Framework(s)

CSS3

What is a Framework?

A framework may have any number of reasons for existing, but in programming most commonly they offer a way to code more quickly, enhance backwards compatibility, or offer a more standardized way to code to encourage easier teamwork.

These purposes are present in numerous CSS frameworks. They are especially prevalent in rapid web application prototyping and often even allow those who are not very experienced in CSS to accomplish layouts that may otherwise have been very difficult for them.

Popular CSS Frameworks

tailwindcss

Tailwind CSS is one of the most recent big CSS frameworks and is seeing rapid adoption by the industry.

BULMA

Bulma is a fairly recent very lightweight CSS framework. Being so lightweight means fast load-times, but fewer features.

B

Bootstrap is by far the most widely used CSS framework on the web. In its maturity, it features a plethora of different pre-made styles, layouts, tutorials, and support.

ZURB Foundation

Foundation is Bootstrap's most serious competitor. It boasts ease of responsive web development.

Popular CSS Frameworks (Continued)

Materialize is one of the most popular, as it is heavily inspired by Google's popular material design concepts and rules.

Semantic UI makes its goal to provide simple and effective elements for rapid user interface development.

So Many Frameworks, So Little Time

Is it possible to even learn all of these frameworks? Maybe.

But you don’t need to. It comes down to picking one or two that will match your needs most often. If you join a development team, they may already have a favourite they’ll ask you to become familiar with.

No one knows all of them in-depth! Though, it is a useful skill to be able to learn about any of them and to feel comfortable in experimenting with more than one if/when needed.

This can be where research and experimentation come in. If you know you’re going to be working on a blog, for instance, you should be checking each of the top frameworks to see which one has the most styles oriented to your specific cause.

Perhaps one features individual post layouts that’d be really helpful, and others don’t. You may want to engage with that one as it looks more and more promising.

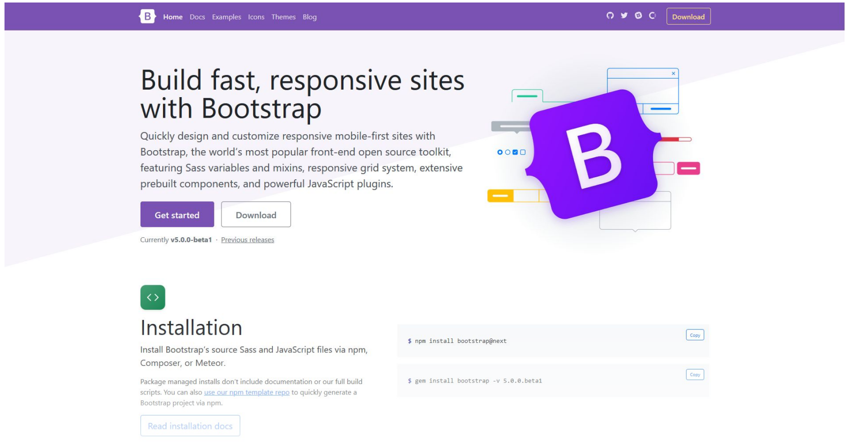
Installing a Framework

Let’s try the most popular CSS framework out there: Bootstrap!

Upon visiting their website, you’ll be met with the “Get Started” and “Download” options.

For most frameworks and tools you look at online, you’ll want to review the “Get Started” section or documentation.

Let’s have a look at the “Get Started” page…



Build fast, responsive sites with Bootstrap

Quickly design and customize responsive mobile-first sites with Bootstrap, the world’s most popular front-end open source toolkit, featuring Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful JavaScript plugins.

Currently **v5.0.0-beta1** · Previous releases

Installation

Install Bootstrap’s source Sass and JavaScript files via npm, Composer, or Meteor.

