sitepoint

BOOTSTRAP: RELATED TOOLS & SKILLS

BUILD RESPONSIVE, MOBILE-FIRST SITES WITH EASE

Bootstrap: Related Tools & Skills

- **Product Manager:** Simon Mackie
- **English Editor:** Ralph Mason
- **Project Editor:** Maria Antonietta Perna
- **Cover Designer:** Alex Walker

Notice of Rights

Notice of Liability

Trademark Notice

Published by SitePoint Pty. Ltd.

48 Cambridge Street Collingwood
VIC Australia 3066
Web: www.sitepoint.com
Email: books@sitepoint.com

Printed and bound in the United States of America

About SitePoint

SitePoint specializes in publishing fun, practical, and easy-to-understand content for web professionals. Visit http://www.sitepoint.com/ to access our blogs, books, newsletters, articles, and community forums. You'll find a stack of information on JavaScript, PHP, Ruby, mobile development, design, and more.

Table of Contents

Preface

Bootstrap stands as one of the most popular, open-source, front-end frameworks on the Web. Since its official release in 2011, it has undergone several changes, and it's now one of the most stable and responsive frameworks available. It's loved by web developers of all levels, as it gives them the capability to build a functional, attractive website design within minutes. A novice developer with just some basic knowledge of HTML and little CSS can easily get started with Bootstrap.

In this book, we'll look at some related tools and skills that will help you with Bootstrap development.

Who Should Read This Book?

This book is for all Frontend developers that want to build sites and apps that run faster. You'll need to be familiar with HTML and CSS and have a reasonable level of understanding of JavaScript in order to follow the discussion.

Conventions Used

You'll notice that we've used certain typographic and layout styles throughout this book to signify different types of information. Look out for the following items.

Code Samples

Code in this book is displayed using a fixed-width font, like so:

```
<h1>A Perfect Summer's Day</h1>
<p>It was a lovely day for a walk in the park.
```

```
The birds were singing and the kids were all back at school.</p>
```

Where existing code is required for context, rather than repeat all of it, ⋮ will be displayed:

```
function animate() {
  ⋮
new_variable = "Hello";
}
```

Some lines of code should be entered on one line, but we've had to wrap them because of page constraints. An ↪ indicates a line break that exists for formatting purposes only, and should be ignored:

```
URL.open("http://www.sitepoint.com/responsive-web-
↪design-real-user-testing/?responsive1");
```

Tips, Notes, and Warnings

Hey, You!

Tips provide helpful little pointers.

Ahem, Excuse Me ...

Notes are useful asides that are related—but not critical—to the topic at hand. Think of them as extra tidbits of information.

Make Sure You Always ...

... pay attention to these important points.

Watch Out!

Warnings highlight any gotchas that are likely to trip you up along the way.

Live Code

This example has a Live Codepen.io Demo you can play with.

Github

This example has a code repository available at Github.com.

Chapter 1

15 Bootstrap Tools and Playgrounds

David Attard

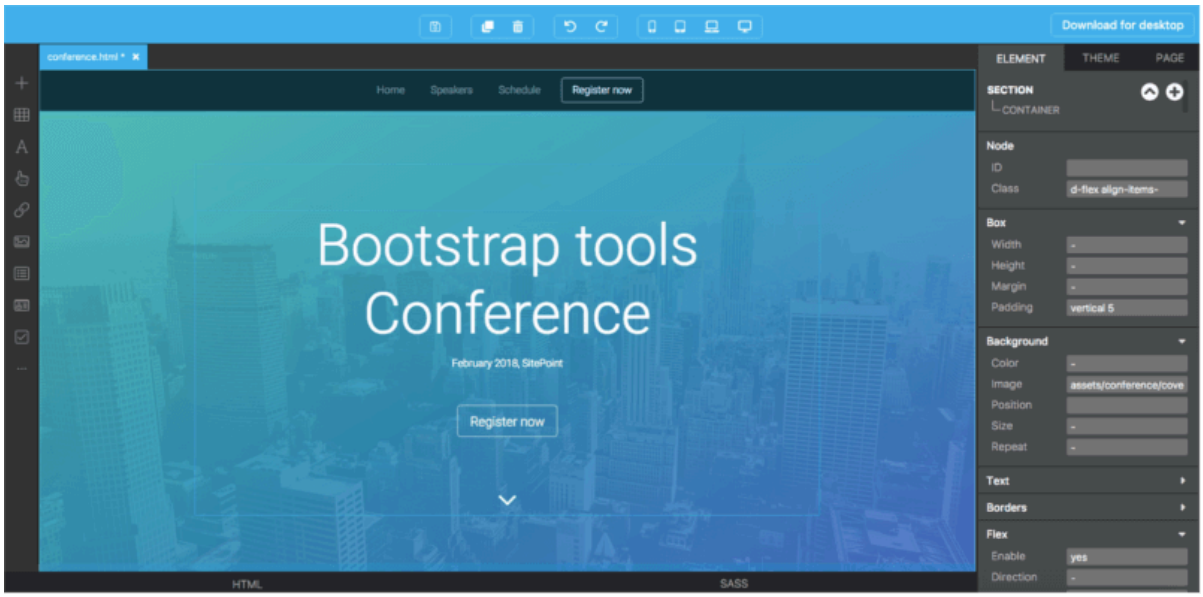
In this post, we've scoured the web for Bootstrap tools and playgrounds and shared with you only the very best out there.

Web designers and developers operate in a great industry. Our expertise and access to affordable development resources gives us the ability to do something unique — something which is found in few (if any) industries: the ability to release tools for other web designers and developers.

Tools for people like us are plentiful. Many of them are free, some are paid. All of them are awesome.

There are tools and playgrounds for almost everything — including Bootstrap. Let's review the best of them.

1. Pingendo



1-1. Pingendo

Price: Free for non-commercial use or $99 one-time payment

Pingendo is a Bootstrap 4 builder which is available in two flavors, an online

playground and a desktop version available for Windows, macOS and Linux.

Pingendo comes with quite a nice selection of templates to get you really bootstrapped with your web design. Amongst the available templates, you'll find an App intro site, a Conference site template and a Restaurant template, which comes in various themes.

There's also a number of wireframes ready for use, including a photo album, a cover page, a checkout form page, a landing page, a product page and a pricing table.

2. Brix

1-2. Brix

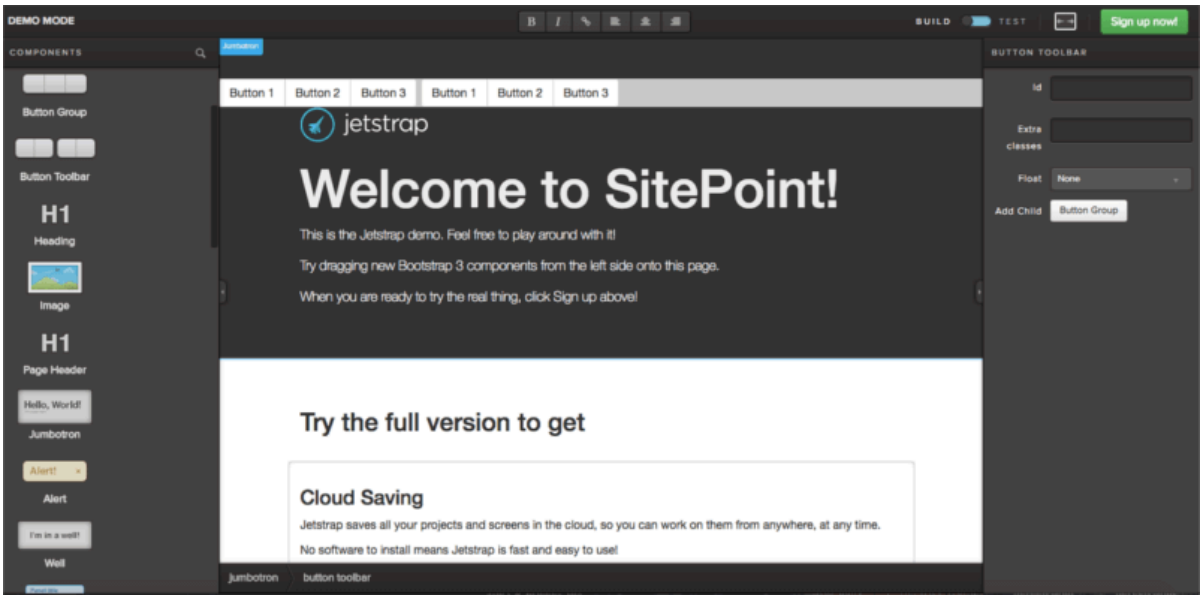
Price: from $14.90/month

Brix is a Bootstrap builder for designing, creating and editing responsive websites and UIs. The service is completely cloud-based, built as a rapid prototyping tool for the Bootstrap framework.

The tool derives from experience gathered since the beginning of Bootstrap.

More than 20 templates are also available to use as a starting point for your web pages.

3. Jetstrap



1-3. Jetstrap

Price: from $16/month for 3 projects

Jetstrap is a Bootstrap Interface builder which is a cross between a mockup tool and an interface-building tool, bringing a bit of both to the table. Actually, the great thing is that rather than mocking up your screens, you're actually building them on the fly.

The tool is fully web-based and includes drag-and-drop components and snippets of good clean markup ready for creating complicated components easily.

4. Pinegrow

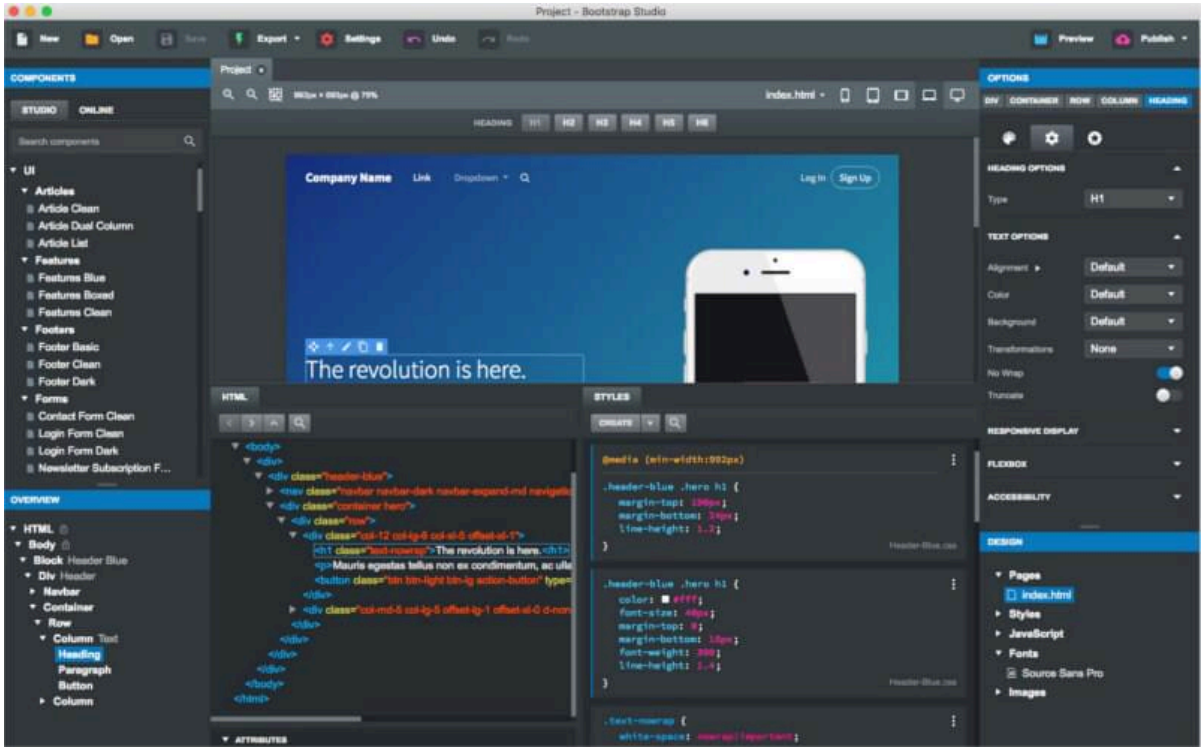
1-4. Pinegrow

Price: from $49

Pinegrow is a desktop web editor that allows you to build responsive sites using live, multi-page editing, CSS and Sass styling, and components for Bootstrap, Foundation and WordPress.

Available for macOS, Windows and Linux, you can develop using Bootstrap 3, 4, or other frameworks as you prefer.

