

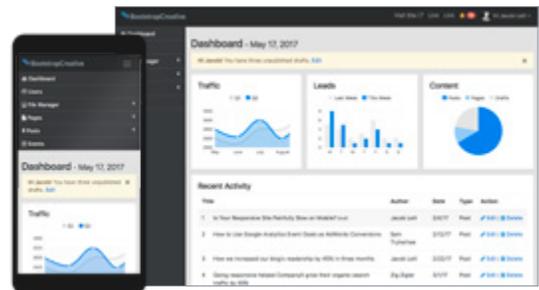
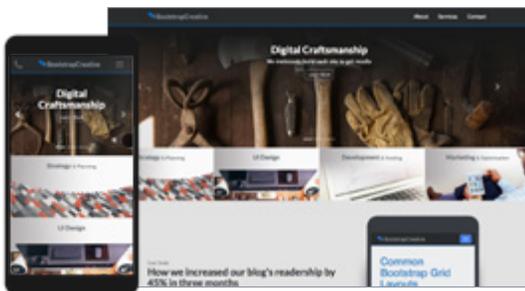
Bootstrap 4 Quick Start

Learning web development is a lot more challenging than it used to be.

Responsive web design adds more layers of complexity to design and develop websites. In this book you will become familiar with the new cards component, setting up the new flexbox grid layout, customizing the look and feel, how to follow the mobile-first development workflow, and more!

This book does not duplicate what you can already find in the official documentation but instead shows you how to reference the documentation and use it effectively in your projects to save time.

**Learn How to Build a Marketing Site
and an Admin Dashboard... Step by Step.**



Also includes access to a recorded training video on responsive design basics.
See page 24 for details.

*The training documents were great and **got me started quickly**. I was able to jump into getting my website framework started.*

— Andrew Diaz, web developer new to Bootstrap

*It is clear and well structured. **A good start for Bootstrap 4 beginners.***

— Uwe, web developer beginner

*An excellent and concise introduction to **Bootstrap 4**. Includes complete examples/code that is easily modifiable for your own uses.*

— Pedrovski, web design beginner

*The book uses metaphors throughout which **helps make technical concepts clear and easier to understand.***

— Max, Web developer new to Bootstrap

Bootstrap 4 Quick Start

Copyright © 2018 by Jacob Lett | Published on June 28, 2018

Publisher

Bootstrap Creative | Sterling Heights, Michigan 48314 | (586) 894-8024

To report errors, please send an email to support@bootstrapcreative.com

Find us on the web at: bootstrapcreative.com

Notice of Rights

All rights reserved. Please do not distribute, share, or sell without the written permission of the publisher. You may print pages for personal use. If you would like to share this with a group, please contact support@bootstrapcreative.com.

Notice of Liability

The information in this book is distributed on an "as is" basis without warranty. While every precaution has been made in the preparation of the book, neither the author nor Bootstrap Creative shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the instructions contained in this book or by the computer software products described in it.

Trademarks

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and Bootstrap Creative was aware of a trademark claim, the designations appear as requested by the owner of the trademark. All other product names and services identified throughout this book are used in editorial fashion only and for the benefit of such companies with no intention of infringement of the trademark. No such use, or the use of any trade name, is intended to convey endorsement or other affiliation with this book. Bootstrap is released under the MIT license and is copyright 2018 Twitter.

ISBN: 978-1-7322058-1-9

Contents

INTRODUCTION	1
THE MOBILE WEB & BOOTSTRAP	9
WHAT'S NEW IN BOOTSTRAP 4	25
THE DESIGN PROCESS	47
BUILD A HOMEPAGE	53
BUILD A DASHBOARD	105
RESPONSIVE LAYOUT TIPS	155
CONCLUSION	175
APPENDIX	179

Dedicated to my wife Colleen.

Thank you for your persistent love and encouragement.

Thank you!

I would like to thank the following for their advice and support with this project: Greg Vance, Dan Joseph, Max, and my encouraging parents. Marcus and Joshua thank you for your patience as your Dad was working on this project. Also my mentors for their wisdom and inspiration: Zig Ziglar, Dan Miller, Seth Godin, and Matthew 25:14-30.

Finally, thank you reader for purchasing this book.

My hope is to help you be successful in your web development projects and career.

About the Author

Jacob Lett is the author of the *Bootstrap 4 Quick Start*, and the *Bootstrap Reference Guide*. His books and training help web developers save time learning how to design and build responsive websites.

www.linkedin.com/in/jacoblett/

Introduction

Do you remember learning how to write a research paper?

Perhaps you learned how to follow the MLA or APA Style Guide. Think how different the reports would look if each student made up their own style and format?

Using the MLA Style guaranteed consistency for anyone who followed that same format even if they were thousands of miles away or written five years apart.

I believe Bootstrap works the same way for web design. It helps developers work more efficiently and write CSS in a clean and consistent manner regardless of where you live or who you work for.

It also ensures your website adheres to a mobile first approach and works well across browsers and devices.

I am convinced Bootstrap will greatly improve how you build websites and save you countless time. So if you're ready, let's get started.

A handwritten signature in blue ink, appearing to read 'Jacob', with a long, sweeping underline.



Hi.

My name is Jacob Lett and it's my mission to help you save time learning how to design and build responsive websites.

I earned a bachelors degree in graphic design around the time CSS and web standards were just starting to take hold. Out of frustration not knowing how to fix broken websites generated by Dreamweaver, I learned how to hand code HTML/CSS. Then in 2009 I got my first job as a web designer writing a ton of CSS and realizing I had a long journey of learning ahead.

 www.linkedin.com/in/jacoblett/

Initial Setup

Before we begin, we need to make sure you have all of the necessary software and tools installed on your computer. So let's get started.

1. Install Google Chrome

I recommend testing your sites in Google Chrome because of their nice set of DevTools to help you debug problems and inspect CSS styles and HTML elements. If you do not have this it installed you can do so here^[1].

In my *Bootstrap 4 Toolkit*,^[2] I include a tutorial video titled *Inspect and Test CSS* that demonstrates how to use Chrome DevTools.

2. Install Chrome Extensions

These are optional but very helpful.

live.js Extension

Install the live.js Chrome extension^[3] to toggle the ability to automatically reload your browser when you make code changes. Or you can manually add the script found at <http://livejs.com/>.

Web Developer Extension

Install the Web Developer Chrome extension^[4] to help you test responsive breakpoints and perform other developer focused tasks.

1 <https://www.google.com/chrome/browser/desktop/index.html>

2 <http://bootstrapcreative.com/b4toolkit>

3 <https://bootstrapcreative.com/livejschrome>

4 <https://bootstrapcreative.com/webdevchrome>

3. Setup a Local Testing Server

If you are new to web servers, a local server is on your desktop/laptop and a remote server is located someplace else accessible via SFTP. Web servers are just big computers with software installed for the sole purpose of hosting websites and data. The benefit of having a local test server is you can eliminate the step of transferring changed files to a remote server and avoid the public seeing your unfinished work.

