Absolute Beginner's Guide to Web Development Lab Contents

Absolute Beginner's Guide to Web Development Workshop

Workshop Overview

For this workshop, we are going to use Github to create, edit, and manage HTML, CSS and JavaScript (jQuery) content.

1. Go to github.com
2. Sign in with the following username and password

   username: shonnadorsey402

   password: Pass@word123!

3. Once you are signed in, type the following website address in the address bar and click **enter: tiny.cc/sept-web**
4. Click on the folder with your name on it
5. Click on the file titled index.html

HTML Updates

Edit following content in your index.html document:

1. Change the content attribute on line 7 to your name, change the title tag to a title you prefer

```
1   <!DOCTYPE html>
2   <html>
3     <head>
4       <meta charset="UTF-8">
5       <meta name="description" content="Training Website">
6       <meta name="keywords" content="HTML,CSS,JavaScript">
7       <meta name="author" content="Shonna Dorsey">
8       <meta name="viewport" content="width=device-width, initial-scale=1.0">
9
10      <title>My Page</title>
11
```

2. Change the color on line 19 between the parentheses to a color you prefer:

```
16
17      </head>
18      <body>
19        <p id="gold">This is some text that will appear on the page.
20          <br />
21          This is another line of text below the first.
22        </p>
23
24        <p>This is additional content without the formatting from the first section.</p>
```

3. Follow the instructions on lines 26 – 36 to add a photo to your page on line 37. Update the link content on lines 43 and 45:

```
37      <img src="INSERT IMAGE LINK" alt="text that will appear for broken links, screen readers" />
38      <br />
39      <button>Click Me</button>
40      <br />
41
42      <!--add a couple of hyperlinks-->
43      Click <a href="here">here</a> to access the best website ever.
44        <br />
45      Click <a href="here" target="_blank">here</a> to access the best website in a new window.
```

4. Update the lists starting on lines 48 and 56 to your own content:

```
<h2>Grilled Cheese Ingredients</h2>
<ul>
  <li>Bread</li>
  <li>Cheese</li>
  <li>Bacon</li>
</ul>

<h3>List Title</h2>
<ol>
  <li>Step 1</li>
  <li>Step 2</li>
  <li>Step 3</li>
</ol>
```

CSS Updates

Return to Github.com. Go to the following address: tiny.cc/sept-web. Click on your folder name and click on the css folder. Open style.css and click the edit button to make the following updates:

1. On line 2 is a reference to a massive photo of a breakfast sandwich 😊. Change it to something else using the pexels.com instructions from the previous section:

```
1   body {
2     background: url("https://images.pexels.com/photos/725993/pexels-photo-725993.jpeg");
3     background-repeat: no-repeat;
4     background-size: cover;
5   }
```

2. On line 7 is a reference to the original gold ID in the index.html file. Change it to the color your chose and update the color reference on line 8:

```
7   #gold {
8     color: gold;
9   }
```

3. Check out the image section. Notice the rounded corners (border-radius) and display:none on line 16. We will use the display property on jQuery to display photos on page load:

```
11  /*make the image corners round and resize all images*/
12  img {
13    border-radius: 4em;/*1 em = 16px*/
14    height: 20em; /*force image height to be uniform*/
15    width: auto; /*automatically resize width in proportion to image height*/
16    display:none;
17  }
```

4. On line 25, change the background color to a different color. On line 26, change the box-shadow color to a different color. On line 31, change the text color to a different color:

```
23  a:hover {
24    text-decoration: underline;
25    background: red; /*change to a different color*/
26    box-shadow: 5px 5px 5px green; /*change to a different color*/
27  }
28
29  h3 {
30    color: orange; /*change to a different color*/
31    font-weight: 73em; /*bold text*/
32  }
```

jQuery Updates

Return to Github.com. Go to the following address: tiny.cc/sept-web. Click on your folder name and click on the js folder. Open script.js and click the edit and type the following text on specified lines (**click the enter key after each line**). You are going to use jQuery to make images appear when the page loads and hide images when the user clicks the button.

Line 2: $('img').fadeIn(slow);

Line 3: $('button').click(function() {

Line 4: $('img).hide();

Line 5: });

Absolute Beginner's Guide to Web Development Lab

Lab Overview

During this lab, you will use Github Pages to create a live, working website you can access from any device. By working through the steps in this lab, you will create a website from scratch using the tools and technologies discussed in the introduction to web development workshop.

The page we will be building for this section of the workshop will feature an animated CSS button that when clicked, changes the entire look and feel of the page (we will use jQuery to achieve this outcome).