5. Bootstrap Studio



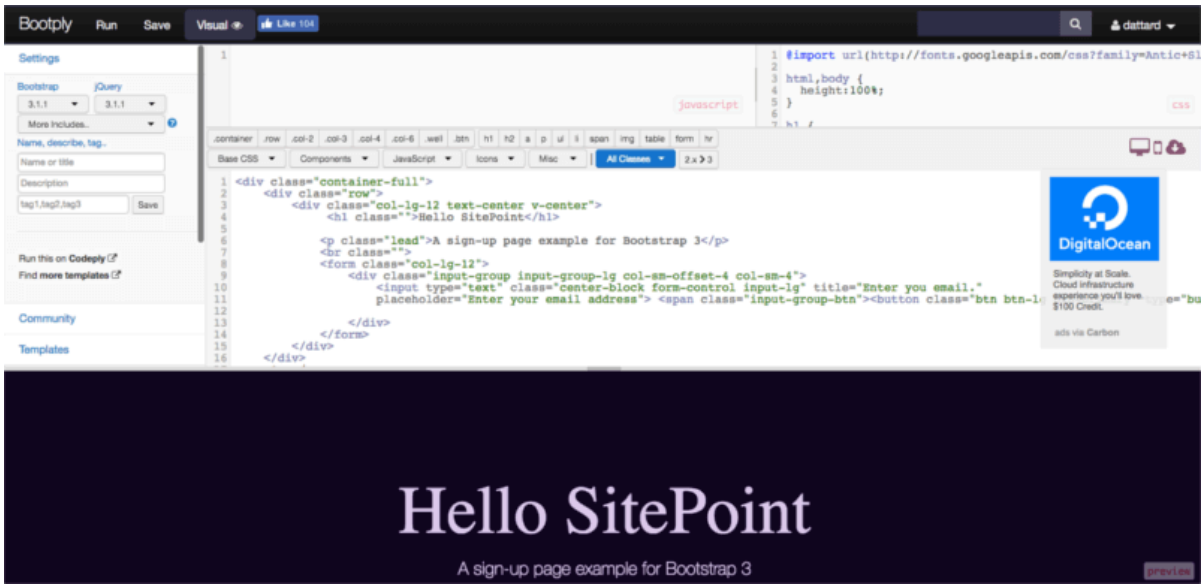
1-5. Bootstrap Studio

Price: from $25

Bootstrap Studio is a desktop app, but it does offer an online demo of its capabilities.

It's built around drag-and-drop functionality and comes with quite a good set of built-in components, including headers, footers, galleries, and slideshows.

6. Bootply



1-6. Bootply

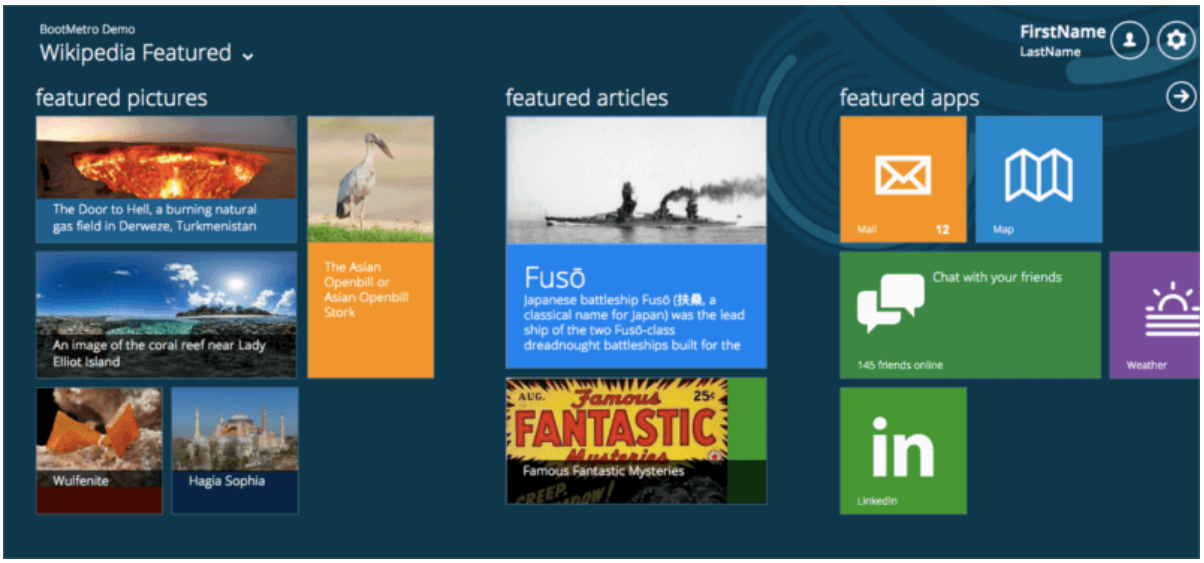
Price: Free with ads, $9/month

Bootply touts itself as a Bootstrap playground, editor and builder.

Of all the tools we've seen so far, this seems like the one which is most suited for those who like having the power of drag and drop but with the full ability of coding at hand. It allows you to switch between the Code Editor and the Preview so you can quickly check your build.

Bootply also comes with a number of pre-build starter templates to get you up and running quickly. Besides your run-of-the-mill landing page, single page app or article, you've also got more complex templates such as the Control Panel and the Dashboard templates and a modern layout for a tech news site.

7. BootMetro

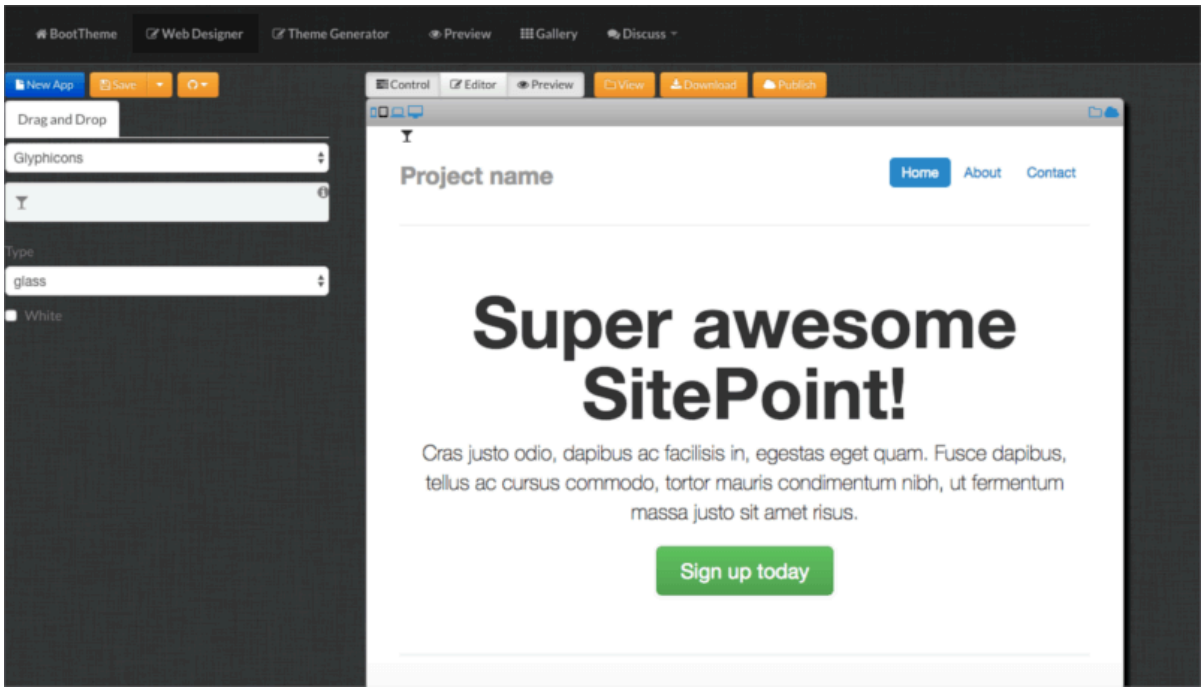


1-7. BootMetro

Price: Free

This is a simple UI framework which allows you to create a Metro-like interface using Bootstrap.

8. BootTheme



1-8. BootTheme

Price: Free or $9.99/month

With a tag line of *build a web page in 5 minutes, no coding required*, you can get an idea of where this tool is strongest.

Essentially, it's another UI builder that allows you to generate whole UIs using drag and drop and code tweaks as necessary.

9. JSFiddle with Bootstrap

Price: Free

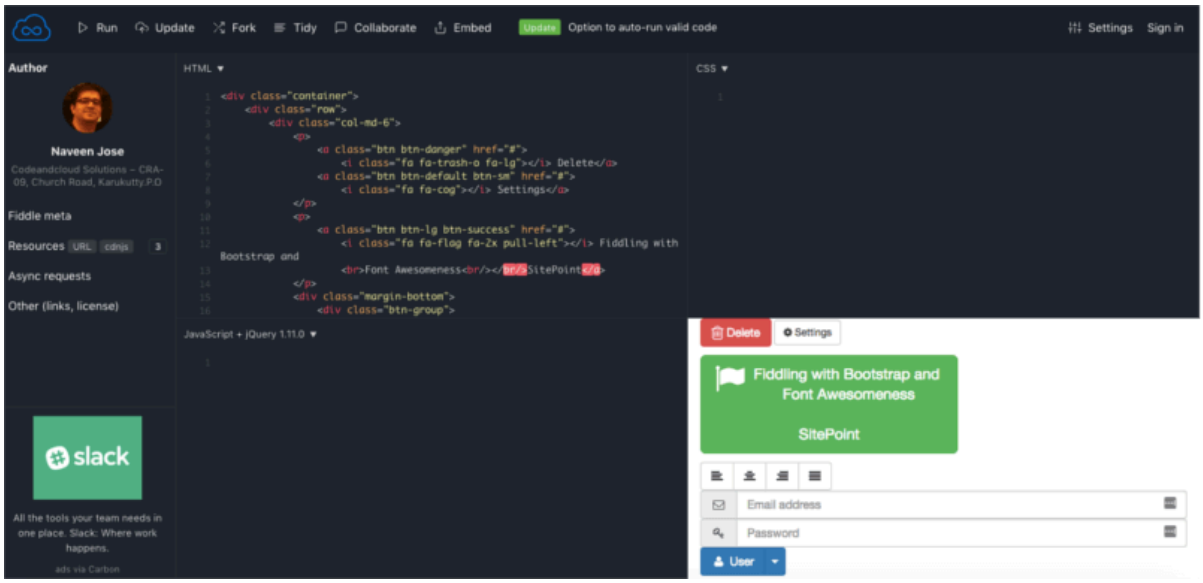
You've probably already come across JSFiddle, a popular JS playground. But what if you want to also fiddle around with Bootstrap?

The above is a JSFiddle workspace with Bootstrap resources and its dependencies added as external resources so you can JSFiddle in a Bootstrap environment.

This is for the control freaks — all those of you who want to do it all yourselves — so there are no components, no drag and drop, no nothing.

This is a blank slate for you to fiddle with, perfect for running a few tests to get stuff working your way, quickly and cheaply, with all the control you need.

10. JSFiddle (Bootstrap + Font Awesome)

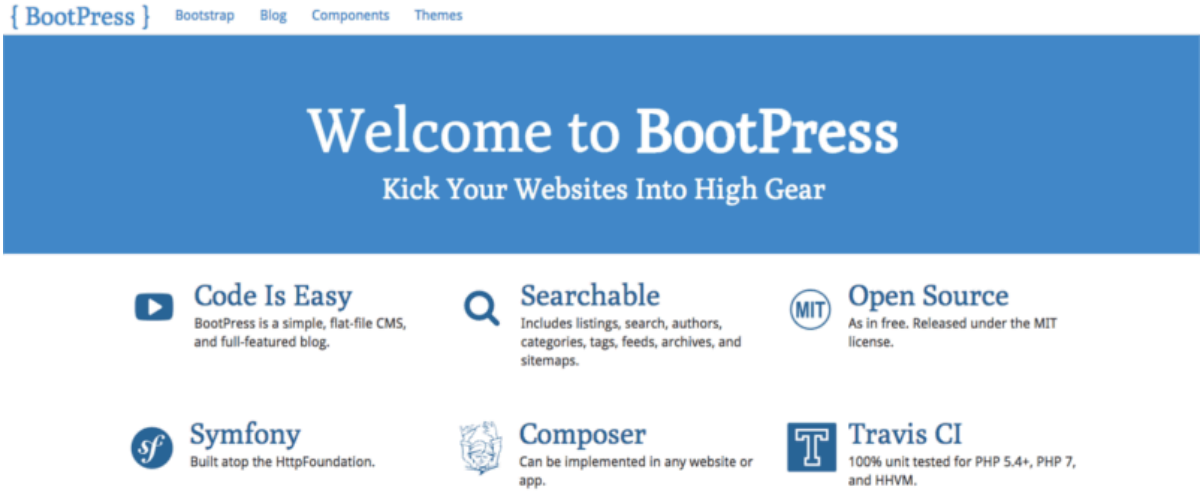


1-9. JSFiddle with Bootstrap and Font Awesome

Price: Free

While the previous tool involved just Bootstrap, this one is Bootstrap + Font Awesome, so you can also embed fonts and fiddle away to your heart's content.

