

Bootstrap





Docs Examples Icons Themes Blog

Search

CTRL K



v5.3



A responsive CSS framework for designing mobile-first, flexible, and modern web applications

Build fast, responsive sites
with Bootstrap

Powerful, extensible, and feature-packed frontend toolkit. Build and customize with
Sass, utilize prebuilt grid system and components, and bring projects to life with
JavaScript plugins.

easily.

\$ npm i bootstrap@5.3.3

Read the docs

Currently v5.3.3 · [Download](#) · [All releases](#)



Docs Examples Icons Themes Blog

Search

CTRL K

Q Twitter CI

v5.3 ▾

O ▾

Get Security Updates for Bootstrap 3 & 4 →



Build fast, responsive sites with Bootstrap

Powerful, extensible, and feature-packed frontend toolkit. Build and customize with Sass, utilize prebuilt grid system and components, and bring projects to life with powerful JavaScript plugins.

\$ npm i bootstrap@5.3.3

Read the docs

Currently v5.3.3 · [Download](#) · [All releases](#)

Bootstrap is an open-source frontend framework for building responsive, mobile-first websites using CSS, JavaScript, and components.

Why Use Bootstrap?



Pre-designed components, responsive grid system,
browser compatibility, customization, and easy-to-use
classes for fast development.

Responsive Design

getbootstrap.com/docs/5.3/layout/grid/#example

Docs Examples Icons Themes Blog

Search CTRL K

v5.3

On this page

Example

How it works

Grid options

Auto-layout columns

Equal-width

Setting one column width

Variable width content

Responsive classes

All breakpoints

Stacked to horizontal

Mix and match

Row columns

Nesting

CSS

Sass variables

Sass mixins

Example usage

Customizing the grid

Columns and gutters

Grid tiers

Grid system

View on GitHub

Overview

Sass

Options

Color

Color modes

Components

CSS variables

Optimize

Layout

Breakpoints

Containers

Grid

Columns

Gutters

Utilities

Z-index

CSS Grid

Content

Reboot

Typography

Images

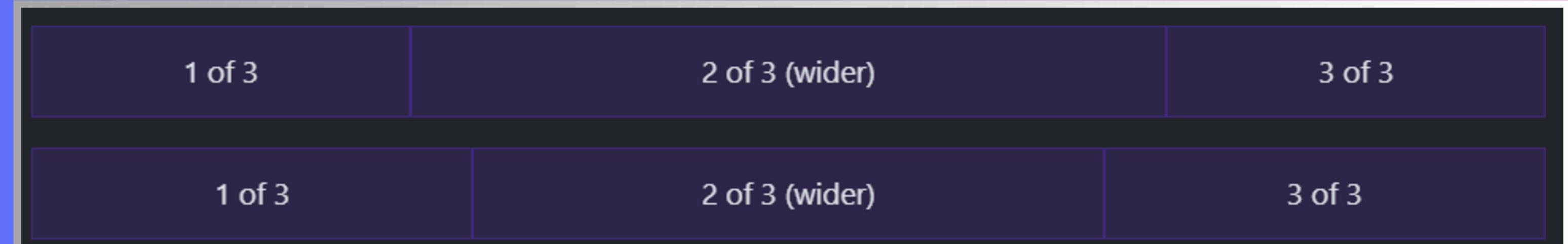
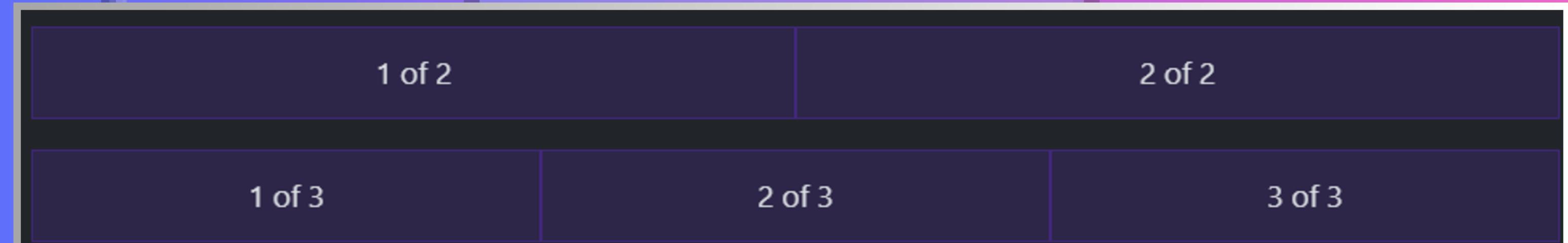
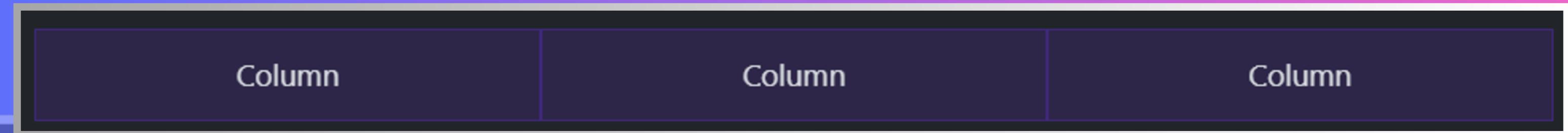
Tables