A live version of the final product is available at the following link: https://tiny.cc/shonnaweb

This project will include static and interactive content. All of content required to achieve the final product is included in this document. You can view the code for this project at: http://tiny.cc/sept-web

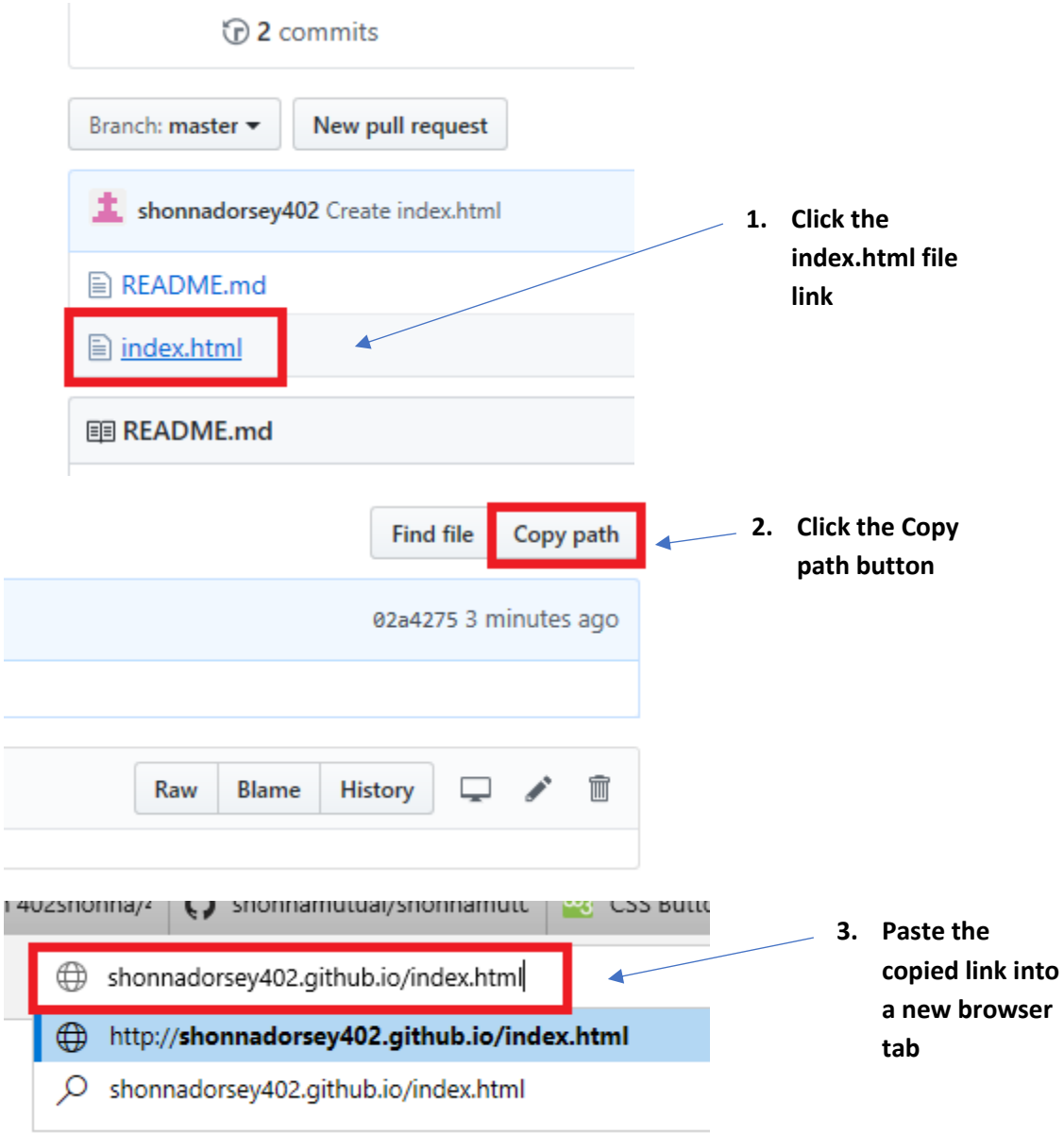
Equipment and Implementation Requirements

- Internet access
- Testing devices (laptop, smartphone)
- Recommended browser - Google Chrome

Let's get started!

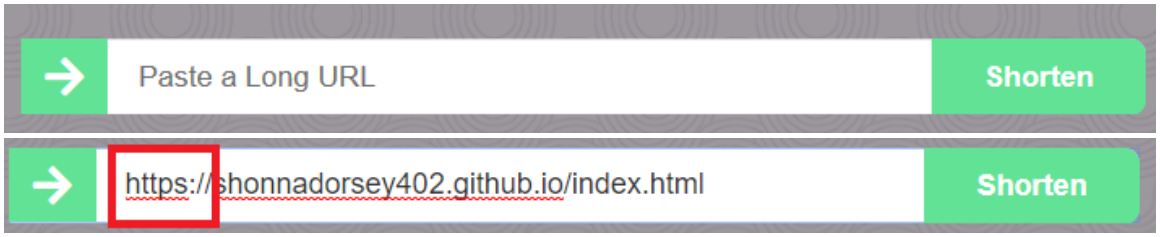
Create a tiny URL for easy site reference

1. On your project dashboard, click the index.html file link, then click **Copy Path** and **paste the copied link into a new browser tab and click enter**:

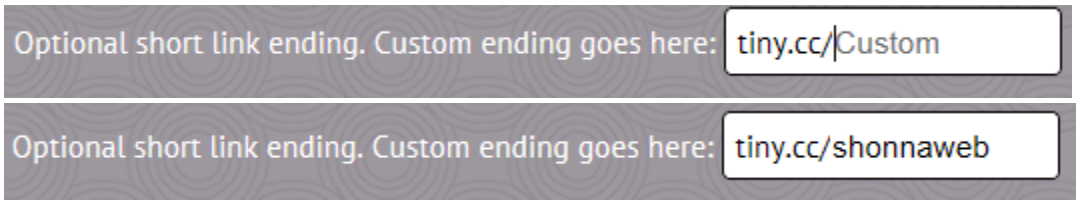


2. Once your page loads, you should see the following – this is a live website:

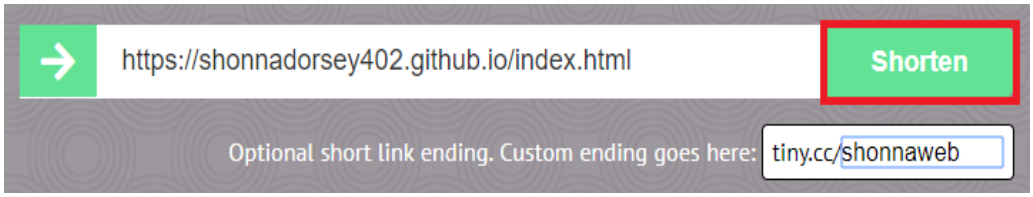
3. In a new tab, type **tiny.cc** and press the enter key.
4. Paste the link you copied from Github into the **Paste a long URL** field. Be sure to add **https://** before your URL if it does not already appear there. You will receive an error message if this prefix is missing.



5. In the custom section of the tiny.cc URL, type a short and easy to remember short name for your new file:



6. After you'd updated the custom URL field, click the **Shorten** button next to the long url field:



7. In a new/existing browser window, type in your new short URL. **In this case, it would be tiny.cc/shonnaweb.**Press enter. You will notice that your page loads with the original URL.

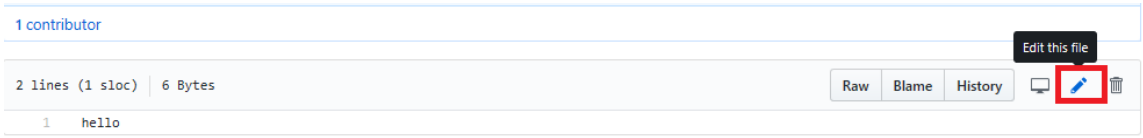
8. **Return to Github – let's build out the content for the site!**

Build Your Home Page

Throughout the rest of this document, the purpose and options to edit content in each of the files you will create is described in detail.

1. Go to the following link: **tiny.cc/sept-web** to view the content for this project. You can use this link as a reference throughout this lab.

2. In a separate tab/window, open your personal index.html file. Click the **edit** button (looks like a pencil) to begin making changes to the file:



Setup the <head> section

3. On the index.html file located at **tiny.cc/sept-web**, copy lines 1 – 10 and paste the content into your index.html file. A description of the content on each line follows this image:

```
1  <!DOCTYPE html>
2  <html>
3     <head>
4        <meta name="viewport" content="width=device-width, initial-scale=1.0">
5        <title>My Page</title>
6        <link href="https://fonts.googleapis.com/css?family=Raleway|Calligraffitti|Homemade+Apple" rel="stylesheet">
7        <link rel='stylesheet' href='css/style.css'>
8        <script src='https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js'></script>
9        <script src='js/script.js'></script>
10    </head>
```

Line 1: <!DOCTYPE html>

This declaration must be the very first thing in your HTML document, before the <html> tag. It is not an HTML tag; it is an instruction to the web browser about what version of HTML the page is written in.

Line 2: <html>

This tag tells the browser that this is an HTML document, represents the root of an HTML document, and is the container for all other HTML elements (except for the <!DOCTYPE> tag).

Line 3 and Line 10: <head> </head>

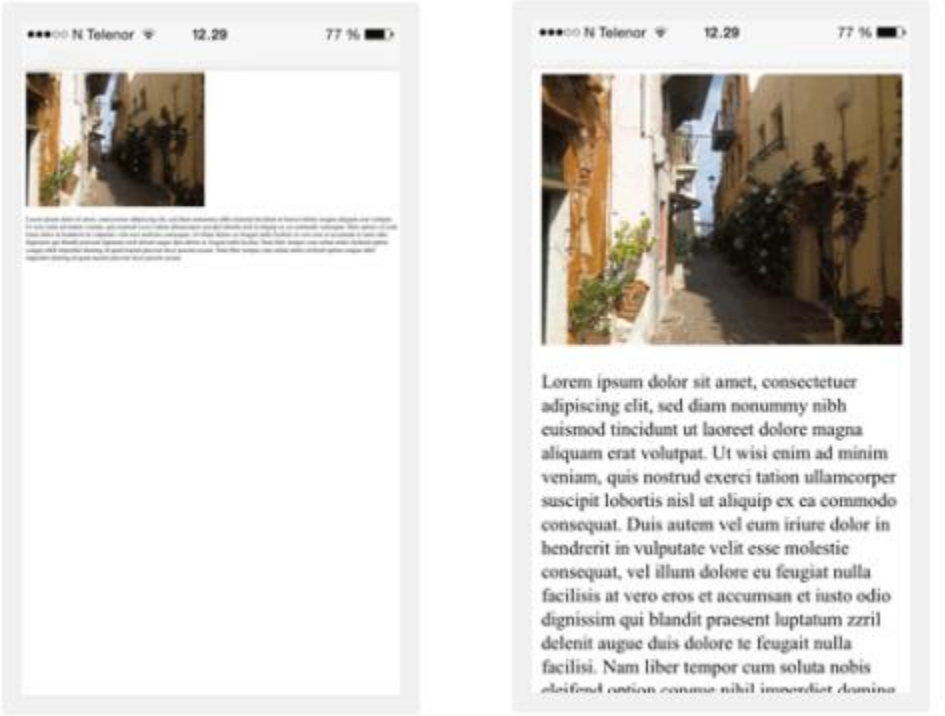
This element is a container for all the head elements and can include a title for the document, scripts, styles, meta information, and more.

