HTML5 Authoring with Mark Lassoff

Section 7: CSS Libraries and Frameworks

In modern day development, many of the problems traditionally faced by developers and designers have been solved. Many of the solutions are publicly available for you to include in your work as a library or Framework.

I often say, "A good developer is a lazy developer." Finding and integrating completed solutions into your work will allow you to work more quickly and often give your work a professional finish that you'd have difficulty accomplishing yourself.

After completing this section you will:

- ☐ Understand Where to Find CSS Libraries and Frameworks
- ☐ Know how to access a library from your own work
- ☐ Be able to use the library and its features in your own work.

> Note: While there are differences between libraries and Frameworks, I am using the terms interchangeably here and we're covering examples from both categories.

Watch This: HTML5 Section 007 Video

As always your course videos are available on YouTube, Roku and other locations. However, only

those officially enrolled have access to this course guide, are able to submit assignments, work with the instructor, and get this guide.

Watch the section video at: https://www.youtube.com/watch?v=98RiXCss_9o

Choosing a CSS Library

There are dozens of CSS libraries and Frameworks to choose from. The job of a library is to take care of heavy lifting and leave you, as the developer, free to work on implementation.

> As you proceed through the Framework Program, you learn to hand develop functions and actions that some of these libraries do for you automatically. Some skip learning the pure HTML5, CSS and Javascript required to execute these functions.
>
> While this may seem like a more expedient way to achieve professional goals, skipping the basics will leave deficits in your abilities that will render you unable to solve problems that aren't represented in existing libraries.

The first step in choosing a library is determining what exactly you want to use the library for? Some possible justifications for using a CSS library include:

- Using the library's graphic design scheme to make your site appear more uniform and attractive
- Taking advantage of the library's page layout system to render a complex multi-column web layout
- Implementing the library strategy for working in the realm of "responsive design" where different

variants of screen size and screen resolution can be supported.

- Ensuring your site maximizes usability on mobile screens.

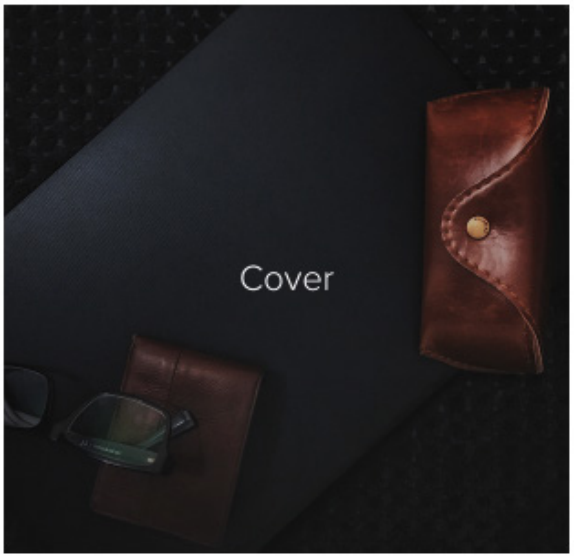
Using a library simply because you find it easier may be a poor long-term approach as you inevitably run into limitations of the library as your project expands.

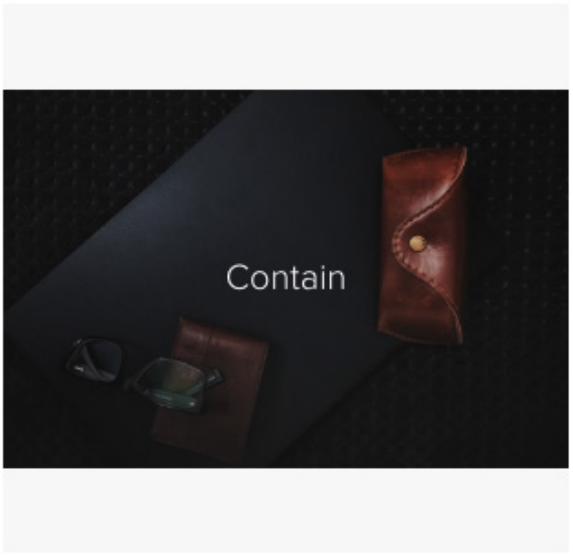
Do This: Examine These CSS Libraries

As a digital professional you should be aware of some of the most common libraries in use today. When working with nontechnical stakeholders, you are likely to be responsible for selecting appropriate CSS libraries for a project.