New to or unfamiliar with flexbox? [Read this CSS Tricks flexbox guide](#) for background, terminology, guidelines, and code snippets.

Column Column Column

HTML

```
<div class="container text-center">
  <div class="row">
    <div class="col">
```



Column

Column

Column

HTML



```
<div class="container text-center">
  <div class="row">
    <div class="col">
      Column
    </div>
    <div class="col">
      Column
    </div>
    <div class="col">
      Column
    </div>
  </div>
</div>
```

1 of 2

2 of 2

1 of 3

2 of 3

3 of 3

HTML



```
<div class="container text-center">
  <div class="row">
    <div class="col">
      1 of 2
    </div>
    <div class="col">
      2 of 2
    </div>
  </div>
  <div class="row">
    <div class="col">
      1 of 3
    </div>
    <div class="col">
      2 of 3
    </div>
    <div class="col">
      3 of 3
    </div>
  </div>
</div>
```

Breakpoints

getbootstrap.com/docs/5.3/layout.breakpoints/#core-concepts

Docs Examples Icons Themes Blog Search CTRL K

Getting started

- Introduction
- Download
- Contents
- Browsers & devices
- JavaScript
- Webpack
- Parcel
- Vite
- Accessibility
- RFS
- RTL
- Contribute

Customize

- Overview
- Sass
- Options
- Color
- Color modes
- Components
- CSS variables
- Optimize

Breakpoints

View on GitHub

Breakpoints are customizable widths that determine how your responsive layout behaves across device or viewport sizes in Bootstrap.

Core concepts

- **Breakpoints are the building blocks of responsive design.** Use them to control when your layout can be adapted at a particular viewport or device size.
- **Use media queries to architect your CSS by breakpoint.** Media queries are a feature of CSS that allow you to conditionally apply styles based on a set of browser and operating system parameters. We most commonly use `min-width` in our media queries.
- **Mobile first, responsive design is the goal.** Bootstrap's CSS aims to apply the bare minimum of styles to make a layout work at the smallest breakpoint, and then layers on styles to adjust that design for larger devices. This optimizes your CSS, improves rendering time, and provides a great experience for your visitors.

Available breakpoints

Bootstrap includes six default breakpoints, sometimes referred to as *grid tiers*, for building responsively. These breakpoints can be customized if you're using our source Sass files.

Breakpoint	Class infix	Dimensions
Extra small	<code>None</code>	<576px

On this page

- Core concepts
- Available breakpoints
- Media queries
- Min-width
- Max-width
- Single breakpoint
- Between breakpoints

Available breakpoints

Bootstrap includes six default breakpoints, sometimes referred to as *grid tiers*, for building responsively. These breakpoints can be customized if you're using our source Sass files.

Breakpoint	Class infix	Dimensions
Extra small	<code>None</code>	<576px
Small	<code>sm</code>	≥576px
Medium	<code>md</code>	≥768px
Large	<code>lg</code>	≥992px
Extra large	<code>xl</code>	≥1200px
Extra extra large	<code>xxl</code>	≥1400px

Each breakpoint was chosen to comfortably hold containers whose widths are multiples of 12. Breakpoints are also representative of a subset of common device sizes and viewport dimensions—they don't specifically target every use case or device. Instead, the ranges provide a strong and consistent foundation to build on for nearly any device.