Line 4: **<meta name="viewport" content="width=device-width, initial-scale=1.0">**
A <meta> viewport element gives the browser instructions on how to control the page's dimensions and scaling.

The width=device-width part sets the width of the page to follow the screen-width of the device (which will vary depending on the device).

The initial-scale=1.0 part sets the initial zoom level when the page is first loaded by the browser.

Examples from w3schools of a site with and without meta viewport data:



Line 5: <title>Page Title</title>

Defines a title in the browser toolbar, provides a title for the page when it is added to favorites, and displays a title for the page in search-engine results.

Line 6: **<link href="https://fonts.googleapis.com/css?family=Raleway|Calligraffitti|Homemade+Apple" rel="stylesheet">**
Reference to Google Fonts an open source library of customizable fonts for your web projects. You can view other font options at fonts.google.com.

Line 7: `<link rel='stylesheet' href='css/style.css'>`

Reference to the custom stylesheet we will use to modify the look and feel of content in our HTML document.

Line 8: `<script src='https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js'></script>`

This is a reference to a jQuery library. A reference to the library is required before a custom JS file reference. Remember, you cannot checkout a book without first going into the library.

Line 9: `<script src='js/script.js'></script>`

Reference to the custom JavaScript file we will create to make our site interactive. Notice that it is in a directory called js. It is important to organize your files in a way that is easy to manage.

Setup the <body> and <container>

As mentioned during the walkthrough, the <body> section of your page is where all content that will appear on the page belongs. In this section, you will add text, hyperlinks, images, buttons, tables, lists, etc.

```
11      <body>
12          <div class="container">
13              <header>
14                  <h2>I love fall!</h2>
15              </header>
16                  <hr />
```

Line 11: <body>

This is the beginning of the section where visible content will appear on your site.

Line 12: `<div class="container">`

The container will include all visible content within the layout we develop for this site.

Line 13 – 15: Header section

Content in this part of the site will appear at the top of the layout

Line 16: <hr />

HR = horizontal row and is a self closing element. It is a single line separating the header from the content below it.

Setup <section> within <container>

```
18            <section>
19              <nav>
20                 <h3>Fall Activities and Images</h3>
21                <ul>
22                  <li><a href="https://www.omaha.com/calendar//search/?l=25&unrolled=1&s=start_ti
23                  <li><a href="https://www.pexels.com/search/fall%20nature/">Beautiful Fall Photo
24                  <li><a href="http://www.midwestliving.com/food/holiday/25-fabulous-fall-recipes
25                </ul>
26              </nav>
```

Line 18: <section>

Semantic tag to describe the content included in the main section of the page.

Lines 19-26: <nav>…</nav>

Semantic tag used to describe the navigation section of the page.

Line 20: <h3>

Special header tag to bring attention to this section of the page.

**Lines 21 – 25: <ul> … **

Unordered list with hyperlinks to external content.

Lines 22 – 24: <li> and <a>

Each <li> describes a single line in this list.

Each <a> references an external link. Instructions on how to add a hyperlink follow:

Format: Word/words you want user to click on

Example: Best Search Engine!

Include target=”_blank” after your URL if you want the user to go to the link in a new tab. Very useful feature for external content.

New Window/Tab Example:

Best Search Engine!

Setup <article> within <section>

```
28              <article>
29                <h1>Fall is the best</h1>
30                <p>Not only is the weather better, but the fashion is the best! Layers anyone?</p>
31                <p>Click the button below for a beautfiul fall background!</p>
32                 <button id="animated">Fall is near!!</button>
33              </article>
34            </section>
```

Lines 28 – 33: <article>…</article>
Content within this section will appear to the right of the <nav> section on lines 19 – 26.

Line 29: <h1>
Largest default heading in HTML.

**Lines 30-31: ... **
Two lines of paragraph content. Please add as much content as you'd like to here. Only use a closing tag when you have finished adding content.

Line 32: <button>
This button will be customized using CSS. Please update the button text to your own text.
Current (the highlighted section can be customized):
<button id="animated">**Fall is near!!**</button>

Line 34: </section>
End of the main content on the page

Setup <footer> within <container>, close the document, commit changes, view initial results

```
36              <footer>
37                <p id="footer-text">Shonna Dorsey
38                 <br/>
39                 <a href="mailto:shonna.dorsey@gmail.com">shonna.dorsey@gmail.com</a>
40                 <br/>
41                   <a href="tel:14023511536">402.351.1536</a></p>
42              </footer>
43            </div>
44        </body>
45    </html>
```

Lines: 36 – 42: <footer>…</footer>
Content that will appear at the bottom of the page.

