

Exercise 3: screen-to-semantic-to-screen

Existing website: <https://fablepets.com/>

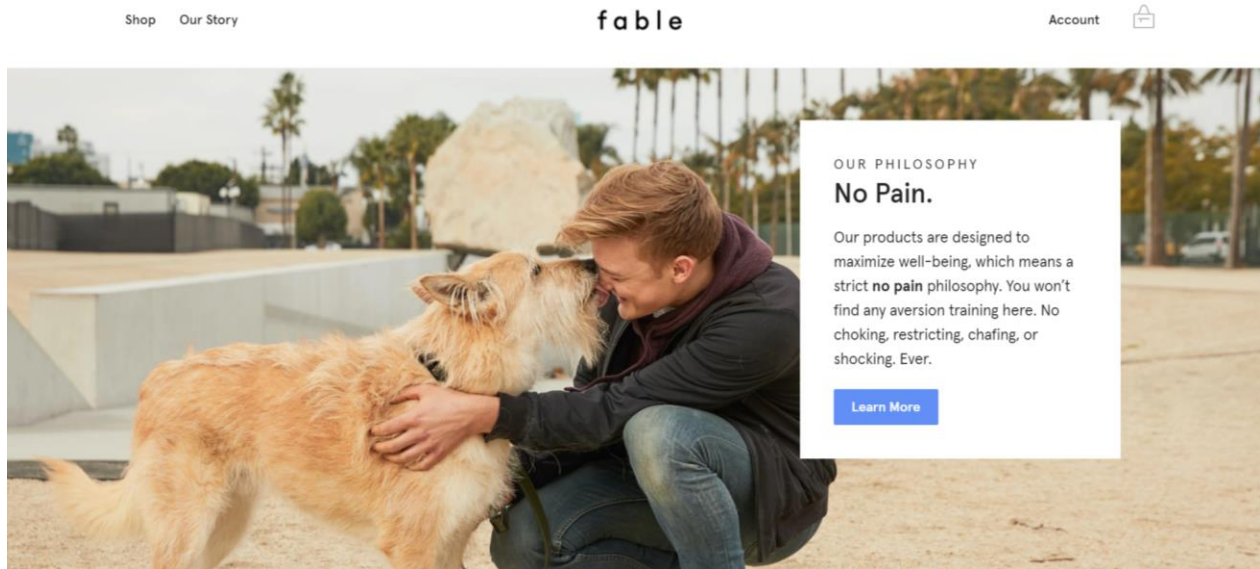


Table of Contents

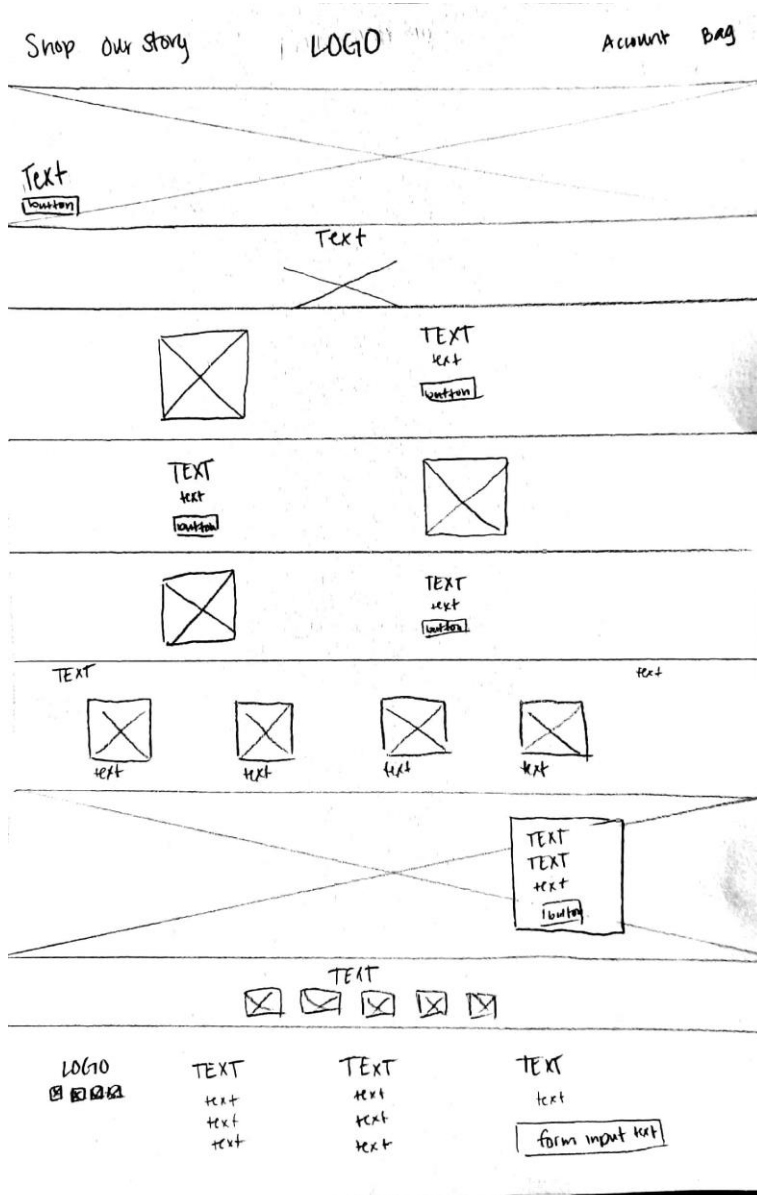
Project Goals	1
Semantic Markup	2
Responsiveness	3
Elements & Selectors (per section)	
Reflection	11

Project Goals

- Understand **semantic** markup
- **Recreate** the home page of a website using HTML and CSS
- Create **responsiveness** for two screens

Semantic Markup

This is a sketch of the webpage from top to bottom, including the entire container semantic name and other semantics within the container broken down.



CS Scanned with CamScanner

Header

- Logo is h1
- Other elements are wrapped in nav li

Section

- Text is h1

Section

- Text is h3

3 Sections

- Images are wrapped in <figure>
- Text & button wrapped in <article>
- Title text is h1
- Other text is p

Section

Aside

- Text is h1 or p

Main

- Each image & text is wrapped in <main>
- text is p

Section

Figcaption

- h2, h1, p

Aside

- text is h3

Footer

- Logo wrapped in h2
- Images wrapped in <nav>
- Each column wrapped in nav with h3 and li elements

Aside

- Fourth column is aside with h3, p, and form

Responsiveness

This is a breakdown of each section of the website. The image shows the *responsiveness* of the website at two different widths (desktop and mobile), and the caption talks about each element and selector. The code is in black, the explanation of HTML element in red, and the explanation of the CSS selectors in blue. Responsiveness is shown in the image.

