



Forms, Bootstrap Themes Bootstrap CSS

Forms

Bootstrap 5 Forms

Examples and usage guidelines for form control styles, layout options, and custom components for creating a wide variety of forms.

This is a general overview with a summary of the most fundamental knowledge. For the more detailed information regarding specific form, types have a look at the dedicated documentation pages.

- [Checkbox](#)
- [File](#)
- [Input fields](#)
- [Select](#)
- [Switch](#)
- [Radio](#)
- [Range](#)
- [Validation](#)

Basic example

A basic example of a simple login form with input fields (email and password), checkbox and submit button.

Checkbox and "forgot password" link are positioned inline by using 2 column grid layout.

Note: Most of the demo examples have a **fixed width** for the demo purpose. Included code examples **do not** have a fixed width, so they'll naturally fill the full width of its parent element. To control the width of the form place it in the grid, use the sizing utilities, or set the width inline.

Email address

 Password

Remember me

[Forgot password?](#)

SIGN IN

Every group of form fields should reside in a `<form>` element. MDB provides no default styling for the `<form>` element, but there are some powerful browser features that are provided by default.

- New to browser forms? Consider reviewing [the MDN form docs](#) for an overview and complete list of available attributes.
- `<button>`s within a `<form>` default to `type="submit"`, so strive to be specific and always include a type.
- You can disable every form element within a form with the `disabled` attribute on the `<form>`.

Since MDB 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.

Examples

Login form

Typical login form with additional register buttons.

Email address

Password

Remember me

[Forgot password?](#)

SIGN IN

Not a member? [Register](#)

or sign up with:

Register form

Typical register form with additional register buttons.

First name

Last name

Email address

Password

Subscribe to our newsletter

SIGN UP

or sign up with:

Contact form

Typical contact form with textarea input as a message field.

Name

Email address

Message

Send me a copy of this message

SEND

Subscription form

A typical subscription form used when subscribing to the newsletter.

Name

Email address

I have read and agree to the terms

SUBSCRIBE

Checkout form

An example of the extended form with typical checkout inputs.

First name

Last name

Company name

Address

Email

Phone

Additional information

Create an account?

PLACE ORDER

Login-register

By using [pills](#) you can place login and register forms into a single component.

- [LOGIN](#)
- [REGISTER](#)

Sign in with:

or:

Email or username

Password

Remember me

[Forgot password?](#)

SIGN IN

Not a member? [Register](#)

Layout

There are multiple ways to structure forms and provide them the desired layout. Have a look at the examples below to learn more about forms layout.

Utilities

[Margin utilities](#) are the easiest way to add some structure to forms. They provide basic grouping of labels, controls, optional form text, and form validation messaging. We recommend sticking to margin-bottom utilities, and using a single direction throughout the form for consistency.

Feel free to build your forms however you like, with `<fieldset>`s, `<div>`s, or nearly any other element.

In the example below, we add `.mb-4` class to provide a proper margin between two input fields.

Name

Email address

Form grid

More complex forms can be built using our [grid classes](#). Use these for form layouts that require multiple columns, varied widths, and additional alignment options. **Requires the `$enable-grid-classes` Sass variable to be enabled** (on by default).

Name

Email address

First name

Last name

Email address

Gutters

By adding gutter modifier classes, you can have control over the gutter width in as well the inline as block direction. **Also requires the \$enable-grid-classes Sass variable to be enabled** (on by default).

 Name

Email address

 Name

Email address

Column sizing

As shown in the previous examples, our grid system allows you to place any number of `.cols` within a `.row`. They'll split the available width equally between them. You may also pick a subset of your columns to take up more or less space, while the remaining `.cols` equally split the rest, with specific column classes like `.col-sm-7`.

 Name Name Name

Auto-sizing

The example below uses a flexbox utility to vertically center the contents and changes `.col` to `.col-auto` so that your columns only take up as much space as needed. Put another way, the column sizes itself based on the contents.

Label

Checked

Label

Checked switch checkbox input

SUBMIT

Inline forms

Use the `.row-cols-*` classes to create responsive horizontal layouts. By adding [gutter modifier classes](#), we'll have gutters in horizontal and vertical directions. On narrow mobile viewports, the `.col-12` helps stack the form controls and more. The `.align-items-center` aligns the form elements to the middle, making the `.form-checkbox` align properly.

Username

@

Preference

Remember me

SUBMIT