Line 37: <p id="footer-text">
Custom ID that will be used in CSS to customize this section of the page

**Line 38:
**
Self closing line break element. Used instead of <p> to create a single line break vs. double line break.

Line 39: `<a href="mailto:shonna.dorsey@gmail.com">shonna.dorsey@gmail.com</a>`
Using an href attribute with mailto:email address results in the default email client opening with the email address that appears after mailto in the 'to' field.

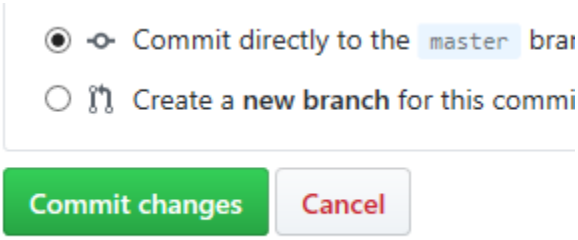
Line 41: `<a href="tel:14023511536">402.351.1536</a></p>`
Using an href attribute with tel:phone number results in the default phone app opening with the phone number in the phone number field. Try this on your smartphone after you save the content.

Line 42: </footer>
Closing footer tag.

**Line 43: **
Closing div for the container class on line 12

Line 44: </body>
Closing body tag for all content on the page

Line 45: </html>
Closing html tag signifies the end of the document
Click the commit changes button at the bottom of the page once you are ready to submit your index.html file.



Go to your page (tiny.cc/**NameYouChose**) to view the current results. It may take a minute, but once the page content is loaded, you should see something like this:

I love fall!

Fall Activities and Images

- Fall Activities in Omaha
- Beautiful Fall Photos
- Fall Recipes

Fall is the best

Not only is the weather better, but the fashion is the best! Layers anyone?

Click the button below for a beautfiul fall background!

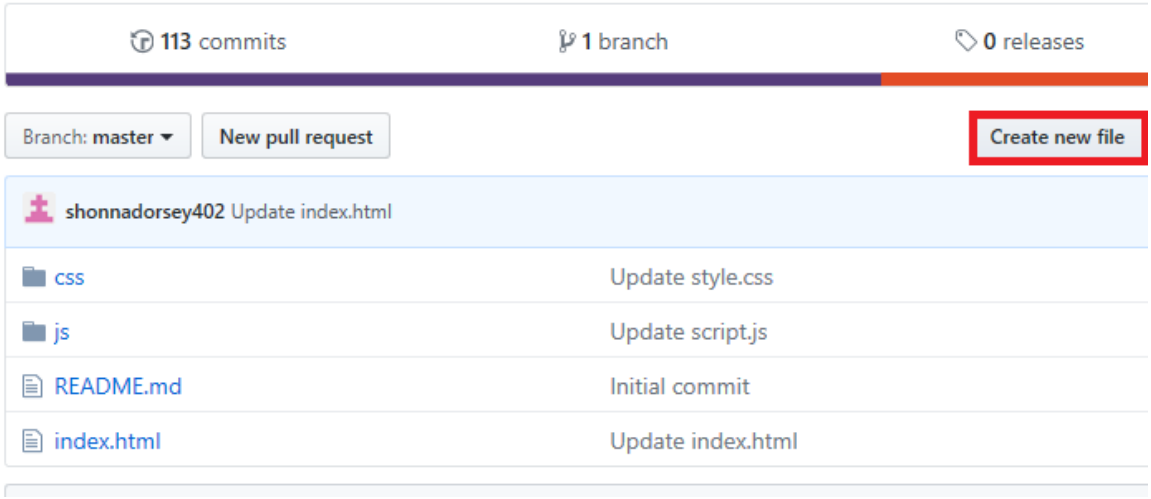
Fall is near!!

Shonna Dorsey
shonna.dorsey@gmail.com
402.351.1536

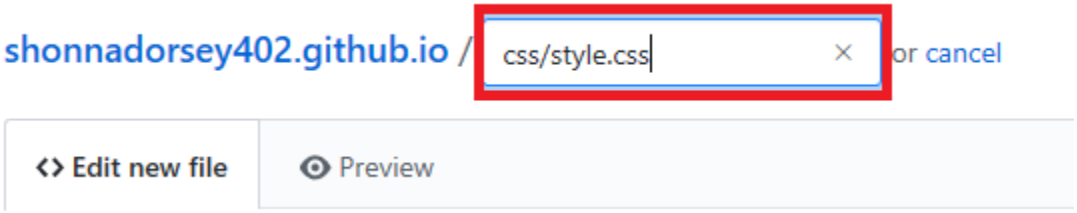
Not very exciting yet! We are going to change that with some CSS.

Create your CSS file

1. Go back to the root of your github repository for this project. (You should see your index.html file and your README.MD file). Click the **Create New File** button:



2. Once you have clicked the **Create New File** button, you are going to create a CSS folder with a file called style.css (all lowercased). To create a folder, simply type the folder name followed by an "/" then add the name of the file you'd like to add (see example below).