Header



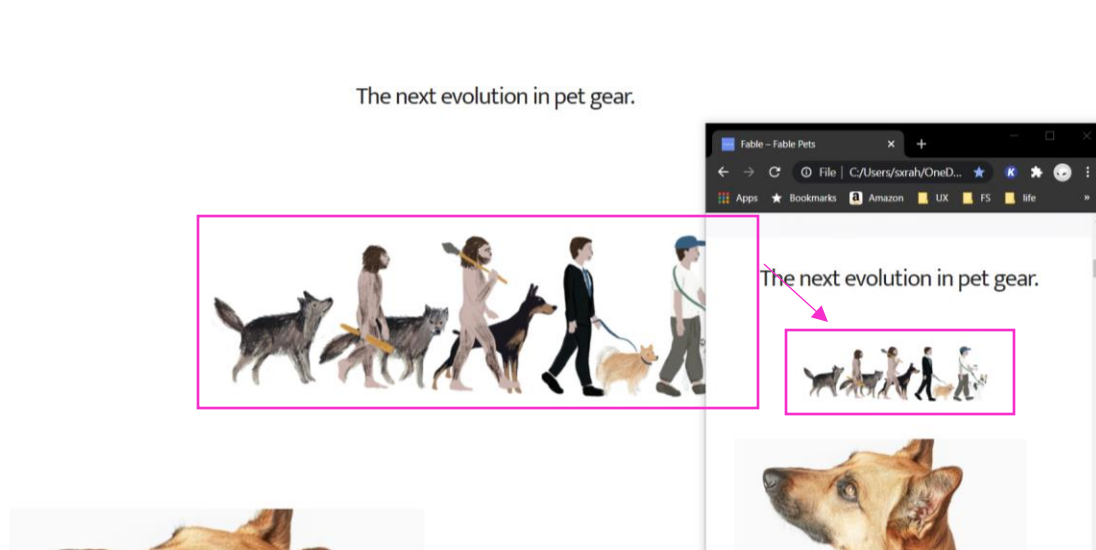
Code	HTML element	CSS selector
<code><header class="container"></code>	Header is top of page, includes nav bar.	"container" is main centered wrapper in skeleton
<code><nav class="row"></code>	Navigation bar inside header.	"row" keeps all elements within the nav bar in a single row.
<code><ul class="nav_bar"> <li class="four columns"></code>	ul within navigation to group similar elements (all navigation <a> elements)	"nav_bar" defines entire nav bar with padding, flex box, and margins. "four columns" creates responsiveness by making them narrow when the screen gets smaller.
<code></code>	a element creates links within each ul.	"nav_elements" defines color, font, etc. of links within the nav bar. "nav_left" & "nav_right" defines float of each column within the nav bar.
<code><h1></h1></code>	H1 shows hierarchy of logo (most importance)	N/A

Section 1



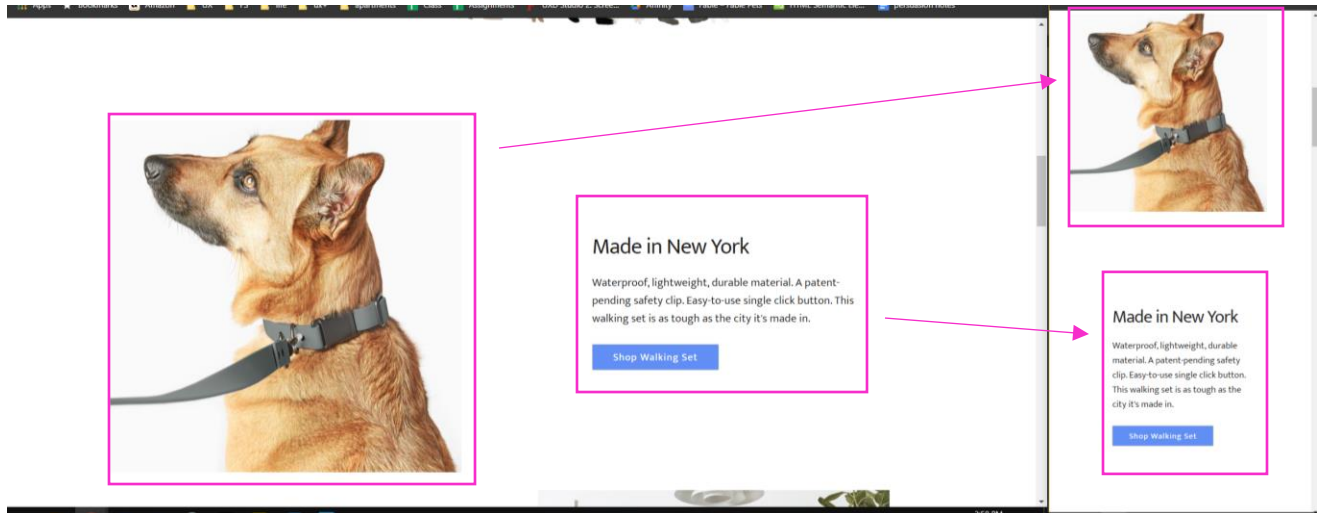
Code	HTML element	CSS selector
<pre><section class="background_image"> <div class="container"> <div class="row"></pre>	<p>Section defines image + text. Div used to apply CSS selectors.</p>	<p>“background_image” applies background image and size. “container” is main centered wrapper in skeleton “row” keeps elements in a single row.</p>
<pre><figcaption id="nobackground"></pre>	<p>Figcaption is used to define text on top of an image.</p>	<p>“nobackground” removes inherited background image and defines alignment.</p>
<pre><h1 class="six columns img_text bkg_img_text"></pre>	<p>H1 to show hierarchy of text.</p>	<p>“six columns” makes text take up half the screen. “bkg_img_text” defines top-margin of text.</p>
<pre><button></pre>	<p>The link is a button, which has preset design in skeleton.</p>	<p>N/A</p>

