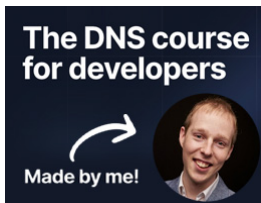


[View on GitHub](#)

# Layout

Give your forms some structure—from inline to horizontal to custom grid implementations—with our form layout options.



I've spent 2 years learning DNS while building NSLookup.io. Now, I'm teaching everything I know.

ads via Carbon

## On this page

Forms

Utilities

Form grid

Gutters

Horizontal form

Horizontal form label sizing

Column sizing

Auto-sizing

Inline forms

## Forms

Every group of form fields should reside in a `<form>` element. Bootstrap 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 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.

## 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.

Example label

Another label

```
<div class="mb-3">
  <label for="formGroupExampleInput" class="form-label">Example label</label>
  <input type="text" class="form-control" id="formGroupExampleInput" placeholder="Ex
</div>
<div class="mb-3">
  <label for="formGroupExampleInput2" class="form-label">Another label</label>
  <input type="text" class="form-control" id="formGroupExampleInput2" placeholder="A
</div>
```

## 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).

```
<div class="row">
  <div class="col">
    <input type="text" class="form-control" placeholder="First name" aria-label="Fir
  </div>
  <div class="col">
    <input type="text" class="form-control" placeholder="Last name" aria-label="Last
  </div>
</div>
```

## 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).

```
<div class="row g-3">
  <div class="col">
    <input type="text" class="form-control" placeholder="First name" aria-label="Fir
  </div>
  <div class="col">
    <input type="text" class="form-control" placeholder="Last name" aria-label="Last
  </div>
</div>
```

More complex layouts can also be created with the grid system.

Address

Address 2

City

State

Zip

☐ Check me out

```
<form class="row g-3">
  <div class="col-md-6">
    <label for="inputEmail4" class="form-label">Email</label>
    <input type="email" class="form-control" id="inputEmail4">
  </div>
  <div class="col-md-6">
    <label for="inputPassword4" class="form-label">Password</label>
    <input type="password" class="form-control" id="inputPassword4">
  </div>
  <div class="col-12">
    <label for="inputAddress" class="form-label">Address</label>
    <input type="text" class="form-control" id="inputAddress" placeholder="1234 Main
  </div>
  <div class="col-12">
    <label for="inputAddress2" class="form-label">Address 2</label>
    <input type="text" class="form-control" id="inputAddress2" placeholder="Apartmen
  </div>
  <div class="col-md-6">
    <label for="inputCity" class="form-label">City</label>
    <input type="text" class="form-control" id="inputCity">
```

```

</div>
<div class="col-md-4">
  <label for="inputState" class="form-label">State</label>
  <select id="inputState" class="form-select">
    <option selected>Choose...</option>
    <option>...</option>
  </select>
</div>
<div class="col-md-2">
  <label for="inputZip" class="form-label">Zip</label>
  <input type="text" class="form-control" id="inputZip">
</div>
<div class="col-12">
  <div class="form-check">
    <input class="form-check-input" type="checkbox" id="gridCheck">
    <label class="form-check-label" for="gridCheck">
      Check me out
    </label>
  </div>
</div>
<div class="col-12">
  <button type="submit" class="btn btn-primary">Sign in</button>
</div>
..

```

## Horizontal form

Create horizontal forms with the grid by adding the `.row` class to form groups and using 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.

At times, you maybe need to use margin or padding utilities to create that perfect alignment you need. For example, we've removed the `padding-top` on our stacked radio inputs label to better align the text baseline.

Email

Password

Radios

☒ First radio
 ☐ Second radio
 ☐ Third disabled radio
 ☐ Example checkbox

Sign in

```
<form>
  <div class="row mb-3">
    <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">
    </div>
  </div>
  <div class="row mb-3">
    <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">
    </div>
  </div>
  <fieldset class="row mb-3">
    <legend class="col-form-label col-sm-2 pt-0">Radios</legend>
    <div class="col-sm-10">
      <div class="form-check">
        <input class="form-check-input" type="radio" name="gridRadios" id="gridRadio
        <label class="form-check-label" for="gridRadios1">
          First radio
        </label>
      </div>
      <div class="form-check">
        <input class="form-check-input" type="radio" name="gridRadios" id="gridRadio
        <label class="form-check-label" for="gridRadios2">
          Second radio
        </label>
      </div>
      <div class="form-check disabled">
        <input class="form-check-input" type="radio" name="gridRadios" id="gridRadio
        <label class="form-check-label" for="gridRadios3">
          Third disabled radio
        </label>
      </div>
    </div>
  </fieldset>
  <div class="row mb-3">
    <div class="col-sm-10 offset-sm-2">
      <div class="form-check">
        <input class="form-check-input" type="checkbox" id="gridCheck1">
        <label class="form-check-label" for="gridCheck1">
          Example checkbox
        </label>
      </div>
    </div>
  </div>
```

```

    </div>
  </div>
  <button type="submit" class="btn btn-primary">Sign in</button>
  --

```

## Horizontal form label sizing

Be sure to use `.col-form-label-sm` or `.col-form-label-lg` to your `<label>s` or `<legend>s` to correctly follow the size of `.form-control-lg` and `.form-control-sm`.