3. Click into the editor. Let's start styling!

```
/*toggleClass content for jQuery */
.fall {
    background-image: url("https://images.pexels.com/photos/33109/fall-autumn-red-season.jpg");
}
.text {
    color: #fff;
    background: #6B0200;
}
.content {
    background-image: url("https://images.pexels.com/photos/589840/pexels-photo-589840.jpeg");
    color: #fff;
}
.navigation {
    background:#6B0200;
    color: white;
}

.links {
        color: #fff;
}
```

Create toggleClass() references for jQuery

Line 1: `/*`toggleClass content for jQuery `*/`

This is an example of a comment. Comments in CSS start with /* and end with */. All content within a comment will only be seen by the developer or when a user clicks on your page source. Use these throughout your CSS document whenever you want to leave notes for yourself/anyone else who will be reviewing your content.

Line 2 – 4: Fall Class

This content will be used by jQuery to change the background of the page when the user clicks on the button. The format of the background image property in CSS follows. You can use any image you'd like. Pexel.com is recommended and includes thousands of open source photos for the purpose of this project.

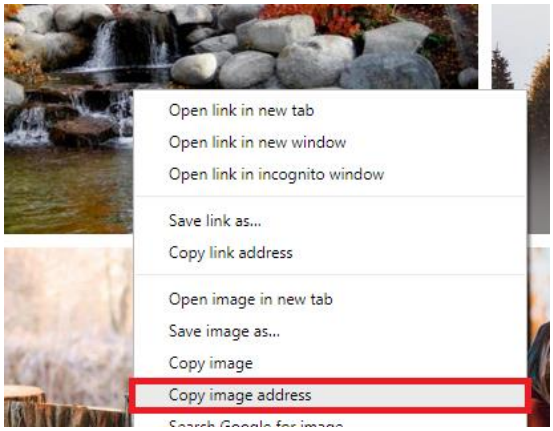
Format:

background-image: url('picture link');

Example:

```
background-image: url("https://images.pexels.com/photos/33109/fall-autumn-red-season.jpg");
```

On pexels.com, search for an image then right click on any image you'd like to use then click on **copy image address**:



Go back to your CSS document and paste the address in the field – **important –** be sure to remove all content after the ? (please also delete the question mark) in the URL.

Example:

Original Image Link Text:

https://images.pexels.com/photos/230629/pexels-photo-230629.jpeg?auto=compress&cs=tinysrgb&h=350

Updated Image Link Text:

https://images.pexels.com/photos/230629/pexels-photo-230629.jpeg

Correct format for CSS background-image property:

background-image: url('https://images.pexels.com/photos/230629/pexels-photo-230629.jpeg')

Lines 5 – 8: .text

The **.text** class includes an updated font color (color:) and background color (background:) for header and footer. Feel free to provide select a different font color and background color.

Lines 9 – 12: .content

The **.content** class includes an updated font color (color:) and background image (background:) for article section of index.html. Feel free to select a different font color and background image.

Lines 13 – 16: .navigation

The **.navigation** class defines an updated font color (color:) and background color (background:) for <nav> section of index.html. Feel free to select a different font color and background color.

Lines 18 – 20: .links

The **.links** class defines an updated font color (color:) for hyperlinks in index.html. Feel free to select a different font color for links.

Container Class Modifications and Intro to Flexbox for Responsive Design

```
22  .container {
23      /*top right bottom left*/
24      padding: 5% 3% 5% 3%;
25      font-family: 'Raleway';
26  }
27
28  header {
29      font-family: 'Homemade Apple', cursive;
30      text-align: center;
31      font-size: 35px;
32      color: #333;
33      padding: 30px 30px 5px 30px;
34  }
35
36  /*flexbox content required for media queries*/
37  section {
38      display: -webkit-flex;
39      display: flex;
40  }
41
```

Lines 22-26: .container

Padding is added to the container class so that the background image will be visible when a user clicks the button on your page. Font family is set to the Google Font Raleway. You can use any Google Font you list in the <head> section of your HTML document. If you did not add fonts here, you can simply leave it as is.

Lines 28 – 34: header

The header section defines the font, padding, alignment, text size and color for the header section. You may update the **font-family** and **color** properties for the header selector.

Lines 36 – 40: section (the comment on line 36 is optional)

This is an important segment as it will determine how your page appears on different screen sizes by utilizing flexbox. Float and position were used in the past, but are difficult to control. Flexbox (Flexible Box Layout Model) is an updated approach to manage page flow on different screen sizes.

Navigation Section Modifications

```
42  /* Style the navigation menu */
43  nav {
44      -webkit-flex: 20%;
45      -ms-flex: 20%;
46      flex: 20%;
47      background: #ccc;
48      padding: 10px;
49      height: 1000px;
50      overflow-y: hidden;
51  }
52
53  /* Style the list inside the menu */
54  nav ul {
55      list-style-type: none;
56      padding: 0;
57  }
58
59  a {
60      text-decoration: none;
61  }
62
63  a:hover {
64      text-decoration: underline;
65  }
66
67  /* Style the content */
68  article {
69      -webkit-flex: 80%;
70      -ms-flex: 80%;
71      flex: 80%;
72      background-color: #f1f1f1;
73      padding: 10px;
74  }
```

Lines 43 – 51: nav

Describes the left navigation section of the page. Recommend keeping all content as it is. You may consider increasing/reducing the size of the flexbox (currently set to 20% of the container width – lines 44-46)

Lines 54 – 57: nav ul

This selector targets any <ul> tags under the <nav> tag. A <ul> outside of <nav> would not be impacted by this selector. **list-style-type: none;** indicates that this list will not have a bullet (<ul>) or a number (<ol>).