Section 2



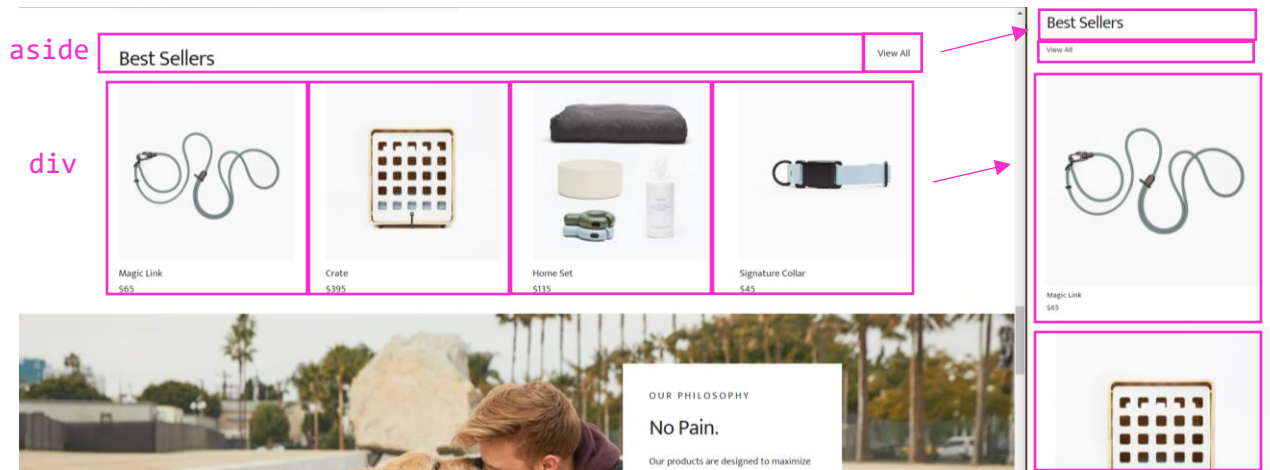
Code	HTML element	CSS selector
<code><aside class="container"></code>	Aside is an element that has content only indirectly related.	“container” is the main centered wrapper in skeleton
<code><div class="row evolution_image"></code>	Figcaption is used to define text on top of an image. Div used in order to apply css selectors to a more specific section within the aside.	“row” keeps all elements within the same row. “evolution_image” puts image and header into flexbox, and aligns content to horizontal center.
<code><h3 class="margin_top"></code>	H3 to show hierarchy (less than other titles)	“margin_top” applies space between this section and the section above it. Used throughout website.