Package managed installs don’t include documentation or our full build scripts. You can also use our npm template repo to quickly generate a Bootstrap project via npm.

```
$ npm install bootstrap@next
```

```
$ gem install bootstrap -v 5.0.0.beta1
```

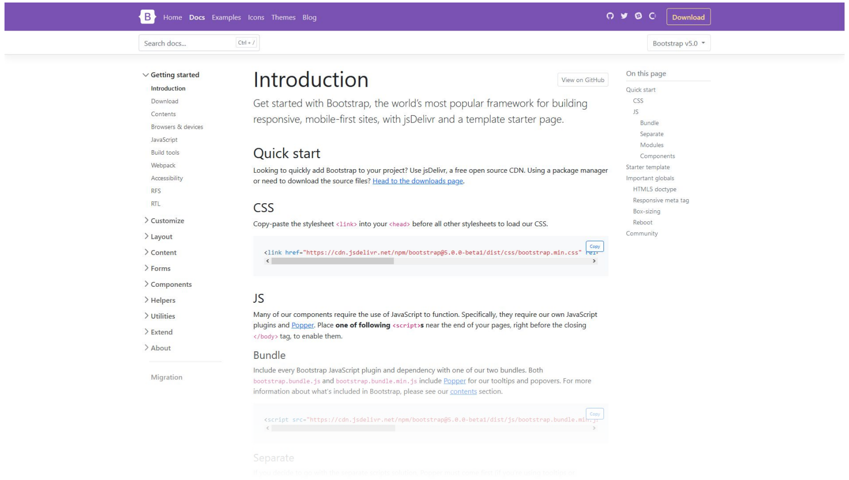
Locate the Download

For us to install the framework, we firstly have to locate the CSS code file(s) that make it up.

On the “Quick Start” page, there are two quick and easy ways to include Bootstrap in your project.

There is the anchor leading to the downloads page, but there is also the copy-paste <link> element snippet. The advantage of the external linked asset is that their content delivery network resource can likely serve the stylesheet faster than your web server.

The downside is that if they do maintenance, are hacked, or the service otherwise goes down: you’re out of luck!



Introduction

Get started with Bootstrap, the world’s most popular framework for building responsive, mobile-first sites, with jsDelivr and a template starter page.

Quick start

Looking to quickly add Bootstrap to your project? Use jsDelivr, a free open source CDN. Using a package manager or need to download the source files? Head to the downloads page.

CSS

Copy-paste the stylesheet `<link>` into your `<head>` before all other stylesheets to load our CSS.