The easiest way to start testing sites locally is using a tool called XAMPP on windows or MAMP on a mac. It lets you run PHP, apache, and even phpMyAdmin for MySQL database configuration. Commonly referred to as the LAMP web stack^[5] of bundled software.

Windows



Install XAMPP^[6] following the instructions in this guide. It is written for WordPress users so you can ignore the part of setting up a database and installing WordPress. You want the ability to make HTTP requests (or access files on other servers). I created this video tutorial^[7] on how to install XAMPP on Windows 10.

5 [https://en.wikipedia.org/wiki/LAMP_\(software_bundle\)](https://en.wikipedia.org/wiki/LAMP_(software_bundle))

6 <https://www.apachefriends.org/index.html>

7 <https://bootstrapcreative.com/installxampp>

Mac

Install MAMP^[8] following the instructions in this guide. It is written for WordPress users so you can ignore the part of setting up a database and installing WordPress. You want the ability to make HTTP requests (or access files on other servers).

Linux

Install Mongoose Binary^[9]. A really light weight server ideal for web design testing.

4. Install a Code Text Editor

Install a code text editor so you can benefit from code syntax coloring and other features that make the job of writing code easier. There are a lot of editors available and it comes down to your preference.

I personally use Visual Studio Code because it's free, cross platform, and works really well without having to add a lot of addons.

Below are some other popular code editors available:

- **Atom** (free, cross platform): <https://atom.io/>
- **Brackets** (free, cross platform): <http://brackets.io/>
- **Dreamweaver** (paid, cross platform): <http://www.adobe.com/products/dreamweaver.html>
- **Notepad++** (free, Windows only): <https://notepad-plus-plus.org/download/v7.3.3.html>
- **TextWrangler** (free, mac only): <http://www.barebones.com/products/textwrangler/download.html>
- **Visual Studio Code** (free, cross platform): <https://code.visualstudio.com/>
- **Other editors:** https://en.wikipedia.org/wiki/Source_code_editor#Some_well-known_source_code_editors

8 <https://bootstrapcreative.com/installmamp>

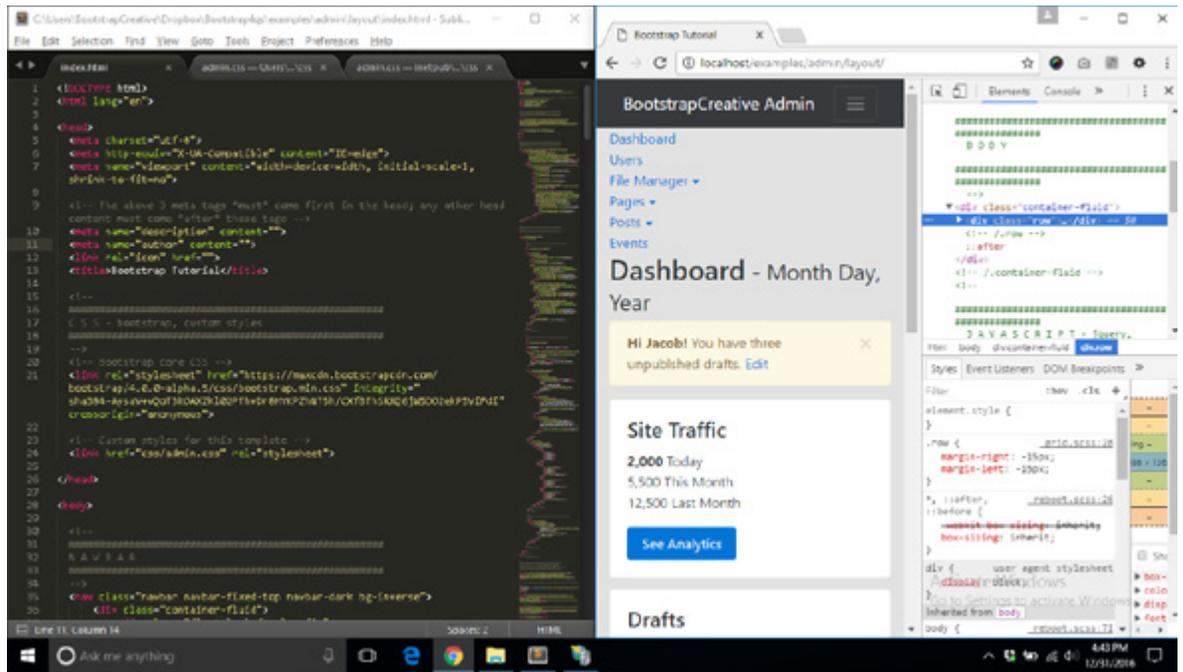
9 <https://www.cesanta.com/products/binary>

5. Mobile First Desktop Workspace

Next, open up your Chrome web browser then press **F12** to open Chrome DevTools. Next, press **CTRL + SHIFT + D** (win) or **CMD + SHIFT + D** (mac) to move the dock to the right side of your browser window.

Reference for all of Chrome DevTools shortcuts: <https://developers.google.com/web/tools/chrome-devtools/shortcuts>

Then move your text editor window to the left half of your screen and your browser to the right half of your screen using the keyboard shortcuts **WIN + →** and **WIN + ←** (win) you will need to adjust the windows manually on a mac.



Start coding/testing mobile first - *Your screen should now look like this.*



FREE Cheat Sheets Bundle

Do you have trouble remembering all of the Bootstrap class names?

Includes the following three references:

Bootstrap 4 Cheat Sheet | Flexbox Cheat Sheet | CSS3 Cheat Sheet

<https://bootstrapcreative.com/b4bundle>

The Mobile Web & Bootstrap

Building websites today is a lot more challenging and time consuming than it used to be.

Some of my first websites were first designed in Adobe Photoshop®, exported to HTML tables (yes tables) and then linked together with Adobe Dreamweaver®. If your website did not exceed the width of common monitor resolutions (1024px by 768px) everything would work out fine.

Web standards^[1] were quickly introduced because using table markup for grid layout is just bad practice. So HTML tables were replaced with floated divs and tag markup that had meaning – referred to as semantics. This also shifted things away from the majority of the visual design being baked into images and now relying on CSS3 to create borders, shadows, rounded corners, etc.

The first widely used CSS grid system was the 960 grid system (Fig. 1) created by Nathan Smith. This 12,16, 24 column grid system was designed to work well for a fixed desktop resolution of 1024px x 768px. This grid system was widely used and helped designers and developers work from the same grid pixel dimensions.



Fig. 1: The 960 grid system helped bring consistency between grid design in Photoshop and the web.

1 https://www.goodreads.com/book/show/259072.Designing_With_Web_Standards

Then in 2007, Steve Jobs introduced the world to the iPhone with Multi-Touch gestures^[2]. Now people could access websites anywhere using just their fingers.