Section 3,4,5



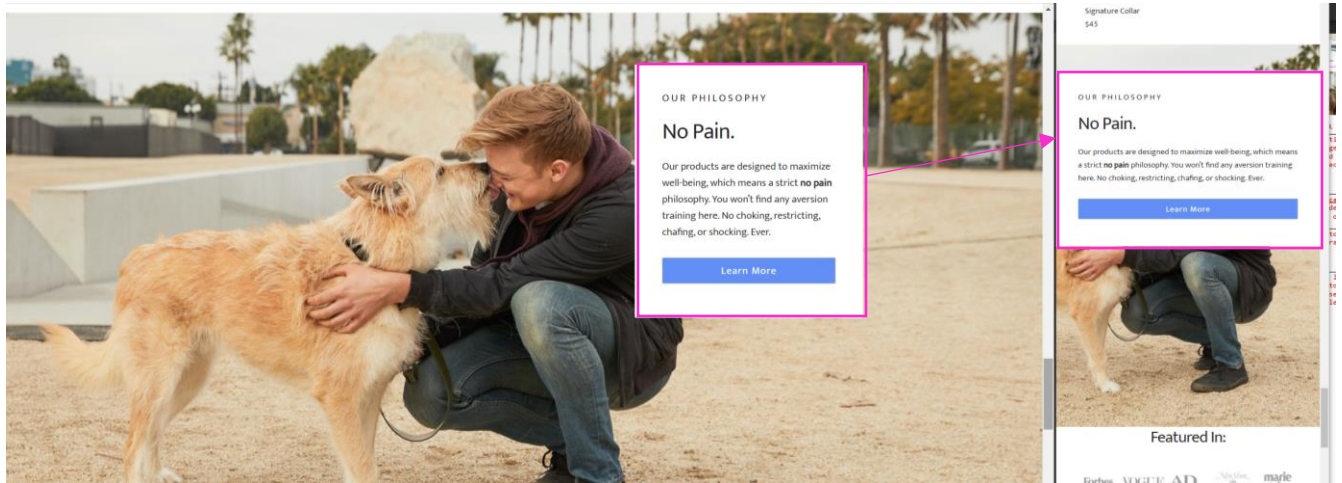
Code	HTML element	CSS selector
<pre><section class="container"> <div class="row"></pre>	Section defines image + text. Div used to apply CSS selectors.	“container” is main centered wrapper in skeleton “row” keeps elements in a single row.
<pre><figure class= "six columns"> <article class= "six columns"></pre>	Figure defines image within main section of document, and article is used for the text that accompanies it.	“six columns” creates responsiveness by making them stack when the screen gets smaller.
<pre></pre>	Img is used to insert an image.	“two_col_img” is used to make image get a little smaller in order to maintain responsiveness on mobile.
<pre><h1> <p> <button></pre>	H1 to show hierarchy of text, p used for main content, and button used to create the same style of button as used through entire website.	N/A

Collection



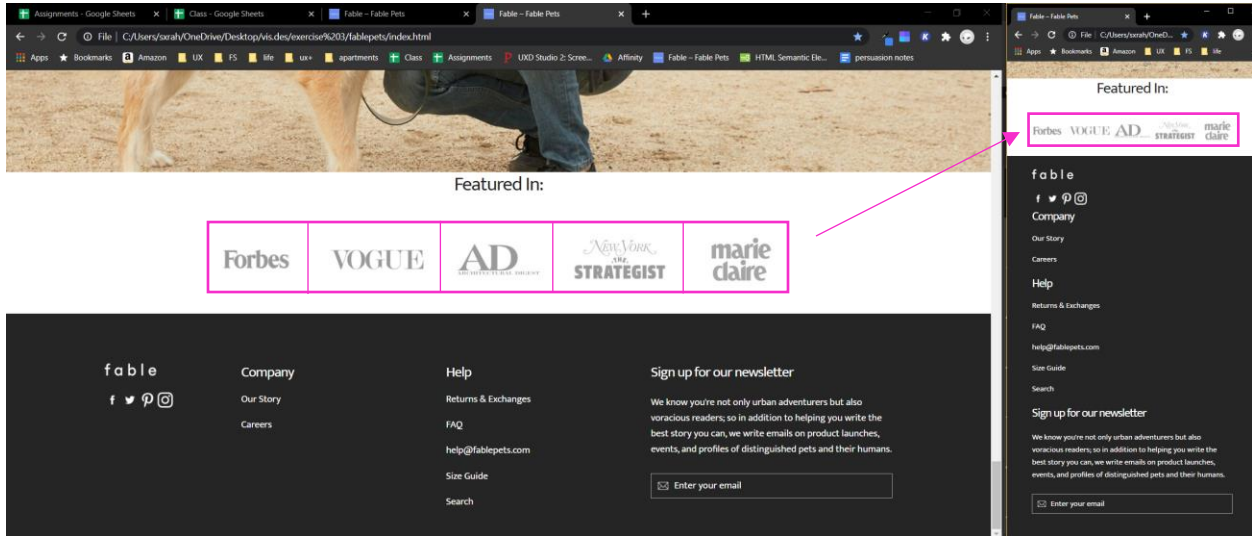
Code	HTML element	CSS selector
<pre><section class="container"> <aside class="row"> <div class="row"></pre>	<p>Section defines images + text. Aside is text only. Div used to apply CSS selectors to all main elements.</p>	<p>“container” is main centered wrapper in skeleton “row” keeps elements in a single row.</p>
<pre><h1 class="eleven columns"> <p class="one column"></pre>	<p>H1 and p used to show hierarchy.</p>	<p>“eleven columns” and “one column” creates responsiveness by making them stack when the screen gets smaller.</p>
<pre><main class="three columns" id="nomargin"></pre>	<p>Main is used to show that this is main content. Each image & text is wrapped in main.</p>	<p>“three columns” creates responsiveness by stacking each of these main elements on top of each other when screen is smaller. “no margin” removes margin, which is used throughout the website.</p>
<pre></pre>	<p>Img used to insert image.</p>	<p>“u-max-full-width” is used to make the image take up the entirety of the column.</p>

