to vary this layout on a per-form basis.

Form groups

The `.form-group` class is the easiest way to add some structure to forms. Its only purpose is to provide `margin-bottom` around a label and control pairing. As a bonus, since it's a class you can use it with `<fieldset>`s, `<div>`s, or nearly any other element.

Example label

Another label

```

<form>
  <div class="form-group">
    <label for="formGroupExampleInput">Example label</label>
    <input type="text" class="form-control" id="formGroupExampleInput" placeholder="Example input">
  </div>
  <div class="form-group">
    <label for="formGroupExampleInput2">Another label</label>
    <input type="text" class="form-control" id="formGroupExampleInput2" placeholder="Another input">
  </div>
</form>

```

Inline forms

Use the `.form-inline` class to display a series of labels, form controls, and buttons on a single horizontal row. Form controls within inline forms vary slightly from their default states.

- Controls are `display: flex`, collapsing any HTML white space and allowing you to provide alignment control with `spacing` and `flexbox` utilities.
- Controls and input groups receive `width: auto` to override the Bootstrap default `width: 100%`.
- Controls **only appear inline in viewports that are at least 576px wide** to account for narrow viewports on mobile devices.

You may need to manually address the width and alignment of individual form controls with `spacing utilities` (as shown below). Lastly, be sure to always include a `<label>` with each form control, even if you need to hide it from non-screenreader visitors with `.sr-only`.

@

 Remember me
 Submit

```
<form class="form-inline">
  <label class="sr-only" for="inlineFormInput">Name</label>
  <input type="text" class="form-control mb-2 mr-sm-2 mb-sm-0" id="inlineFormInput" placeholder="J">

  <label class="sr-only" for="inlineFormInputGroup">Username</label>
  <div class="input-group mb-2 mr-sm-2 mb-sm-0">
    <div class="input-group-addon">@</div>
    <input type="text" class="form-control" id="inlineFormInputGroup" placeholder="Username">
  </div>

  <div class="form-check mb-2 mr-sm-2 mb-sm-0">
    <label class="form-check-label">
      <input class="form-check-input" type="checkbox" checked=""> Remember me
    </label>
  </div>

  <button type="submit" class="btn btn-primary">Submit</button>
</form>
```

Custom form controls and selects are also supported.

Choose... ▾
 Remember my preference
 Submit

```
<form class="form-inline">
  <label class="mr-sm-2" for="inlineFormCustomSelect">Preference</label>
  <select class="custom-select mb-2 mr-sm-2 mb-sm-0" id="inlineFormCustomSelect">
    <option selected>Choose...</option>
    <option value="1">One</option>
    <option value="2">Two</option>
    <option value="3">Three</option>
  </select>

  <label class="custom-control custom-checkbox mb-2 mr-sm-2 mb-sm-0">
    <input type="checkbox" class="custom-control-input" checked="">
    <span class="custom-control-indicator"></span>
    <span class="custom-control-description">Remember my preference</span>
  </label>

  <button type="submit" class="btn btn-primary">Submit</button>
</form>
```

Alternatives to hidden labels

Assistive technologies such as screen readers will have trouble with your forms if you don't include a label for every input. For these inline forms, you can hide the labels using the `.sr-only` class. There are further alternative methods of providing a label for assistive technologies, such as the `aria-label`, `aria-labelledby` or `title` attribute. If none of these are present, assistive technologies may resort to using the `placeholder` attribute, if present, but note that use of `placeholder` as a replacement for other labelling methods is not advised.

Using the Grid

For more structured form layouts that are also responsive, you can utilize Bootstrap's [predefined grid classes](#) or [mixins](#) to create horizontal forms. Add the `.row` class to form groups and use the `.col-*-*` classes to specify the width of your labels and controls.

Be sure to add `.col-form-label` to your `<label>`s as well so they're vertically centered with their associated form controls. For `<legend>` elements, you can use `.col-form-legend` to make them appear similar to regular `<label>` elements.

