**Design Challenge: Setting Goals for Your Site**

This is an exercise to get you thinking critically about web design, as well as prepare you for the project in this unit.

Find two websites whose layouts and designs you admire, that share similar goals to the business website you began building in Unit 1.

**Step 1**

Ask yourself the following questions:

* How does this site exhibit effective use of space, size and balance?
* How could this site benefit from a reexamination of space, size and balance?

**Step 2**

Compare the two layouts. What makes them different? What are the similarities in both that appeal to you?

**Step 3**

Search Google and Google Image Search for “thumbnail layout sketches” and compare and contrast various solutions.

Take note of thumbnail sketch designs that you think are particularly effective.

**Step 4**

Now that you've analyzed the layouts of two sites and seen examples of thumbnail sketches, think about the site you began creating in Unit 1, grab a pencil and paper, and draw two different versions of what you think would make an effective homepage for your site.

If you're not the best artist, it's okay! Just use rectangles and basic shapes.

As you draw, keep in mind the importance of demonstrating focal point, hierarchy, and flow across different grid-based layouts.

These thumbnail drawings will help you prepare for the project at the end of this unit.

**Advanced Study**

Using Google Chrome's Inspect Element feature, explore the HTML and CSS of one of your selected pages.

It's okay if you don't understand all of the code you're seeing. There's going to be some elements we haven't discussed yet.

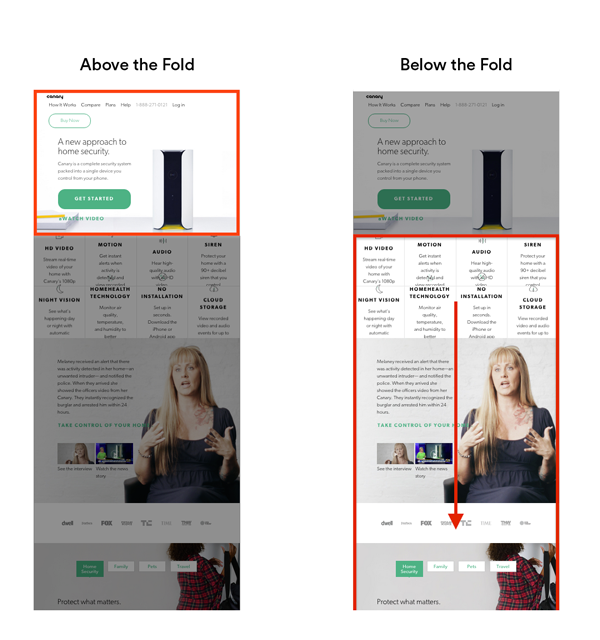
If you're unfamiliar with Google Chrome's Inspect Element feature, check out our videos "Chrome, Gmail, and Google Drive" and "Thinking Like a Developer" from our Plugging into Circuits pre-work section.

# Review Guide: How Design Works

Let’s review some of the key ideas introduced in this unit. You can also download all this content for future reference by clicking on the attached file.

## Key Definitions

* **Call To Action**: The element on a page prompting the primary desired user action.
* **Pixel**: The smallest “picture element” of screen based images. Often represented relative to size, i.e. ppi, or pixels per inch.
* **Above The Fold**: The top section of a website — the part that's visible without scrolling down.
* **Below The Fold**: The part of a website that can’t be seen without scrolling down.



* **Index Page**: Also called the home page, this is the main page on a website.
* **Prototyping**: A part of the design process, usually for the purposes of testing and iterating, in either a digital or analog space.
* **Iteration**: Typical "design-speak" for the process of creation and destruction whereby new solutions are found while others are abandoned.
* **Mockup**: The presentation of the basic visual layout of a website. Designers use mockups in order to quickly present ideas and concepts for testing, allowing them to make rapid changes to the design based on feedback.
* **Comps**: Short for "composites," are proposed design directions. These may take a number of forms, including sketches, wireframes, and thumbnails.
* **Thumbnail Sketch**: In terms of web design comps, thumbnail sketches are most commonly used to refer to basic shapes and drawings that represent page elements.
* **Target Audience**: The viewers you anticipate gaining the maximum value from the site.
* **Responsive Design**: The construction of web sites to render in the optimal setting based on the size of the user's browser & device.
* **GitHub Repository**: A folder that lives online in the cloud where you can store web files.
* **Cloned Local Folder**: A folder that lives on your computer that connects with your GitHub Repository. Any changes made to files in this folder must be committed and synced in the GitHub Desktop app to connect them with the GitHub Repository.

