INTERNET DESIGN 1

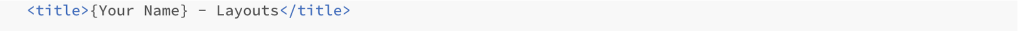
Styleguide

PART 2

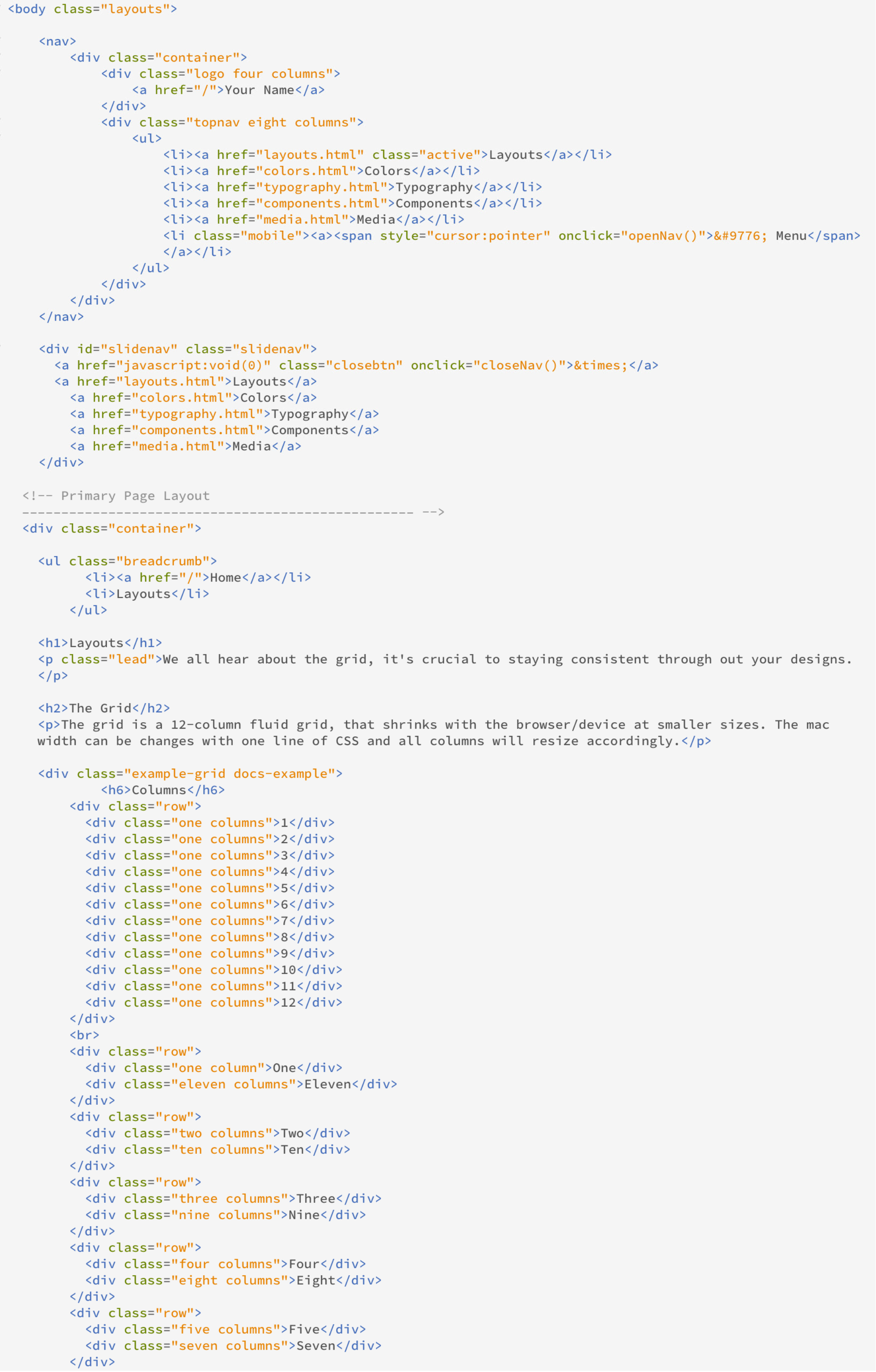
Layouts

The grid, we’ve all heard of it, and in web design and specifically coding it can be extremely helpful. Daunting at times but the consistency will make you life a lot better. So let’s set one up and see how it works out for us.