Email

Password

Radios

Option one is this and that—be sure to include why it's great

Option two can be something else and selecting it will deselect option one

Option three is disabled

Checkbox Check me out

Sign in

```
<div class="container">
  <form>
    <div class="form-group row">
      <label for="inputEmail3" class="col-sm-2 col-form-label">Email</label>
      <div class="col-sm-10">
        <input type="email" class="form-control" id="inputEmail3" placeholder="Email">
      </div>
    </div>
  </form>
</div>
```

```

        </div>
    </div>
    <div class="form-group row">
        <label for="inputPassword3" class="col-sm-2 col-form-label">Password</label>
        <div class="col-sm-10">
            <input type="password" class="form-control" id="inputPassword3" placeholder="Password">
        </div>
    </div>
    <fieldset class="form-group row">
        <legend class="col-form-legend col-sm-2">Radios</legend>
        <div class="col-sm-10">
            <div class="form-check">
                <label class="form-check-label">
                    <input class="form-check-input" type="radio" name="gridRadios" id="gridRadios1" value="Option one is this and that—be sure to include why it's great">
                </label>
            </div>
            <div class="form-check">
                <label class="form-check-label">
                    <input class="form-check-input" type="radio" name="gridRadios" id="gridRadios2" value="Option two can be something else and selecting it will deselect option one">
                </label>
            </div>
            <div class="form-check disabled">
                <label class="form-check-label">
                    <input class="form-check-input" type="radio" name="gridRadios" id="gridRadios3" value="Option three is disabled">
                </label>
            </div>
        </div>
    </fieldset>
    <div class="form-group row">
        <label class="col-sm-2">Checkbox</label>
        <div class="col-sm-10">
            <div class="form-check">
                <label class="form-check-label">
                    <input class="form-check-input" type="checkbox"> Check me out
                </label>
            </div>
        </div>
    </div>
    <div class="form-group row">
        <div class="offset-sm-2 col-sm-10">
            <button type="submit" class="btn btn-primary">Sign in</button>
        </div>
    </div>
</form>
</div>

```

Grid-based form layouts also support large and small inputs.

```

Email      you@example.com
Email      you@example.com

<div class="container">
<form>
    <div class="form-group row">
        <label for="lgFormGroupInput" class="col-sm-2 col-form-label col-form-label-lg">Email</label>
        <div class="col-sm-10">
            <input type="email" class="form-control form-control-lg" id="lgFormGroupInput" placeholder="Email">
        </div>
    </div>
    <div class="form-group row">
        <label for="smFormGroupInput" class="col-sm-2 col-form-label col-form-label-sm">Email</label>
        <div class="col-sm-10">
            <input type="email" class="form-control form-control-sm" id="smFormGroupInput" placeholder="Email">
        </div>
    </div>
</form>

```

Checkboxes and radios

Default checkboxes and radios are improved upon with the help of `.form-check`, a single class for both input types that improves the layout and behavior of their HTML elements. Checkboxes are for selecting one or several options in a list, while radios are for selecting one option from many.

Disabled checkboxes and radios are supported, but to provide a `not-allowed` cursor on hover of the parent `<label>`, you'll need to add the `.disabled` class to the parent `.form-check`. The disabled class will also lighten the text color to help indicate the input's state.

Default (stacked)

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

Option one is this and that—be sure to include why it's great

Option two is disabled

```
<div class="form-check">
  <label class="form-check-label">
    <input class="form-check-input" type="checkbox" value="" checked>
    Option one is this and that—be sure to include why it's great
  </label>
</div>
<div class="form-check disabled">
  <label class="form-check-label">
    <input class="form-check-input" type="checkbox" value="" disabled>
    Option two is disabled
  </label>
</div>
```

Copy

Option one is this and that—be sure to include why it's great

Option two can be something else and selecting it will deselect option one

Option three is disabled

```
<div class="form-check">
  <label class="form-check-label">
    <input class="form-check-input" type="radio" name="exampleRadios" id="exampleRadios1" value="o1" checked>
    Option one is this and that—be sure to include why it's great
  </label>
</div>
<div class="form-check">
  <label class="form-check-label">
    <input class="form-check-input" type="radio" name="exampleRadios" id="exampleRadios2" value="o2">
    Option two can be something else and selecting it will deselect option one
  </label>
</div>
<div class="form-check disabled">
  <label class="form-check-label">
    <input class="form-check-input" type="radio" name="exampleRadios" id="exampleRadios3" value="o3" disabled>
    Option three is disabled
  </label>
</div>
```