## Visual Design Elements

Use the following elements as guides for articulating and defending your choices when creating the design for you website.

* **Visual Balance**: The distribution of an object’s “visual weight” (often used in relation to positive and negative space on a site). Visual balance can also refer to the overall arrangement of the objects on a web page.
* **Scale**: Refers to the size of an object on a page, and in turn, its visual relationship to the size of other page elements. Sizing objects on a page is a common method of attracting visual attention or organizing elements based on relative importance.
* **Space**: Positive space, or the content of the page, is framed by the negative space (often called "white space", even if the color isn't white).
* **Focal Point**: The primary point of visual emphasis, or in other words, the element on the site that the users should focus on. Examples of this include logos, headlines, and large images (often called "hero images").
* **Hierarchy**: In web design, hierarchy may refer to the organization of elements on a page, or more broadly as the organization of different pages on a site.
* **Flow**: The movement of the user’s eye through the site content, initiated by a focal point and influenced by hierarchy.

## Design Process

**1. Understand The Problem**   
Key idea: Each website has a goal, or a problem its attempting to solve. For example, a portfolio site is solving the problem of a designer's work not being visible online; or a healthcare site is solving the problem of a lack of information for patients. If you clearly understand the problem, it will be easier to communicate your idea (solution) to the end user.

**2. Brainstorm**  
Key idea: When you become comfortable with the practice of brainstorming and not feeling as if only “good” ideas are worth further consideration, you will shed the anxiety that often occurs when faced with a blank page.

**3. Create a Sitemap**  
Key idea: By building the site’s structure as well as the overall goals for the viewer ahead of time, this will prevent unnecessary and time-consuming site restructuring later.

**4. Thumbnail Sketching**  
Key idea: Rapid generation of clear ideas is the purpose of thumbnail sketching. Relationships of size, space, and balance should be evident at this stage.

**5. Iterate**  
Key idea: Don’t be afraid to “kill your darlings,” or rather, don’t be afraid to abandon an unsuccessful design even if it holds special personal significance.

**6. Deploy**  
Key idea: Remember, the design process is iterative in nature. Once your page is live on the web, continue acquiring user feedback to inform future changes on the site.

## The Visual Design Toolset

Although we've decided to use [wireframe.cc](https://wireframe.cc/) — a free, browser-based sketching tool — there are a number of other great tools available, ranging in both price and complexity. Below is a list of software programs commonly used by professional user-experience designers and web designers.

**1. Sketch**  
Designed to rapidly concept nearly any type of digital execution—allows for the rapid updating of reusable elements such as buttons and other interface styles.  
[Check it out here.](https://bohemiancoding.com/sketch/)

**2. Abobe Illustrator, Photoshop, and InDesign**  
The most common software found across design disciplines. Adobe Creative Cloud mobile and desktop solutions are utilized at all steps of design, development, and deployment.  
[Check it out here.](http://www.adobe.com/creativecloud.html)

**3. OmniGraffle**  
Useful for diagramming website structure, refining webpage sketches, adding annotations.  
[Check it out here.](http://www.omnigroup.com/omniGraffle)

**4. Apple Keynote**  
Useful for laying out rough comps or more detailed drafts as well as adding basic interactivity at the between individual elements and/or pages.  
[Check it out here.](http://www.apple.com/mac/keynote/%20target=)

**5. Inkscape**  
A free, open-source version of Adobe Illustrator.  
[Check it out here.](https://inkscape.org/en/)

## Questions to Review With Your Mentor

1. What are some examples of “good” web design and “bad” design?
2. Can you show me an example of a website that’s effective in guiding the user’s eye through the page?
3. What software, hardware, and other tools do you use to design websites? When comparing tools, what features and functionality should I look for in web design hardware/software?
4. What are the most common industry tools for designing web pages?
5. Where do you find web design inspiration?