HTML



```
  <title>{Your Name} - Layouts</title>
```



```
<body class="layouts">

    <nav>
        <div class="container">
            <div class="logo four columns">
                <a href="/">Your Name</a>
            </div>
            <div class="topnav eight columns">
                <ul>
                    <li><a href="layouts.html" class="active">Layouts</a></li>
                    <li><a href="colors.html">Colors</a></li>
                    <li><a href="typography.html">Typography</a></li>
                    <li><a href="components.html">Components</a></li>
                    <li><a href="media.html">Media</a></li>
                    <li class="mobile"><a><span style="cursor:pointer" onclick="openNav()">&#9776; Menu</span>
                    </a></li>
                </ul>
            </div>
        </div>
    </nav>

    <div id="slidenav" class="slidenav">
      <a href="javascript:void(0)" class="closebtn" onclick="closeNav()">&times;</a>
      <a href="layouts.html">Layouts</a>
        <a href="colors.html">Colors</a>
        <a href="typography.html">Typography</a>
        <a href="components.html">Components</a>
        <a href="media.html">Media</a>
    </div>

  <!-- Primary Page Layout
  --------------------------------------------------- -->
  <div class="container">

    <ul class="breadcrumb">
          <li><a href="/">Home</a></li>
          <li>Layouts</li>
        </ul>

    <h1>Layouts</h1>
    <p class="lead">We all hear about the grid, it's crucial to staying consistent through out your designs.
    </p>

    <h2>The Grid</h2>
    <p>The grid is a 12-column fluid grid, that shrinks with the browser/device at smaller sizes. The mac
    width can be changes with one line of CSS and all columns will resize accordingly.</p>

    <div class="example-grid docs-example">
            <h6>Columns</h6>
        <div class="row">
          <div class="one columns">1</div>
          <div class="one columns">2</div>
          <div class="one columns">3</div>
          <div class="one columns">4</div>
          <div class="one columns">5</div>
          <div class="one columns">6</div>
          <div class="one columns">7</div>
          <div class="one columns">8</div>
          <div class="one columns">9</div>
          <div class="one columns">10</div>
          <div class="one columns">11</div>
          <div class="one columns">12</div>
        </div>
        <br>
        <div class="row">
          <div class="one column">One</div>
          <div class="eleven columns">Eleven</div>
        </div>
        <div class="row">
          <div class="two columns">Two</div>
          <div class="ten columns">Ten</div>
        </div>
        <div class="row">
          <div class="three columns">Three</div>
          <div class="nine columns">Nine</div>
        </div>
        <div class="row">
          <div class="four columns">Four</div>
          <div class="eight columns">Eight</div>
        </div>
        <div class="row">
          <div class="five columns">Five</div>
          <div class="seven columns">Seven</div>
        </div>
```

1. Make a duplicate of the index.html file
2. Replace the title text to include “Layouts”
3. Change the body class to layouts
4. The nav section we are defining our menu the will show up across the top. We are also building a duplicate menu which will populate our mobile sliding menu.
5. Breadcrumbs are those useful little links at the top of most pages that helps you get back to the parent page. We don’t have deep nested pages here but it’s good practice to include them.
6. Now starts the fun where you’ll see the columns come together. The classes that are added to these divs in crucial to how the grid works. Once you copy the code down and get the css added you’ll see it come to life.