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/css/bootstrap.min.css" rel=
```

JS

Many of our components require the use of JavaScript to function. Specifically, they require our own JavaScript plugins and Popper. Place **one of following** `<script>`**s** near the end of your pages, right before the closing `</body>` tag, to enable them.

Bundle

Include every Bootstrap JavaScript plugin and dependency with one of our two bundles. Both `bootstrap.bundle.js` and `bootstrap.bundle.min.js` include Popper for our tooltips and popovers. For more information about what’s included in Bootstrap, please see our contents section.

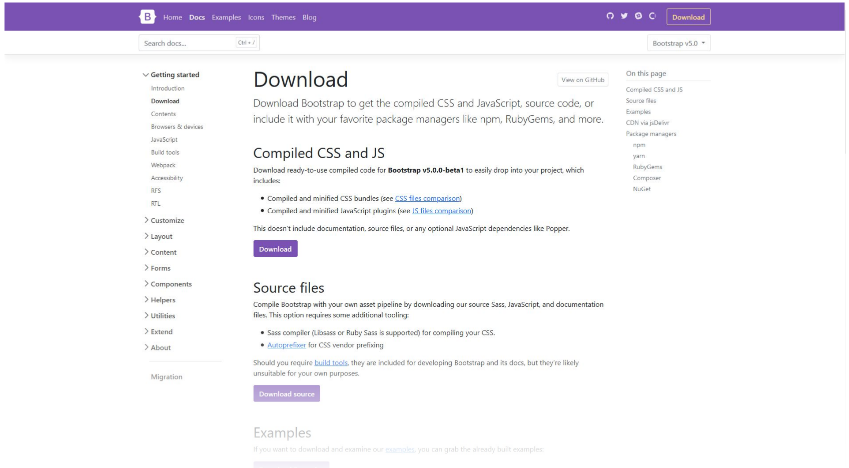
```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/js/bootstrap.bundle.min.js
```

Separate

Download the Framework File(s)

Compiled and minified assets are best, as these are smaller and will download quicker on a production website.

Once you download the ZIP, unpack it using a decompression tool. 7Zip is popular and free.



Download

Download Bootstrap to get the compiled CSS and JavaScript, source code, or include it with your favorite package managers like npm, RubyGems, and more.

Compiled CSS and JS

Download ready-to-use compiled code for **Bootstrap v5.0.0-beta1** to easily drop into your project, which includes:

- Compiled and minified CSS bundles (see CSS files comparison)
- Compiled and minified JavaScript plugins (see JS files comparison)

This doesn't include documentation, source files, or any optional JavaScript dependencies like Popper.

Source files

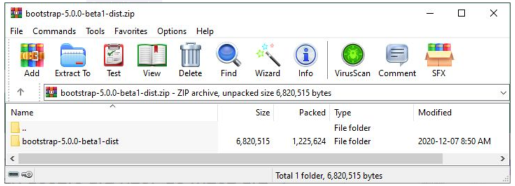
Compile Bootstrap with your own asset pipeline by downloading our source Sass, JavaScript, and documentation files. This option requires some additional tooling:

- Sass compiler (Libsass or Ruby Sass is supported) for compiling your CSS.
- Autoprefixer for CSS vendor prefixing

Should you require build tools, they are included for developing Bootstrap and its docs, but they're likely unsuitable for your own purposes.

Examples

If you want to download and examine our examples, you can grab the already built examples:



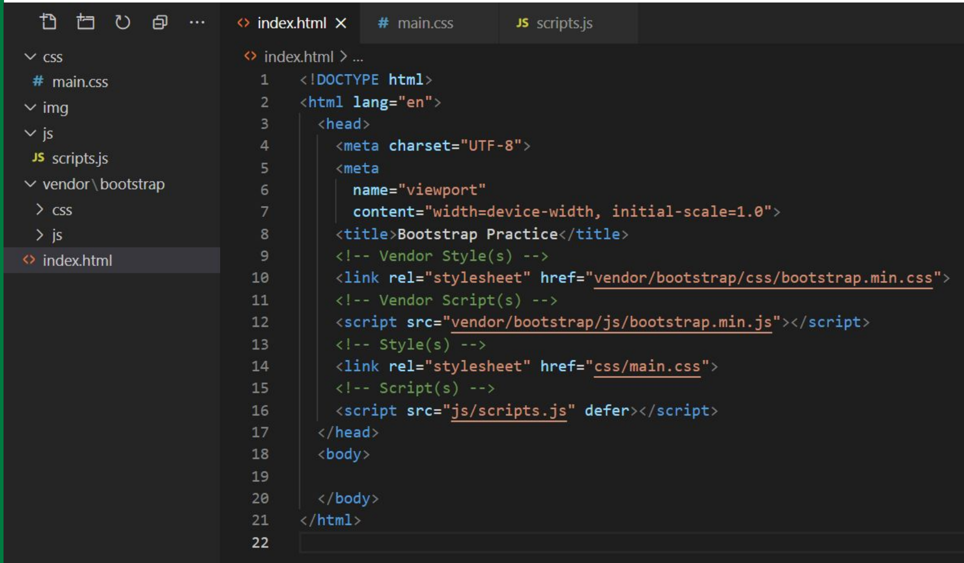
```
bootstrap-5.0.0-beta1-dist.zip - ZIP archive, unpacked size 6,820,515 bytes
Name                          Size     Packed   Type         Modified
..                                              File folder
bootstrap-5.0.0-beta1-dist    6,820,515 1,225,624 File folder 2020-12-07 8:50 AM
                                        Total 1 folder, 6,820,515 bytes
```

Organization

Staying organized is very important when working with frameworks, as it is easy to get lost if you're not careful.

It is recommended you keep a "vendor" folder in your project, and have separate folders inside for each framework or library you're using.

It can be helpful to also include subfolders for each framework or library to more easily coordinate image, font, CSS, and JavaScript assets.



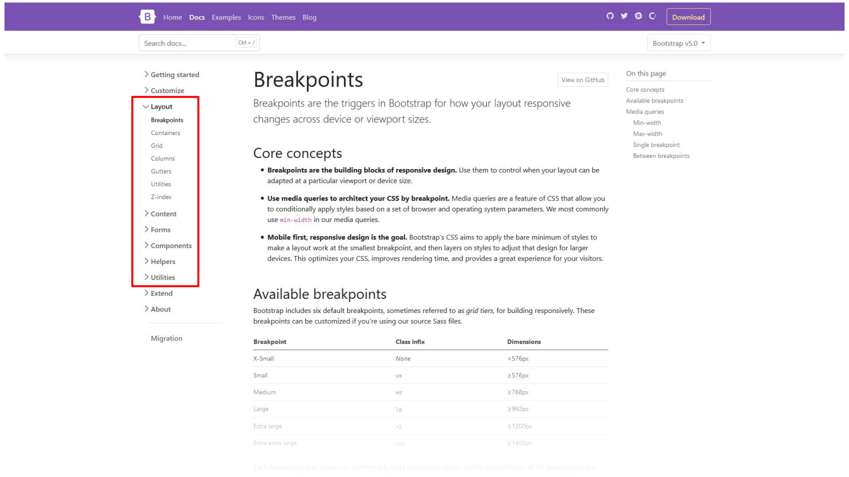
```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta
      name="viewport"
      content="width=device-width, initial-scale=1.0">
    <title>Bootstrap Practice</title>
    <!-- Vendor Style(s) -->
    <link rel="stylesheet" href="vendor/bootstrap/css/bootstrap.min.css">
    <!-- Vendor Script(s) -->
    <script src="vendor/bootstrap/js/bootstrap.min.js"></script>
    <!-- Style(s) -->
    <link rel="stylesheet" href="css/main.css">
    <!-- Script(s) -->
    <script src="js/scripts.js" defer></script>
  </head>
  <body>