Copy

Inline

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

1 2 3

```
<div class="form-check form-check-inline">
  <label class="form-check-label">
    <input class="form-check-input" type="checkbox" id="inlineCheckbox1" value="option1" checked> 1
  </label>
</div>
<div class="form-check form-check-inline">
  <label class="form-check-label">
    <input class="form-check-input" type="checkbox" id="inlineCheckbox2" value="option2"> 2
  </label>
</div>
<div class="form-check form-check-inline disabled">
  <label class="form-check-label">
    <input class="form-check-input" type="checkbox" id="inlineCheckbox3" value="option3" disabled>
  </label>
</div>
```

Copy

1 2 3

```
<div class="form-check form-check-inline">
  <label class="form-check-label">
    <input class="form-check-input" type="radio" name="inlineRadioOptions" id="inlineRadio1" value="radio1" checked>
  </label>
</div>
<div class="form-check form-check-inline">
  <label class="form-check-label">
    <input class="form-check-input" type="radio" name="inlineRadioOptions" id="inlineRadio2" value="radio2">
  </label>
</div>
<div class="form-check form-check-inline disabled">
  <label class="form-check-label">
    <input class="form-check-input" type="radio" name="inlineRadioOptions" id="inlineRadio3" value="radio3" disabled>
  </label>
</div>
```

Copy

Without labels

Should you have no text within the `<label>`, the input is positioned as would expect. [View example code](#)

Should you have no text within the `<label>`, the input is positioned as you'd expect. **Carefully only works on non-inline checkboxes and radios.** Remember to still provide some form of label for assistive technologies (for instance, using `aria-label`).

The screenshot shows a form with two radio buttons. The first radio button is labeled "Can't check this". The second radio button is labeled "Submit". A "Copy" button is located in the top right corner of the code block.

```
<div class="form-check">
  <label class="form-check-label">
    <input class="form-check-input" type="checkbox" id="blankCheckbox" value="option1" aria-label="Can't check this">
  </label>
</div>
<div class="form-check">
  <label class="form-check-label">
    <input class="form-check-input" type="radio" name="blankRadio" id="blankRadio1" value="option1">
  </label>
</div>
```

Static controls

When you need to place plain text next to a form label within a form, use the `.form-control-static` class on an element of your choice.

The screenshot shows a form with two fields: "Email" and "Password". The "Email" field contains the value "email@example.com". The "Password" field is a placeholder for a password. A "Copy" button is located in the top right corner of the code block.

```
<form>
  <div class="form-group row">
    <label class="col-sm-2 col-form-label">Email</label>
    <div class="col-sm-10">
      <p class="form-control-static">email@example.com</p>
    </div>
  </div>
  <div class="form-group row">
    <label for="inputPassword" class="col-sm-2 col-form-label">Password</label>
    <div class="col-sm-10">
      <input type="password" class="form-control" id="inputPassword" placeholder="Password">
    </div>
  </div>
</form>
```


The screenshot shows a form with three fields: "Email", "Password", and a "Confirm identity" button. The "Email" field contains "email@example.com". The "Password" field is a placeholder for a password. The "Confirm identity" button is blue. A "Copy" button is located in the top right corner of the code block.

```
<form class="form-inline">
  <div class="form-group">
    <label class="sr-only">Email</label>
    <p class="form-control-static">email@example.com</p>
  </div>
  <div class="form-group mx-sm-3">
    <label for="inputPassword2" class="sr-only">Password</label>
    <input type="password" class="form-control" id="inputPassword2" placeholder="Password">
  </div>
  <button type="submit" class="btn btn-primary">Confirm identity</button>
</form>
```

Disabled states

Add the `disabled` boolean attribute on an input to prevent user interactions. Disabled inputs appear lighter and add a `not-allowed` cursor.

The screenshot shows a single input field with the placeholder "Disabled input here...". A "Copy" button is located in the top right corner of the code block.

```
<input class="form-control" id="disabledInput" type="text" placeholder="Disabled input here..." disabled="disabled" />
```

Add the `disabled` attribute to a `<fieldset>` to disable all the controls within.