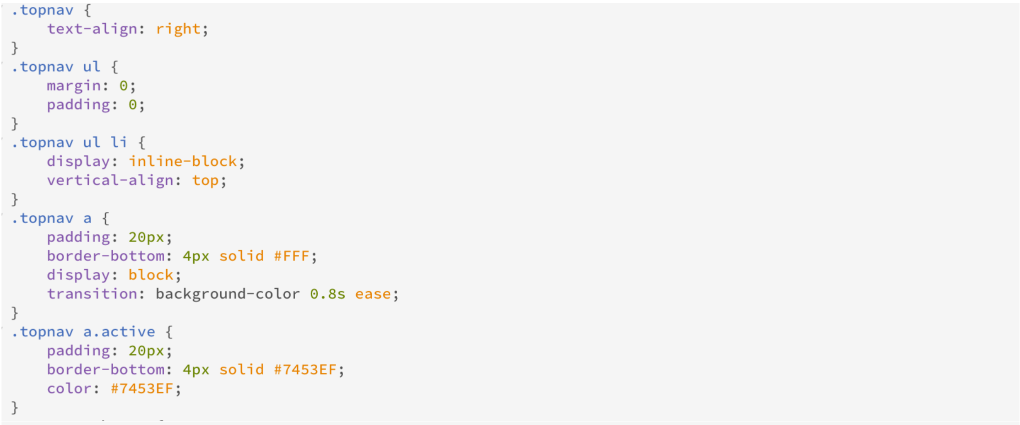
```
        <div class="row">
          <div class="six columns">Six</div>
          <div class="six columns">Six</div>
        </div>
        <div class="row">
          <div class="seven columns">Seven</div>
          <div class="five columns">Five</div>
        </div>
        <div class="row">
          <div class="eight columns">Eight</div>
          <div class="four  columns">Four</div>
        </div>
        <div class="row">
          <div class="nine columns">Nine</div>
          <div class="three columns">Three</div>
        </div>
        <div class="row">
          <div class="ten columns">Ten</div>
          <div class="two columns">Two</div>
        </div>
        <div class="row">
          <div class="eleven columns">Eleven</div>
          <div class="one column">One</div>
        </div>
        <br>
        <br>
        <div class="row">
          <div class="three columns">Three</div>
          <div class="three columns">Three</div>
          <div class="three columns">Three</div>
          <div class="three columns">Three</div>
        </div>
        <br>
        <div class="row">
          <div class="four columns">Four</div>
          <div class="four columns">Four</div>
          <div class="four columns">Four</div>
        </div>
      </div>
  </div>

  <footer>
    <div class="container">
        <p>{Your Name}   |   Copyright © {Year}</p>
    </div>
  </footer>

<!-- End Document
  -------------------------------------------------- -->

  <script src="./script.js"></script>

</body>
</html>
```

7. Finish the page off with the footer which will be the same on all pages going forward.
8. Save and lets move onto the css.