Web designers and developers had to quickly develop creative solutions to work within the new constraints presented by smartphones and tablets.

These Constraints Include:

- Smaller screens
- Increased pixel densities with retina displays
- Ability to switch between portrait and landscape orientation
- Multi-touch gestures
- Slower data connections
- Distracted user attention (one eyeball and one thumb).

At the start, the concept of responsive design did not exist. And so mobile devices had to scale down websites to fit the smaller screens. For the user, in order to read the text they would have to double tap the screen or pinch and zoom.

Website owners quickly realized it was not a good experience to display their homepage at a zoomed in level. The meta tag below was introduced to remove this default scaling and give the site creator more control. When this meta tag is added to the `<head>` of a page, it instructs the web browser to scale the document 100% to prevent pinch/zoom on mobile.

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

2 <http://www.lukew.com/ff/entry.asp?1071>

Different Mobile Strategies

Mobile Applications

One approach is to build a dedicated experience as a mobile app. This gives the developer the most control and could utilize the device user interface components and to help with navigation. Major drawbacks include: it requires an app developer, considerable amount of marketing to direct existing traffic to download the mobile app, and overcoming low rates of user adoption. Also, any links to outside pages required them to open in a web browser window.

Adaptive Design

Another approach is to build multiple versions of a website and use server side detection to then present custom code for that device or viewport size.

You could decide to have your mobile site on a separate domain for example m.domain.com. The server will then automatically serve all mobile traffic to that domain. The server could also perform dynamic serving of page content so that you have just one domain name. The downsides to this approach is it requires complex server side detection code and is harder to maintain multiple site versions.

Responsive Design

Responsive design was introduced to help designers build one site on one domain that responds to a users viewport. The two necessary elements for a responsive design are a meta viewport tag to disable scaling and media queries to alter the design as the page gets smaller. Responsive design is a lot less expensive and easier to maintain than the other mobile strategies. This has added to its rapid growth and adoption.

A big challenge with responsive design is finding a balance between the content needs for both mobile and desktop. A desktop site has a lot of visual real estate that is often filled with carousels, videos, large parallax background images, and large blocks of text.

If you load a feature-rich website on a mobile device you often increase the page load for mobile visitors. This is due to the large images and videos which are scaled down to mobile.

End-users don't care about your responsive web or your separate sites, they just want to be able to get stuff done quickly.

— Brad Frost, author of Atomic Design

Mobile First

In the desktop first approach, you sacrifice the mobile experience because you have a lot of images and text content. In an article from Zurb on mobile first design it said, "Roughly 80% of the screen size is taken away when you start with mobile first design, you have to think about how to utilize your space in a much more conservative manner."

Fig. 2 - Desktop First Responsive Site

	Desktop	Mobile
Data Speed	Fast	Slow
Width	Wide	Narrow
Height	Unlimited	Unlimited
Retina Display Probability	Medium	High
Page File Size	Large	Large

A mobile first approach considers the goals of a mobile user and presents the content to help them achieve those goals. It removes all of the fluff and filler content and presents a concise collection of content that loads fast and is easy to use.

Fig 3. Mobile First Responsive Site

	Mobile	Desktop
Data Speed	Slow	Fast
Width	Narrow	Wide
Height	Unlimited	Unlimited
Retina Display Probability	High	Medium
Page File Size	Small	Medium +

The chart above shows the workflow flipped so the site is built mobile first and then enhancements are added as the viewport gets wider. Notice how the mobile site is loading a small file size on a slow data speed? That is as Google would say, being mobile friendly.

But some might say. "Ok now the mobile site looks good but now the desktop looks too basic and lacks flair."

Progressive Enhancement

A great way to solve this is to progressively enhance the page as your data speed and screen width increases. Everything you add to the page will be enhancing the design and if it doesn't load for some reason your page is still usable.

*Screens are small, connections are slow, and people often only give you their partial attention or short bursts of their time. **Designing for mobile first forces you to embrace these constraints***

— *Luke Wroblewski, Mobile First*

The best way to do this is with JavaScript media queries^[3] to determine viewport width and then load in content to the page. I created a small plugin called `IfBreakpoint.js`^[4] to help detect Bootstrap 4 breakpoints with JavaScript. I also recommend reading this article^[5] on ways to progressively load images with media queries.

One creative solution that has transformed the web and made responsive design easier for web designers has been the Bootstrap frontend framework. We will take a closer look at Bootstrap in the next section.

3 <https://jacoblett.github.io/IfBreakpoint/>

4 <https://jacoblett.github.io/IfBreakpoint/>

5 <https://timkadlec.com/2012/04/media-query-asset-downloading-results/>

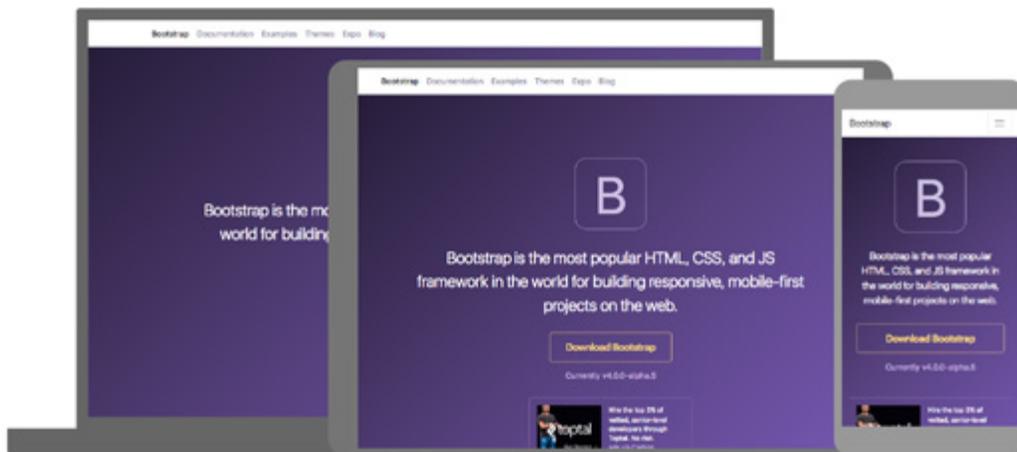
What is Bootstrap?

I remember building my first few responsive websites. I wasted so much time writing the same type of styles over and over for each new project. I also found it difficult to find plugins that worked well together and had cohesive design style

I then heard about Bootstrap and I liked how it included javascript components and had really comprehensive documentation. The documentation was extremely detailed and easy to follow. At first, it was hard to know what classes did what but after using it on a few projects I was amazed at how quickly I could create a working prototype of a design. The time I saved enabled me to complete more projects in less time and make more money in the process.

The more I used Bootstrap, the more I felt like it could be a global standard because it removes a lot of routine tasks when building responsive sites.