The screenshot shows several disabled form controls: a "Disabled input" field, a "Disabled select menu" dropdown, and a "Disabled select" dropdown. It also includes a checkbox labeled "Can't check this" and a "Submit" button. A "Copy" button is located in the top right corner of the code block.

```
<input class="form-control" id="disabledInput" type="text" placeholder="Disabled input here..." disabled="disabled" />
<select class="form-control" id="disabledSelect" disabled="disabled">
  <option>Disabled select</option>
</select>
<input type="checkbox" id="disabledCheck" checked="checked" disabled="disabled" /> Can't check this
<button type="submit" class="btn btn-primary">Submit</button>
```

```

<form>
  <fieldset disabled>
    <div class="form-group">
      <label for="disabledTextInput">Disabled input</label>
      <input type="text" id="disabledTextInput" class="form-control" placeholder="Disabled input">
    </div>
    <div class="form-group">
      <label for="disabledSelect">Disabled select menu</label>
      <select id="disabledSelect" class="form-control">
        <option>Disabled select</option>
      </select>
    </div>
    <div class="checkbox">
      <label>
        <input type="checkbox" value="1" checked=""> Can't check this
      </label>
    </div>
    <button type="submit" class="btn btn-primary">Submit</button>
  </fieldset>
</form>

```

Copy

Caveat about link functionality of `<a>`

By default, browsers will treat all native form controls (`<input>`, `<select>` and `<button>` elements) inside a `<fieldset disabled>` as disabled, preventing both keyboard and mouse interactions on them. However, if your form also includes `<a ... class="btn btn-*">` elements, these will only be given a style of `pointer-events: none`. As noted in the section about [disabled state for buttons](#) (and specifically in the sub-section for anchor elements), this CSS property is not yet standardized and isn't fully supported in Opera 18 and below, or in Internet Explorer 11, and won't prevent keyboard users from being able to focus or activate these links. So to be safe, use custom JavaScript to disable such links.

Cross-browser compatibility

While Bootstrap will apply these styles in all browsers, Internet Explorer 11 and below don't fully support the `disabled` attribute on a `<fieldset>`. Use custom JavaScript to disable the fieldset in these browsers.

Readonly inputs

Add the `readonly` boolean attribute on an input to prevent modification of the input's value. Read-only inputs appear lighter (just like disabled inputs), but retain the standard cursor.

```



```

Copy

Control sizing

Set heights using classes like `.form-control-lg`, and set widths using grid column classes like `.col-lg-*`.

```





```

Copy

```

<select class="form-control form-control-lg">
  <option>Large select</option>
</select>
<select class="form-control">
  <option>Default select</option>
</select>
<select class="form-control form-control-sm">
  <option>Small select</option>
</select>

```

Copy

Column sizing

Wrap inputs in grid columns, or any custom parent element, to easily enforce desired widths.



```
<div class="row">
  <div class="col-2">
    <input type="text" class="form-control" placeholder=".col-2">
  </div>
  <div class="col-3">
    <input type="text" class="form-control" placeholder=".col-3">
  </div>
  <div class="col-4">
    <input type="text" class="form-control" placeholder=".col-4">
  </div>
</div>
```

Copy

Help text

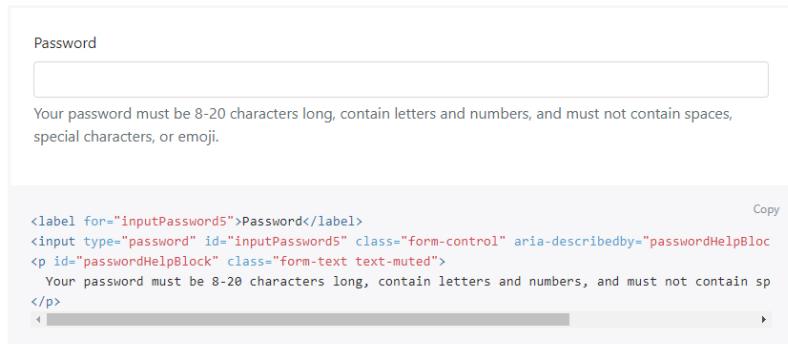
Block-level help text in forms can be created using `.form-text` (previously known as `.help-block` in v3). Inline help text can be flexibly implemented using any inline HTML element and utility classes like `.text-muted`.

