INTERNET DESIGN 1

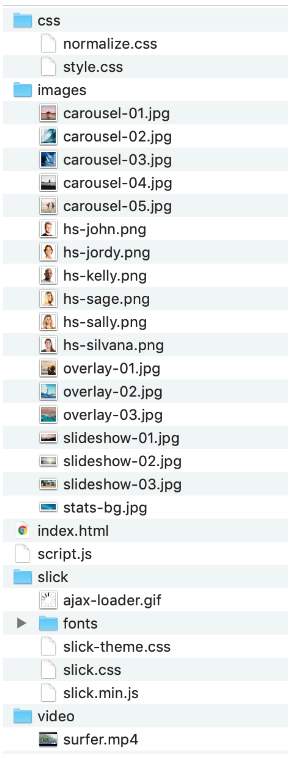
Styleguide

INTRODUCTION

Getting Started

To start things off lets first look at what's included in the folder and what you have to work with. The index.html, css, js, image and video files are provided to help you just focus on the coding side of the project. Here's an overview of the file structure and some details about specific files and how they work.

FILE STRUCTURE



```
css
    normalize.css
    style.css
images
    carousel-01.jpg
    carousel-02.jpg
    carousel-03.jpg
    carousel-04.jpg
    carousel-05.jpg
    hs-john.png
    hs-jordy.png
    hs-kelly.png
    hs-sage.png
    hs-sally.png
    hs-silvana.png
    overlay-01.jpg
    overlay-02.jpg
    overlay-03.jpg
    slideshow-01.jpg
    slideshow-02.jpg
    slideshow-03.jpg
    stats-bg.jpg
index.html
script.js
slick
    ajax-loader.gif
  ▶ fonts
    slick-theme.css
    slick.css
    slick.min.js
video
    surfer.mp4
```

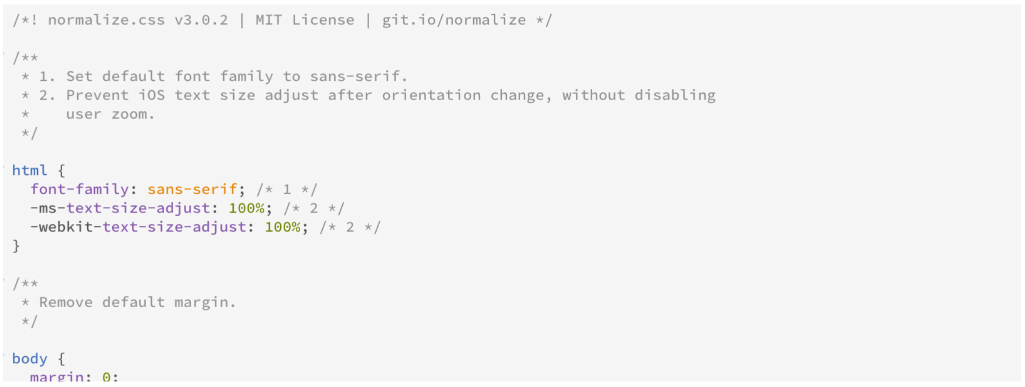
CSS folder contains the normalize.css file which I'll go into detail in the next section and style.css which is where we'll spend all our time inputing the styles for our web pages.

Images obviously contain all the images you'll be referencing during this project.

Script.js and the slick folder at the cool interactions that we'll be adding to the site. We won't go into a ton of detail on how these work since it's a little above this class. Though we want to have some fun experiences with how elements work, so it's a little intro into it. The slick folder specifically is for the slideshow and carousel components you'll implement later.

Video folder contains the one video we'll use in this project.