Bootstrap was created by Mark Otto and Jacob Thornton^[6] at Twitter as a framework to encourage consistency across internal tools. It is now an open source project hosted on GitHub^[7] and has seen rapid growth and global use in web applications and websites.



6 <https://v4-alpha.getbootstrap.com/about/history/>

7 <https://github.com/twbs/bootstrap>

Bootstrap CSS Framework History

- Before 2011 An internal Twitter tool
- August 2011 Released as open source
- January 2012 Bootstrap 2
- August 2013 Bootstrap 3
- August 2015 Bootstrap 4 Alpha
- August 2017 Bootstrap 4 Beta
- January 2018 Bootstrap 4

A Toolkit Built in Style Guide Form

When Bootstrap was first created at Twitter it was built as a toolkit of reusable components with additional documentation and code snippets on how to use them. This helped a team of multiple developers work on a project and have a cohesive methodology on how to build layouts. The documentation and ease of implementation, made it easy to share and reference with others, regardless of their skill level.

So, the initial intent of Bootstrap was to be a living style guide documentation for a team of developers to code in the same way following a set of pre-defined rules and components.

Today, Bootstrap Can Be Used in Two Main Ways:

1. Linking to a precompiled version via CDN or locally
2. Linking to a customized build using the Sass source files

On the next page (Fig. 5) I explore the pros and cons of each method and also break them into smaller sub-methods to help you decide which is best for your project.

A System of Components

At this point, you might be wondering what is a component and why does Bootstrap use them? Well, one definition I found was, "A component is a minimal software item that can be tested in isolation." The keyword in that phrase is **isolation**.

Since CSS cascades down to child elements, how do you isolate things and write styles to target specific components and leave everything as is? The solution Bootstrap presented is the use of prefix class naming and sub-classes for variations.



Fig. 4: The International Space Station is made up of isolated components that perform a specific purpose. Combined they create a system of linked components.

Mark Otto wrote on his blog^[8], "Each class name begins with a prefix. Class name prefixing makes our code more durable and easier to maintain, but it also better enables us to **scope styles to only the relevant elements.**"

8 <http://markdotto.com/2012/03/02/stop-the-cascade/>

Is Bootstrap Even Necessary?

If you are an experienced web designer or developer you are probably wondering what the benefits are using Bootstrap in your project. Prior to using Bootstrap, I used a boilerplate I wrote myself that consisted of a reset, basic grid, typography, utilities, and media queries. Below are the benefits I have experienced from now using Bootstrap for my projects.

Helps You Save Time

I admit I was the worst at documenting my own work. I would use my boilerplate on a project and then want to make an update to it a month later. But by then, I totally forgot my naming convention. So I would have to spend time reading my code to try to understand what I did. If I couldn't figure it out, I would add new code and leave the old code alone to prevent breaking something. Yup, sound the code bloat alarm.

Bootstrap has amazing documentation on each component. So if I want to update a project I worked on a few months ago that uses Bootstrap I know where to go to find documentation if I get stuck. Also, the more I use Bootstrap the more it is burned into my brain and the less time is spent searching the documentation.

Helps You Avoid Cross-browser Bugs

Prior to using Bootstrap I would get the dreaded emails from clients saying their website they just paid me for doesn't look good on X device. And of course, it is a device I do not currently own or have access to. After hours or searching on Google you finally find a fix on Stack Overflow. You find comfort knowing it is a common problem with Android devices and not something you caused.

Being an open source project, anyone can submit browser bugs and code fixes for it. This is an extremely valuable asset to a developer because you gain confidence knowing your code has been improved by a community to address common browser bugs. No matter how good you are, there is no way you can be aware of every browser inconsistency and the fix necessary. Using Bootstrap, you're standing on the shoulders of giants.

Fig. 5 - The Different Ways to Use Bootstrap

Method		Use Cases
1a	<p>Link to CDN Minified Difficulty: beginner</p>	<ul style="list-style-type: none"> • Cases where custom branding is not a priority • Backend layouts • Prototypes
1b	<p>Link to CDN Minified + Custom Stylesheet Difficulty: beginner</p>	<ul style="list-style-type: none"> • The method used in this book • Production sites that require unique branding
1c	<p>Link to CDN Minified + Custom Stylesheet from Sass Files Difficulty: intermediate</p>	<ul style="list-style-type: none"> • Production sites that require unique branding
2a	<p>Link to Custom Build Difficulty: advanced</p>	<ul style="list-style-type: none"> • Production sites that require unique branding
2b	<p>Link to Custom Build + Docs Build Difficulty: advanced</p>	<ul style="list-style-type: none"> • Production sites that require unique branding • Multiple developers work on a single project

Pros	Cons
<ul style="list-style-type: none"> • Fast • Easy to setup • No preprocessor needed 	<ul style="list-style-type: none"> • Lacks unique visual style • Some code bloat of unused components
<ul style="list-style-type: none"> • Fast • Easy to setup • No preprocessor needed 	<ul style="list-style-type: none"> • The time to inspect and overwrite Bootstrap styles • Some code bloat of unused components
<ul style="list-style-type: none"> • Some setup time • You gain the benefits of Sass with mixins, variables, and multiple files. 	<ul style="list-style-type: none"> • The time to inspect and overwrite Bootstrap styles • Some code bloat of unused components • Requires knowledge of Sass and compiling
<ul style="list-style-type: none"> • More setup time • Gain the benefits of Sass • Removes code bloat 	<ul style="list-style-type: none"> • Knowledge of Sass and compiling
<ul style="list-style-type: none"> • More setup time • Gain the benefits of Sass • Removes code bloat • Create updated documentation 	<ul style="list-style-type: none"> • Knowledge of Sass and compiling • Knowledge of Jekyll and compiling • Documentation updates

Helps You Follow Best Practices

I studied graphic design in college and self-taught myself HTML & CSS from books, YouTube, and blog posts. This mixture of knowledge worked to some degree but I know there are a lot of knowledge gaps. I hit my lack of understanding head-on when I first learned Bootstrap with all of the new terminology written for software engineers.

*Bootstrap is not just a framework
but a methodology of best practices
for front-end design.*

Gather a room full of the smartest web designers and developers and let them discuss at length what they think is the best way to write CSS and to organize a project. The result being a distilled version of best practices agreed upon by a large collection of your peers.

Helps You Avoid jQuery Plugin Soup

I know some JavaScript but writing a full-fledged plugin is out of my reach. So I often collected various jQuery plugins into a project to achieve the look and functionality I was looking for.

But I often ran into the following problems:

3. Plugins would not work across browsers
4. Plugin CSS styles would conflict with other CSS styles
5. Plugins would be dependent on different versions of jQuery

Bootstrap contains a collection of jQuery components that you know are stable on modern browsers compatible and works with your jQuery version. Also, the styling matches all of the other components in the your project.