Visit the web pages for the following libraries and read about their capabilities.

1. Bootstrap (http://getbootstrap.com/2.3.2/index.html) Bootstrap is quite possibly the most popular library in use today.
2. Foundation (https://foundation.zurb.com/) Foundation is a great Framework if Responsive Design is important to you.
3. Bulma (https://bulma.io/) No Framework makes a column-style layout easier.
4. Materialize (https://materializecss.com/) A responsive framework with a modern material design aesthetic.
5. UIkit (https://getuikit.com/) UIkit is popular for the number of powerful components on the library. This library makes elements like full-screen cover images (seen below) easy.





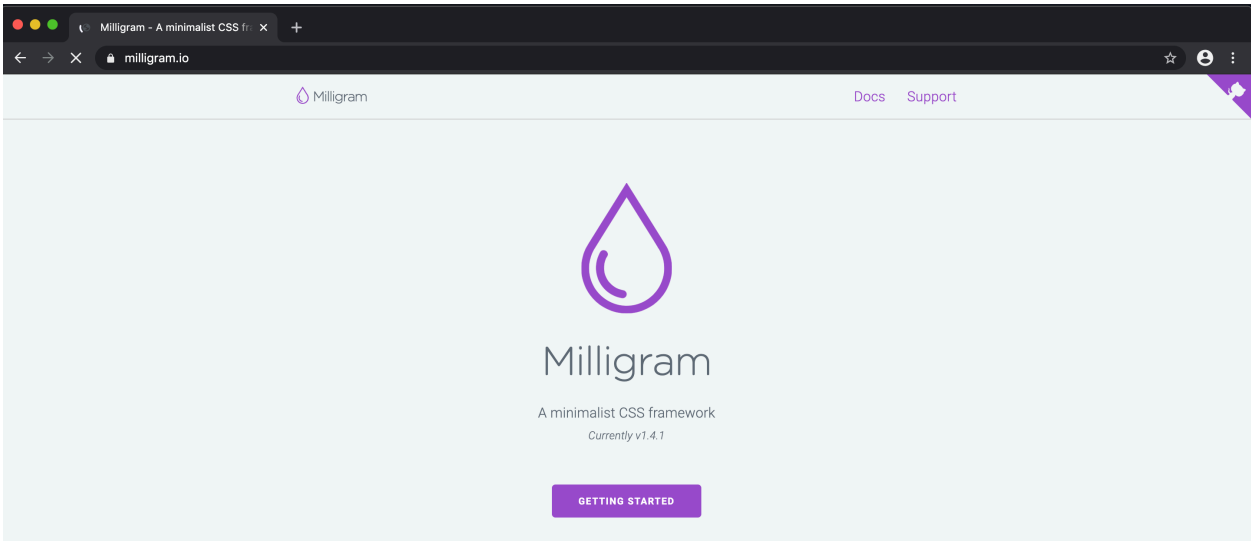
You might be surprised to know that most libraries and frameworks are free to download and use. Community volunteers maintain many of these libraries as well.

Accessing a Libary from Your Code

There are generally two ways that these external libraries are accessed from your code. Some libraries have a complete installation routine in which the library files must be installed in your project folder. Others will allow you to directly download the library files and include them using the `<link>` and `<script>` tags. Each library should have instructions for its use on the library website.

Of course, the installation of all libraries differs, but, as an example, we're going to use Milligram, a "minimalist CSS framework."

Milligram is a very small library that's designed to help you create clean code as well as a clean interface! Milligram's web page is an example of the minimalism the company supports.



Visit www.milligram.io and click the button marked "Getting Started." You'll now see the download button which is used to download Milligram's very small library to your computer.

Once you have the library downloaded, unzip the files. From the `dist` folder, grab the file entitled `milligram.min.css` and place that in the folder in which you're going to store your code.

