Acme Final Project Directions

Purpose

This final project is a review of the concepts, skills and techniques that should have been learned throughout the semester. It requires you to build a small 5-page site (home page and four content "pages") for a fictitious company - Acme, Inc.

Notice!

All HTML and CSS must be of your own creation. No outside templates, libraries or frameworks are acceptable for this project.

Expectations

This project expects that you can:

- Write semantic and valid HTML5 markup.
- Write valid CSS3 and use media queries with Flexbox to implement responsive layouts.
- Implement the basic design principles of Proximity, Alignment, Repetition and Contrast.
- Implement basic usability concepts in the web content and validated using the WAVE tool.
- Use JavaScript with JSON data to deliver content to web pages.
- **Do your own work, but you don't have to do it alone.**

The Big Picture

Refer to the Overview Video found in the Contents > Web Project folder of I-Learn for a video overview of the finished project.

Tasks

1. Download

A zip file containing a base folder along with subfolders of starter assets should be downloaded, unzipped and placed at the root of your repository folder. All subsequent work with the Acme web site will be performed here. The download can be found in the *Web Project folder*.

2. The JSON File

The content for the four content pages is stored in a JSON file found in the "**js**" folder of the downloaded seed code and will be used to provide content data using JavaScript (*you may want to review the fetch function used in previous weeks*).

Note: All needed content already exists in the JSON file, but to make your work easier you may want to add an additional array or object to the JSON file in order to build the navigation bar. If you do change the JSON file you will need to check that it is valid and has correct formatting using the JSON Editor Online tool at *http://www.jsoneditoronline.org/* .

3. Build a Template

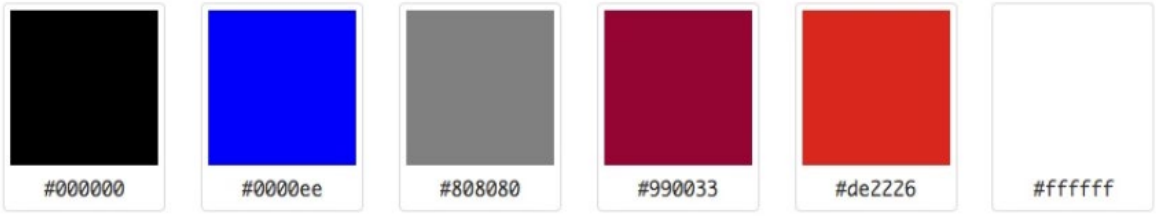
You are to build a generic template for the Acme web site (a template contains no content, it is a generic file from which actual content pages can be built) named "**template.html**" and stored in the provided "**support**" folder. The template must:

HTML

1. Use HTML5 semantic structures (header, nav, main and footer) for page sectioning.
2. Be valid HTML5 when completed.
3. Include the viewport meta element.
4. Include the **media="screen"** attribute for the css link element.

CSS

5. Use external CSS for all styling. The CSS must:
   - be stored in the provided css folder,
   - start with a mobile first approach,
   - use flexbox to provide responsive layouts to small and large screen sizes,
   - use a minimum of one media query for larger than mobile screen sizes,
   - be valid to CSS3 standards.
6. Use the provided images for the logo and background (found in the images > site folder).
7. Use the color scheme shown below:



8. Use a font set of your own choosing appropriate to the Acme site.

JavaScript for Navigation

9. Build the navigation bar:
   - Write a JavaScript function that will execute when the page loads.
   - The function must build a link to the "home" page first.
   - Then, read the four navigation items from the JSON file and build the remaining four links of the navigation bar (Hint: a loop will make this fast – think about the hourly weather data).
   - Each link should point to a page that matches the name of the content object from the JSON file.
   - In order to match web naming rules, the object names in the URL must be made lowercase. But the displayed value in the navigation menu must be title case (starts with a capital letter).
   - All five links must be in list items within an unordered list.
   - Include the completed navigation bar list into the template file.
   - If JavaScript is disabled, use a <noscript> element in the <nav> to display a message that JavaScript must be enabled to use the site.
   - The navigation links are: 1) Home, 2) Anvils, 3) Explosives, 4) Decoys and 5) Traps

Finished Template

10. 10. Weigh less than 150Kb total weight (this leaves 250Kb for content).