## Further Reading

Looking for more help with the design basics? Here are a few places you can check out:

American Institute of Graphic Arts, the oldest and largest professional membership organization for design:  
[http://www.aiga.org](http://www.aiga.org/)

An interesting read on how good design promotes understanding:  
[https://www.vitsoe.com/rw/about/good\-design](https://www.vitsoe.com/rw/about/good%5C-design)

William Lidwell’s Universal Principles of Design is a wide-ranging survey of the root components shared across disciplines:   
[http://www.amazon.com/Universal\-Principles\-Design\-William\-Lidwell/dp/1592530079](http://www.amazon.com/Universal%5C-Principles%5C-Design%5C-William%5C-Lidwell/dp/1592530079)

Design agencies worth following: R/GA, Razorfish, AKQA, Huge, Blue State Digital, and Possible:

* [http://www.rga.com](http://www.rga.com/)
* [http://www.razorfish.com](http://www.razorfish.com/)
* [http://www.akqa.com](http://www.akqa.com/)
* [http://www.hugeinc.com](http://www.hugeinc.com/)
* [http://www.bluestatedigital.com](http://www.bluestatedigital.com/)
* [http://www.possible.com](http://www.possible.com/)

SmashingMagazine.com, a content provider specific to web designers and developers, frequently posts “how-to” articles in addition to articles covering code and design techniques, inspirational sites, and the business of web development:  
[http://www.smashingmagazine.com](http://www.smashingmagazine.com/)

Communication Arts is standard reading for professionals working in various design disciplines:  
<http://www.commarts.com/>

A good site to examine flexibility and continuity. Although the number of columns defining the site’s grid appears to change between pages, what the viewer is experiencing is the merging of smaller column units into larger fields, a design common to print media:  
<http://www.ffmark.com/>

[←](https://circuits.generalassemb.ly/student/209/pages/792?direction=back) [→](https://circuits.generalassemb.ly/student/209/quizzes/233)

### Downloads

[Summary](https://ga-sprites.s3.amazonaws.com/uploads/production/page/downloadable_content/144/Summary.zip)

© 2016 General Assembly. All rights reserved. [Regulatory Information](https://generalassemb.ly/regulatory-information) [Terms](https://generalassemb.ly/terms_and_conditions) [Privacy](https://generalassemb.ly/privacy_policy) [Assembled](http://nytm.org/made-in-nyc)

You just learned so much about design technique... But there’s even more to learn in the unit 2 lecture! Check it out before you start your project.

Earlier in this unit, you researched websites and created some simple thumbnail drawings. We hope those steps helped you gain a clearer vision of the website you’re planning to develop.

In this project, you’ll take your design one step further by turning your thumbnail drawings into digital mockups. You’ll also learn how to host images online using your Github account.

This exercise will not only help you plan, it will help your mentor understand what key areas he or she will need to help you develop.

## Step 1: Reflection

Look back at your work for the design challenge. Think about a more concrete visualization of your thumbnail drawings.

## Step 2: Creating Your Mockups

Using [wireframe.cc](http://wireframe.cc/) (or the design program of your choice), create two mockups based on your design challenge thumbnails, keeping in mind focal point, hierarchy, and flow.

Don't know where to start? Don't worry! For design strategies and instructions on how to use wireframe.cc to create your mockups, continue to the next activity, where a slideshow will walk you through the process.

Following that slideshow is an additional set of steps that will instruct you how to host your finished design mockups on GitHub.