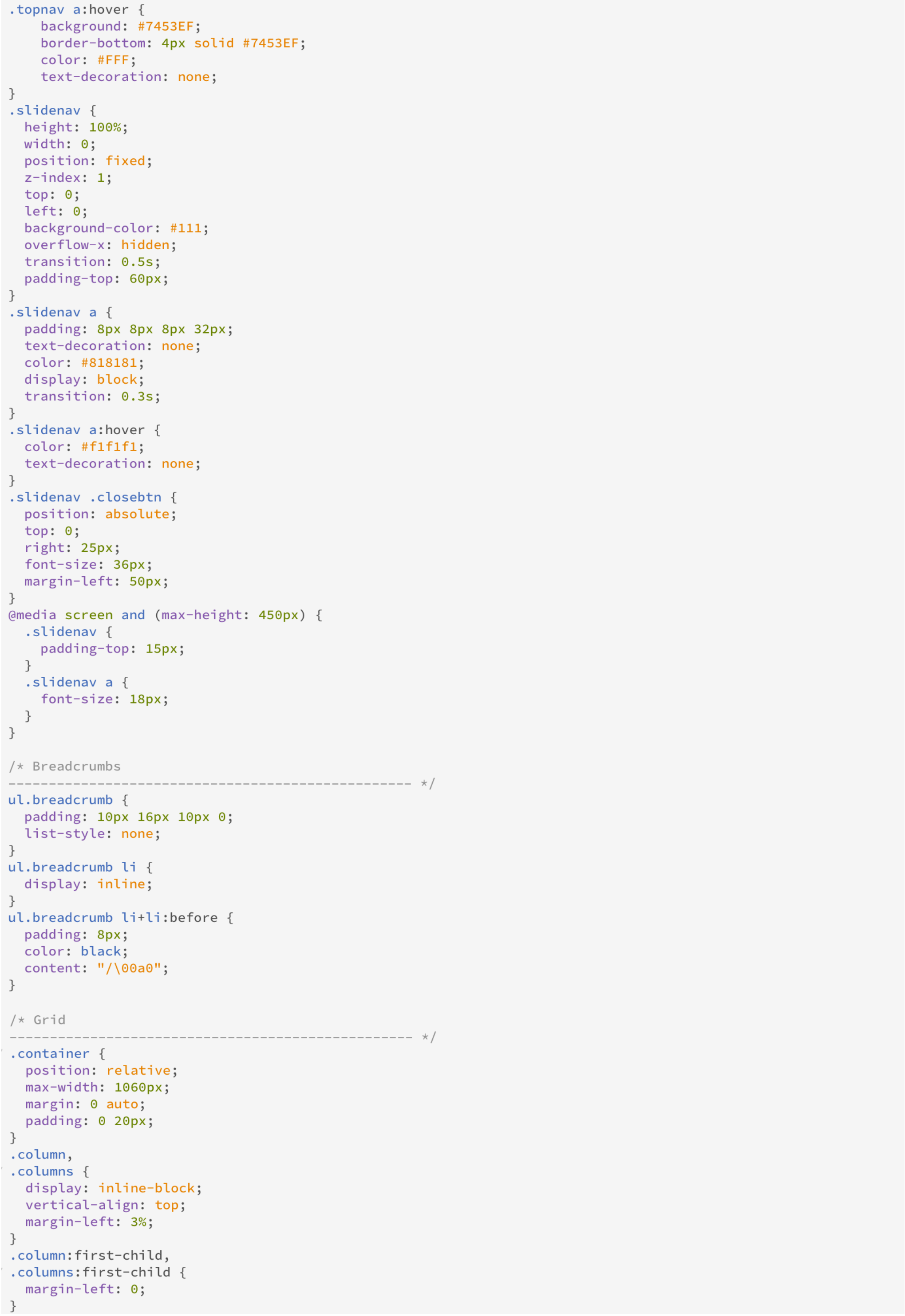
CSS

```
/* Navigation
-------------------------------------------------- */
nav {
    position: fixed;
    background: #FFF;
    top: 0;
    left: 0;
    width: 100%;
    z-index: 1;
    box-shadow: 0 2px 8px 0 rgba(0,0,0,0.25);
}
nav a {
    color: #333;
    font-weight: bold;
}
nav .topnav li.mobile {
    display: none;
}
@media (max-width: 903px) {
    nav .topnav ul li {
        display: none;
    }
    nav .topnav ul li.mobile {
        display: inline-block;
    }
    nav .eight.columns,
    nav .four.columns {
            width: 48%;
    }
}
.logo {
    font-size: 24px;
    margin-top: 16px;
}
.logo a:hover {
    color: #7453EF;
    text-decoration: none;
}
```

9. Let’s start with styling the navigation.
10. You’ll notice a section with @media. That is where we can adjust our page to work in a specific way on smaller screens. We can have as many or few media queries as we want. In this case we are saying, any screen 903px or smaller will have these styles applied to the selected elements contained in the curly brackets.



```css
.topnav {
    text-align: right;
}
.topnav ul {
    margin: 0;
    padding: 0;
}
.topnav ul li {
    display: inline-block;
    vertical-align: top;
}
.topnav a {
    padding: 20px;
    border-bottom: 4px solid #FFF;
    display: block;
    transition: background-color 0.8s ease;
}
.topnav a.active {
    padding: 20px;
    border-bottom: 4px solid #7453EF;
    color: #7453EF;
}
```