Helps You Be More Marketable

Bootstrap has 73% of the design framework market share^[9] as of May 2017. This popularity correlates to the demand for people to know the framework to either update existing systems and or create new ones.

So this will make you more marketable to prospective employers. Indeed.com, a popular job search engine, shows Bootstrap has a lot of job postings compared to other CSS frameworks.

Framework Name	Total Sites
Bootstrap CSS (<i>update to v4?</i>)	12,559,226
HTML5 Boilerplate	4,219,959
960 Grid System	437,120
Unsemantic	74,386
Semantic UI	10,803

Source: BuiltWith as of May, 2017^[10]

9 <https://trends.builtwith.com/docinfo/design-framework>

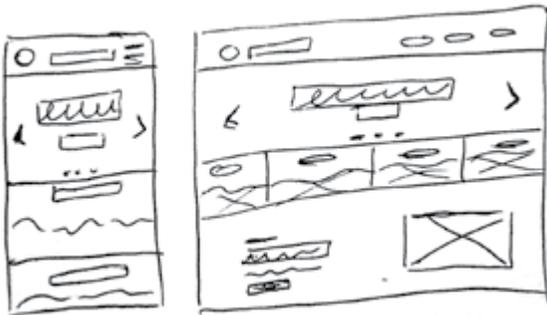
10 <https://trends.builtwith.com/docinfo/design-framework>

The Design Phases Used in This Book

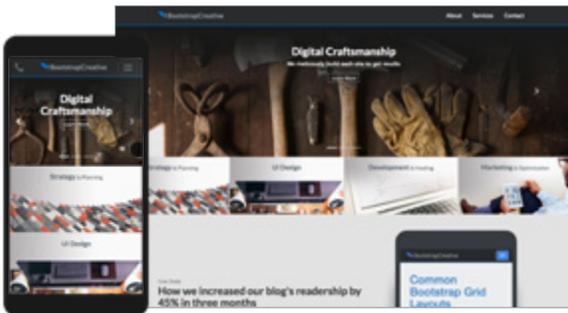
In this book, we will be spending our time focusing on the prototype and design phases.

These phases are closely related and could also be broken up into UX and UI. Prototypes focus on identifying content, naming, hierarchy and site architecture. The design phase converts wireframes into final layouts with graphics and branding.

This book will not be able to demonstrate how to generate pencil thumbnail sketches or perform design exploration in a graphics program. Both of these require a deep dive into design theory, branding, and design software. If you would like to learn these skills, please signup on this page^[21] to be notified of a future book on this subject.