NORMALIZE.CSS



```css
/*! normalize.css v3.0.2 | MIT License | git.io/normalize */

/**
 * 1. Set default font family to sans-serif.
 * 2. Prevent iOS text size adjust after orientation change, without disabling
 *    user zoom.
 */

html {
  font-family: sans-serif; /* 1 */
  -ms-text-size-adjust: 100%; /* 2 */
  -webkit-text-size-adjust: 100%; /* 2 */
}

/**
 * Remove default margin.
 */

body {
  margin: 0;
```

You don't know it but different browsers can display the same information slightly different. Normalize.css makes browsers render all elements more consistently and in line with modern standards. It precisely targets only the styles that need normalizing.

SCRIPTS.JS

As I mentioned above I'll run through the different parts of the script.js file so you can see whats happening. This file is already created for you and there's no need to type this code out.

```
/* Navigation */
function openNav() {
  document.getElementById("slidenav").style.width = "250px";
}

function closeNav() {
  document.getElementById("slidenav").style.width = "0";
}
```

This navigation section is what tells the mobile menu we'll implement when to show and when to hide.

```
/* Tabs */
function openCity(evt, cityName) {
  var i, tabcontent, tablinks;
  tabcontent = document.getElementsByClassName("tabcontent");
  for (i = 0; i < tabcontent.length; i++) {
    tabcontent[i].style.display = "none";
  }
  tablinks = document.getElementsByClassName("tablinks");
  for (i = 0; i < tablinks.length; i++) {
    tablinks[i].className = tablinks[i].className.replace(" active", "");
  }
  document.getElementById(cityName).style.display = "block";
  evt.currentTarget.className += " active";
}

// Get the element with id="defaultOpen" and click on it
document.getElementById("defaultOpen").click();
```

You'll be coding a tab section and this adds the necessary classes and shows the proper section. It also allows for the first tab to be open and showing its content by default.

```
/* Accordions */
var acc = document.getElementsByClassName("accordion");
var i;

for (i = 0; i < acc.length; i++) {
  acc[i].addEventListener("click", function() {
    this.classList.toggle("active");
    var panel = this.nextElementSibling;
    if (panel.style.maxHeight){
      panel.style.maxHeight = null;
    } else {
      panel.style.maxHeight = panel.scrollHeight + "px";
    }
  });
}
```

You'll be coding an accordion section which does very similar things as the tabs above.

```
/* Expanding Grid */
function openTab(tabName) {
  var i, x;
  x = document.getElementsByClassName("containerTab");
  for (i = 0; i < x.length; i++) {
    x[i].style.display = "none";
  }
  document.getElementById(tabName).style.display = "block";
}
```

The expanding grid is kind of like an accordion but cooler. I really like this kind of functionality and it's a really fun way to give a glimpse of more information without having to navigate to another page just yet.

Good Luck!

Continue on to part 1 of the styleguide project

INTERNET DESIGN 1

Styleguide

PART 1

The Homepage

For any website to work we need to set up the index.html file. When you navigate to any website the first page you will see is the index.html page. The reason for this specific name is to keep things consistent. Every browser knows what page to look for first. If it was named something else you'd never be able to get to the homepage.

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>

  <!-- Basic Page Needs
  –––––––––––––––––––––––––––––––––––––––––––––––––– -->
  <meta charset="utf-8">
  <title>{Your Name} - Home</title>

  <!-- Mobile Specific Metas
  –––––––––––––––––––––––––––––––––––––––––––––––––– -->
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <!-- CSS
  –––––––––––––––––––––––––––––––––––––––––––––––––– -->
  <link rel="stylesheet" href="css/normalize.css">
  <link rel="stylesheet" href="css/style.css">

</head>
<body class="home">

  <!-- Primary Page Layout
  –––––––––––––––––––––––––––––––––––––––––––––––––– -->
  <div class="container">
    <div class="row">
        <p>{Your Name}</p>
        <h1>Coded Styleguide</h1>
        <a class="button" href="layouts.html">Get Started</a>
    </div>
  </div>