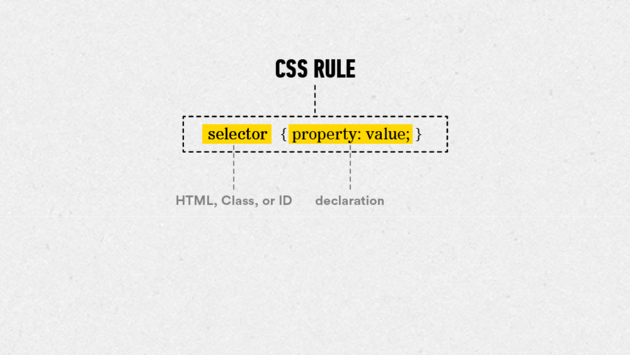
If you have any questions, reach out to your mentor.

# Review Guide: Principles of HTML & CSS

Let’s review some of the key ideas introduced in this unit. You can also download all this content for future reference by clicking on the attached file.

## Key Definitions

* **Rule**: The building block of a CSS stylesheet. A rule consists of a selector and a declaration block (one or more declarations).
* **Declaration**: A declaration is made up of a property and a value, separated by a colon and punctuated by a semi-colon."
* **Selector**: The actual HTML object the declaration(s) apply to.
* **Property**: The characteristic of the selector that will be changed.
* **Value**: The amount or type of change to be applied to the corresponding property of the matched selector.

  
- **Marking-Up**: The process of assigning HTML tags to given text content in order to indicate its relation to the rest of the text or dictate how it should be displayed.  
- **Serif Font**: One of two general categories of fonts (typefaces) that uses marks (called “serifs”) to embellish characters. A common serif font is “Times New Roman”.  
- **Sans-Serif Font**: One of two general categories of fonts that have cleaner line due to not using marks to embellish characters. (Sans Serif literally meaning “Without serif”) A common sans-serif font is “Helvetica”.  
- **Class**: A class attribute is added to an HTML element in order to give you a “hook” to refer to that element in your CSS. CSS class selectors begin with a “.”. Classes can be used multiple times per page.  
- **ID**: An id attribute is added to an HTML element in order to give you a “hook” to refer to that element in your CSS. CSS id selectors begin with a “#”. IDs can only be used one time per page.  
- **Horizontal Rule:** Add a horizontal line across your page using the hr tag.  
- **Line Break:** Break up a block of text using the br tag.  
- **Image:** Add images to your HTML using the img tag. Tell the browser the source of the image file with an src attribute.  
- **Absolute File Path**: A path to a website or file that includes a full web address (starting with “http”) that the browser loads from the remote location directly. For example:

<img src="http://imgur.com/awesomedog.jpg">

* **Relative File Path**: A path to a website or file that gives you the path to the resource you are looking for as it relates to your website's local file structure. For example, if you wanted to retrieve an image called newlogo.png that resides in a file called img, you would enter the following relative address:

<img src="images/awesomedog.jpg">

## Why Separate HTML from CSS?

Separating HTML from CSS offers you scalability and versatility. If you separate how your site looks from what your site says, things become more flexible.

By separating your HTML and CSS, you can make the change in one place and have it apply to your whole site. If you separate what your site says from how it looks, you can apply any number of different styles to the same content.

## CSS Color Treatment

While color names are fine when you're just beginning, there's a number of reasons you'll want to switch over to something more advanced.

First, color names are rendered differently by different browsers. Secondly, there are only 147 color names accepted as standard, meaning your options are going to be pretty limited.

Instead, you'll want to use either RGB or hexadecimal codes. Both of these are built on a system of entering values for the colors red, green, and blue.

By mixing different intensities of these three colors, you can create millions of different colors and shades. Intensity values range from 0 (no intensity) to 255 (full intensity) in the RGB system.

In hex, they range from 0-9, then continue from A-F, with two characters each for red, green, and blue.   
This is clearer with examples, so [here are some common colors with their RGB and Hex equivalents](http://www.w3schools.com/tags/refcolortryit.asp?color=White).