Example Template

An example of what the finished template:

4. Build the Acme Home page

Having built the template, you will now use it to build the Acme home page. This page is the sole physical page of the web site. It will display both the “home page” and “content pages”. The content pages will be built dynamically using JavaScript as described below. The page must:

“Home” Page Markup

1. The index.html page will be created and stored at the root of the provided acme folder,
2. Have a container to hold the “home page” content shown in the images (large and mobile sizes) below (the page content is in a text file included in the support folder and images are found in the images folder):

3. the "Home" page content should be visible by default.

“Content Page” Markup

4. The index.html page will contain a second container that designates it as "product content". This container will be hidden by default and must contain the following:
    - an <h1> element to display the product name.
    - an element to display the product picture.
    - an element to display the product description.
    - an element to display the product manufacturer.
    - an element to display the product reviews.
    - an element to display the product price.

Entire Page

5. be fully responsive so that the content is consumable across all devices without horizontal scrolling or zooming (including all images),
6. use usability concepts (refer to the Usability Checklist in the week 9 module) and validate that no usability errors exist using the WAVE tool,
7. be valid HTML5 and CSS3 code.

5. "Build" the Content Pages

The Acme site is to have four "content" pages. However, since the data is stored in JSON format, this means that you will reuse the index page to display the content based on the navigation link that is clicked. In short, the web site will have five "pages" - Home, Anvils, Explosives, Decoys, Traps - but will use only a single physical page. In modern web terminology, you are building a single page application.

JavaScript for Content Creation

1. Write a JavaScript event listener that will "listen" to the links in the main navigation bar. When the "home" link is clicked the acme home page content should appear and the “content” container should be hidden. Don’t forget to change the <title> element’s value.
2. When any of the other four navigation links is clicked:
    - the default action of the link should be stopped,
    - the content for that link should be read from the JSON file,
    - the content should be injected to the appropriate element within the "content" container in the index page
    - The "home container" hidden and the "content" container made visible, and
    - the page displayed in the browser for viewing.
3. If JavaScript is disabled, a message telling the site client that JavaScript is required to use the site must be displayed. Use a <noscript> element to do this.
4. The content page should display the full-size product image on the left and the other content on the right (similar to what you would find on an Amazon.com content detail page).
5. The content must use appropriate markup and be formatted to provide a professional appearance.
6. The content page must be responsive, usable and valid HTML5 and CSS3.
7. Here is an example of what a finished "content" page could look like:

6. Peer Reviews

A peer review will be conducted in class to find errors or omissions and give you feedback. The peer review will be done on the day specified in the class calendar. You must be present. Failure to participate in the peer review will penalize your final project 50%!

7. Submission

- Work with your learning team to complete the project and post to and read the weekly discussion board to help one another.
- Build and test your code in Brackets.
- Sync the code to your GitHub repository online and test there as well.
- Add two links to your home page: 1) to the Acme home page and 2) to the Acme template file. Sync the home page to your repository. Test that the links work.
- Check your work to ensure that it meets all the expectations listed in the grading matrix.
- When satisfied that everything is working:
  1. Create a short video (5 minutes maximum) of the Acme site in operation and post the video to your YouTube channel as an “Unlisted” video. Include in the video:
     - a demonstration of the navigation bar being clicked and the content being delivered.
     - a demonstration of the responsiveness of both "home" and "content" pages.
     - a demonstration of the validity of the HTML5 and CSS3 code for the "home" and "content" pages.
     - a demonstration of the usability validation using the WAVE tool
  2. Add the link to your YouTube video to your temporary home page adjacent to the links to the Acme home page and the Acme template.
  3. Sync the temporary home page to your github repository and test the link to make sure it works.

Grading Matrix

Consult the Project Grading Matrix for grading details - *found in the Web Project folder*.