11. BootPress

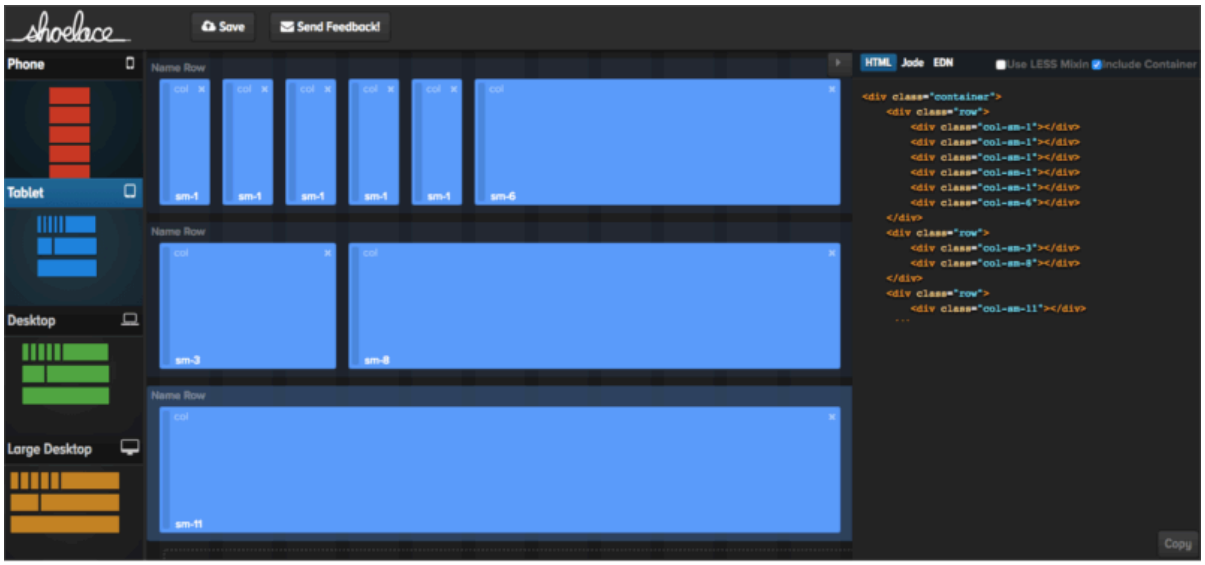


1-10. BootPress

Price: Free

BootPress (inspired by WordPress?) is actually a simple, flat-file CMS and blog framework which you can add lock-stock into your projects to quickly add this functionality to your Bootstrapped projects. Also, it has a number of wrappers and components for such stuff as pagination, tables, hierarchies, validation etc., together with a number of ready-made themes that you can use to start a blog or website.

12. Shoelace



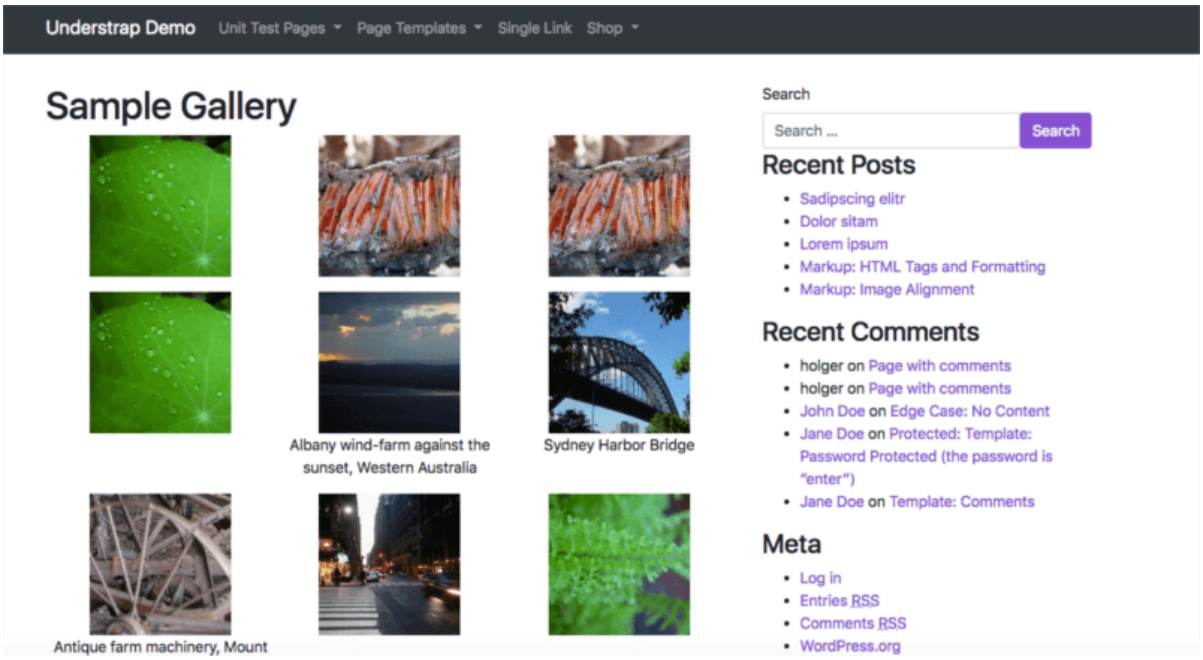
1-11. Shoelace

Price: Free

If you're struggling to get your Bootstrap grids done, you could try designing them visually and getting the code auto-generated for you, ready for pasting right into your project.

This is what Shoelace does for you, and it also includes all responsive Bootstrap classes taken care of for you.

13. UnderStrap

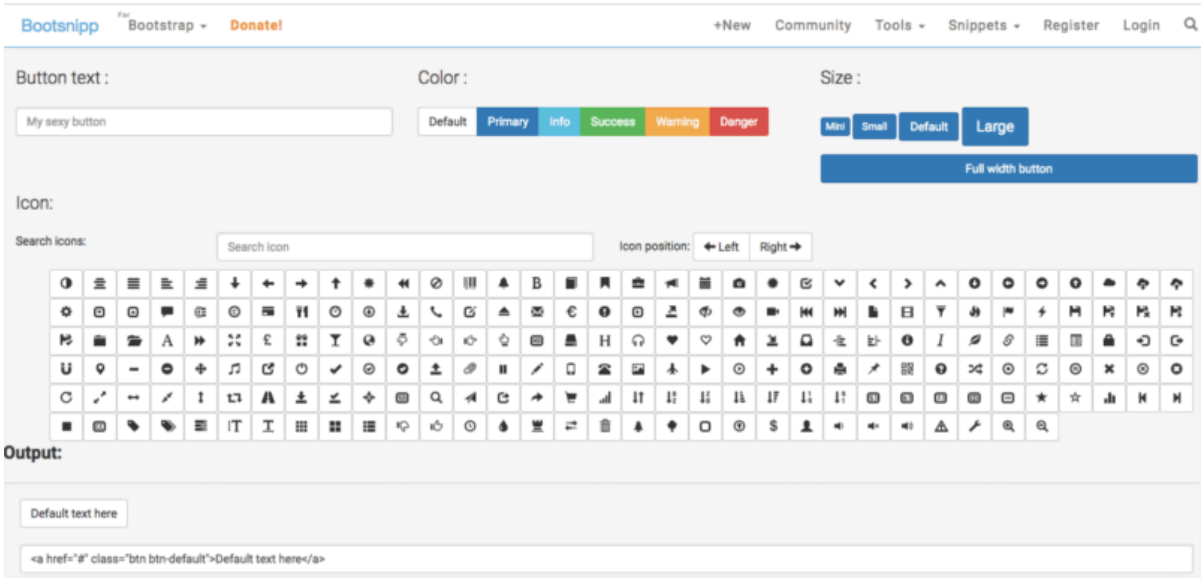


1-12. UnderStrap

Price: Free

Understrap is a bit of a combination of themes, WordPress and Bootstrap — a theme that's based on the Underscores theme from Automattic and Bootstrap 4, thus being an excellent starting point if you want to create a Bootstrap-powered WordPress theme.

14. Bootstrap Button Generator



1-13. Bootstrap Button Generator

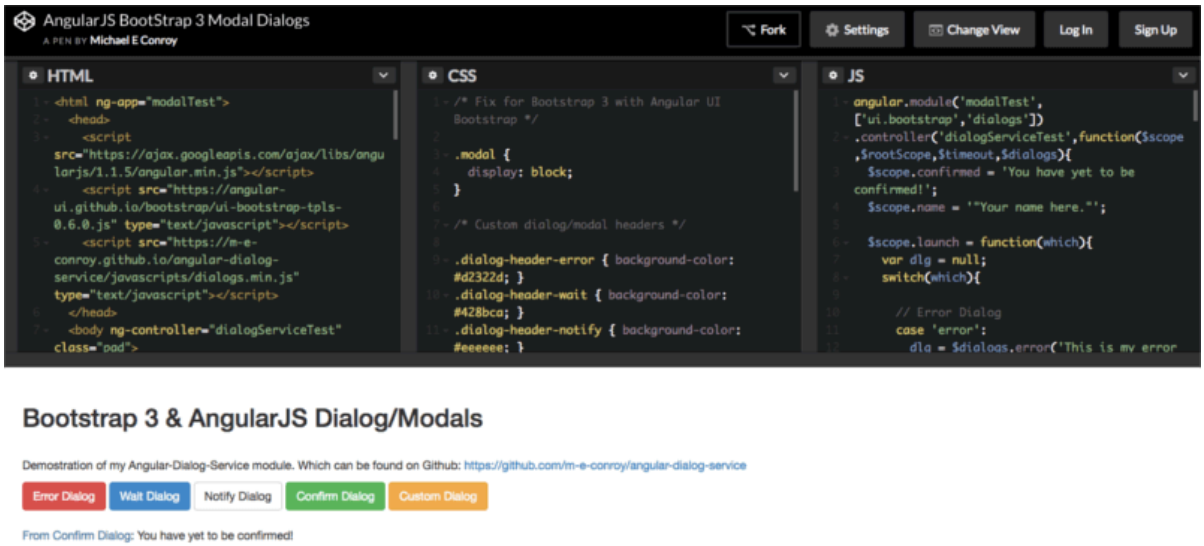
Price: Free

I strive to have buttons looking perfect in my projects. A good set of buttons is the real difference between a good UI and a great UI.

However, fiddling around with buttons can be time-consuming.

This bootstrap button generator shortcuts that time by providing an easy-to-use generator that you can tweak as you require.

15. CodePen



1-14. CodePen

Price: Free, Pro $8/month

We've left the best for last.

This is without a shadow of a doubt the most fully-fledged code playground out there and of course, it fully supports Bootstrap. Chris Coyier's pet project has quickly evolved into one of the most popular sites for web designers and developers out there.

CodePen is not just a playground. It's also a repository of other people's experiments that can be shared. So just to get you started with Bootstrap-built Pens, click on this link to find the most popular Pens tagged using Bootstrap.

As you have seen, Bootstrap tools and playgrounds are quite plentiful. You might actually be at a bit of a loss as to which to choose.

We'd highly recommend trying each of them out. Eventually you'll find your own preferred set of tools that you'll keep going back to. At that point, it's a question

of digging in and starting to design and code with Bootstrap!

Chapter

2

An Introduction to Grid Systems in Web Design

Callum Hopkins

In web design, grid systems are invisible structures that collect all the elements within a web page together. In this article, I'll provide an introduction to grid systems, explaining what they are, their purpose, and some of the theory behind them. Wikipedia explains a grid system as —

> a structure (usually two-dimensional) made up of a series of intersecting straight (vertical,horizontal, and angular) or curved guide lines used to structure content. The grid serves as an armature or framework on which a designer can organize graphic elements (images, glyphs, paragraphs, etc.) in a rational, easy-to-absorb manner.

The Nature of Grid Systems

Grid systems are never properly visible, but traces of a grid's "discipline" can be seen by the placement of elements within a web page. Grid systems also dictate the size of such design elements as widths of column texts, repeated placement of elements, padding around imagery, word spacing, line height, etc. A grid's main goal is to create a connection of unity within a design, which in turn makes web page content flow better, producing a more readable and enjoyable web page design.

Unity in Grid Systems

Alex White's Elements of Graphic Design explains the use of unity through a grid system:

> Unity in design exists where all elements are in agreement. Elements are made to look like they belong together, not as though they happened to be placed randomly ... So, without unity a design becomes chaotic and unreadable. But without variety a design becomes inert, lifeless, and uninteresting. A

> balance must be found between the two.

However, the benefits of a grid system on a particular design will only take effect if the grid is used at the initial stages of the design process. Attempting to implement a grid into an existing design will not create the same fluid layout or unity of content. This issue is discussed in Josef Muller-Brockmann's Grid System is Graphic Design. He writes:

> A suitable grid in visual design makes it easy: a) to construct the argument objectively with the means of visual communication; b) to construct the text and illustrative material systematically and logically; c) to organize the text and illustrations in a compact arrangement with its own rhythm; d) to put together the visual material so that it is readily intelligible and structured with a high degree of tension.

Grid Systems Beyond Web Design

Grids aren't just restricted to web and graphic design. Almost every profession where any form of design is implemented has a grid system, which professionals use as a guarantee for positive element positioning. It has become almost professionally vital to use grids in all modern design practices.

What Grid Systems Are Available?