The format for color names, RGB, and hexadecimal should look like the following, respectively:

p {

color: red;

}

p {

color: (255,0,0);

}

p {

color: #FF0000;

}

## CSS Text Treatment

* **font-family**

To adjust the font of your selected text element, use the font-family property. For the value, enter the name of the font to which you’d like to alter your text.

To be safe, try putting a comma after your selected font and enter a generic family as a fallback. If the web browser doesn’t support the font you selected, it will choose the fallback.

h1 {

font-family: Arial, sans-serif;

}

If your selected font is more than one word, capitalize both words and put them in quotation marks.

h1 {

font-family: "Courier New", monospace;

}

* **font-size**

To increase or decrease the spacial dimensions of your chosen text, use the font-size property. As a beginner, you’ll want to enter pixel values for your font-size values.

h3 {

font-size: 24px;

}

As you become more advanced, try using percentages or ems instead of pixels. These can be pretty tricky, though, so you may want to wait until we cover them in Unit 9.

* **font-weight**

To adjust the thickness of your selected text, use the font-weight property.

As a beginner, you can enter values like “normal” to make your text thin and “bold” to make your text thick. As these values aren’t very specific, different browsers may interpret their display with slightly different outputs.

h1 {

font-weight: normal;

}

h1 {

font-weight: bold;

}

As you become more advanced, try using the numbers 100, 200, 300, 400, 500, 600, 700, 800, and 900 as values to gain more granular control. With this system, 400 is roughly equivalent to “normal” and 700 roughly equals “bold.”

h1 {

font-weight: 400;

}

h1 {

font-weight: 700;

}

* **font-style**

To make normal text italic, use the property font-style and the value “italic.” To reverse this effect, use the value “normal.”

a {

font-style: italic;

}

a {

font-style: normal;

}

* **text-align**

To adjust the positioning of a text element, use the CSS property text-align and one of the following values: left, right, center, or justify.

body {

text-align: center;

}

* **text-decoration**

To add an underline to normal text, use the CSS property text-decoration and the value “underline.”

h1 {

text-decoration: underline;

}

To remove underlines, use the value “none.” This declaration is often applied to anchor tags.

a {

text-decoration: none;

}

Less commonly used values include “overline” to add a line above text and “line-through” which strikes a line through your text.

* **text-transform**

To adjust capitalization in a selected text element, use the text-transform property.

Values for this property include “uppercase” to make every letter capitalized, “lowercase” to make every letter uncapitalized, and “capitalize” to make the first letter of every word in the selected text uppercase.

h1 {

text-transform: uppercase;

}

## Sublime Text Shortcuts

* **Quick Save:** ⌘+S (for Mac) or CTRL+S (PC)
* **Instant Boilerplate:** Type “html” + TAB (Mac and PC)
* **Instant Lorem Ipsum:** Type “Lorem” + TAB (Mac and PC)
* **Add Comment:** ⌘+/ (for Mac) or CTRL+/ (for PC)
* **Undo:** ⌘+Z (for Mac) or CTRL+Z (for PC)
* **Redo:** ⌘+Y (for Mac) or CTRL+Y (for PC)

## Classes vs. IDs

Classes and IDs, also called selectors, are ways of targeting the style of specific HTML elements on your page.

So what’s the difference between them?

In short, classes allow you to style many elements with a particular style, while IDs are only capable of styling a single element.

Using these selectors allows you flexibility and control of styling individual, as well as groups, of elements on your page.

## HTML and Images

When you’d like to add images to your site, you should use the tag with the “src” attribute. “src” stands for “Source” and works just like the tag’s “href” attribute. It tells the image tag where to find the image you’d want to include on your page.

## Relative vs. Absolute Addressing

Relative addressing basically tells your browser: “Look in our project folder for the file specified. Or, if a folder is specified, look within that folder for the file specified.”

On the other hand, when we link to a source outside of our project, it is referred to as an absolute address.