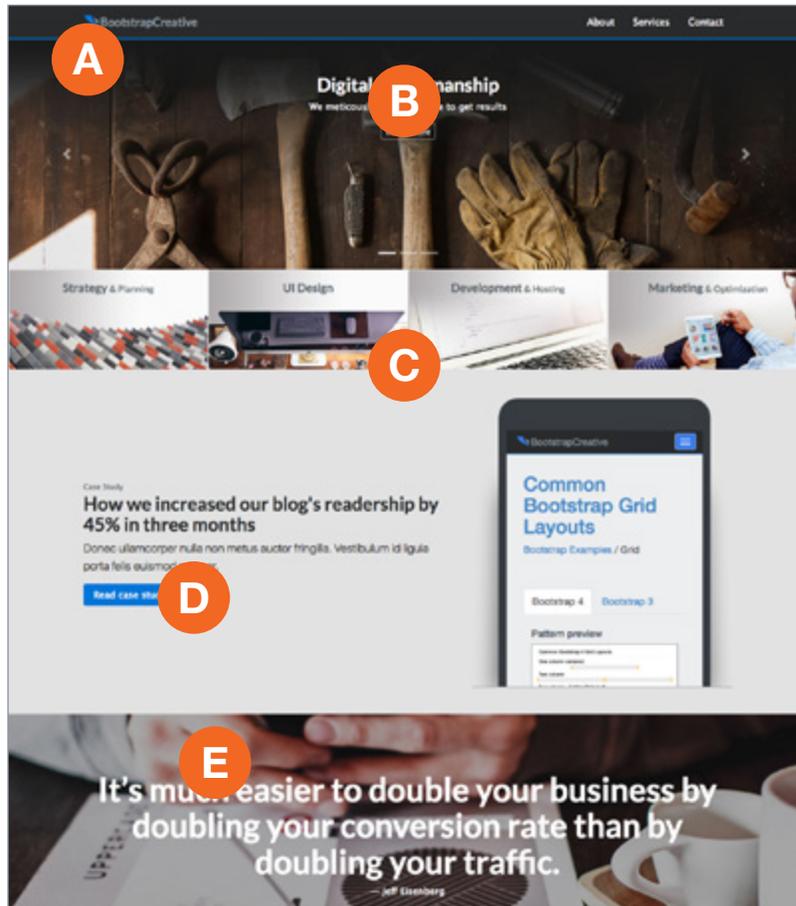
2. Prototype



3. Design

21 <https://bootstrapcreative.com/shop/web-design-book/>

Build a Homepage



A. Navbar Branding

Add blue border to the bottom and change link color

B. Carousel Captions

Move the captions to the top

C. Image Cards

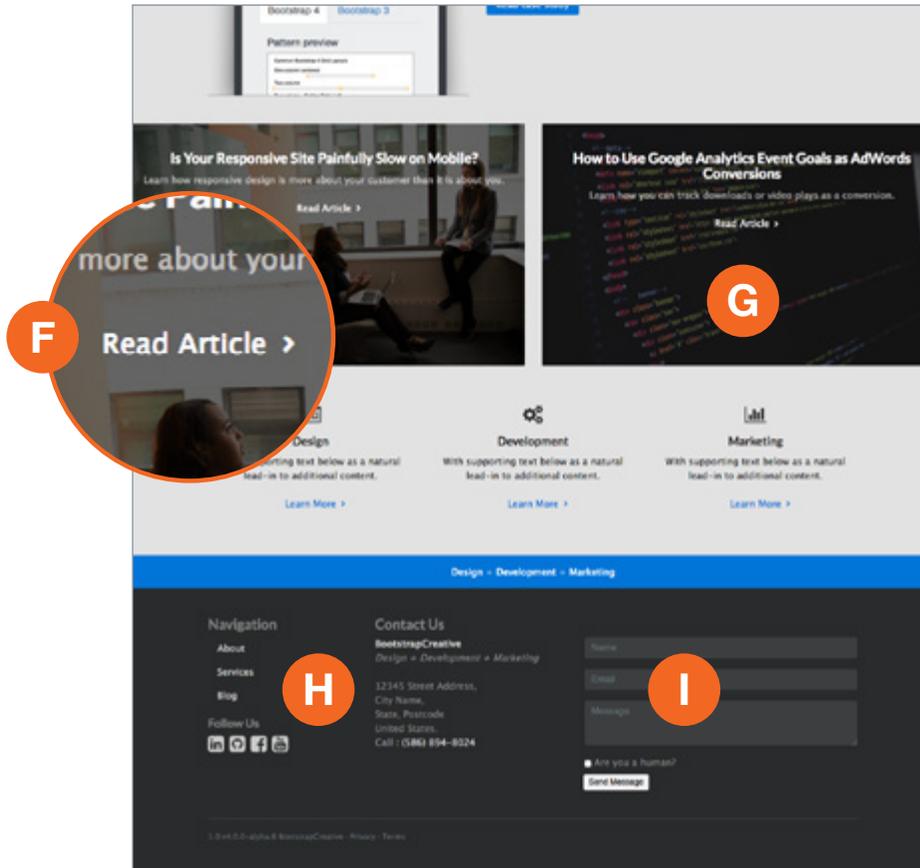
Adjust image widths

D. Vertical Centered Content

Make the text vertically centered with the image

E. Parallax Background Image

Set a background image that is fixed on scroll



F. Add Right Arrow to Nav Links

Add right arrow with CSS and not HTML

G. Text Overlay on Image Cards

Add a black overlay on top of the image to improve text readability

H. Nested Columns

Add columns with breakpoints that make sense

I. Muted Form Inputs

The form inputs have too much contrast and are hard on the eyes

Build a Dashboard

BUILD A DASHBOARD

The image shows a Bootstrap dashboard with a dark sidebar and a light main content area. The sidebar contains a navigation menu with items: Dashboard, Users, File Manager, Pages, Posts, and Events. The main content area has a header with the title 'Dashboard - May 17, 2017' and a yellow alert box that says 'Hi Jacob! You have three unpublished drafts. Edit'. Below the header is a 'Traffic' chart showing data for Q1 and Q2 from May to August. The chart shows Q1 (grey) and Q2 (blue) lines with shaded areas. Below the chart is a 'Recent Activity' section with a table of five items.

Annotations on the left side of the dashboard:

- Add a dismissible alert (points to the yellow alert box)
- Convert the default dropdown items to be a navigation tree that toggles sub items when clicked (points to the 'Posts' item in the sidebar)
- Sidebar to always be 100% height (points to the sidebar area)
- Add responsive data charts (points to the 'Traffic' chart)

Q1	Q2
2500	3000
2500	2500
3000	3500
3000	2500
3500	3000

Title
1 Is Your Responsive Site Painfully Slow on Mobile?
2 How to Use Google Analytics Event Goals as AdW
3 How we increased our blog's readership by 45% i
4 Going responsive helped CompanyX grow their o 45%
5 About Us

Responsive Layout Tips

Introduction

This chapter is a collection of tips to help you address common responsive layout challenges.

- **How Does CSS Flexbox Work?**
Page 157
- **How Do I Prevent a Full-width Carousel from Being Super Tall on Desktop?**
Page 164
- **How Do I Change the Order of Columns on Mobile?**
Page 169
- **How Do I Change the Default Colors and Fonts?**
Page 172
- **Can I Adjust Text Size with Bootstrap in a Responsive Design?**
Page 174

Content precedes design. Design in the absence of content is not design, it's decoration.

— Jeffrey Zeldman

How Does CSS Flexbox Work?

How to Take Command of CSS Flexbox

As a kid I loved playing with toy plastic army men. I would line them up and pretend to be in command giving orders to my soldiers in order to defeat the enemy.

Since I was in full command, the soldiers would look to me for their orders. Sometimes a hot shot recruit would overstep my orders and go rogue, He would ignore and override my orders and just do what he wanted. My example demonstrates the basic principle of the CSS display property flexbox which is...



Flex items (soldiers) follow the orders given by their flex container (commander).

In addition, an individual flex item has the ability to override the orders given out by the flex container if required. All it needs is a unique class name or ID with a unique set of properties.

In this flexbox tutorial, I will be answering some of the most common questions and show you how to start using flexbox in your projects.

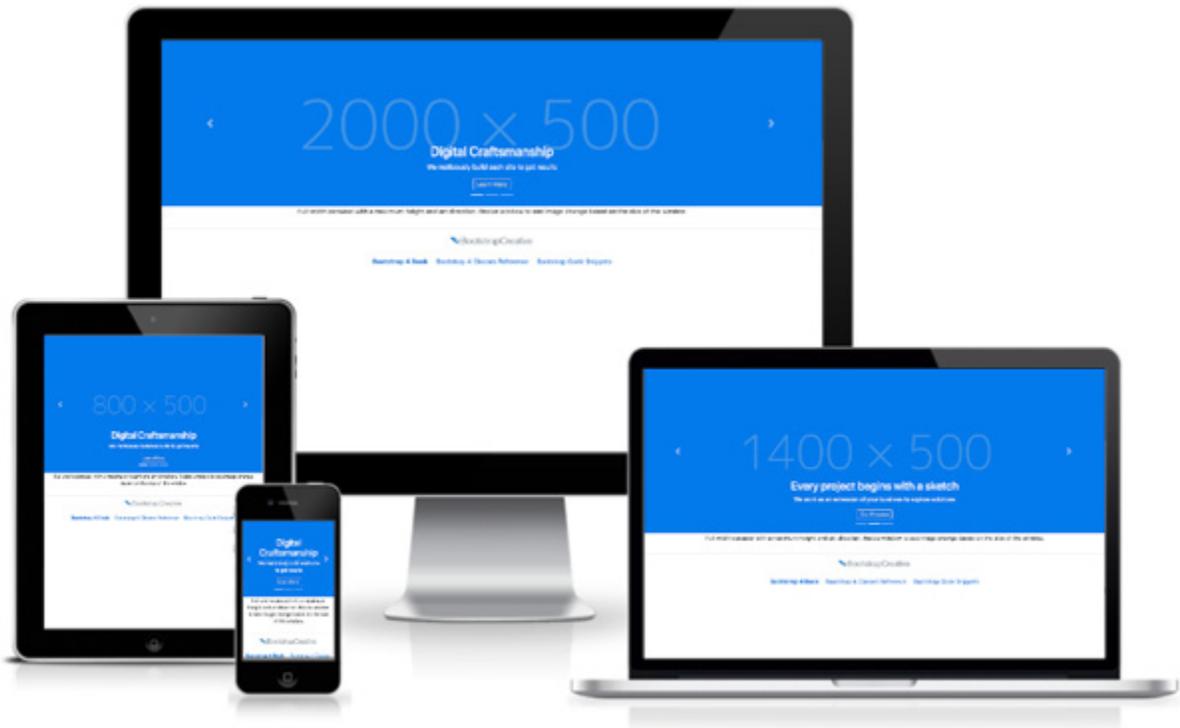
What is Flexbox?

Flexbox is a CSS display property that was introduced in CSS3. If you are familiar with an UL and LI relationship in HTML unordered lists, flexbox is very similar to how it has sub-items or children inside a parent wrapping container. With flexbox they are called flex items.