  </body>
</html>
```

Finding What You Need

Bootstrap, being the most widely-used and mature CSS framework out on the web right now, has a lot of history and buzz—this makes it easy to find resources on!

If you visit their documentation, take a careful look at the sidebar. There are major topics covered on how many layouts and common appearances can be achieved.

When learning any technology, it is important to become familiar with any documentation associated with it.



Breakpoints

Breakpoints are the triggers in Bootstrap for how your layout responsive changes across device or viewport sizes.

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
X-Small	*None*	<576px
Small	`sm`	≥576px
Medium	`md`	≥768px
Large	`lg`	≥992px
Extra large	`xl`	≥1200px
Extra extra large	`xxl`	≥1400px

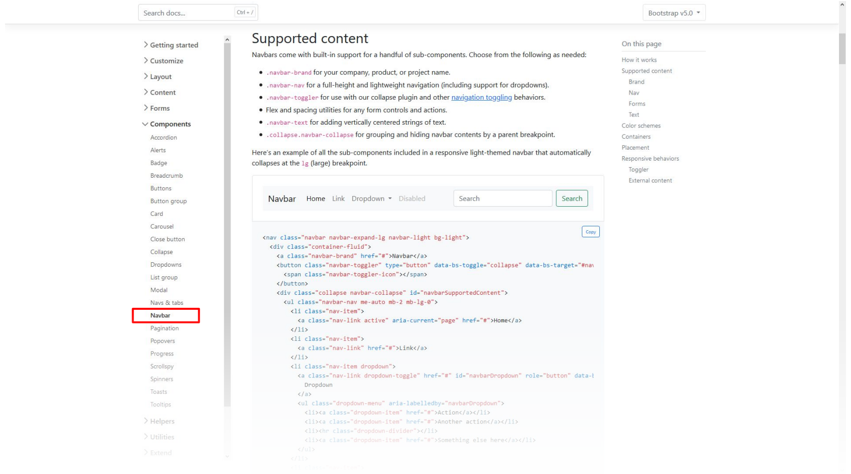
Each breakpoint was chosen to comfortably hold containers whose widths are multiples of 12. Breakpoints are

Navbar

If, for example, we want to add a navigation bar to our Bootstrap-enabled web page, locate the "Navbar" item in the sidebar and click it.

This page details how to use the related classes, and offers sample HTML for building out a navbar of your own.

Copy and paste the navbar code into the <body> of your HTML and observe the result. Note how easy it is to manipulate that HTML if you'd like to change the wording, number of links, etc.



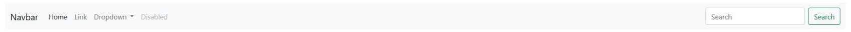
Supported content

Navbars come with built-in support for a handful of sub-components. Choose from the following as needed:

- `.navbar-brand` for your company, product, or project name.
- `.navbar-nav` for a full-height and lightweight navigation (including support for dropdowns).
- `.navbar-toggler` for use with our collapse plugin and other navigation toggling behaviors.
- Flex and spacing utilities for any form controls and actions.
- `.navbar-text` for adding vertically centered strings of text.
- `.collapse.navbar-collapse` for grouping and hiding navbar contents by a parent breakpoint.

Here's an example of all the sub-components included in a responsive light-themed navbar that automatically collapses at the `lg` (large) breakpoint.

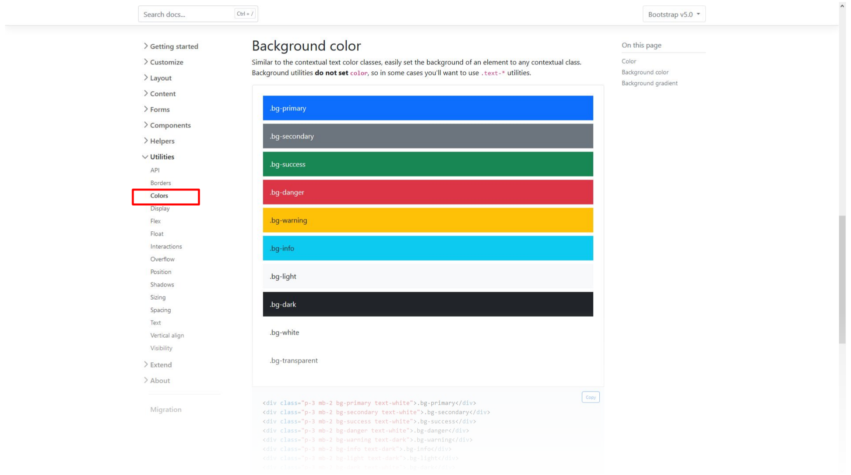
```html
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Navbar</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#na
      <span class="navbar-toggler-icon"></span>
    </button>
    <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="#" id="navbarDropdown" role="button" data-
            Dropdown
          </a>
          <ul class="dropdown-menu" aria-labelledby="navbarDropdown">
            <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>
```



Colours

For quick layout testing, using some of the default colouring can be useful as a visual indicator.

Note that you'll want to set up your own classes for colours later on that match your branding, opposed to these common defaults.



Background color

Similar to the contextual text color classes, easily set the background of an element to any contextual class. Background utilities **do not set** `color`, so in some cases you'll want to use `.text-*` utilities.

.bg-primary
.bg-secondary
.bg-success
.bg-danger
.bg-warning
.bg-info
.bg-light
.bg-dark
.bg-white
.bg-transparent