Grid systems for web designers to use are everywhere on the Internet. The king of the grids has been deemed "The 960 Grid". The 960 Grid has the following structure:

- Total width 960px.
- 12 columns maximum, 60px each wide.

- Each column has 10px left & right margin. In turn, this makes 20px of gutter space on the edges.
- Total content area is 940px.

One of the main reasons for the 960 grid's popularity with designers is its flexibility. Designers can use a wide range of columns, with the maximum number being 12. The most popular examples are: 9 x 3, 3 x 3 x 3, 4 x 4 x 4 x 4 and 10 x 2. The grid has also been used on a wide range of websites, such as Assistly, Drupal, 51bits and Sony Music. To see these sites' grid systems in action, head over to the official 960 Grid home page. The 960 grid has always had a very close connection to "The Blueprint Grid", but they do have a lot of differences. To begin with, Blueprint has an in-depth setup for typography, whereas the 960 grid system setup only has minimal typography in place for a guideline rather than a shipped setup. I prefer this approach, because both grid systems were released some years ago and the web typography front has changed rapidly. Therefore the 960 grid system allows for more flexibility when using modern typography in designs. Another notable feature in the 960 grid framework is the lack of pull quotes in the setup. Nathan Smith (creator of the 960 grid) states:

> Those are things I rarely use, and I consider that to be a bit more design and contextually content oriented than related to page layout and prototyping.

I'm not entirely sure about the pull/push quote design, and frankly, I get slightly put off by their appearance in websites, because it makes me feel like someone is over compensating. In addition, if the pull/push quote design is produced without much thought or guidance, it can really pull the overall design of a website down. However, I do understand that it is a popular design element in modern websites, so appreciate Blueprint's inclusion of a template for the feature. This means that if I were creating a scoring system for 960 vs Blueprint, it's currently 1-1 and down to the my last key difference — vertical rhythm. **Vertical rhythm** is the consistent spacing between sentences and paragraphs

in website content, or in other words, line height. Off the bat, Nathan Smith again states that, while he feels vertical rhythm is an important factor in web design, he believes “it is fragile” and can easily be damaged by a content editor’s placing of an image within the text. My verdict: I’m going with Nathan Smith purely because, as a developer, I know how many CMSs will allow users to place images of any size into any position within the text. They have no regard for intricate vertical rhythm. There are many workarounds to deal with this, but I feel that for it to truly work is circumstantial. If you have a project where it will be respected, great, but for the majority of cases I think 960 has a more realistic attitude. It’s my opinion that the 960 grid framework is the better option, but honestly try them both out and see which one works best for you.

Other Grid Systems and Further Reading

There are loads of grid systems available for use. It’s all about personal opinion, so if you’re looking for grid systems, there’s only one place to go: The Grid System. For further reading, here’s a small list of articles that cover the grid systems I’ve talked about in more depth:

- 960 Grid System
- 960 Grid System is Getting Old
- Setting Type on the Web
- Gridding the 960
- The Funniest Grid System You Ever Saw
- The Use of Grids in Website Design

Chapter

3

Bootstrap UI Libraries for Angular, React and Vue.js

Maria Antonietta Perna

This chapter lists a number of resources that provide Bootstrap components for your Angular, React, and Vue.js apps.

These days, the web is populated with single page applications (SPAs) which feel fast and responsive to user interaction. Often, they're powered by JavaScript frameworks like Angular, React, and Vue.js.

These libraries are great tools that help keep the JavaScript code needed to build SPAs more organized and maintainable, while using clever techniques to update the DOM without slowing down the web page.

When it comes to styling the components you can create with these libraries, you're spoilt for choice: you can write your own global CSS document as you normally would on regular websites, you can write inline styles, or you can use UI frameworks — which is what I'll focus on here.

Bootstrap Interface Elements for Angular Developers

Angular is the oldest of the frameworks I deal with in this article. Its maturity and long life in the brutal life cycle of JavaScript frameworks is a sign of its robust and reliable foundation for web projects.

What tools are out there if you want to use Bootstrap to style your Angular app?

ng-bootstrap

3-1. ng-bootstrap

ng-bootstrap offers a set of Bootstrap-styled components that makes building Bootstrap apps with Angular a breeze.

More specifically, this resource contains a set of Angular directives based on Bootstrap's markup and styles. To use it, it's important that you don't include any Bootstrap JavaScript files. You only need Angular and Bootstrap CSS files.

ng-bootstrap is a free and open-source project. Feel free to contribute to it or ask for new features on GitHub.

ngx-bootstrap

3-2. ngx-bootstrap

This is another great, open-source modular library of Bootstrap components powered by Angular. You don't need to use jQuery or any Bootstrap JS code, although the markup and CSS are provided by Bootstrap.

To install the library using `npm`, type this line into the command line:

```
npm install ngx-bootstrap --save
```

Then include the link to Bootstrap's stylesheet in `index.html` document:

```
<link href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
↪rel="stylesheet">
```

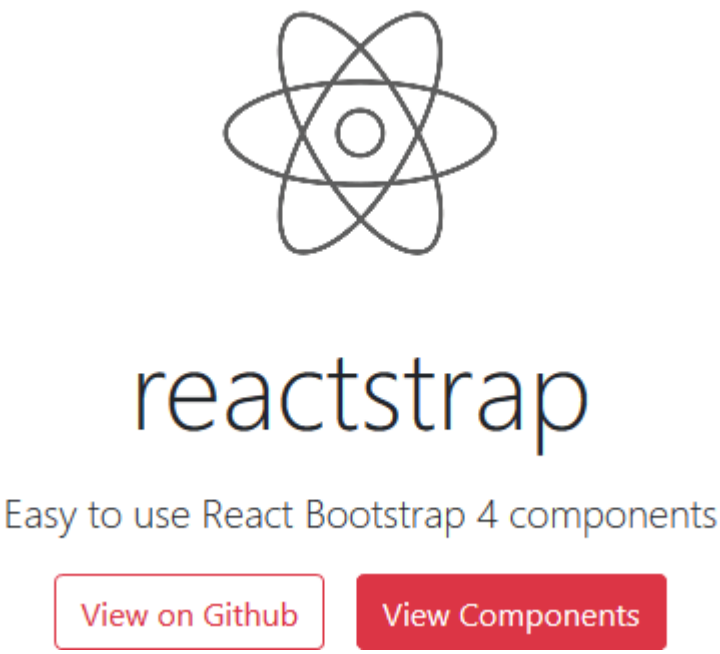
You can refer to the awesome docs pages on the ngx-bootstrap website for detailed explanations and tutorials.

Bootstrap Interface Elements for React Developers

React was created by the Facebook devs and is one of the trendiest JavaScript libraries for building modern user interfaces.

If you're looking to give your React app a Bootstrap makeover, here are some of the options available to you out there.

reactstrap



3-3. reactstrap

reactstrap is a fast and convenient library that lets you add a Bootstrap 4 look and feel to your React app.

The styling is provided by Bootstrap, while the JavaScript functionality is built into reactstrap components. This means that you don't need to include any jQuery or Bootstrap JS code. However, you need to include react-popper, since some dynamic Bootstrap components like dropdowns depend on Popper.js to work properly.

You can install reactstrap via a CDN:

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/reactstrap/
↪4.8.0/reactstrap.min.js">
```

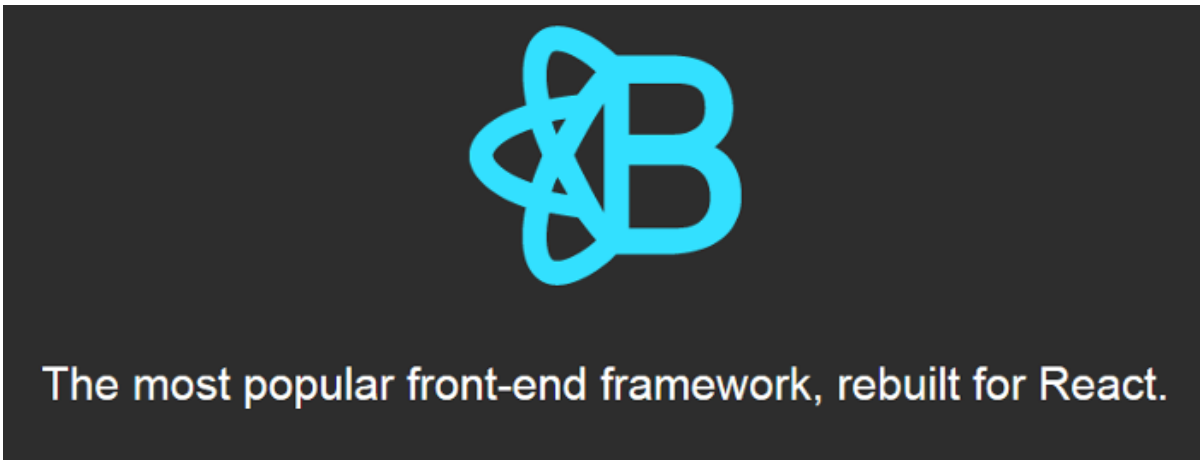
In this case, you need to include the React library and ReactTransitionGroup before reactstrap.

You can also install the library and all dependencies with NPM:

```
npm install --save reactstrap@next react react-dom
```

If you're curious, hop over to the docs and learn more about each component and how to integrate reactstrap with the Create React App setup.

React-Bootstrap



3-4. React-Bootstrap

React-Bootstrap is an open-source project which is still under active development to reach the 1.0.0 release.

Since this library doesn't support any specific Bootstrap version, you're free to include any Bootstrap stylesheet you think best suits your project. Generally speaking, including the latest Bootstrap release will ensure that you enjoy the benefits of improvements and bug fixes, so this should be your preferred option.

The most straightforward way of including Bootstrap in your React app is via a CDN:

```
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/
↪bootstrap.min.css">
```

On the JavaScript front, you just need React; no jQuery or Bootstrap JS is required for Rect-Bootstrap to work. You can install this Bootstrap UI library using NPM:

```
npm install --save react react-dom
```

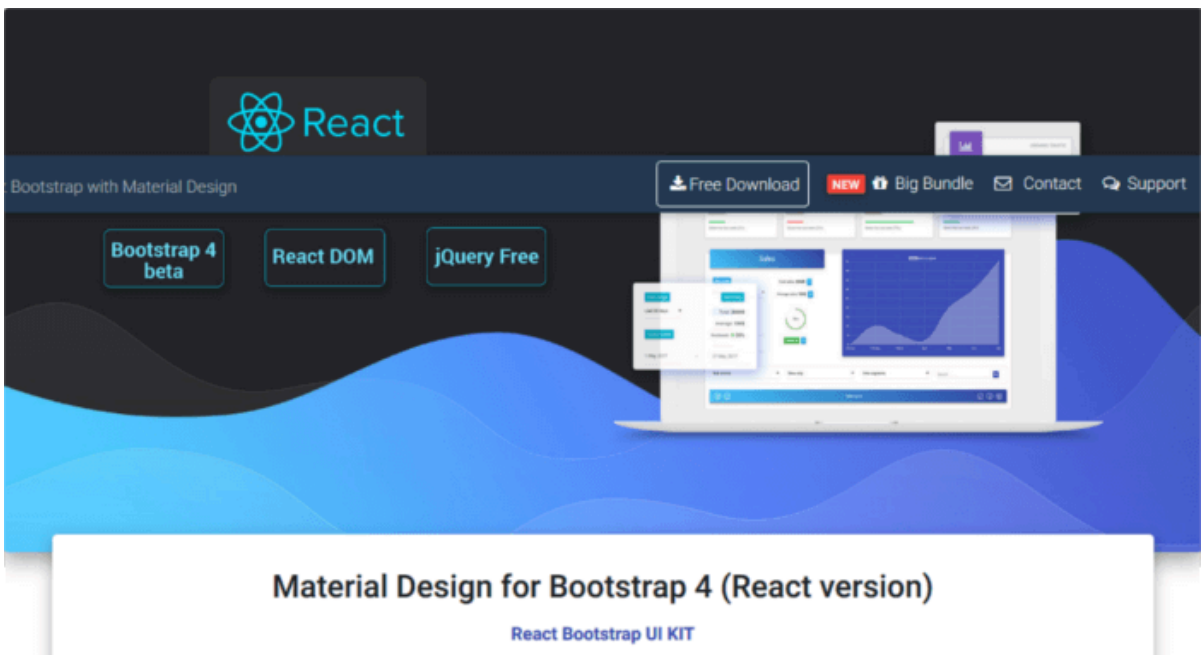
```
npm install --save react-bootstrap
```

Installing the library this way will allow you to import individual components from `react-bootstrap/lib` rather than the entire library. Doing so pulls in only the individual components you need rather than the entire package, which is great for keeping the file size under control.

Alternatively, you can include React-Bootstrap and the React library as a bundle using CDNJS and reference the files in your `<script>` tags:

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/react/<react-version>/react.min
↪.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/react/<react-version>/react-dom
↪.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/react-bootstrap/<version>/react
↪-bootstrap.min.js"></script>
```

Material Design for Bootstrap React Version



3-5. Material Design for Bootstrap (React version)

Material Design for Bootstrap (also available for Angular and Vue) is a UI library for React that combines the look and feel of Bootstrap and Google's Material Design.