Lines 59 – 65: a, a:hover

The first selector (a{) includes text-decoration: none; which means, do not underline. However, with **a:hover** we reintroduce the underline, but it will only appear when the user hovers over a hyperlink

Lines 68 – 74: article

This section defines the size (80%), the background color and padding around the content.

Media Queries to Customize Content Based on Screen Size

```
85    @media (max-width: 600px) {
86        section {
87          -webkit-flex-direction: column;
88          flex-direction: column;
89        }
90
91        nav{
92            height: 100%;
93        }
94
95        #animated {
96            animate: 0;
97        }
98
99        hr {
100           display: none;
101       }
102   }
```

Lines 85: @media (max-width: 600px)
This is a media query and will include a different set of properties and values for a variety of selectors for screen sizes that are 600px wide and smaller.

Lines 86 -89: section
Sets the sections to be stacked vs. the original layout.

Lines 91 – 93: nav
Adjusts the nav height from 1000px to 100% which means that it will be tall enough to accommodate its content.

Lines 99 – 101: hr
display: none; is the same as **hide**

Line 102: }
Important: remember to close the media query!

Animated Button Setup

```
104   button:hover {
105       transition: box-shadow 1s ease-in-out;
106       box-shadow: 5px 5px 5px grey;
107   }
108
109   #animated {
110   font-family: 'Raleway';
111   width:200px;
112   padding: 5px;
113   background: #6B0200;
114   color: #fff;
115   position: relative;
116   font-weight:bold;
117   font-size:20px;
118   padding:10px;
119   animation:animated 5s 1;
120   border-radius:5px;
121   -webkit-border-radius:5px;
122   }
```

Lines 104 – 107: button:hover

This section adds a shadow behind the button when a user hovers over the button. Feel free to change the color of the hover on line 106 (**currently grey)**

Lines 109 – 122: #animated

This section includes styling for the #animated button on the page. Feel free to change:

- **Line 110: font-family** (remember: if you are using a different Google Font, it has to be listed in the head section of your index.html document)
- **Line 111: width -** you may need to adjust the width of the button if your button text is much longer or shorter than “Fall is near!!” as provided in the example.
- **Line 113: background –** this is the color of the button. The button can be any color you choose
- **Line 114: color** - this is the text color of the button. The button text can be any color you choose
- **Line 117: font-size -** the font can be any size you choose.
- **Line 119: animation** – **5s** is the duration of the animation described in the next section. You can increase/decrease the length of the animation. **1** is the number of times the animation will run. Change to **infinite** to see what happens.

Finally – the most fun! **Button animation!**

```
124   @keyframes animated
125   {
126   0% {transform: rotate(0deg);left:0px;}
127   25% {transform: rotate(20deg);left:0px;}
128   50% {transform: rotate(0deg);left:500px;}
129   55% {transform: rotate(0deg);left:500px;}
130   70% {transform: rotate(0deg);left:500px;background:#C95101; color:#fff;}
131   100% {transform: rotate(-720deg);left:0px;}
132   }
```

Lines 124: @keyframes animated

The @**keyframes CSS** at-rule controls the intermediate steps in a **CSS** animation sequence by defining styles for **keyframes** (or waypoints) along the animation sequence (source: http://tiny.cc/mozilla-keyframes). For this animation, we are targeting the animated button in index.html which is also called **animated**.