## Sublime Text Shortcuts

Bookmark [this guide](https://docs.google.com/document/d/1LugbQxfLpG5yFNfTz22hnGclN1Fwp_PH5QZfk0KLjO0/edit?usp=sharing) or download it for review.

## Questions to Ask Your Mentor

1. Why is it important to create separate HTML and CSS documents? Can you explain separation of concerns in greater detail?
2. How do I get started with creating an external style sheet? What elements should I define the style for first?
3. Where can I find examples of external style sheets to use as templates?
4. What are some best practices for organizing my files and folders on my computer? Should I be saving my work elsewhere?
5. Which fonts should I use for my project?

## Further Reading

How the RGB Color System Works  
<http://en.wikipedia.org/wiki/RGB_color_model>  
<http://www.rgbworld.com/color.html>

More on Colors  
<http://www.w3schools.com/cssref/css_colornames.asp>  
<http://www.color-hex.com/>

Adobe Color CC (formerly known as Adobe Kuler)  
<https://color.adobe.com/create/color-wheel/>

More on Fonts   
<http://www.w3schools.com/css/css_font.asp>  
<http://www.w3schools.com/cssref/css_websafe_fonts.asp>

Difference Between Serif and Sans-Serif   
<http://www.urbanfonts.com/blog/2013/02/serif-vs-sans-the-final-battle>

Classes and IDs   
<http://css-tricks.com/the-difference-between-id-and-class/>

File Paths   
<http://css-tricks.com/quick-reminder-about-file-paths/>

Absolute vs. Relative Paths  
<http://www.coffeecup.com/help/articles/absolute-vs-relative-pathslinks>  
<http://www.boogiejack.com/server_paths.html>

# Building Your Home Page

In this unit, you learned how to expand your color palette by a couple million values, how to style text and font, how to label and manipulate elements with classes and IDs, how to add images, and more! Did you also know semantic HTML is a vital step in defining your content? If not, don’t worry! The Unit 3 lecture will tell you all about it.

In this project, we’ll revisit the page you made in Unit 1, and we’ll give it a few upgrades.

Let's get started!

## Step 1:

If you haven’t already created a **unit\_3** subfolder in your local GitHub folder, do so now. Within that folder, create two subfolders. Name one subfolder **images** and the other **css**.

Then, create a new file in Sublime Text and save it in this folder as **index.html**. Don’t worry that it has the same filename as the Unit 1 project. It’s okay to have multiple files in your GitHub folder named index.html as long as they’re not in the same unit subfolder. Your homepage should always be named index.html.

Next, recreate the HTML you had in your Unit 1 project. If you want to save time, feel free to copy and paste your original code. If you’d like some extra practice, you should recreate it from scratch.

If you start from scratch, try using the time-saving Sublime Text technique we showed you to create your HTML boilerplate (after saving the file as a .html file, type **html** and hit the **tab** key).

At this point, your HTML should contain at least the following:  
- The HTML boilerplate  
- A title your page "[Your business name]"  
- An <h1> with your business name  
- A <p> tag with a little information about your business

Okay, now that we’re all on the same page, let’s get building!

## Step 2:

First, let’s update the copy in your <p> element. Instead of just a description of your business, let’s make it a slogan! In fact, let’s give it the ID “slogan.” Try to come up with a fun, catchy phrase that captures the spirit of your business. If you’re short of ideas, try using one of the following slogan generators: <http://slogangenerator.co/> or <http://www.sloganizer.net/en/>

Beneath your slogan, add a hero image. This should be a large image that sets the tone of your site. For example, if your business is a restaurant, add a photo of something delicious. If it’s a personal blog, maybe add a photo of yourself or an image representative of the topics about which you’ll write.

Remember, there are two ways to add an image:

Option 1 (best practice) — Host your image locally by putting it in the **images** subfolder inside your \*\* unit\_3\*\* folder, and use a \*\* relative file path\*\* to reference the image.

Option 2 — Find an image online. Right click on the image (or press **CTRL** + \*\* Click\*\* ) and select “Copy Image URL.” Then, use an \*\* absolute file path\*\* to reference the image.