Associating help text with form controls

Help text should be explicitly associated with the form control it relates to using the `aria-describedby` attribute. This will ensure that assistive technologies – such as screen readers – will announce this help text when the user focuses or enters the control.

Block level

Block help text—for below inputs or for longer lines of help text—can be easily achieved with `.form-text`. This class includes `display: block` and adds some top margin for easy spacing from the inputs above.



```
<label for="inputPassword5">Password</label>
<input type="password" id="inputPassword5" class="form-control" aria-describedby="passwordHelpBlock">
<p id="passwordHelpBlock" class="form-text text-muted">
  Your password must be 8-20 characters long, contain letters and numbers, and must not contain spaces, special characters, or emoji.
</p>
```

Copy

Inline

Inline text can use any typical inline HTML element (be it a `<small>`, ``, or something else).



```
<form class="form-inline">
  <div class="form-group">
    <label for="inputPassword4">Password</label>
    <input type="password" id="inputPassword4" class="form-control mx-sm-3" aria-describedby="passwordHelpInline">
    <small id="passwordHelpInline" class="text-muted">
      Must be 8-20 characters long.
    </small>
  </div>
</form>
```

Copy

Validation

Bootstrap includes validation styles for danger, warning, and success states on most form controls.

How it works

Here's a rundown of how they work:

- To use, add `.has-warning`, `.has-danger`, or `.has-success` to the parent element. Any `.col-form-label`, `.form-control`, or custom form element will receive the validation styles.
- Contextual validation text, in addition to your usual form field help text, can be added with the use of `.form-control-feedback`. This text will adapt to the parent `.has-*` class. By default it only includes a bit of `margin` for spacing and a modified `color` for each state.
- Validation icons are `url()`'s configured via Sass variables that are applied to `background-image` declarations for each state.

- You may use your own base64 PNGs or SVGs by updating the Sass variables and recompiling.
- Icons can also be disabled entirely by setting the variables to `none` or commenting out the source Sass.

Defining states

Generally speaking, you'll want to use a particular state for specific types of feedback:

- **Danger** is great for when there's a blocking or required field. A user *must* fill in this field properly to submit the form.
- **Warning** works well for input values that are in progress, like password strength, or soft validation before a user attempts to submit a form.
- And lastly, **success** is ideal for situations when you have per-field validation throughout a form and want to encourage a user through the rest of the fields.

Examples

Here are some examples of the aforementioned classes in action. First up is your standard left-aligned fields with labels, help text, and validation messaging.

Input with success

✓

Success! You've done it.
Example help text that remains unchanged.

Input with warning

⚠

Shucks, check the formatting of that and try again.
Example help text that remains unchanged.

Input with danger

✗

Sorry, that username's taken. Try another?
Example help text that remains unchanged.

```
<div class="form-group has-success">
  <label class="form-control-label" for="inputSuccess1">Input with success</label>
  <input type="text" class="form-control form-control-success" id="inputSuccess1">
  <div class="form-control-feedback">Success! You've done it.</div>
  <small class="form-text text-muted">Example help text that remains unchanged.</small>
</div>
<div class="form-group has-warning">
  <label class="form-control-label" for="inputWarning1">Input with warning</label>
  <input type="text" class="form-control form-control-warning" id="inputWarning1">
  <div class="form-control-feedback">Shucks, check the formatting of that and try again.</div>
  <small class="form-text text-muted">Example help text that remains unchanged.</small>
</div>
<div class="form-group has-danger">
  <label class="form-control-label" for="inputDanger1">Input with danger</label>
  <input type="text" class="form-control form-control-danger" id="inputDanger1">
  <div class="form-control-feedback">Sorry, that username's taken. Try another?</div>
  <small class="form-text text-muted">Example help text that remains unchanged.</small>
</div>
```

Those same states can also be used with horizontal forms.

Email

Success! You've done it.
Example help text that remains unchanged.

Email

Shucks, check the formatting of that and try again.
Example help text that remains unchanged.

