



CodeFirst:Girls Beginners Coding course - Front end Web development

Week 5 - Bootstrap

WHAT IS HARD ABOUT CSS?

As designs get more and more complex, they need more and more code!

© CodeFirstGirls 2017

5

What's hard about CSS?

We have learned quite a bit of CSS by now and it all seems fairly straightforward. You write some CSS, tweak it here and there until it looks the way you want, and you're done! In theory that is exactly how CSS is meant to work and why it is brilliant.

As designs get more and more complex, for larger and larger sites - you will need to write more and more CSS code. For simple designs like navbar, tables and modals, lots of other people will have written CSS to create them on their website, so it is counter-productive for everyone to write their own version.

What else is hard about CSS?

RESPONSIVE WEB DESIGN

Making websites look good on all the different screen sizes we now have.

What else is hard about CSS?

Up until about 5 years ago cross-browser difference were the only thing you had to worry about. Since then the mobile web has exploded and this brought another challenge. Now your website does not only need to look good on a desktop computer, but also on mobiles, tablets and any other device that has a web browser to access the web.

Making websites look good on all these different screen sizes adds a whole new level of complexity. The name for this is responsive web design.

WHAT ARE THE PROBLEMS WE NEED TO SOLVE?

- We need to make sure our websites work on any device shown, regardless of screen size
- We need to become more efficient at writing code

Case study sites - finalised - Google Sheets

© CodeFirst Girls 2017

7

So what problem are we trying to solve?

- We need to make sure our websites work on any device it is shown on, whatever its screen size;
- And we need to become more efficient in writing code. Websites often reuse bits of code for common sections and functions, and we don't want to write the same things over and over again.

The solution? A framework.

The solution: a framework

- A skeleton of code providing generic functionality
- You can use it to extend your own code
- You can override the framework code with your own

What is a framework?

A framework is a skeleton of code providing generic functionality. It's a collection of code you can add to your website project to speed up development. And it has several advantages:

- You can use it as a base for your styles
- You can override the framework code

Twitter Bootstrap

- It's a framework
- Made and maintained by the team behind Twitter
- A set of ready-made CSS & JavaScript files with style solutions and functions to common requirements
- It's cross browser
- And it's responsive!

What is Bootstrap

An example of a framework is Bootstrap. It's a collection of CSS and JavaScript files developed by the makers of Twitter and it is maintained by its developers.

It provides a set of ready-made CSS files with pre-built functions for common web development and presentation requirements. All the solutions are cross-browser compatible - you don't have to worry about coding your own fallbacks - and it is responsive.

Using Twitter Bootstrap

1. Link the Twitter Bootstrap files in the head of your HTML pages
2. Attach the relevant Twitter Bootstrap classes to your HTML elements

<http://getbootstrap.com/getting-started/>

Using Bootstrap

To use Bootstrap you need to do two things:

1. Link the Bootstrap stylesheet in the head of your HTML pages.
2. Attach the relevant Bootstrap classes to your HTML elements.

Bootstrap has extensive documentation on how it works, how to use and examples of what you can do with it. In order to understand the framework it is vital that you read the documentation. Basically it's a cookbook containing lots of different recipes to flavour your projects. The documentation is available on the Bootstrap website.

<http://getbootstrap.com/getting-started/>

Getting started with Bootstrap

There are two ways to include Bootstrap in your projects:

- Link to the Bootstrap CDN
- Download the Bootstrap files and add them to your website folder

Getting started with Bootstrap

There are two ways to include Bootstrap to your projects:

- Link to the Bootstrap CDN
- Download the Bootstrap files and add them to your website folder

CONTENT DELIVERY NETWORKS

- What is a Content Delivery Network?
- Why would you need one?
- CDNs speed the delivery of content to users when websites are facing high traffic across a global reach



© CodeFirst Girls 2017

Content delivery network

A **content delivery network** or **content distribution network (CDN)** is a geographically distributed network of servers. The goal is to distribute service spatially relative to end-users to provide high availability and high performance. CDNs serve a large portion of the Internet content today, including web objects (text, graphics and scripts), downloadable objects (media files, software, documents), applications (e-commerce, portals).

However using a CDN can be risky:

- If for some reason your call to the CDN is not working, your crucial files will not be loaded by the browser.
- If the owner of the CDN moves the files to a different location and you are not aware of that, your crucial files will not be loaded by the browser.

However, they are much easier to use than downloading everything to your machine, so that's what we will be using today!