```
<div class="p-3 mb-2 bg-primary text-white">.bg-primary</div>
<div class="p-3 mb-2 bg-secondary text-white">.bg-secondary</div>
<div class="p-3 mb-2 bg-success text-white">.bg-success</div>
<div class="p-3 mb-2 bg-danger text-white">.bg-danger</div>
<div class="p-3 mb-2 bg-warning text-dark">.bg-warning</div>
<div class="p-3 mb-2 bg-info text-dark">.bg-info</div>
<div class="p-3 mb-2 bg-light text-dark">.bg-light</div>
```

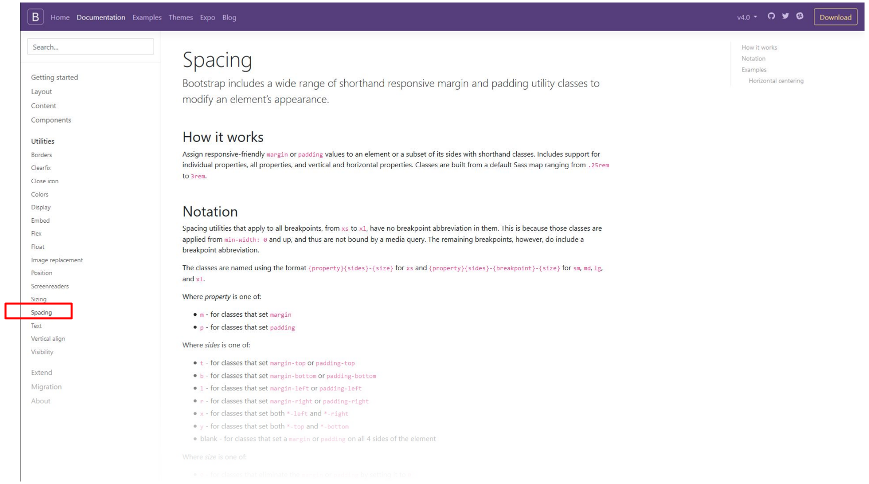
Spacing

There are some utility classes available for spacing in Bootstrap.

These include a few for both margin and padding. Either lets you choose if it will affect all sides, or a specific side of the element.

Pay close attention to the pattern in these examples…

- m-3
  Adds **margin: 1em;** as a style, affecting all sides.
- mt-2
  Adds **margin-top: 0.5em;** as a style.
- pr-0
  Adds **padding-right: 0;** as a style.



Spacing

Bootstrap includes a wide range of shorthand responsive margin and padding utility classes to modify an element's appearance.

How it works

Assign responsive-friendly `margin` or `padding` values to an element or a subset of its sides with shorthand classes. Includes support for individual properties, all properties, and vertical and horizontal properties. Classes are built from a default Sass map ranging from `.25rem` to `3rem`.

Notation

Spacing utilities that apply to all breakpoints, from `xs` to `xl`, have no breakpoint abbreviation in them. This is because those classes are applied from `min-width: 0` and up, and thus are not bound by a media query. The remaining breakpoints, however, do include a breakpoint abbreviation.

The classes are named using the format `{property}{sides}-{size}` for `xs` and `{property}{sides}-{breakpoint}-{size}` for `sm`, `md`, `lg`, and `xl`.

Where *property* is one of:

- `m` - for classes that set `margin`
- `p` - for classes that set `padding`

Where *sides* is one of:

- `t` - for classes that set `margin-top` or `padding-top`
- `b` - for classes that set `margin-bottom` or `padding-bottom`
- `l` - for classes that set `margin-left` or `padding-left`
- `r` - for classes that set `margin-right` or `padding-right`
- `x` - for classes that set both `*-left` and `*-right`
- `y` - for classes that set both `*-top` and `*-bottom`
- blank - for classes that set a `margin` or `padding` on all 4 sides of the element

Where *size* is one of:

- `0` - for classes that eliminate the `margin` or `padding` by setting it to `0`

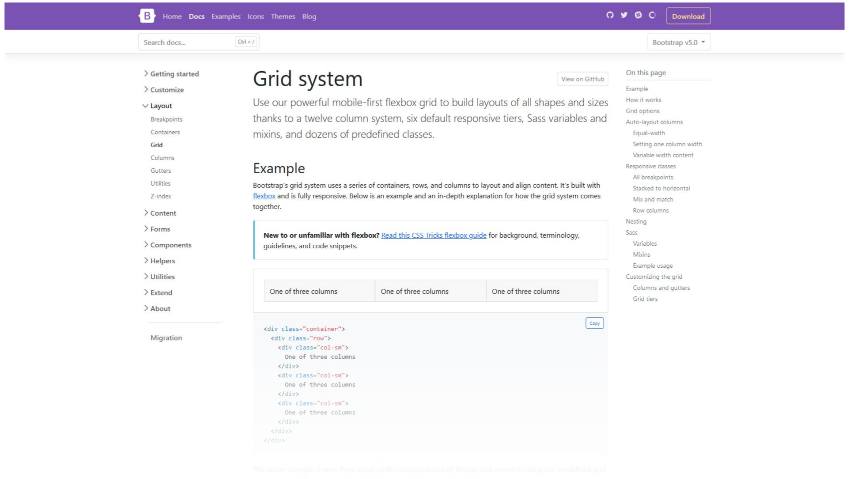
Grid System

One of Bootstrap's more powerful features, especially for rapid prototyping of web applications, is its grid system.

Using this system you can create all sorts of different modern layouts.

The important pattern and hierarchy of classes to note for this is as follows…