Email

Sorry, that username's taken. Try another?
Example help text that remains unchanged.

```
<div class="container">
  <form>
    <div class="form-group row has-success">
      <label for="inputHorizontalSuccess" class="col-sm-2 col-form-label">Email</label>
      <div class="col-sm-10">
        <input type="email" class="form-control form-control-success" id="inputHorizontalSuccess">
        <div class="form-control-feedback">Success! You've done it.</div>
        <small class="form-text text-muted">Example help text that remains unchanged.</small>
      </div>
    </div>
    <div class="form-group row has-warning">
      <label for="inputHorizontalWarning" class="col-sm-2 col-form-label">Email</label>
      <div class="col-sm-10">
        <input type="email" class="form-control form-control-warning" id="inputHorizontalWarning">
        <div class="form-control-feedback">Shucks, check the formatting of that and try again.</div>
        <small class="form-text text-muted">Example help text that remains unchanged.</small>
      </div>
    </div>
  </form>

```

```

</div>
<div class="form-group row has-danger">
  <label for="inputHorizontalDnger" class="col-sm-2 col-form-label">Email</label>
  <div class="col-sm-10">
    <input type="email" class="form-control form-control-danger" id="inputHorizontalDnger" placeholder="Sorry, that username's taken. Try another?">
    <small class="form-text text-muted">Example help text that remains unchanged.</small>
  </div>
</div>
</form>
</div>

```

Checkboxes and radios are also supported.

Checkbox with success
 Checkbox with warning
 Checkbox with danger

```

<div class="form-check has-success">
  <label class="form-check-label">
    <input type="checkbox" class="form-check-input" id="checkboxSuccess" value="option1">
    Checkbox with success
  </label>
</div>
<div class="form-check has-warning">
  <label class="form-check-label">
    <input type="checkbox" class="form-check-input" id="checkboxWarning" value="option1">
    Checkbox with warning
  </label>
</div>
<div class="form-check has-danger">
  <label class="form-check-label">
    <input type="checkbox" class="form-check-input" id="checkboxDanger" value="option1">
    Checkbox with danger
  </label>
</div>

```

Copy

Custom forms

For even more customization and cross browser consistency, use our completely custom form elements to replace the browser defaults. They're built on top of semantic and accessible markup, so they're solid replacements for any default form control.

Checkboxes and radios

Each checkbox and radio is wrapped in a `<label>` for three reasons:

- It provides a larger hit areas for checking the control.
- It provides a helpful and semantic wrapper to help us replace the default `<input>`s.
- It triggers the state of the `<input>` automatically, meaning no JavaScript is required.

We hide the default `<input>` with `opacity` and use the `.custom-control-indicator` to build a new custom form indicator in its place. Unfortunately we can't build a custom one from just the `<input>` because CSS's `content` doesn't work on that element.

We use the sibling selector (`~`) for all our `<input>` states—like `:checked`—to properly style our custom form indicator. When combined with the `.custom-control-description` class, we can also style the text for each item based on the `<input>`'s state.

In the checked states, we use **base64 embedded SVG icons** from [Open Iconic](#). This provides us the best control for styling and positioning across browsers and devices.

Checkboxes

Check this custom checkbox

```

<label class="custom-control custom-checkbox">
  <input type="checkbox" class="custom-control-input">
  <span class="custom-control-indicator"></span>
  <span class="custom-control-description">Check this custom checkbox</span>
</label>

```

Copy

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

Check this custom checkbox

Copy

If you're using jQuery, something like this should suffice:

```

$( '.your-checkbox' ).prop('indeterminate', true)

```

Radios

Toggle this custom radio Or toggle this other custom radio

```
<label class="custom-control custom-radio">
  <input id="radio1" name="radio" type="radio" class="custom-control-input">
  <span class="custom-control-indicator"></span>
  <span class="custom-control-description">Toggle this custom radio</span>
</label>
<label class="custom-control custom-radio">
  <input id="radio2" name="radio" type="radio" class="custom-control-input">
  <span class="custom-control-indicator"></span>
  <span class="custom-control-description">Or toggle this other custom radio</span>
</label>
```

Copy

Disabled

Custom checkboxes and radios can also be disabled. Add the `disabled` boolean attribute to the `<input>` and the custom indicator and label description will be automatically styled.