DOWNLOAD THE BOOTSTRAP STARTER EXERCISE

<https://github.com/CodeFirstGirls/week-five>

- Click on the **fork** icon, to copy it onto your github account
- On your copy, Click **clone or download**
- **Download ZIP**
- Go to your downloads folder
- Unzip the folder
- Place the folder in your `coding_course` folder

© CodeFirstGirls 2017

14

Task

Get students to fork the exercise for week 5 (it is important that they fork it, and not clone it). The first task is to add bootstrap to the project using the CDN.

We will then use the repo later on as a base to add/play and experiment with different bootstrap elements.

The essence of Bootstrap

A huge collection of classes

- In the Bootstrap documentation find the style you want to add to your project
- Copy the Bootstrap class(es) exactly as they are listed
- Add the class(es) to your HTML element
- Save the file and have a look

The essence of Bootstrap

Now that we have added the Bootstrap CSS file in our project we can start using Bootstrap. Essentially Bootstrap is a huge collection of styles attached to CSS classes. If you want to add Bootstrap styling to your HTML you need to do the following:

- Go to the Bootstrap documentation and find the style you want to add
- Copy the Bootstrap class exactly as it is listed in the documentation
- Add the class to your HTML element

KEY BOOTSTRAP COMPONENTS!

<https://getbootstrap.com/docs/4.1/components/>

- Nav bars
- Alerts
- Grid layouts
- modals

There many different bootstrap components, but the most common (and therefore the ones we will focus on today) are nav bars, grid layouts, buttons, modals and carousels.

The `.container` div

This class will center the `<div>` inside the viewport

```
...  
<body>  
  <div class="container">  
    <!-- page content goes here -->  
  </div>  
</body>  
</html>
```

Page margins

Bootstrap makes it easy to center content inside the viewport by using the `.container` class. You can add the class to a `<div>` element and then “contain” all the HTML you want to be centered inside this `<div>`.

You’ll be using container divs a lot in your Bootstrap projects.

Columns

- Bootstrap works on a grid layout with 12 columns
- Each set of columns sit inside a
`<div class="row"></div>`
- Each column `<div>` has a class to indicate the size of the column
`<div class="col-xs-4"></div>`

Columns

Inside a `.container` you might want to divide things up further and maybe create columns. Bootstrap works on a grid layout with 12 columns. To create a column layout you use a combination of `.row` and `.col` classes

1. First you need to create a `.row` which is going to contain your columns. You do this by adding a `<div>` with a class of `.row`.
2. Inside this `.row` div you then create other divs, one for each column.
3. Each column `<div>` needs a class starting with `col`
4. The next part of the class indicates from which viewport width this class applies.

You can see a table of the Grid options in the Bootstrap documentation.

- a. `-xs` is for extra small devices like phones
- b. `-sm` is for tablet devices
- c. `-md` is for medium sized desktops and laptops
- d. `-lg` is for large desktop devices
- e. If you leave it without - it will apply it to all devices

COLUMNS

Example of two even-sized columns

```
<div class="container">
  <div class='row'>
    <div class='col-sm'>
      <!-- Column content goes here -->
    </div>
    <div class='col-sm'>
      <!-- Column content goes here -->
    </div>
  </div>
</div>
```

20

This is an example of two even sized columns would look like in your HTML.

What if you don't like the Bootstrap styles?

- Override it!
- But it's best to stick with the Bootstrap layout - it's tried and tested in browsers
- Keep overrides to fonts, colours and other small things

Modifying Bootstrap

Our site looks pretty good now by only adding a couple of things, but maybe you don't like how Bootstrap makes some things look? Also, imagine if all websites in the world used Bootstrap out-of-the-box without tweaking anything. The web would look a little tedious.

Well, here is the good news. We can override the styling Bootstrap gives us with our own styles. But a word of warning first.

Bootstrap has been designed and heavily tested for good cross-browser compatibility. Unless you know what you are doing, or have a lot of time, it's probably best to stick with the Bootstrap layout and keep your overrides to fonts, colours and general small things that leave the layout structure well alone.

- Create a custom CSS file
- Link to your file AFTER the Bootstrap files

The correct way of overriding Bootstrap styles is by creating a new CSS file of your own and linking to it in your HTML. Never ever make changes in the Bootstrap files themselves.

- ## Badges and Buttons

Badges and Buttons

16

CODEFIRST Girls

TASKS

1. Add a Carousel to the top of your page, with pictures!
([instructions here](#))
2. Add a Modal to your page ([instructions here](#))
3. Write over bootstrap! In the 'styles.css' file, change the design of the default CSS styles (this could be colors, fonts, images - the website is your oyster!)
4. Add any more bootstrap components you think look cool!
(The slides after this one can be inspiration)

© CodeFirst Girls 2017 23

Task!

Carry out the instructions using the bootstrap template we downloaded earlier in the class. If in doubt, reference bootstrap for help!