Among its advantages are:

- being up to date with the latest release of Bootstrap
- its awesome documentation and tutorials
- being compatible with the latest release of React
- its use of JSX
- its easy installation
- being free for personal and commercial projects.

To get started with Material Design for Bootstrap, just download it straight from the project's website, or use NPM:

```
npm install mdbreact
```

or use yarn:

```
yarn add mdbreact
```

Once you've got the files, everything is linked appropriately and ready to go. Run `npm install` then `npm start` and get coding.

Bootstrap Interface Elements for Vue Developers

Vue.js hasn't been around long, but it's steadily been gaining ground in popularity due to a number of benefits that come with it:

- it's easy to get started: you can quickly include it in an HTML project via a CDN and incorporate it in your regular HTML document using its directives
- you can decide to incorporate Vue incrementally by starting with a few

features and eventually embrace it as a full-blown framework
- it has a small file size and it's very performant out of the box.

To integrate Bootstrap into your Vue-based app, Bootstrap-Vue is a nice option.

Bootstrap-Vue

3-6. Bootstrap-Vue

Bootstrap-Vue introduces itself on its website as the UI library that:

> *provides one of the most comprehensive implementations of Bootstrap V4 components and grid system available for Vue.js 2.4+, complete with extensive and automated WAI-ARIA accessibility markup.*

The documentation is great and you can also try out the library before including it in your project on the Bootstrap-Vue playground.

Conclusion

This chapter has listed a number of UI libraries to help you give your SPA's components a Bootstrap look and feel. These are fully functional Bootstrap components that work with Angular, React, and Vue out of the box, without extra work required from you.

Chapter

4

Bootstrap Grid: Mastering the Most Useful Flexbox Properties

Ahmed Bouchefra

In this article, I'll introduce you to the key Bootstrap CSS classes for building layouts with the Bootstrap grid system.

Bootstrap 4 uses Flexbox as the basis for its grid system. I'll explain the Flexbox CSS properties that underlie the new grid's functionality, and define how the Bootstrap flex utility classes work to help you build awesome layouts quickly and painlessly.

What is Flexbox?

Let's first introduce Flexbox. It stands for *flex*ible *box*, and it's a cutting-edge CSS layout system that makes it easy to create layouts for dynamic or unknown screen sizes. (The flex container has the ability to adjust and control the size of its child elements to adapt to different viewports.)

You can easily create a Flexbox layout using a set of CSS properties designed for this task.

Bootstrap makes it even easier to create Flexbox-based layouts by providing a set of wrapper classes on top of Flexbox properties, which you can simply apply to your markup to achieve your desired result.

Introduction to the Bootstrap Grid System

Grid systems are an important element of a CSS framework, since building complex layouts without a powerful and flexible grid system can be an intimidating task.

Among the new features of the latest Bootstrap grid system, you'll find the `xl` (extra large) grid breakpoint, corresponding to an extra large screen size, and the use of Flexbox instead of floats as the underlying layout mechanism.

Key Classes of the Bootstrap Grid System

You can build a layout using Bootstrap's grid system by applying a bunch of Bootstrap classes: `.container`, `.row` and `.col-*-*`. (The first * in `.col-*-*` needs to be replaced by the breakpoint specifier — such as `xs, sm, md, lg, xl` — and the second * needs to be filled with the column span size. The sum of all columns has to be equal to 12.)

Let's now look at the core components of the Bootstrap grid.

Container

The container is the outer wrapper for the grid layout. It's a `div` that has either the class `.container` for fixed width or `.container-fluid` for a *100%* full width.

Row

A row serves as a logical container for columns.

Column

A column is what makes a block in the grid. It should be contained in a row.

The Bootstrap grid system provides these additional column classes:

- `.col-xs-*`: designed for extra small screens of less than *576px* width
- `.col-sm-*`: designed for small screens with the width equals to or greater than *576px*
- `.col-md-*`: designed for medium screens with the width >= *768px*
- `.col-lg-*`: designed for large screens with the width >= *992px*
- `.col-xl-*`: designed for extra large screens that have a width that equals to or is greater than *1200px*.

You don't need to add multiple classes if you want to specify the same width for different screen sizes; just add the class with the smallest breakpoint. So, for example, instead of `.col-sm-6` and `.col-md-6`, you only need to apply `.col-sm-6`.

Bootstrap Grid Layouts with Flexbox vs Floats

Thanks to Flexbox, you can easily achieve things like same-height columns or same-width columns, which you could only accomplish with CSS hacks before.

CSS *float* and *clearfix* techniques to build layouts have been among such hacks, which made it hard to build and debug complex layouts.

For example, consider a two-column layout. If you build this layout with Bootstrap 3, it will look like this:

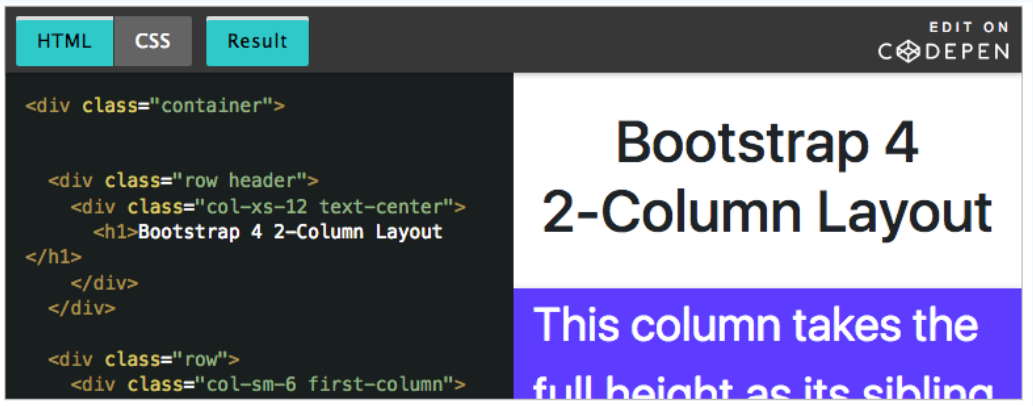
Live Code

See the Pen Bootstrap 3 2-Column Layout.

If you build the same layout with Bootstrap 4, this is what you have:

Live Code

See the Pen Bootstrap 4 2-Column Layout.



With Bootstrap 4 and its flexbox-based grid, you achieve a more realistic column (just like in a table), as columns in the same row will take the same height.

Let's tackle same-width column layouts. Thanks to Flexbox, you can easily divide the available space between multiple columns in the same row. If you create a grid layout with multiple columns without specifying the column width (i.e using the `.col-*` classes), the available space will automatically and equally be divided among those columns.

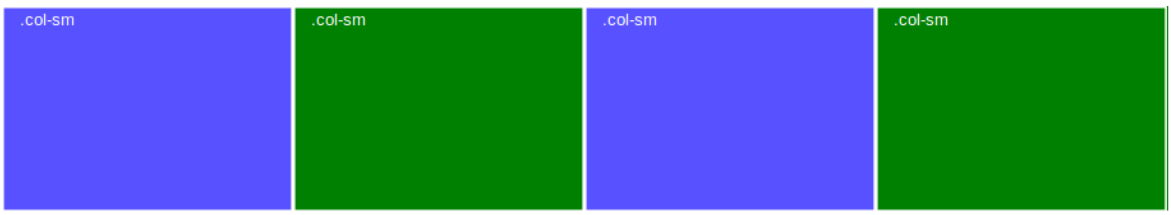
Here's an easy and quick example:

```
<div class="row">
   <div class="col-sm" style="min-height:200px;">
     .col
   </div>
   <div class="col-sm">
     .col
   </div>
   <div class="col-sm">
     .col
   </div>
```

```
    <div class="col-sm">
      .col
    </div>

  </div>
```

The four instances of `.col-sm` will each automatically be *25%* wide from the small breakpoint and up.

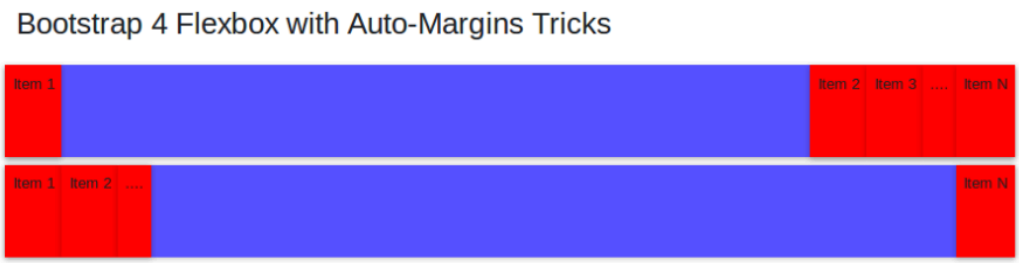
With some minimal styling, here's what you get:



4-1. Bootstrap grid: four columns

Flexbox with Auto Margins

Combining Flexbox with auto margins results in some cool tricks.



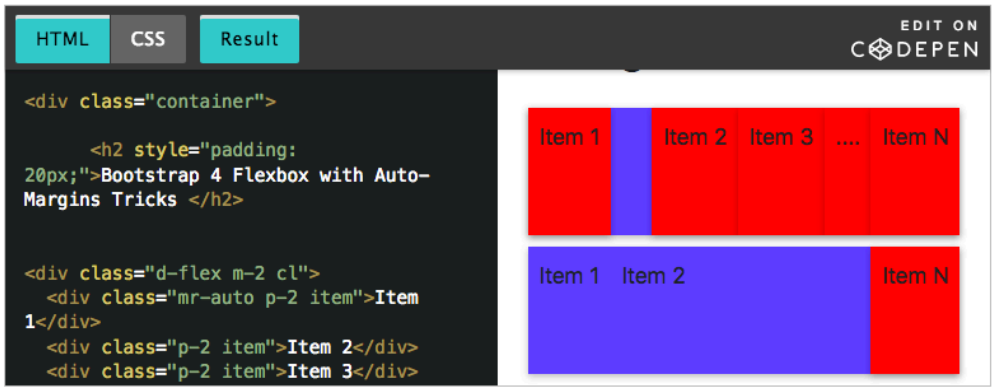
4-2. Bootstrap grid: combining Flexbox with auto margins

For example, look at the above layout: you can position elements to the right of

an item by adding a Bootstrap `.mr-auto` class to the item, which stands for `margin-right: auto;` in regular CSS, or also position some elements to the left of a specified item by using the Bootstrap `.ml-auto` class (`margin-left: auto;` in regular CSS). You can see this as moving the item with the `.mr-auto` or `.ml-auto` classes to the right-most or left-most position respectively, and the other elements to the opposite direction.

Live Code

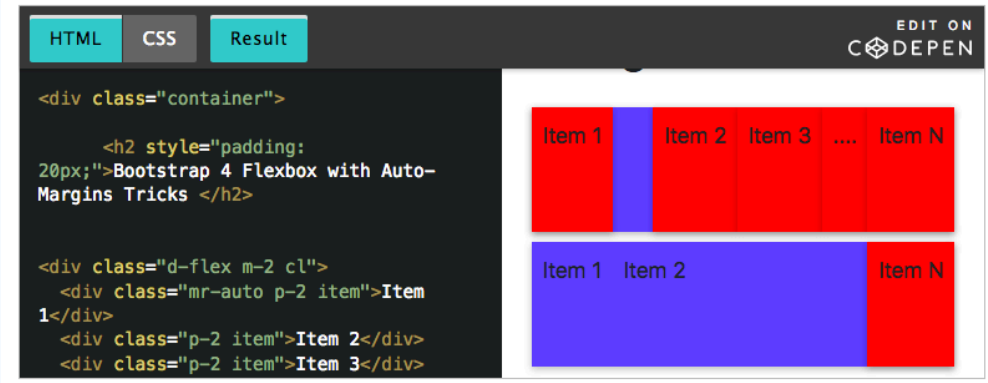
See the Pen Bootstrap 4 Flexbox with Auto-Margins Tricks.



You can accomplish this result either horizontally or vertically. To achieve the same behavior moving flex elements to the top or bottom (rather than to the right or to the left), you need to use `mb-auto` (`margin-bottom: auto;`) and `mt-auto` (`margin-top: auto;`), set `flex-direction` to `column` and apply the `align-items-(start|end)` class.

Live Code

See the Pen Bootstrap 4 Flexbox with Auto-Margins Tricks.



Useful Flexbox Concepts to Handle Bootstrap Flex Utility Classes

Bootstrap 4 uses specific flex utility classes, which might seem somewhat esoteric to someone who's never heard of Flexbox or doesn't know how flex containers and flex items behave.

For instance, Bootstrap now applies the `display:flex` property to its grid container elements. Also, Bootstrap lets you turn any HTML container into a flex container by simply applying the `.d-flex` class to your chosen element.

There are also available responsive classes such as `.d-sm-flex` and `.d-md-flex`, etc.