Summary

In this tutorial, you learned how to customize the carousel code snippet to use responsive images and go full width. This optimization achieves better load times on mobile and gives the designer control on how the images look at the various breakpoints.

The only downsides are that it requires more image editing time and needs a polyfill to support IE11 and below. But in my opinion, the pros greatly outweigh the cons.

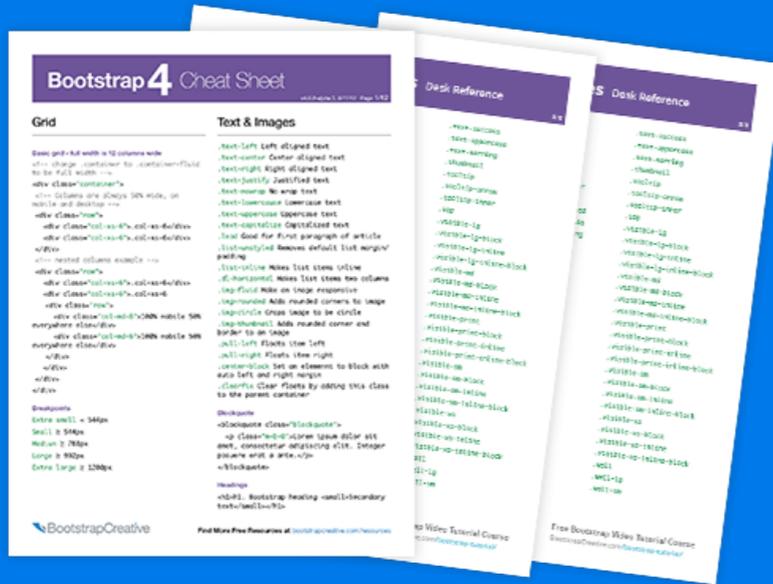


The logo for BootstrapCreative features a stylized icon on the left consisting of a blue curved shape and a grey curved shape. To the right of the icon, the text "BootstrapCreative" is displayed in a light grey, sans-serif font. The word "Creative" has a small blue dot above the letter 'i'.

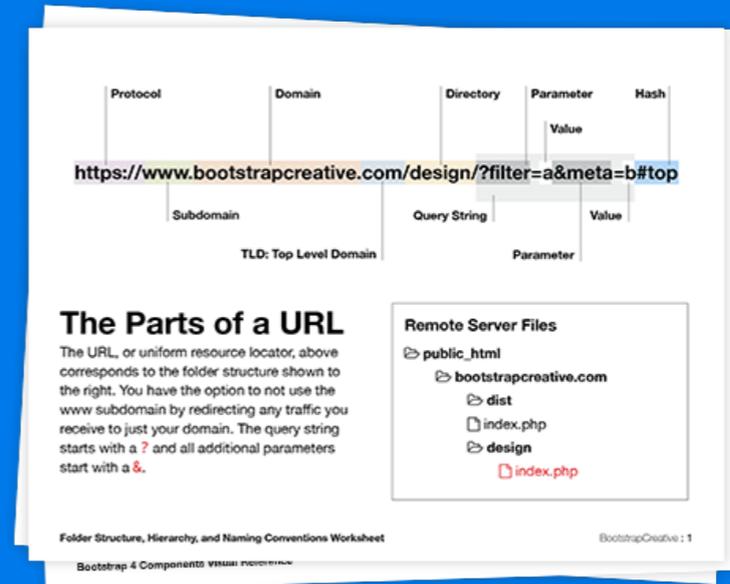
BootstrapCreative

Videos & Quick Reference Guides

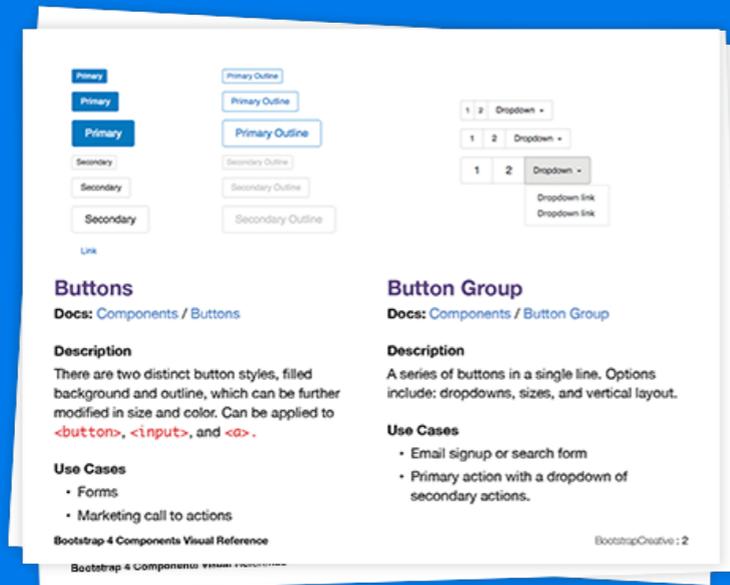
In addition to the book, there are videos and visual reference guides available to help you save time and be more efficient.