Figcaption



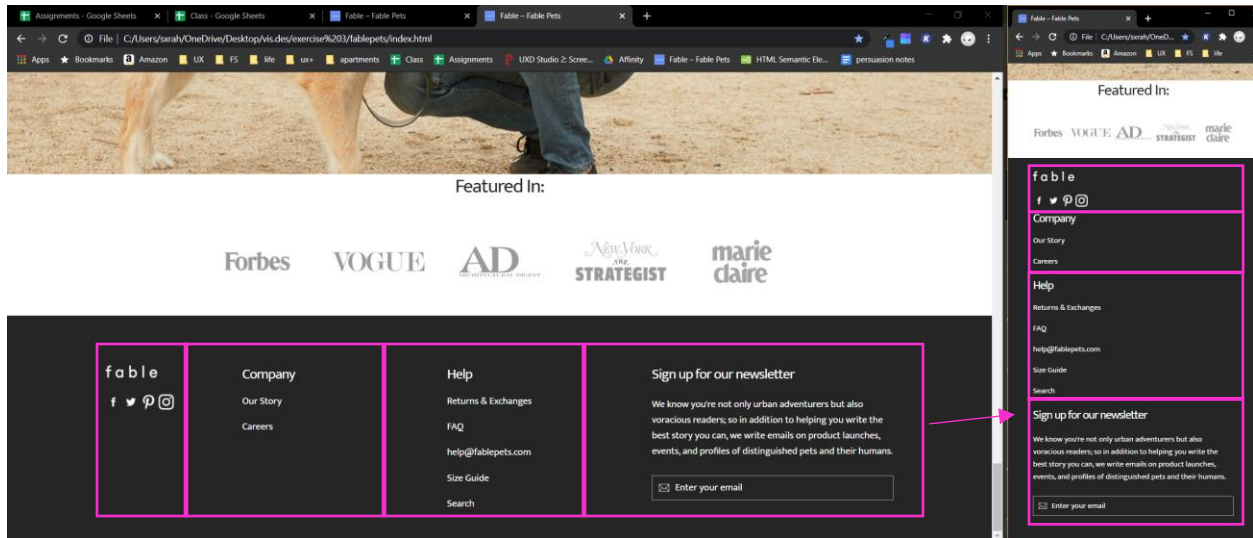
Code	HTML element	CSS selector
<pre><section class="background_image container" id="two"> <div class="row"></pre>	<p>Section defines image + text. Div used to apply CSS selectors.</p>	<p>“background_image” applies background image and size. “container” is main centered wrapper in skeleton. “two” specifies which picture to use for background image. “row” keeps elements in a single row.</p>
<pre><div class="seven columns">&nbsp; </div> <figcaption class="three columns dumbcap"></pre>	<p>Div used to apply CSS selectors. Figcaption is used to define text on top of an image.</p>	<p>“seven columns” and “three columns” creates responsiveness by making them stack when the screen gets smaller. (the first element is a “faked” element.) “dumbcap” sets top-margin (Used only for this section.)</p>
<pre><div class="caption"></pre>	<p>Div used to apply CSS selectors.</p>	<p>“Caption” is used for entire white section. This ensures box will be responsive, content is centered, and text is right aligned, and sets margins.</p>
<pre><h2> <h1> <p> <button></pre>	<p>H1 and H2 used to show hierarchy of text, p used for main content, and button used to create the same style of button as used through entire website.</p>	<p>N/A</p>