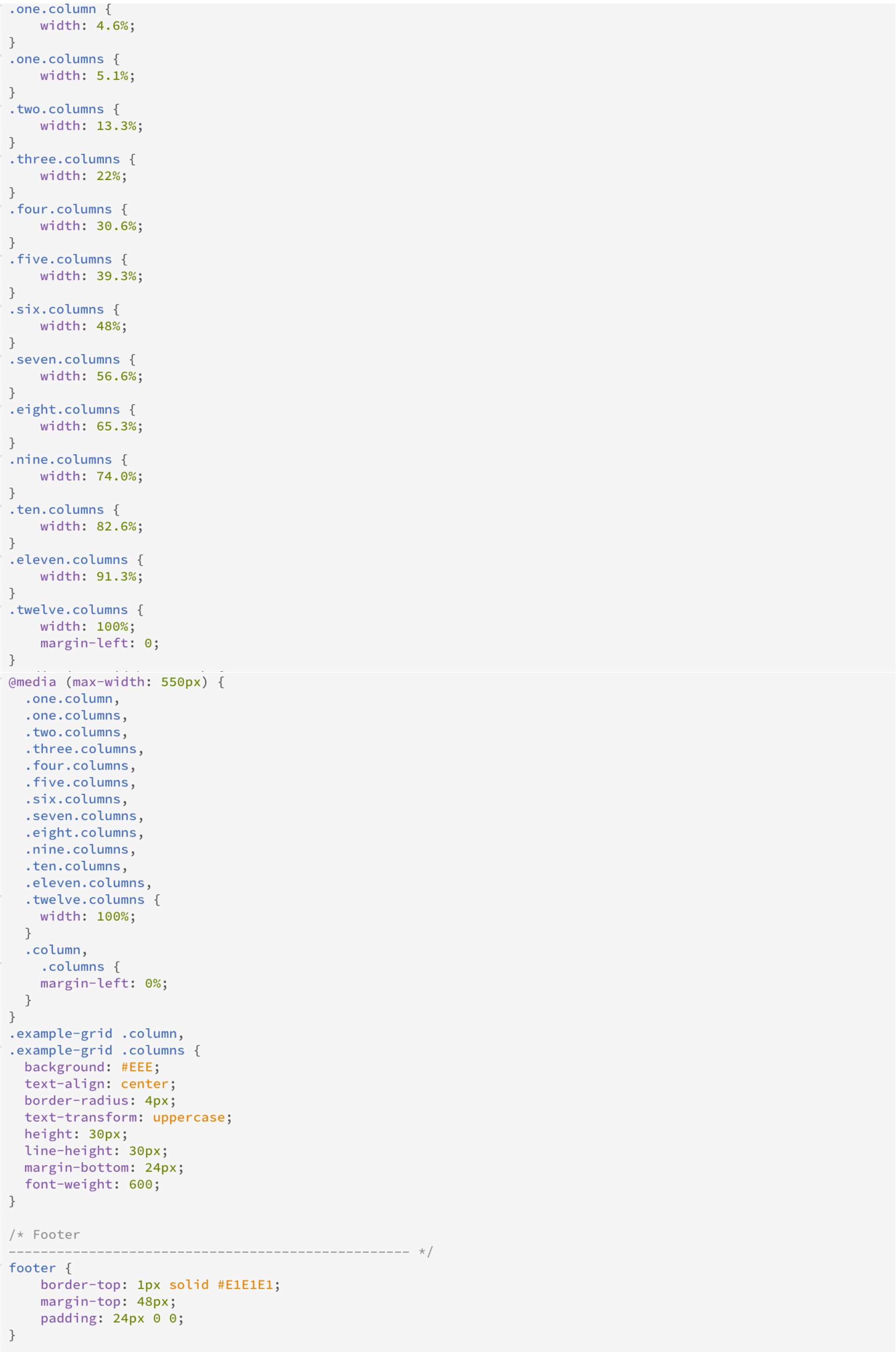


```css
.topnav a:hover {
    background: #7453EF;
    border-bottom: 4px solid #7453EF;
    color: #FFF;
    text-decoration: none;
}
.slidenav {
  height: 100%;
  width: 0;
  position: fixed;
  z-index: 1;
  top: 0;
  left: 0;
  background-color: #111;
  overflow-x: hidden;
  transition: 0.5s;
  padding-top: 60px;
}
.slidenav a {
  padding: 8px 8px 8px 32px;
  text-decoration: none;
  color: #818181;
  display: block;
  transition: 0.3s;
}
.slidenav a:hover {
  color: #f1f1f1;
  text-decoration: none;
}
.slidenav .closebtn {
  position: absolute;
  top: 0;
  right: 25px;
  font-size: 36px;
  margin-left: 50px;
}
@media screen and (max-height: 450px) {
  .slidenav {
    padding-top: 15px;
  }
  .slidenav a {
    font-size: 18px;
  }
}

/* Breadcrumbs
--------------------------------------------------- */
ul.breadcrumb {
  padding: 10px 16px 10px 0;
  list-style: none;
}
ul.breadcrumb li {
  display: inline;
}
ul.breadcrumb li+li:before {
  padding: 8px;
  color: black;
  content: "/\00a0";
}

/* Grid
--------------------------------------------------- */
.container {
  position: relative;
  max-width: 1060px;
  margin: 0 auto;
  padding: 0 20px;
}
.column,
.columns {
  display: inline-block;
  vertical-align: top;
  margin-left: 3%;
}
.column:first-child,
.columns:first-child {
  margin-left: 0;
}
```