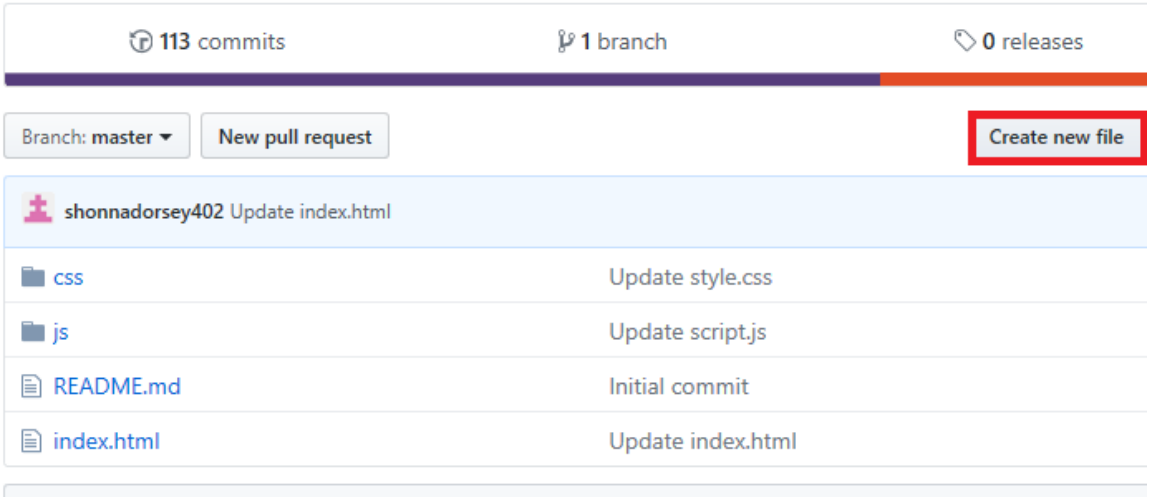
Lines 125 – 132: Animation

- Line 126: starts the animation
- Line 127: at the 25% point of the animation, rotate the button by 20 degrees
- Line 128: at the 50% point of the animation, move the button 500px to the right (add 500px of space to the left) and rotate it back to 0 degrees
- Line 129: at the 55% point of the animation, move the button 500px to the right (add 500px of space to the left)
- Line 130: at the 70% point of the animation, move the button 500px to the right (add 500px of space to the left), change the background color (you can update the color here), and change the text color (you can change the text color here)
- Line 131: at the 100% point of the animation, rotate the button by -720 degrees and move it back to its original position. (you may update the rotation by any multiple of 360 – be sure to include a negative sign in front of your rotation.

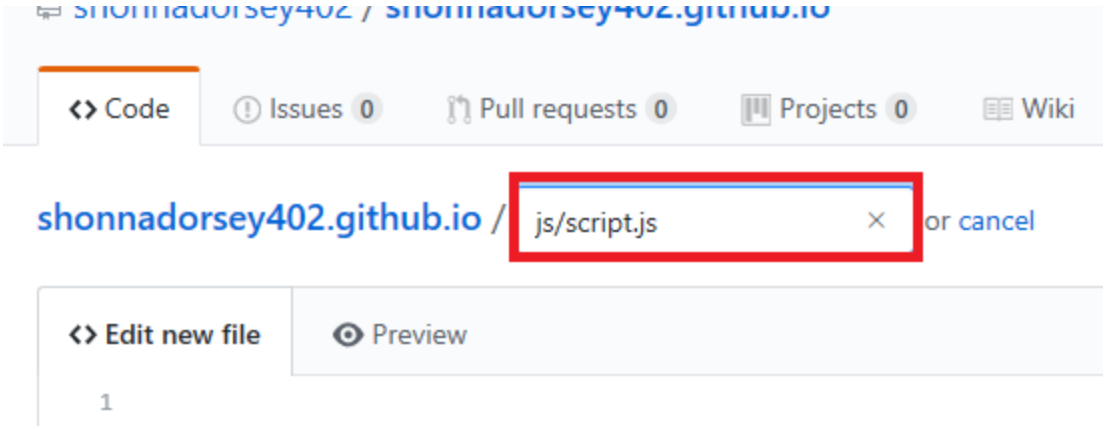
That's it for CSS! Commit your file. Reload your page at tiny.cc/NameYouChose to see the completed version. Refer to tiny.cc/sept-web to view the code if you run into issues.

Add jQuery to your site (final step!)

1. Go back to the root of your github repository for this project. (You should see your index.html file, your css folder and your README.MD file). Click the Create New File link:



2. Once you have clicked the create new file button, you are going to create a js folder with a file called script.js (all lowercased). To create a folder, simply type the folder name followed by an "/" then add the name of the file you'd like to add (see example below).



3. Add content to your script.js file

```
1   $(document).ready(function(){
2           $("button").click(function(){
3                   $("body").toggleClass("fall");
4                   $("header, footer").toggleClass("text");
5                   $("nav").toggleClass("navigation");
6                   $("a").toggleClass("links");
7                   $("article").toggleClass("content");
8                   $("#update").toggleClass("hide");
9       });
10  });
```

Line 1: $(document).ready(function(){

Ensures the document it is attached to has loaded before any of the functions kick off. In this case, we do not have any functions that kick off on page load.

Line 2: $("button").click(function(){

Once the button is clicked, do everything that is contained within the click function (lines 3 – 8). Important: do not miss any semicolons, opening/closing parentheses or opening/closing braces. Your code may not work if any of the symbols described are not present.

Lines 3 -8: toggleClass method

The toggleClass method allows us target elements and add/remove CSS classes. In this case we are transforming the look and feel of the body, header, footer, navigation bar, article background, and hyperlinks.

Lines 9 – 10: close your functions });

Each function needs to be closed. In this case, we have two functions, so you should see)}; twice at the end of the script.js document.

That's it for this project! Commit your file. Reload your page at tiny.cc/NameYouChose to see the completed version. Refer to tiny.cc/sept-web to view the code if you run into issues.

Additional Resources

Free online coding courses:

https://www.codecademy.com/ - Codecademy – free online web development tutorials

https://learn.freecodecamp.org/ - FreeCodeCamp – learn to code for free. Opportunities to help nonprofits.

Code Snippets

https://codepen.io/ - creative uses of jQuery, JavaScript, CSS

https://css-tricks.com/ - tips and tricks for CSS. Great help forum.

https://www.w3schools.com/ - excellent resource for front end development

Code Editors

https://atom.io/ - Developed by Github.

http://brackets.io/ - lightweight, powerful