Email	<input type="text" value="col-form-label-sm"/>
Email	<input type="text" value="col-form-label"/>
Email	<input type="text" value="col-form-label-lg"/>

```

<div class="row mb-3">
  <label for="colFormLabelSm" class="col-sm-2 col-form-label col-form-label-sm">Email
  <div class="col-sm-10">
    <input type="email" class="form-control form-control-sm" id="colFormLabelSm" placeholder="col-form-label-sm" />
  </div>
</div>
<div class="row mb-3">
  <label for="colFormLabel" class="col-sm-2 col-form-label">Email</label>
  <div class="col-sm-10">
    <input type="email" class="form-control" id="colFormLabel" placeholder="col-form-label" />
  </div>
</div>
<div class="row">
  <label for="colFormLabelLg" class="col-sm-2 col-form-label col-form-label-lg">Email
  <div class="col-sm-10">
    <input type="email" class="form-control form-control-lg" id="colFormLabelLg" placeholder="col-form-label-lg" />
  </div>
</div>

```

## 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 `.col`s equally split the rest, with specific column classes like `.col-sm-7`.

```
<div class="row g-3">
  <div class="col-sm-7">
    <input type="text" class="form-control" placeholder="City" aria-label="City">
  </div>
  <div class="col-sm">
    <input type="text" class="form-control" placeholder="State" aria-label="State">
  </div>
  <div class="col-sm">
    <input type="text" class="form-control" placeholder="Zip" aria-label="Zip">
  </div>
</div>
```

## 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.

@

☐ Remember me

```
<form class="row gy-2 gx-3 align-items-center">
  <div class="col-auto">
    <label class="visually-hidden" for="autoSizingInput">Name</label>
    <input type="text" class="form-control" id="autoSizingInput" placeholder="Jane D
  </div>
  <div class="col-auto">
    <label class="visually-hidden" for="autoSizingInputGroup">Username</label>
    <div class="input-group">
      <div class="input-group-text">@</div>
      <input type="text" class="form-control" id="autoSizingInputGroup" placeholder=
    </div>
  </div>
  <div class="col-auto">
```



```

<label class="visually-hidden" for="autoSizingSelect">Preference</label>
<select class="form-select" id="autoSizingSelect">
  <option selected>Choose...</option>
  <option value="1">One</option>
  <option value="2">Two</option>
  <option value="3">Three</option>
</select>
</div>
<div class="col-auto">
  <div class="form-check">
    <input class="form-check-input" type="checkbox" id="autoSizingCheck">
    <label class="form-check-label" for="autoSizingCheck">
      Remember me
    </label>
  </div>
</div>
<div class="col-auto">
  <button type="submit" class="btn btn-primary">Submit</button>
</div>
--

```

You can then remix that once again with size-specific column classes.

☐ Remember me

```

<form class="row gx-3 gy-2 align-items-center">
  <div class="col-sm-3">
    <label class="visually-hidden" for="specificSizeInputName">Name</label>
    <input type="text" class="form-control" id="specificSizeInputName" placeholder="Name" />
  </div>
  <div class="col-sm-3">
    <label class="visually-hidden" for="specificSizeInputGroupUsername">Username</label>
    <div class="input-group">
      <div class="input-group-text">@</div>
      <input type="text" class="form-control" id="specificSizeInputGroupUsername" placeholder="Username" />
    </div>
  </div>
  <div class="col-sm-3">
    <label class="visually-hidden" for="specificSizeSelect">Preference</label>
    <select class="form-select" id="specificSizeSelect">
      <option selected>Choose...</option>
      <option value="1">One</option>
      <option value="2">Two</option>
    </select>
  </div>
  <div class="col-sm-3">
    <input type="checkbox" /> Remember me
  </div>
  <div class="col-sm-3">
    <button type="submit" class="btn btn-primary">Submit</button>
  </div>
</form>

```

```

    <option value="3">Three</option>
  </select>
</div>
<div class="col-auto">
  <div class="form-check">
    <input class="form-check-input" type="checkbox" id="autoSizingCheck2">
    <label class="form-check-label" for="autoSizingCheck2">
      Remember me
    </label>
  </div>
</div>
<div class="col-auto">
  <button type="submit" class="btn btn-primary">Submit</button>
</div>
--

```

## 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.

```

<form class="row row-cols-lg-auto g-3 align-items-center">
  <div class="col-12">
    <label class="visually-hidden" for="inlineFormInputGroupUsername">Username</label>
    <div class="input-group">
      <div class="input-group-text">@</div>
      <input type="text" class="form-control" id="inlineFormInputGroupUsername" placeholder="Username">
    </div>
  </div>

  <div class="col-12">
    <label class="visually-hidden" for="inlineFormSelectPref">Preference</label>

```

```
<select class="form-select" id="inlineFormSelectPref">
  <option selected>Choose...</option>
  <option value="1">One</option>
  <option value="2">Two</option>
  <option value="3">Three</option>
</select>
</div>

<div class="col-12">
  <div class="form-check">
    <input class="form-check-input" type="checkbox" id="inlineFormCheck">
    <label class="form-check-label" for="inlineFormCheck">
      Remember me
    </label>
  </div>
</div>

<div class="col-12">
  <button type="submit" class="btn btn-primary">Submit</button>
</div>
..
```





Designed and built with all the love in the world by the Bootstrap team with the help of our contributors.

Code licensed MIT, docs CC BY 3.0.

Currently v5.0.2.

Analytics by Fathom.

## Links

[Home](#)

[Docs](#)

[Examples](#)

[Themes](#)

[Blog](#)

[Swag Store](#)

## Projects

[Bootstrap 5](#)

[Bootstrap 4](#)

[Icons](#)

[RFS](#)

[npm starter](#)

## Guides

[Getting started](#)

[Starter template](#)

[Webpack](#)

[Parcel](#)

## Community

[Issues](#)

[Discussions](#)

[Corporate sponsors](#)

[Open Collective](#)

[Stack Overflow](#)