However, if you don't know what a flex container is and how it affects its child elements, using Bootstrap flex utility classes could be a bit problematic. The same could be said of all the other flex utilities like `.flex-row`, `.flex-row-reverse`, `.flex-column` and `.flex-column-reverse`.

Let's go over a quick review of how Flexbox works. It's most likely you'll find this useful when handling Bootstrap flex utility classes.

Flex Containers

Flexbox defines a flex container by applying the `display` property with the `flex` or `inline-flex` values:

```
.mycontainer {
   display: flex;
 }
```

The Bootstrap flex utility class for creating a flex container is `d-flex`.

Flex Items

Every *direct child element* of a flex container is turned into a flex item.

You can define the *direction* of flex items using the `flex-direction` CSS property with one of the following values: `row`, `row-reverse`, `column` and `column-reverse`.

- `row` sets the horizontal direction from left to right
- `row-reverse` sets the horizontal direction from right to left
- `column` sets the vertical direction from top to bottom
- `column-reverse` sets the vertical direction from bottom to top.

Bootstrap uses the classes `flex-row`, `flex-row-reverse`, `flex-column`, and `flex-column-reverse` to determine the direction of flex items.

Also, Flexbox lets you explicitly alter the *visual order* of specific flex items by using the `order` property, which defaults to zero:

```
.item {
   order: 1;
 }
```

Bootstrap offers its own order utility classes for making a flex item first, last, or for resetting the order property to the default DOM order. For example, to make a flex item appear first with respect to its siblings, add `order-1` to the markup.

Alignment of Flex Items

Flexbox makes it quick and easy to align flex items in any way you like.

For example, the `justify-content` property on the flex container lets you align flex items on the main axis (the x-axis by default, which you can change by setting the `flex-direction` to `column`). Available values are:

- `flex-start`: this is the initial value, and will line up items at the start of the container
- `flex-end` aligns items to the end of the parent element
- `center` aligns items to the center of the container
- `space-between` creates a space in between flex items after they've been laid out
- `space-around` creates an equal amount of space to the right and left of each flex item.

Bootstrap utility classes for applying `justify-content` values to elements are:

- `justify-content-start`
- `justify-content-end`
- `justify-content-center`
- `justify-content-between`
- `justify-content-around`

The Flexbox `align-items` property allows you to change the alignment of flex items in the cross axis. (If you set the main axis to be horizontal, the cross axis will be vertical, and vice versa.)

Possible values for `align-items` are:

- `stretch` : this is the initial value, which causes flex items to stretch to the height of their tallest sibling elements
- `flex-start` lines up the items at the start of the flex container
- `flex-end` lines up the items at the end of the flex container
- `center` is responsible for centering the flex item inside its container.

You can quickly apply this behavior with the following Bootstrap classes:

- `align-items-stretch`
- `align-items-start`
- `align-items-end`
- `align-items-center`

Check out the code in this pen for an example of how you can apply Bootstrap flex utility classes:

Live Code

See the Pen Bootstrap 4 Vertical Alignment.

You can find other available utilities in the Bootstrap flex docs.

Conclusion

In this article, we looked at how Flexbox makes the Bootstrap grid system more versatile, with features such as automatic same-width layouts and same-height columns. We then looked at some Flexbox properties that are key to mastering

Bootstrap flex utility classes and getting the most out of their powerful layout capabilities.

Chapter

5

Bootstrap Sass Installation and Customization

Reggie Dawson

Bootstrap is a popular, open-source framework. Complete with pre-built components, it allows web designers of all skill levels to quickly build a site. The latest version of Bootstrap uses Sass as the preprocessor of choice. In this article, I'll show you how to configure and customize a Bootstrap Sass-based project.

Bootstrap Installation

There are multiple ways to obtain and install the Bootstrap source files. Just remember that whatever method and package manager you choose, make sure you have a Sass compiler and Autoprefixer up and running. Bootstrap needs both in order to work.

Download Bootstrap files

We can download the Bootstrap files from the Bootstrap download page. Once downloaded, we can extract the contents of the file into our project's folder.

Node

With Node.js installed, we can quickly install the npm package for Bootstrap by typing in the command line:

```
npm install bootstrap
```

Ruby Gems

We can include Bootstrap in a Ruby app using Bundler and Ruby Gems with this command in the Gemfile:

```
gem 'bootstrap', '~> 4.0.0'
```

Bootstrap Sass Setup

Once we have Bootstrap installed we need to set up our project. The first thing we want to do is create a `sass` folder to hold our SCSS files and a `stylesheets` folder to store the compiled CSS. After that, we create a file named `app.scss` inside the `sass` folder.

The `app.scss` file is used to import Bootstrap components.

The next thing we want to do is find the `_variables.scss` file inside the Bootstrap Sass package and copy it into our `sass` folder. Next, we rename the file as `_customVariables.scss` and add an import statement for `_customVariables.scss` to `app.scss`. It's important to import `_customVariables.scss` first for reasons I'll explain shortly.

The last import is an optional `_custom.scss` file. Many people will include custom CSS rules directly after their import statements or in their `app.scss` file, but we're going to separate any custom rules into their own partial.

Notice, we import our `_customVariables.scss` file first. The reason is that Bootstrap's variables are set to `default!` values, so we need to override these values by importing our variables first.

Customize

When we edit variables, it's advisable to make a copy of the original and change the copy. After copying, we just comment out the original variable. That way, we can go back to what it was previously set to in case we don't like the result. For example, let's say we want to change the base font size to 20px. First, we'll look in our `_customVariable.scss` file. The variables are broken down into sections, and we're looking for the Fonts section. There, we want the `$font-size-base:1rem !default;` variable so we can copy it over to our custom file and comment out the original. After that, it's as simple as changing the value

to 20px:

```
//$font-size-base:1rem !default;
$font-size-base:20px;
```

Above, we've commented out the original variable and changed the copy.

When trying to customize Bootstrap, bear in mind there are a lot of variables to deal with. When looking for a variable to modify, it's advisable to make full use of the text editor's search feature. It's also a good idea to look over the `_customVariables.scss` file and get familiar with the variables present.

Another effective method for finding which variables to change is to look at the raw SCSS files that make up Bootstrap before they're compiled. From there, we can see which variables are used in that module. For example, let's say we're not happy with the color of the `.navbar-dark` element. Instead of trying to figure out which variable we need to change, we can look inside the `_navbar.scss` file, scroll down (or use the search function) and look for a reference to a color variable.

```
// navbar-dark
.navbar-dark {
  .navbar-brand {
    color: $navbar-dark-active-color;

    @include hover-focus {
      color: $navbar-dark-active-color;
    }
  }
  // more Sass here
}
```

From looking at this rule, we determine the variable we need to change is `$navbar-dark-active-color`. We would then go into `_customVariables.scss`, copy/comment out the original variable, and create our own.

When using Bootstrap's source files, we can also take advantage of its mixins. Not only will they help with understanding how Bootstrap fits together, they may actually help us build our website. For example, let's look at the `@mixin make-container`:

```
@mixin make-container() {
  width: 100%;
  padding-right: ($grid-gutter-width / 2);
  padding-left: ($grid-gutter-width / 2);
  margin-right: auto;
  margin-left: auto;
}
```

From this mixin we can see which variables affect our container. We now know we can alter the `$grid-gutter-width` to make changes to the paddings of a container element.

Conclusion

Using Bootstrap can be complicated, especially for someone who is not familiar with the framework. With the methods I demonstrated, you should be able to establish a Bootstrap Sass setup and customize the framework with ease. Finding the variables you need to change should be more manageable now that you know what to look for. Just remember, your text editor's search functions are your friend, and when in doubt, looking at the mixins can help.

Chapter

6

The 5 Most Popular Front-end Frameworks Compared

Ivaylo Gerchev

There's a deluge of CSS front-end frameworks available nowadays. But the number of really good ones can be narrowed down to just a few. In this article, I'll compare what I think are the five best front-end frameworks available today.

Each framework has its own strengths and weaknesses, and specific areas of application, allowing you to choose based on the needs of a specific project. For example, if your project is simple, there's no need to use a complex framework. Also, many of the options are modular, allowing you to use only the components you need, or even mix components from different front-end frameworks.

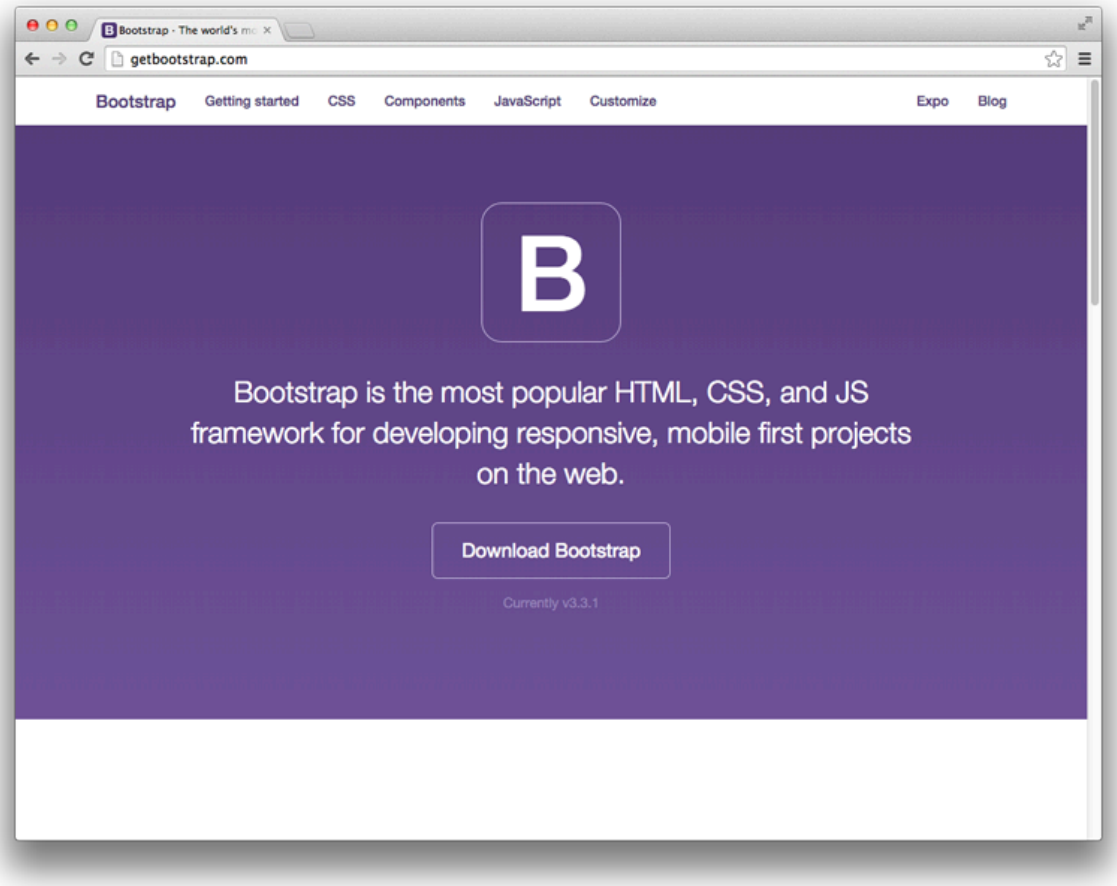
The front-end frameworks I'm going to explore are presented based on their GitHub popularity, beginning with the most popular, which is, of course, Bootstrap.

Ahem, Excuse Me ...

Note that some of the information below will go out of date in the coming weeks and months – such as GitHub stars and version numbers – so be aware of this if you're reading this article long after the publication date. Also note that the framework sizes are the minified sizes of the necessary CSS and JavaScript files.

1. Bootstrap

Bootstrap is the undisputed leader among the available front-end frameworks today. Given its huge popularity, which is still growing every day, you can be sure that this wonderful toolkit won't fail you, or leave you alone on your way to building successful websites.