```
container
  row
    col
```



Grid system

Use our powerful mobile-first flexbox grid to build layouts of all shapes and sizes thanks to a twelve column system, six default responsive tiers, Sass variables and mixins, and dozens of predefined classes.

Example

Bootstrap's grid system uses a series of containers, rows, and columns to layout and align content. It's built with flexbox and is fully responsive. Below is an example and an in-depth explanation for how the grid system comes together.

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

One of three columns	One of three columns	One of three columns

```
<div class="container">
  <div class="row">
    <div class="col-sm">
      One of three columns
    </div>
    <div class="col-sm">
      One of three columns
    </div>
    <div class="col-sm">
      One of three columns
    </div>
  </div>
</div>
```

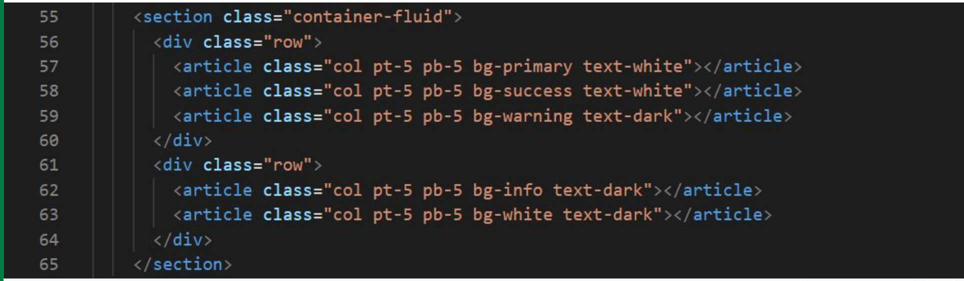
The above example creates three equal-width columns across all devices and viewports using our predefined grid

Using Bootstrap's Grid System

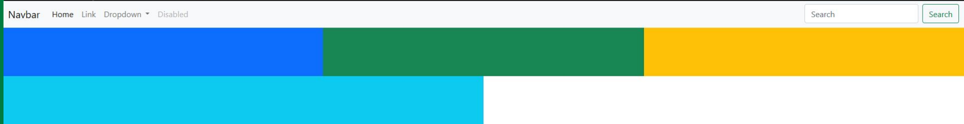
Let's try a basic layout.

Keep in mind the rule we looked at previously. For Bootstrap's grid to work properly, any column (col) should be inside of a row; any row should be inside of a container.

Insert the following snippet after your navbar.



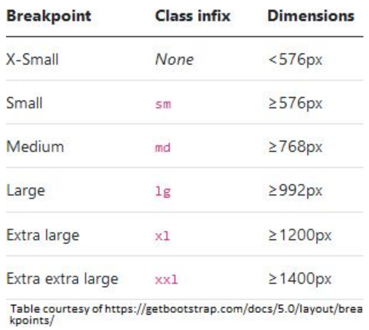
```html
<section class="container-fluid">
  <div class="row">
    <article class="col pt-5 pb-5 bg-primary text-white"></article>
    <article class="col pt-5 pb-5 bg-success text-white"></article>
    <article class="col pt-5 pb-5 bg-warning text-dark"></article>
  </div>
  <div class="row">
    <article class="col pt-5 pb-5 bg-info text-dark"></article>
    <article class="col pt-5 pb-5 bg-white text-dark"></article>
  </div>
</section>
```



Breakpoints

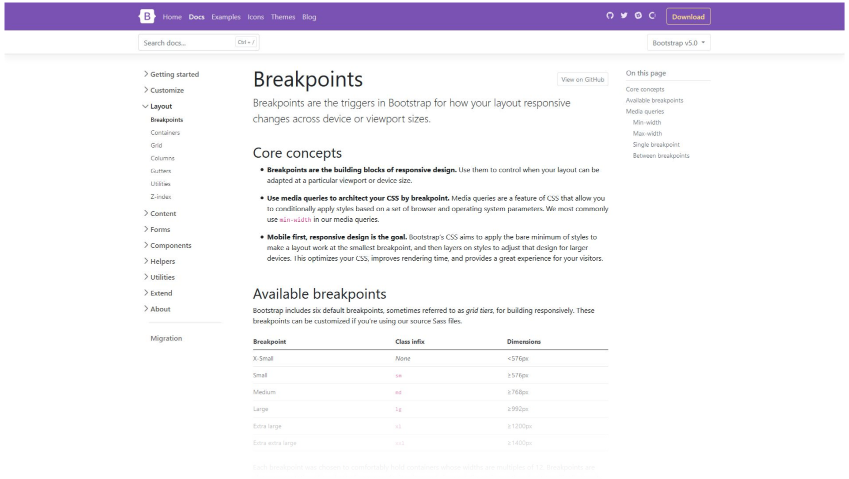
The folks at Bootstrap did not forget about responsive considerations.