Aside



Code	HTML element	CSS selector
<pre><aside class="container"> <div class="row evolution_image"></pre>	<p>Aside is an element that has content only indirectly related. Div used to apply CSS selectors.</p>	<p>“container” is the main centered wrapper in skeleton. “row” keeps all elements within the same row. “evolution_image” puts (only h3) into a flexbox, and aligns content to horizontal center.</p>
<pre><h3 class="margin_top"></pre>	<p>H3 is used to show hierarchy (and maintain consistency)</p>	<p>“margin_top” applies space between this section and the section above it. Used throughout website.</p>
<pre><div class="row dumbstuffatbottom"</pre>	<p>H1 to show hierarchy of text.</p>	<p>“row” keeps all elements within the same row and also defines a new row within the same <aside> tag. “dumbstuffatbottom” places all images into a flex box and centers items horizontally. Useful to maintain responsiveness when screen is shrunk to mobile.</p>
<pre><div class="five columns alpha opacity"></pre>	<p>Div used to apply CSS selectors.</p>	<p>“five columns alpha” makes each image take up one fifth of the row. “opacity” lowers the opacity of the images.</p>

Footer



Code	HTML element	CSS selector
<pre><footer class="container"> <div class="row padding"></pre>	<p>Footer is bottom of page, includes nav bars. Div used to apply CSS selectors.</p>	<p>“container” is main centered wrapper in skeleton. “row” keeps all elements within the nav bar in a single row. “padding” used to apply padding within the footer.</p>
<pre><nav class="two columns"> <nav class="three columns"> <nav class="three columns"> <nav class="four columns"></pre>	<p>Navigation bars inside footer.</p>	<p>“# columns” creates responsiveness by stacking each of these nav elements on top of each other when screen is smaller.</p>
<pre><h3> </pre>	<p>H3 used to show hierarchy and maintain consistency. ul within navigation to group similar elements (all navigation <a> elements)</p>	<p>N/A</p>
<pre><form> <input class="u_full-width"></pre>	<p>Email input form type text.</p>	<p>“u-full-width” ensures form takes up entire four columns.</p>

Reflection

This was not a task I would say I enjoyed completing. I believe I like to **code** but recreating a website that already exists was too difficult for me to achieve. I wanted it to be **perfect, and there was simply no way that was possible** with my current skill set and this website. I also found myself struggling with the skeleton framework more than I found it helping me. I also think I struggled to **understand the point of semantics realistically**- I have many friends in CS and a brother who works in Web Dev, and they are all of the opinion that semantically correct code is just not used and not necessary. Therefore I **cannot talk to these types of people about semantics in HTML, which effectively minimizes its usefulness to me**, as I'd love to be an intersection point between those who need websites and those who can build websites. Perhaps it should be, and that is why there was a focus on semantics. I will say that for me, using semantically accurate tags made it **more difficult** than simply using `<div>`s with CSS classes. However, **I do hope that am wrong about the usefulness of semantics** and someday I will be proven wrong and thank Colin for making me complete this exercise.

Learning about website **responsiveness** was extremely helpful, however! I think I will definitely utilize responsiveness in the future, however, I think skeleton made it more complicated for me than I think it should have. I would be really interested to learn how the code in skeleton works, so that I can use the "columns" as a CSS selector idea in the future. The other built in aspects of skeletons seemed to be constantly causing me problems, and **I would have enjoyed starting from scratch to create a responsive website.**