6-1. Front-end framework 1: Bootstrap

- **Creators:** Mark Otto and Jacob Thornton.
- **Released:** 2011
- **Current version:** 4.0
- **Popularity:** 121,374 stars on GitHub
- **Description:** "Sleek, intuitive, and powerful front-end framework for faster and easier web development."
- **Core concepts/principles:** RWD and mobile first.
- **Framework size:** 578 KB (precompiled zip folder)
- **Preprocessor:** Sass
- **Responsive:** Yes
- **Modular:** Yes
- **Starting templates/layouts:** Yes

- **Icon set:** Not included
- **Extras/Add-ons:** None bundled, but many third-party plugins are available.
- **Unique components:** Jumbotron, Card
- **Documentation:** Excellent
- **Customization:** Option for separate files for Grid system and Reboot, easy customization with Sass; no online customizer
- **Browser support:** Latest releases of Firefox, Chrome, Safari, IE810-11-Microsoft Edge.
- **License:** MIT

Notes on Bootstrap

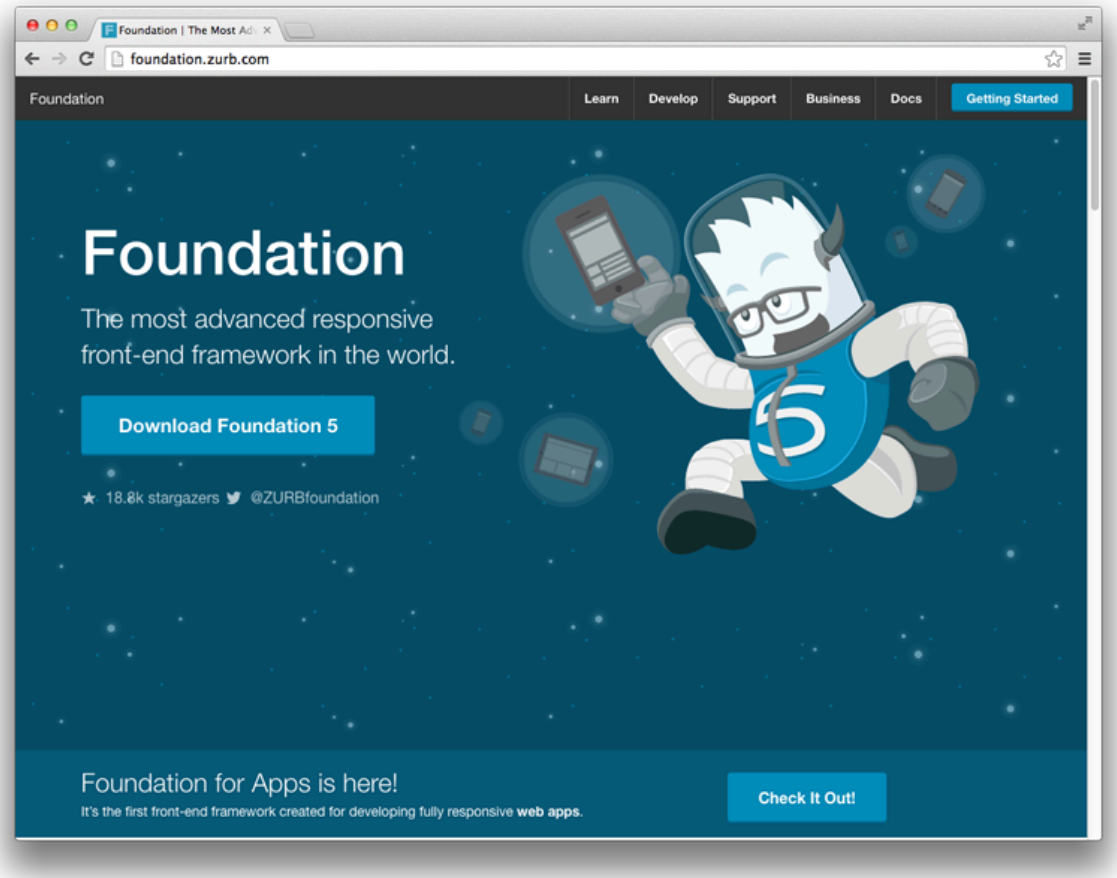
The main strength of Bootstrap is its huge popularity. Technically, it's not necessarily better than the others in the list, but it offers many more resources (articles and tutorials, third-party plugins and extensions, theme builders, and so on) than the other four front-end frameworks combined. In short, Bootstrap is *everywhere*. And this is the main reason people continue to choose it.

Unique?

By saying "unique components", I mean that they're unique compared only to the front-end frameworks mentioned in this list.

2. Foundation by ZURB

Foundation is the second big player in this front-end framework comparison. With a solid company like ZURB backing it, this framework has a truly strong … well … foundation. After all, Foundation is used on many big websites including Facebook, Mozilla, Ebay, Yahoo! and National Geographic, to name a few.



6-2. Front-end framework 2: ZURB Foundation

- **Creators:** ZURB
- **Released:** 2011
- **Current version:**6
- **Popularity:** 26,956 stars on GitHub
- **Description:** "The most advanced responsive front-end framework in the world"
- **Core concepts/principles:** RWD, mobile first, semantic
- **Framework size:** 197.5 KB
- **Preprocessors:** Sass
- **Responsive:** Yes
- **Modular:** Yes
- **Starting templates/layouts:** Yes

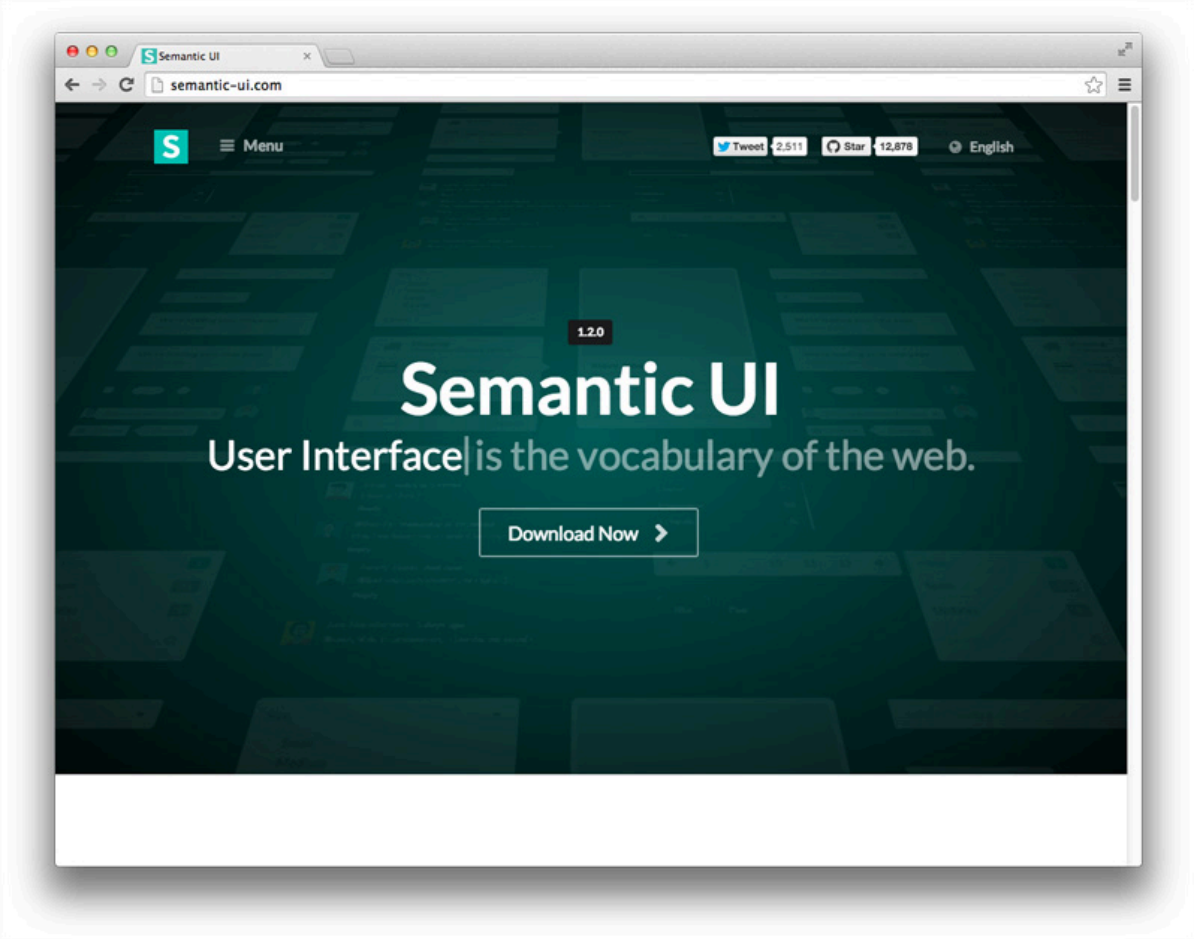
- **Icon set:** Foundation Icon Fonts
- **Extras/Add-ons:** Yes
- **Unique components:** Icon Bar, Clearing Lightbox, Flex Video, Keystrokes, Joyride, Pricing Tables
- **Documentation:** Good, with many additional resources available.
- **Customization:** Basic GUI customizer
- **Browser support:** Chrome, Firefox, Safari, IE9+; iOS, Android, Windows Phone 7+
- **License:** MIT

Notes on Foundation

Foundation is a truly professional front-end framework with business support, training, and consulting offered. It also provides many resources to help you learn and use the framework faster and easier.

3. Semantic UI

Semantic UI is an ongoing effort to make building websites much more semantic. It utilizes natural language principles, thus making the code much more readable and understandable.



6-3. Front-end framework 3: Semantic UI

- **Creator:** Jack Lukic
- **Released:** 2013
- **Current version:** 2.2
- **Popularity:** 39,364 stars on GitHub
- **Description:** "A UI component framework based around useful principles from natural language"
- **Core concepts/principles:** Semantic, tag ambivalence, responsive
- **Framework size:** 806 KB
- **Preprocessors:** Less
- **Responsive:** Yes
- **Modular:** Yes
- **Starting templates/layouts:** Yes, some basic starter templates are offered

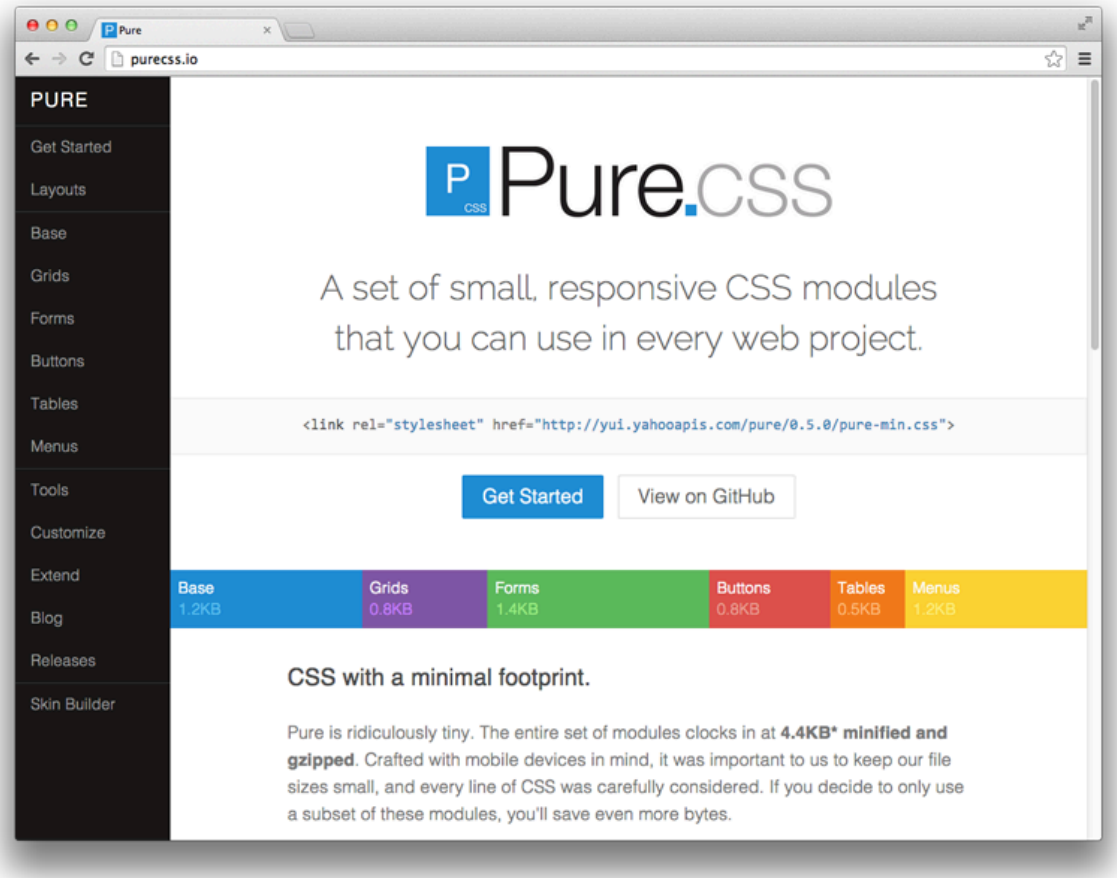
- **Icon set:** Font Awesome
- **Extras/Add-ons:** No
- **Unique components:** Divider, Flag, Rail, Reveal, Step, Advertisement, Card, Feed, Item, Statistic, Dimmer, Rating, Shape
- **Documentation:** Very good, offering very well-organized documentation, plus a separate website that offers guides for getting started, customizing and creating themes
- **Customization:** No GUI customizer, only manual customization
- **Browser support:** Firefox, Chrome, Safari, IE10+ (IE9 with browser prefix only), Android 4, Blackberry 10
- **License:** MIT

Notes on Semantic UI

Semantic is the most innovative and full-featured front-end framework among those discussed here. The overall structure of the framework and the naming conventions, in terms of clear logic and semantics of its classes, also surpasses the others.

4. Pure, by Yahoo!

Pure is a lightweight, modular framework — written in pure CSS — that includes components that can be used together or separately depending on your needs.



6-4. Front-end framework 4: Pure, by Yahoo!