11. The slidenav is the mobile menu that we created a duplicate menu structure for.

12. Here we have our breadcrumbs and you'll notice the :before, which you don't see in the HTML you typed. You're able to activate "before" and "after" elements to style content as needed. You probably wont use it a ton but it can be super helpful when in this case adding the break between links.

13. And here are the columns, we'll be adding backgrounds to these columns but thats just for the layouts display, it wont display on other pages when we add in the columns. Notice the use of percentages everywhere.



```
.one.column {
    width: 4.6%;
}
.one.columns {
    width: 5.1%;
}
.two.columns {
    width: 13.3%;
}
.three.columns {
    width: 22%;
}
.four.columns {
    width: 30.6%;
}
.five.columns {
    width: 39.3%;
}
.six.columns {
    width: 48%;
}
.seven.columns {
    width: 56.6%;
}
.eight.columns {
    width: 65.3%;
}
.nine.columns {
    width: 74.0%;
}
.ten.columns {
    width: 82.6%;
}
.eleven.columns {
    width: 91.3%;
}
.twelve.columns {
    width: 100%;
    margin-left: 0;
}
@media (max-width: 550px) {
  .one.column,
  .one.columns,
  .two.columns,
  .three.columns,
  .four.columns,
  .five.columns,
  .six.columns,
  .seven.columns,
  .eight.columns,
  .nine.columns,
  .ten.columns,
  .eleven.columns,
  .twelve.columns {
    width: 100%;
  }
  .column,
    .columns {
    margin-left: 0%;
  }
}
.example-grid .column,
.example-grid .columns {
  background: #EEE;
  text-align: center;
  border-radius: 4px;
  text-transform: uppercase;
  height: 30px;
  line-height: 30px;
  margin-bottom: 24px;
  font-weight: 600;
}

/* Footer
--------------------------------------------------- */
footer {
    border-top: 1px solid #E1E1E1;
    margin-top: 48px;
    padding: 24px 0 0;
}
```

14. Here the media query is saying that all columns at 550px or smaller will span 100% of the screen. At a certain point have columns just doesn't work. It's better to have the content span the whole screen when it's so small.

15. Style the footer very simply.
16. Save and see what it looks the masterpiece.

GITHUB AND NETLIFY

Just as we did with the first step we'll get these files uploaded to GitHub and a viewable website on Netlify.

1. Navigate to Github.com
2. Upload your files with a new commit message (index files uploaded)
3. Give it a minute to upload the files
4. Netlify should automatically update in a couple minutes. Their site constantly checks for updates to your code on GitHub.
5. Email me (banderson@lcad.edu) the url for the site so I can record this assignment complete.