These breakpoints are customizable via Sass—you'll find them in a Sass map in our `_variables.scss` stylesheet.

Containers

getbootstrap.com/docs/5.3/layout/containers/#how-they-work

Docs Examples Icons Themes Blog Search CTRL K v5.3

Customize

- Overview
- Sass
- Options
- Color
- Color modes
- Components
- CSS variables
- Optimize

Layout

- Breakpoints
- Containers**
- Grid
- Columns
- Gutters
- Utilities
- Z-index
- CSS Grid

Content

- Reboot
- Typography
- Images
- Tables

Containers

View on GitHub

Containers are a fundamental building block of Bootstrap that contain, pad, and align your content within a given device or viewport.

How they work

Containers are the most basic layout element in Bootstrap and are **required when using our default grid system**. Containers are used to contain, pad, and (sometimes) center the content within them. While containers *can* be nested, most layouts do not require a nested container.

Bootstrap comes with three different containers:

- `.container`, which sets a `max-width` at each responsive breakpoint
- `.container-{breakpoint}`, which is `width: 100%` until the specified breakpoint
- `.container-fluid`, which is `width: 100%` at all breakpoints

The table below illustrates how each container's `max-width` compares to the original `.container` and `.container-fluid` across each breakpoint.

See them in action and compare them in our [Grid example](#).

	Extra small	Small	Medium	Large	X-Large	XX-Large
<code>.container</code>	<576px	≥576px	≥768px	≥992px	≥1200px	≥1400px
<code>.container-sm</code>	100%	540px	720px	960px	1140px	1320px
<code>.container-md</code>	100%	540px	720px	960px	1140px	1320px
<code>.container-lg</code>	100%	540px	720px	960px	1140px	1320px
<code>.container-xl</code>	100%	540px	720px	960px	1140px	1320px
<code>.container-xxl</code>	100%	540px	720px	960px	1140px	1320px

	Extra small <code><576px</code>	Small <code>≥576px</code>	Medium <code>≥768px</code>	Large <code>≥992px</code>	X-Large <code>≥1200px</code>	XX-Large <code>≥1400px</code>
<code>.container</code>	100%	540px	720px	960px	1140px	1320px
<code>.container-sm</code>	100%	540px	720px	960px	1140px	1320px
<code>.container-md</code>	100%	100%	720px	960px	1140px	1320px
<code>.container-lg</code>	100%	100%	100%	960px	1140px	1320px
<code>.container-xl</code>	100%	100%	100%	100%	1140px	1320px
<code>.container-xxl</code>	100%	100%	100%	100%	100%	1320px
<code>.container-fluid</code>	100%	100%	100%	100%	100%	100%

Default container

Our default `.container` class is a responsive, fixed-width container, meaning its `max-width` changes at each breakpoint.

```
<div class="container">  
  <!-- Content here -->  
</div>
```

Colors

.text-primary
.text-primary-emphasis
.text-secondary
.text-secondary-emphasis
.text-success
.text-success-emphasis
.text-danger
.text-danger-emphasis
.text-warning
.text-warning-emphasis
.text-info
.text-info-emphasis
.text-light
.text-light-emphasis
.text-dark
.text-dark-emphasis
.text-body
.text-body-emphasis

```
<p class="text-primary">.text-primary</p>
<p class="text-primary-emphasis">.text-primary-emphasis</p>
<p class="text-secondary">.text-secondary</p>
<p class="text-secondary-emphasis">.text-secondary-emphasis</p>
<p class="text-success">.text-success</p>
<p class="text-success-emphasis">.text-success-emphasis</p>
<p class="text-danger">.text-danger</p>
<p class="text-danger-emphasis">.text-danger-emphasis</p>
<p class="text-warning bg-dark">.text-warning</p>
<p class="text-warning-emphasis">.text-warning-emphasis</p>
<p class="text-info bg-dark">.text-info</p>
<p class="text-info-emphasis">.text-info-emphasis</p>
<p class="text-light bg-dark">.text-light</p>
<p class="text-light-emphasis">.text-light-emphasis</p>
<p class="text-dark bg-white">.text-dark</p>
<p class="text-dark-emphasis">.text-dark-emphasis</p>

<p class="text-body">.text-body</p>
<p class="text-body-emphasis">.text-body-emphasis</p>
<p class="text-body-secondary">.text-body-secondary</p>
<p class="text-body-tertiary">.text-body-tertiary</p>

<p class="text-black bg-white">.text-black</p>
<p class="text-white bg-dark">.text-white</p>
<p class="text-black-50 bg-white">.text-black-50</p>
<p class="text-white-50 bg-dark">.text-white-50</p>
```