Using Library Features

We'll implement the Milligram library thorough linking a series of stylesheets to our HTML document. Create an HTML basic document structure and add the code in the listing below.

```
<!DOCTYPE html>
<html>
```

```
<head>
  <title>CSS Library</title>
  <link  rel="stylesheet"
href="http://fonts.googleapis.com/
css?family=Roboto:300,300italic,700,700italic">
  <link  rel="stylesheet"
href="http://cdn.rawgit.com/necolas/normalize.css/master/
normalize.css">
  <link rel="stylesheet"  href="milligram.min.css">
  <style>
    #container{
     margin: 6px;
     }
  </style>
</head>
<body>
  <div id="container">
  <h1>Using a library</h1>
  <p>Once you install it, using a library like <strong>Milligram</
strong>
   is actually quite easy.</p>
   </div> <!-- container -->
</body>
</html>
```

Note that this library takes advantage of Google Fonts, specifically varients of the Roboto typeface. The next link is to a `normailze.css` file. These important files essentially defeat the default style sheet in every browser to give you a zero starting point. CSS rules like space before and after headers is effectively discarded so the library can more easily use it's own CSS rules without fear of accidental overrides.

> Because the `normailze.css` file resets everything to zero, I added a style to pull the content away from the margin of the container.

The third link is to the `milligram.min.css` file you just downloaded. This file is a minified version of the

Milligram library. The file itself is compacted so that as little space is used by the file as possible, making it fast to download every across poor connections.

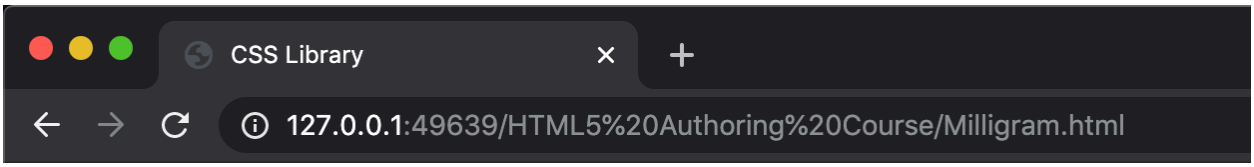
We'll implement Milligram features simply by writing HTML. Let's make some changes to our document.

```
<!DOCTYPE html>
<html>
<head>
  <title>CSS Library</title>
  <link  rel="stylesheet"
href="http://fonts.googleapis.com/
css?family=Roboto:300,300italic,700,700italic">
  <link  rel="stylesheet"
href="http://cdn.rawgit.com/necolas/normalize.css/master/
normalize.css">
  <link rel="stylesheet"  href="milligram.min.css">
  <style>
    #container{
      margin: 6px;
    }
  </style>
</head>
<body>
  <div id="container">
  <h1>Using a library</h1>
  <p>Once you install it, using a library like <strong>Milligram</
strong>
   is actually quite easy.</p>
  <h2>Where I like to get coffee:</h2>
  <ul>
    <li>Starbucks</li>
    <li>Dunkin Donuts</li>
    <li>McDonald's</li>
    <li>Diner</li>
    <li>Train Station Coffee Stand</li>
  </ul>

  </div> <!-- container -->
```

```
</body>
</html>
```

If you've done everything correctly, the file will appear in your browser like this:



Using a library

Once you install it, using a library like **Milligram** is actually quite easy.

Where I like to get coffee:

- Starbucks
- Dunkin Donuts
- McDonald's
- Diner
- Train Station Coffee Stand

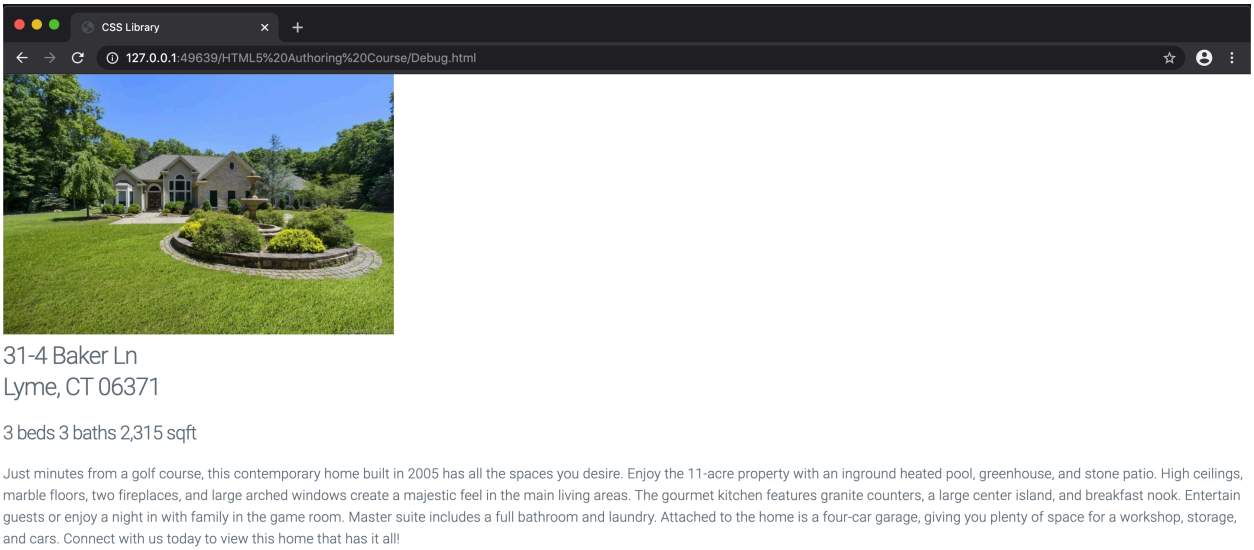
Debug This: Semantic HTML

When this file is corrected it should render its content according to the Milligram library.

```
<!DOCTYPE html>
<html>
<head>
  <title>CSS Library</title>
  <link  rel="stylesheet"
href="//fonts.googleapis.com/
css?family=Roboto:300,300italic,700,700italic">
  <link  rel="stylesheet"
href="//cdn.rawgit.com/necolas/normalize.css/master/
normalize.css">
  <link rel="stylesheet"  href="/milligram.min.css">
  <style>
    .container{
      margin: 6px;
    }
  </style>
</head>
<body>
  <div id="container">
  <img
  src="https://s3.amazonaws.com/coursewareframework/
html5authoring/lyme.jpg"/>
  <h3>31-4 Baker Ln<br/>
    Lyme, CT 06371</h3>
    <h4>3 beds 3 baths 2,315 sqft</h4>
    <p>Just minutes from a golf course, this contemporary
home built in 2005
    has all the spaces you desire. Enjoy the 11-acre property
with an
    inground heated pool, greenhouse, and stone patio. High
ceilings, marble
    floors, two fireplaces, and large arched windows create a
majestic feel
    in the main living areas. The gourmet kitchen features
granite counters,
    a large center island, and breakfast nook. Entertain guests
or enjoy a
    night in with family in the game room. Master suite includes
a full
```

```
        bathroom and laundry. Attached to the home is a four-car
garage, giving
        you plenty of space for a workshop, storage, and cars.
Connect with us
        today to view this home that has it all!</p>
    </div> <!-- container -->
</body>
</html>
```

When debugged, your project should look similar to this:



Submit This: Bulma

Bulma is another popular CSS library. In this exercise, you'll correctly configure Bulma located at www.Bulma.io. The best place to start is with the instructions on the "Getting Started with Bulma" page at https://bulma.io/documentation/overview/start/.

Using Bulma's media object (https://bulma.io/documentation/layout/media-object/) create a document with images and descriptions similar to

the example below. Your page should have three images and three descriptions.

Airbus 220

The Airbus A220, previously known as Bombardier CSeries (or C Series), is a family of narrow-body, twin-engine, medium-range jet airliners marketed by Airbus but designed and originally built by the Canadian manufacturer Bombardier Aerospace. Following Airbus involvement the aircraft are built by CSeries Aircraft Limited Partnership (CSALP).

The 108 to 133-seat A220-100 (formerly CS100) made its maiden flight on 16 September 2013, was awarded an initial type certification by Transport Canada on 18 December 2015 and entered service on 15 July 2016 with Swiss Global Air Lines. The 130 to 160-seat A220-300 (formerly CS300) first flew on 27 February 2015, received an initial type certification on 11 July 2016, and entered service with launch customer airBaltic on 14 December 2016. At service entry operators realized a 21% lower fuel burn for the CS300 in replacing 32-year-old Boeing 737-300s, with a dependability above 99.3%, and 25% lower costs than the RJ100 for the CS100, while the passengers' and pilots' feedback is positive for the cabin and flight controls.

Remember, when submitting the work please use the following naming convention for your file:`HTMLAUTHORING_LastName_SectionNumber.html`. So if your last name is Smith and your submitting section 8, you file name should be `HTMLAUTHORING_Smith_8.html`.

For this course visit https://www.dropbox.com/request/RhW9kBDXtisq2Fsvg3hY to submit your assignments.