- **Creator:** Yahoo
- **Released:** 2013
- **Current version:** 1.0.0
- **Popularity:** 18,183 stars on GitHub
- **Description:** "A set of small, responsive CSS modules that you can use in every web project"
- **Core concepts/principles:** SMACSS, minimalism
- **Framework size:** 3.8 KB minified and gzipped
- **Preprocessors:** None
- **Responsive:** Yes
- **Modular:** Yes
- **Starting templates/layouts:** Yes

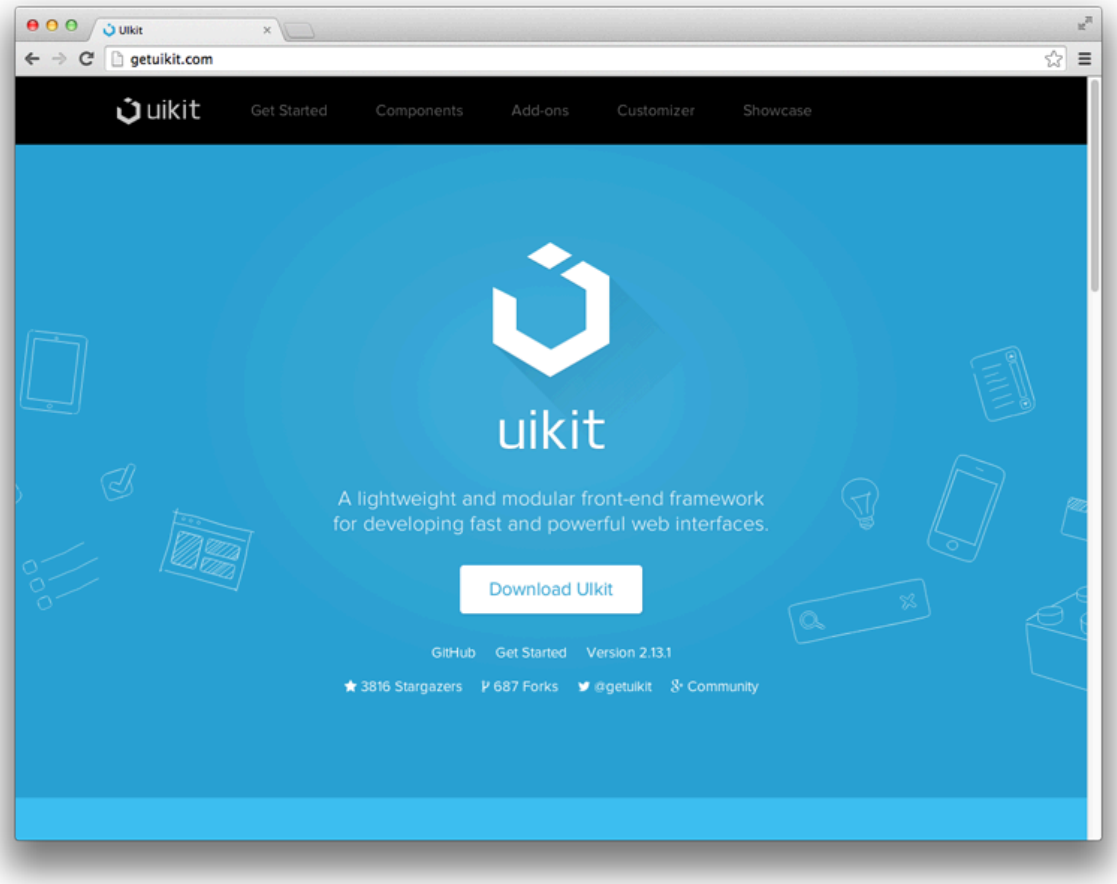
- **Icon set:** None; you can use Font Awesome instead
- **Extras/Add-ons:** None
- **Unique components:** None
- **Documentation:** Good
- **Customization:** Basic GUI Skin Builder
- **Browser support:** Latest versions of Firefox, Chrome, Safari; IE7+; iOS 6.x, 7.x; Android 4.x
- **License:** Yahoo! Inc. BSD

Notes on Pure

Pure offers only bare-bones styles for a clean start to your project. It's ideal for people who don't need a full-featured framework but only specific components to include in their work.

5. UIkit by YOOtheme

UIkit is a concise collection of easy-to-use and easy to customize components. Although it's not as popular as its competitors, it offers the same functionality and quality.



6-5. Front-end framework 5: Ulkit

- **Creator:** YOOtheme
- **Released:** 2013
- **Current version:** 3.0.0
- **Popularity:** 11,604 stars on GitHub
- **Description:** "A lightweight and modular front-end framework for developing fast and powerful web interfaces"
- **Core concepts/principles:** RWD, mobile first
- **Framework size:** 326.9 KB (zipped folder)
- **Preprocessors:** Less, Sass
- **Responsive:** Yes
- **Modular:** Yes
- **Starting templates/layouts:** Yes

- **Icon set:** UIkit comes with its own SVG icon system and library with a growing number of outline icons
- **Extras/Add-ons:** Yes
- **Unique components:** Article, Flex, Cover, HTML Editor
- **Documentation:** Good
- **Customization:** Advanced GUI Customizer only available in version 2 (previous version)
- **Browser support:** Chrome, Firefox, Safari, IE9+
- **License:** MIT

Notes on UIkit

UIkit is successfully used in many WordPress themes. It offers a flexible and powerful manual customization mechanism. (Previous version of the framework also offered an advanced GUI customizer.)

What's the Right Front-end Framework for You?

In the end, let me give you some guidelines for choosing the right framework. Here are some of the more important things to watch out for:

- Does the framework have enough **popularity**? Bigger popularity means more people involved in the project, and thus, more tutorials and articles from the community, more real-world examples/websites, more third-party extensions, and better integration with relative web development products. Great popularity also means that the framework is more future-proof: a framework with a bigger community around it is much less likely to be abandoned.
- Is the framework under **active development**? A good framework needs to level up constantly with the latest web technologies, especially with regards to mobile.
- Has the framework reached *maturity*? If a particular framework is not yet used and tested in real-world projects, then you can freely play with it, but to rely on it for your professional projects would likely be unwise.
- Does the framework offer **good documentation**? Good documentation is

always desirable in order to facilitate the learning process.

- What is the framework's *level of specificity*? The main point here is that a more generic framework is far easier to work with, in comparison to a framework with high-level specificity. In most cases it's better to choose a framework with minimal styles applied, because it's much easier to customize. Adding new CSS rules is a far more convenient and efficient process compared to overwriting or overriding existing ones. Plus, if you add new rules on top of the existing ones, you'll end up with unused rules, which will increase unnecessarily the size of the CSS.

In case you're still unsure, you can adopt a mix-and-match approach. If a particular framework doesn't satisfy your needs, you can mix components from two or more projects. For example, you can get smaller CSS base styling from one framework, a preferred grid system from another, and more complex components from a third. Viva modularity!

Finally, it should be mentioned that nowadays, with Flexbox and Grid Layout having good support in the latest versions of major browsers, it's easier than ever to build complex layouts. This fact alone might encourage more developers to leave the crutch of front-end frameworks and code their layouts from scratch.

What are your thoughts? Do any of these front-end frameworks have any strengths and weaknesses that weren't mentioned here? Do you think there are others that should have been listed? Let us know in the discussion below.

Chapter

7

The CSS Grid Layout vs CSS Frameworks Debate

Maria Antonietta Perna

With cutting-edge CSS standards like CSS Grid Layout and Flexbox, coding a web page layout is no longer such a pain. If you add to this that browser support for both Grid and Flexbox is pretty good too, then the question is bound to come up: Why should consider learning and using CSS frameworks in my development work?

In this article, I'll focus on Bootstrap, since it's arguably one of the most popular among all the CSS UI libraries available out there.

In my view, there are still a number of reasons why it does still make sense to learn and use Bootstrap today.

Here's a few of them for you.

What Is CSS Grid?

Rachel Andrew, a well-known speaker and writer on all things CSS Grid-related, defines it as follows:

> *Grid is a grid system. It allows you to define columns and rows in your CSS, without needing to define them in markup. You don't need a tool that helps you make it look like you have a grid, you actually have a grid!*

The implementation of this CSS standard gives developers the much needed ability to build page layouts with native CSS code, with no dependency from the HTML markup except for the presence of a wrapper element that works as containing grid. Just imagine the flexibility and the potential for creativity in web design!

For instance, you don't need custom classes or extra rows in your markup to build this simple layout:

7-1. Simple web page layout built with CSS Grid

Here's the HTML:

```
<div class="grid">
  <header>Header content</header>
  <main>Main content</main>
  <aside>Sidebar</aside>
  <footer>Footer</footer>
</div>
```

As for the CSS, this is where you're going to build your visual layout. All it takes in this simple case is a few lines of code:

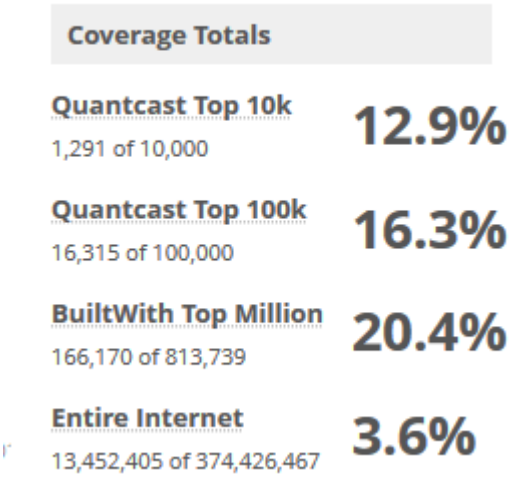
```
.grid {
  display: grid;
  grid-template-columns: repeat(12, 1fr);
  grid-template-rows: 50px 150px 50px;
}

header, footer {
  grid-column: span 12;
}

main {
  grid-column: span 8;
}

aside {
```

```
  grid-column: span 4;
}
```

That's it, you're done! Not bad.

What Is Bootstrap?

At the time of this writing, 3.6% of the entire Internet uses Bootstrap:



7-2. Bootstrap usage

On the Bootstrap website, you'll find this definition:

> *Bootstrap is an open-source toolkit for developing with HTML, CSS, and JS. Quickly prototype your ideas or build your entire app with our Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful plugins built on jQuery.*

In other words, Bootstrap gives you ready-made components that allow you to

whip up a beautiful web page in no time.

Just write the appropriate markup, and your app looks great out of the box.

Bootstrap makes it also super easy to customize its look and feel to your liking and lets you pick and choose the bits you need for your project.

Why Bootstrap When We Have Grid?

The biggest complaint about Bootstrap has always been code bloat. The reasoning was that it included lots of extra CSS code that remained unused in your projects. The second biggest complaint was that Bootstrap components were styled in every detail and this could present some problems when it came to overriding some CSS rules.

Starting with the latest version of this popular front-end component library, both criticisms fall to pieces: Bootstrap is totally modular, so you just include what you need. Also, the Sass files are structured in such a way as to make it very convenient to customize the original styles to your needs.

Today, the main reason against using Bootstrap is the fact that with CSS Grid, CSS has a grid system of its own, which doesn't have any external dependencies, and which, once learned, enables developers to build all sorts of layouts with relative ease.

Although I'm a CSS Grid fan, I think Bootstrap has still its place in front-end development and will have for some time to come.

Here are at least three reasons why.

Bootstrap Is More Than Its Grid System

It's true, the #1 reason to use Bootstrap is the handy grid system, which makes it a breeze to build responsive web pages. However, Bootstrap has some great

components like the versatile new card component, which you can use to display all types of content, such as text, images and videos, and the responsive navbar, which works out of the box. You can also pick a ready-made color scheme of your liking for most components.

And what about the functionality of a good many of these components? With Bootstrap, adding dynamic tooltips, carousels, or dropdown buttons is just a matter of writing the appropriate markup. If JavaScript is not your forte, you can still take advantage of these components without writing a line of JavaScript.

Also, if you're not a CSS wizard, you can still leverage the power of Bootstrap in your web design while you're learning the tricks of the trade.

Bootstrap Is a Great Prototyping Tool

Sometimes you just need to a working prototype for a client. Bootstrap lets you do this in no time and with very little to no custom code. This doesn't only hold for the grid system, but also for all the ready-made components it has to offer.

Just add a little markup and your prototype will boast a sleek responsive navigation bar or a fancy alert box.

Working on Older Websites Built With Bootstrap

One common task for developers is to work on existing websites coded by other developers. There's no denying the fact that a huge number of websites rely on Bootstrap for their front end. Knowing how to work with the framework will come very handy if it's up to you to refactor and maintain the codebase. It's not by chance that still lots of job postings have Bootstrap in their list of desired skills.

Conclusion

To conclude, Bootstrap is not going away any time soon. The latest release

comes with huge improvements over the previous versions, from the clever use of Sass mixins and maps for easy customization to the introduction of new components, utility classes, and an ever more modular architecture.

Add to this great documentation and ease of use, and Bootstrap is still a mighty contender in the front-end ecosystem.