A common technique for slightly more advanced users is to add an image using the \*\* background\*\* property in CSS. This will allow you to have text on top of the image. Try this out if you’re feeling ambitious!

Underneath this new image, create another paragraph with a few sentences telling us about your business. Maybe talk about your history or your mission. Beneath this paragraph, add an unordered list that summarizes a few of your business’s offerings.

Finally, at the bottom of your HTML, add the phrase “Contact me:” with your email address following. Turn your email address into a link. To go above and beyond, see if you can make your email link automatically open an email that’s addressed to you. It’s easier than you think!

## Step 3: Make it Personal

The above content is primarily educational — we want to make sure you can apply the lessons in this unit; however, we understand that you probably have some cool content ideas of your own, so please take this step to add any additional content to your site that you’d like.

The more content you add now, the more interesting style and layout opportunities you'll have later. If you don't have ideas, check out competitor pages and see what content they include on their home page and how it's organized.

Also, add a line or two of space between chunks of content that you imagine will eventually be broken into different sections (as per your design mockup). Don't worry, those lines of space won't affect the layout of your content when it's rendered on the screen. The purpose for such space is simply to make your code more legible and to facilitate future organization of your code.

Additionally, we won’t be adjusting layouts until Unit 6; however, to make the layout process easier, for the time being, list your content as you expect it to eventually appear, top to bottom, left to right.

Once you’re finished adding content, save it. Now, it’s time to style your HTML.

## Step 4: Styling Your Content with CSS

Create a new file in Sublime Text and save it as style.css in the css subfolder of your unit\_3 folder. Make sure to add a link to your CSS in the section of your HTML. Save your HTML again once you’ve added it, then return to your stylesheet.

Consider the selectors you have available:

body

h1

p

ul

li

img

#slogan

Now, let’s give these elements some style! First, give your page’s <body> and <h1> elements unique colors using either the RGB or hexadecimal methods. If you’re having trouble picking the proper hues, check out a site like [Adobe Color CC](https://color.adobe.com/create/color-wheel/?base=2&rule=Custom&selected=4&name=My%20Color%20Theme&mode=rgb&rgbvalues=1,0.13279019591095675,0,0.5,0,0,1,0,0,1,0,0,1,0,0.37792984243083083&swatchOrder=0,1,2,3,4).

Next, make your slogan italicized.

After that, it’s up to you! Play around with the CSS properties you’ve learned thus far and adjust your site however you see fit.

background

background-color

color

font-size

font-style

font-weight

text-align

text-decoration

text-transform

## Step 5: Review the checklist below to make sure your project is ready to submit.

1. I’ve added all content to my page, even if it doesn’t look like my unit 2 sketch
2. I’ve used semantic tags correctly (I learned in the lecture!)
3. I’m using the simple reset (I learned in the lecture!)
4. I made sure to include alt attributes on all of my image tags
5. I only used classes and id's when absolutely necessary

**Thought this unit was the most challenging so far? It was!**  
Book a 1:1 with your mentor by clicking My Mentors.

## Step 6: Host your project for submission

Make sure you’ve saved your finished HTML and CSS files. Then, commit and sync your changes using the GitHub app. The link for your site should be:

<http://your_repository_name.github.io/unit_3/index.html>

If that doesn’t work, go to github.com, open your repository, and copy and paste the URL. Then, discuss whatever issues you had with your mentor.

## Advanced Study:

Try taking your page to the next level with some new CSS properties affecting layout. We're not covering these properties until a later unit, but try doing some quick research and see if you can figure them out!

margin

padding

border

width

text-align

NOTE: If you try to center your list with text-align: center, you'll notice that the list-items and their bullets behave in different ways. There's a reason for that, but it's not one you need to worry about yet.

A good workaround for right now—and one you'll find super-helpful very frequently—is to make the bullets invisible by including the following CSS rule in your stylesheet:

ul {

text-align: center;

list-style-type: none;

}