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

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

</body>
</html>
```

1. Open the index.html file from the project folder
2. Copy the text to the left note there are some spots where you'll put your name.
3. Here we are setting a meta tag in place to help with responsive actions.
4. Our external style sheets are included, note the difference from last weeks assignment of "reset" and "normalize", they basically do the same thing.
5. Setting up the structure of our page, this will make a bit more sense in the next assignment
6. Lastly, including the javascript script file.
7. Be sure to save constantly throughout coding.

CSS

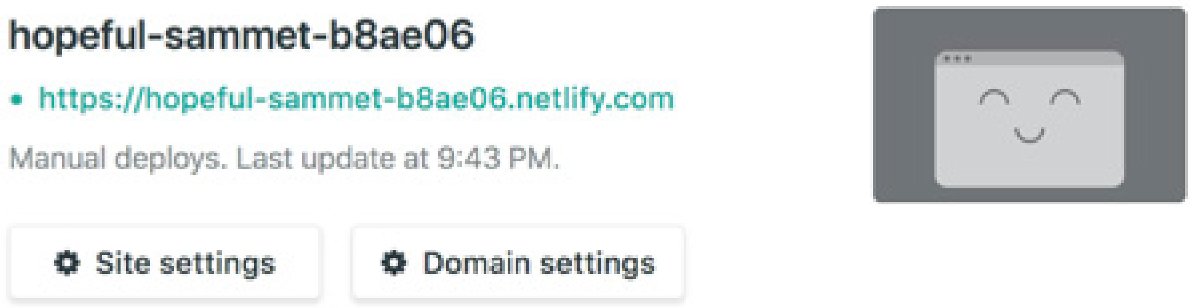
```
/* Base Styles
–––––––––––––––––––––––––––––––––––––––––––––––––– */
body {
  font-size: 16px;
  line-height: 1.6;
  font-weight: 400;
  font-family: Helvetica, Arial, sans-serif;
  color: #333;
  margin-top: 100px;
}
img {
    max-width: 100%;
}

/* Home Styles
–––––––––––––––––––––––––––––––––––––––––––––––––– */
.home {
    background: linear-gradient(180deg, #00C5FF 0%, #B4C873 100%) no-repeat;
    color: #FFF;
    height: 100vh;
    margin: 0;
}
.home .container {
    padding-top: 40vh;
  text-align: center;
}
.home p {
    margin-bottom: 0;
}
.home h1 {
    margin-top: 0;
}
```

8. We are going to work with some really simple CSS to start.
9. Setting some initial font sizes and font families. Feel free to change this to a Google font for extra points.
10. Adding in a gradient background.

GITHUB AND NETLIFY

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

1. Navigate to Github.com
2. Create a new repository
3. Upload your files with a new commit message (index files uploaded)
4. Give it a minute to upload the files
5. Now navigate to netlify.com and great a "New site from Git"
6. Select the first button at the bottom "GitHub"
7. It'll pop up a window for a sec and it'll be linked to your account on GitHub
8. Select the new repository you created on GitHub in the list that appears
9. There will be a few fields for "basic build settings" since our sites are so simple there aren't any settings. So leave these empty and click the "Deploy site"
10. You'll be directed to this page with this information at the top:



hopeful-sammet-b8ae06

https://hopeful-sammet-b8ae06.netlify.com

Manual deploys. Last update at 9:43 PM.

11. You can click the link in green to view your the site.
12. Now to edit some settings since Netlify generates a random domain name for you.
13. Click on the "Settings" tab
14. Scroll down a bit to the site information section, there you'll click the "Change site name" button.
15. Rename you site to {year}-styleguide-{firstname}{last initial}
16. EXAMPLE: 2019-styleguide-bretta
17. Click "Save"
18. Note: it should say example (2019-styleguide-bretta.netlify.com) You're given a free sub domain through Netlify, that's how these are hosted. Later on if you want to add you own domain you can.
19. Email me (banderson@lcad.edu) the url for the site so I can record this assignment complete.