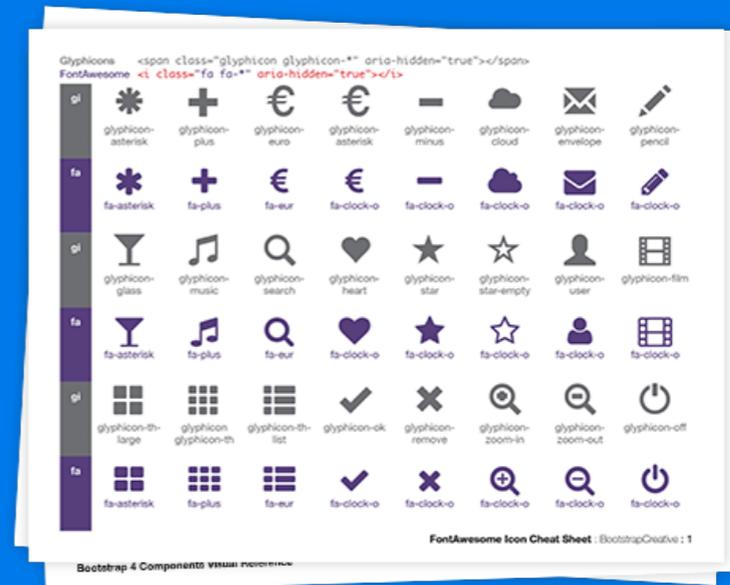
FontAwesome Icon Cheat Sheet



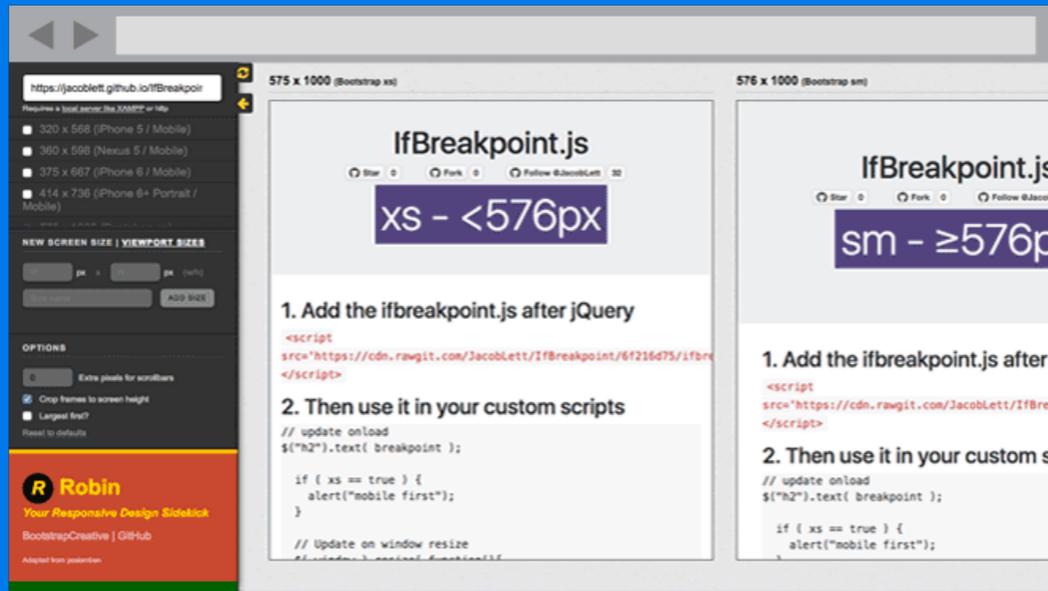
Web Design Primer



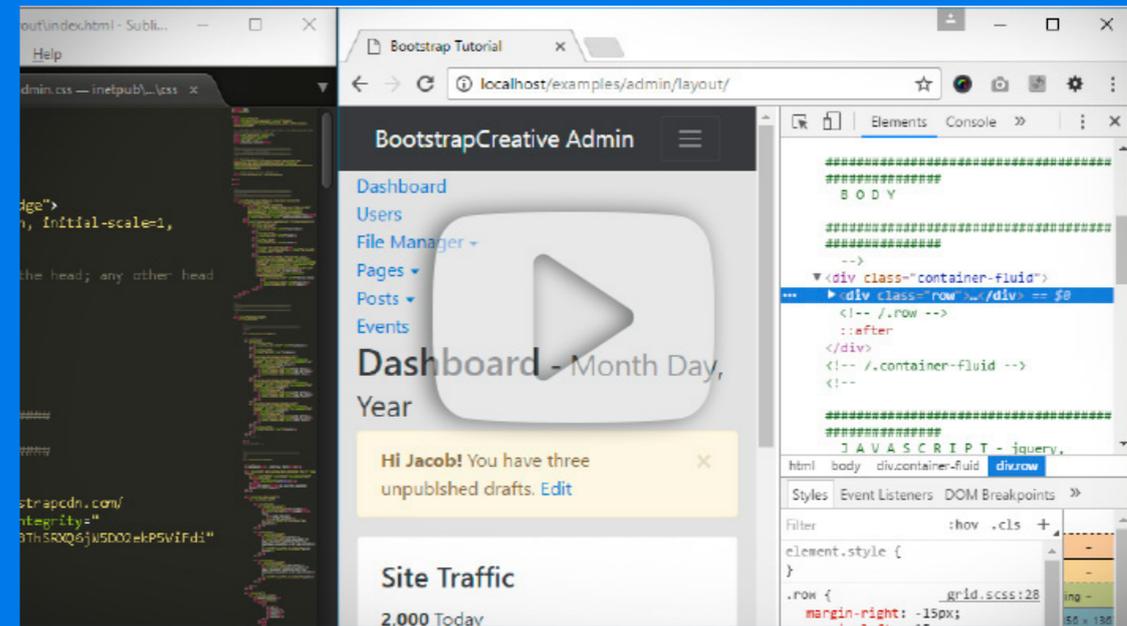
Bootstrap 4 Components Visual Reference



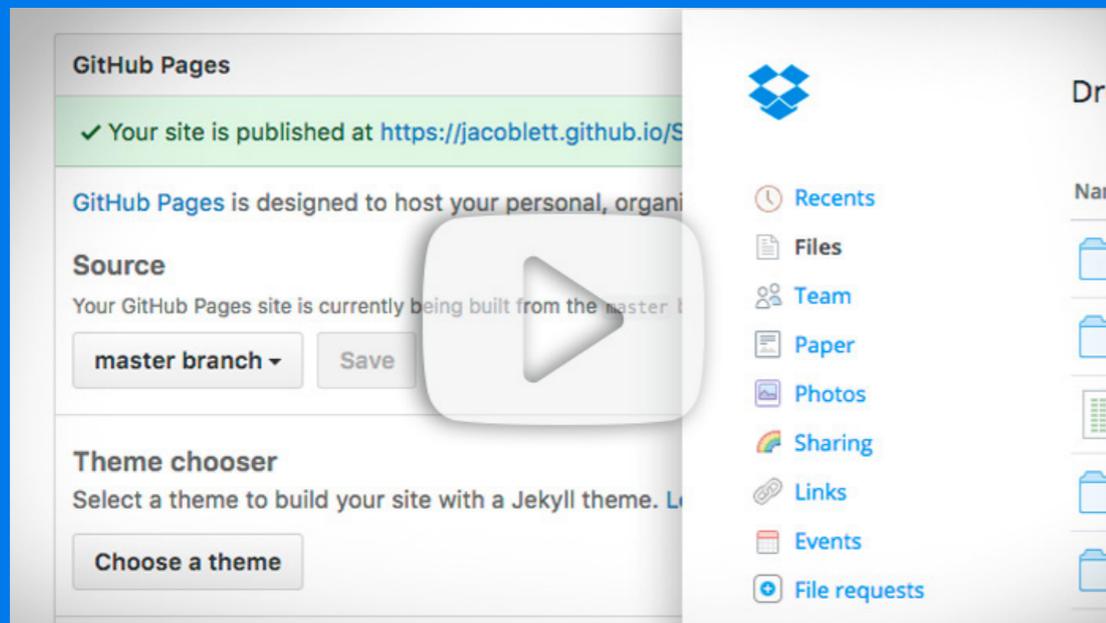
FontAwesome Icon Cheat Sheet



Robin - Responsive Breakpoint Preview Tool



Google DevTools - Inspect and Test CSS



How to Share Prototypes Fast

And More!

**Buy now and start learning
Bootstrap 4 and responsive
web design basics.**



Buy Now