For much of this, you'll be looking at their breakpoints layout classes, following this pattern…



Breakpoint	Class infix	Dimensions
X-Small	*None*	<576px
Small	`sm`	≥576px
Medium	`md`	≥768px
Large	`lg`	≥992px
Extra large	`xl`	≥1200px
Extra extra large	`xxl`	≥1400px

Table courtesy of https://getbootstrap.com/docs/5.0/layout/breakpoints/



Breakpoints

Breakpoints are the triggers in Bootstrap for how your layout responsive changes across device or viewport sizes.

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
X-Small	*None*	<576px
Small	`sm`	≥576px
Medium	`md`	≥768px
Large	`lg`	≥992px
Extra large	`xl`	≥1200px
Extra extra large	`xxl`	≥1400px

Each breakpoint was chosen to comfortably hold containers whose widths are multiples of 12. Breakpoints are

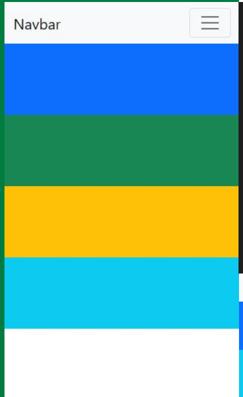
Responsive Columns

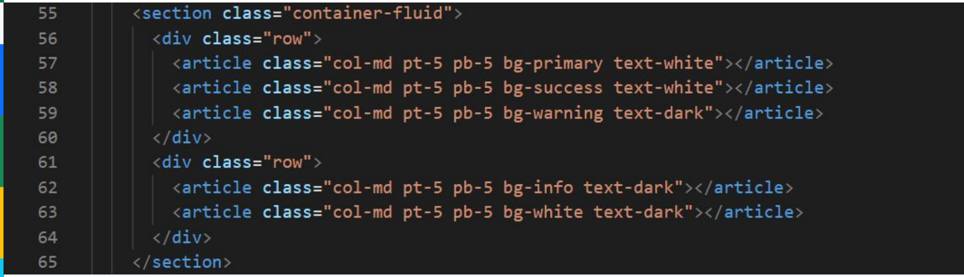
Get in the habit of working mobile-first.

By adding one of the "sizes" from the table previous to our "col" class, we can determine which screen size(s) will behave in the fashion described by the class.

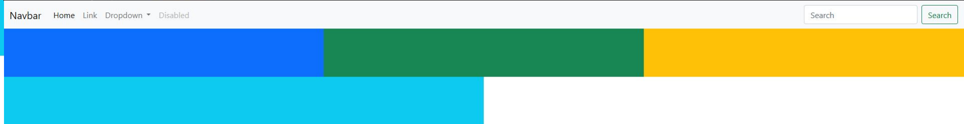
In our case, adding an "-md" to the end of our "col" classes will result in a single column for mobile, and the more complex layout on larger devices like desktop.

Update your code to match the following, and resize your browser.





```html
<section class="container-fluid">
  <div class="row">
    <article class="col-md pt-5 pb-5 bg-primary text-white"></article>
    <article class="col-md pt-5 pb-5 bg-success text-white"></article>
    <article class="col-md pt-5 pb-5 bg-warning text-dark"></article>
  </div>
  <div class="row">
    <article class="col-md pt-5 pb-5 bg-info text-dark"></article>
    <article class="col-md pt-5 pb-5 bg-white text-dark"></article>
  </div>
</section>
```



Exercise

Explore what Bootstrap has to offer. See if you can flesh out the web page a bit more—try adding a form! What other components and features have you found and tried?