Check this custom checkbox Toggle this custom radio

```
<label class="custom-control custom-checkbox">
  <input type="checkbox" class="custom-control-input" disabled>
  <span class="custom-control-indicator"></span>
  <span class="custom-control-description">Check this custom checkbox</span>
</label>

<label class="custom-control custom-radio">
  <input id="radio3" name="radioDisabled" type="radio" class="custom-control-input" disabled>
  <span class="custom-control-indicator"></span>
  <span class="custom-control-description">Toggle this custom radio</span>
</label>
```

Copy

Validation states

Add other states to your custom forms with our validation classes.

Check this custom checkbox
 Check this custom checkbox
 Check this custom checkbox

```
<div class="form-group has-success">
  <label class="custom-control custom-checkbox">
    <input type="checkbox" class="custom-control-input">
    <span class="custom-control-indicator"></span>
    <span class="custom-control-description">Check this custom checkbox</span>
  </label>
</div>
<div class="form-group has-warning">
  <label class="custom-control custom-checkbox">
    <input type="checkbox" class="custom-control-input">
    <span class="custom-control-indicator"></span>
    <span class="custom-control-description">Check this custom checkbox</span>
  </label>
</div>
<div class="form-group has-danger mb-0">
  <label class="custom-control custom-checkbox">
    <input type="checkbox" class="custom-control-input">
    <span class="custom-control-indicator"></span>
    <span class="custom-control-description">Check this custom checkbox</span>
  </label>
</div>
```

Copy

Stacked

Custom checkboxes and radios are inline to start. Add a parent with class `.custom-controls-stacked` to ensure each form control is on separate lines.

Toggle this custom radio
 Or toggle this other custom radio

```
<div class="custom-controls-stacked">
  <label class="custom-control custom-radio">
    <input id="radioStacked1" name="radio-stacked" type="radio" class="custom-control-input">
    <span class="custom-control-indicator"></span>
    <span class="custom-control-description">Toggle this custom radio</span>
  </label>
  <label class="custom-control custom-radio">
    <input id="radioStacked2" name="radio-stacked" type="radio" class="custom-control-input">
    <span class="custom-control-indicator"></span>
    <span class="custom-control-description">Or toggle this other custom radio</span>
  </label>
```

Copy

```
</div>
```

Select menu

Custom `<select>` menus need only a custom class, `.custom-select` to trigger the custom styles.

```
Open this select menu ↗
```

Copy

```
<select class="custom-select">
  <option selected>Open this select menu</option>
  <option value="1">One</option>
  <option value="2">Two</option>
  <option value="3">Three</option>
</select>
```

File browser

The file input is the most gnarly of the bunch and require additional JavaScript if you'd like to hook them up with functional `Choose file...` and selected file name text.

```
Choose file...      Browse
```

Copy

```
<label class="custom-file">
  <input type="file" id="file" class="custom-file-input">
  <span class="custom-file-control"></span>
</label>
```

Here's how it works:

- We wrap the `<input>` in a `<label>` so the custom control properly triggers the file browser.
- We hide the default file `<input>` via `opacity`.
- We use `:after` to generate a custom background and directive (`Choose file...`).
- We use `:before` to generate and position the `Browse` button.
- We declare a `height` on the `<input>` for proper spacing for surrounding content.

In other words, it's an entirely custom element, all generated via CSS.

Translating or customizing the strings

The `:lang()` pseudo-class is used to allow for easy translation of the "Browse" and "Choose file..." text into other languages. Simply override or add entries to the `$custom-file-text` SCSS variable with the relevant `language tag` and localized strings. The English strings can be customized the same way. For example, here's how one might add a Spanish translation (Spanish's language code is `es`):

```
$custom-file-text: (
  placeholder: (
    en: "Choose file...",
    es: "Seleccionar archivo..."
  ),
  button-label: (
    en: "Browse",
    es: "Navegar"
  )
);
```

Copy

You'll need to set the language of your document (or subtree thereof) correctly in order for the correct text to be shown. This can be done using the `lang` attribute or the `Content-Language` HTTP header, among other methods.