[Primary](#)[Secondary](#)[Success](#)[Danger](#)[Warning](#)[Info](#)[Light](#)[Dark](#)[Link](#)

HTML

```
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-dark">Dark</button>

<button type="button" class="btn btn-link">Link</button>
```

Navigation Bars (Navbars)

getbootstrap.com/docs/5.3/components/navbar/#how-it-works

Docs Examples Icons Themes Blog Search CTRL K

View on GitHub

On this page

How it works

Supported content

Brand

- Text
- Image
- Image and text

Nav

- Forms
- Text

Color schemes

Containers

Placement

Scrolling

Responsive behaviors

- Toggler
- External content
- Offcanvas

CSS

- Variables
- Sass variables
- Sass loops

Accordion

Alerts

Badge

Breadcrumb

Buttons

Button group

Card

Carousel

Close button

Collapse

Dropdowns

List group

Modal

Navbar

Navs & tabs

Offcanvas

Pagination

Placeholders

Popovers

Progress

Scrollspy

Spinners

Toasts

Tooltips

Helpers

Navbar

Responsive behaviors

Documentation and examples for Bootstrap's powerful, responsive navigation header, the navbar. Includes support for branding, navigation, and more, including support for our collapse plugin.

How it works

Here's what you need to know before getting started with the navbar:

- Navbars require a wrapping `.navbar` with `.navbar-expand{-sm|-md|-lg|-x1|-xx1}` for responsive collapsing and [color scheme](#) classes.
- Navbars and their contents are fluid by default. Change the [container](#) to limit their horizontal width in different ways.
- Use our [spacing](#) and [flex](#) utility classes for controlling spacing and alignment within navbars.
- Navbars are responsive by default, but you can easily modify them to change that. Responsive behavior depends on our Collapse JavaScript plugin.
- Ensure accessibility by using a `<nav>` element or, if using a more generic element such as a `<div>`, add a `role="navigation"` to every navbar to explicitly identify it as a landmark region for users of assistive technologies.
- Indicate the current item by using `aria-current="page"` for the current page or `aria-current="true"` for the current item in a set.
- New in v5.2.0:** Navbars can be themed with CSS variables that are scoped to the `.navbar` base class. `.navbar-light` has been deprecated and `.navbar-dark` has been rewritten to override CSS variables instead of adding additional styles.

The animation effect of this component is dependent on the `prefers-reduced-motion` media query. See the [reduced motion](#) section of our [accessibility documentation](#).

Navbar Home Link Dropdown ▾ Disabled

Search

Search

HTML



```
<nav class="navbar navbar-expand-lg bg-body-tertiary">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Navbar</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarSupportedContent">
      <ul class="navbar-nav me-auto mb-2 mb-lg-0">
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="#">Home</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Link</a>
        </li>
        <li class="nav-item dropdown">
          <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown" aria-expanded="false">
            Dropdown
          </a>
          <ul class="dropdown-menu">
            <li><a class="dropdown-item" href="#">Action</a></li>
            <li><a class="dropdown-item" href="#">Another action</a></li>
            <li><hr class="dropdown-divider"></li>
            <li><a class="dropdown-item" href="#">Something else here</a></li>
          </ul>
        </li>
        <li class="nav-item">
          <a class="nav-link disabled" aria-disabled="true">Disabled</a>
        </li>
      </ul>
      <form class="d-flex" role="search">
        <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">
        <button class="btn btn-outline-success" type="submit">Search</button>
      </form>
    </div>
  </div>
</nav>
```

Responsive Images

Responsive images

Images in Bootstrap are made responsive with `.img-fluid`. This applies `max-width: 100%;` and `height: auto;` to the image so that it scales with the parent width.



Responsive image

HTML

```
</img>
```



NEXT video

