Assignment: Combining HTML, CSS, JavaScript

Topic: Working with HTML, CSS, and JavaScript
Files, Validation, Editing Environment, Web Server Account, and File / Folder Organization

Assignment Activities

- Working with hosting account & URLs
- Work with files and folders
- Create HTML files
- Create CSS files
- Create JS Files
- Link all files
- Use validation tools for debugging
- Folder Compression
- Upload root folder to your web server / host and submit to BrightSpace

Resources:

Free Web Host: https://www.infinityfree.com/
HTML: https://www.w3schools.com/html/default.asp
CSS: https://www.w3schools.com/css/default.asp
JS: https://www.w3schools.com/js/

Validator JS: https://codebeautify.org/jsvalidate
Validator HTML: https://validator.w3.org/nu
Validator CSS: https://jigsaw.w3.org/css-validator/

Assignments are a great way to practice what you've learned in class, resources you explore online for professional development, and content from labs. This should be YOUR synthesis of knowledge, DON'T copy and paste from the web or other resources. You may reference documentation and resources online ONLY if you document referenced sites.

With this assignment, you will develop a site based on YOUR creativity that meets the assignment requirements. All sites for this course must be professional – how to determine if it's professional; consider showing the site to the Dean of the college and if the content could be shown to a general audience (Rated G).

Task:

Practice your understanding of HTML for structuring content, CSS for presentation (styling and layout), and JavaScript for behavior (interactivity).

- Note all resources referenced for this project.
    - Create a citation .txt file in the root folder.
    - Use comments in html <!—comment -->, in css /* comment */, or in js // comment.
        - Note where you applied the knowledge from the reference sources.

1. Plan your website and decide on its purpose and theme.
    a. Write up a short summary of the website's "point" approx. 2 paragraphs. This could be a personal portfolio, a blog, a restaurant menu, or any other topic you are interested in.
    b. Sketch the wireframe (see Brightspace for wireframing) of the layout of the pages, similar to a storyboard.
2. Next, create the main structure of the website using HTML. This includes creating an index.html file as the homepage and additional HTML files for each page included – at least 3 pages.
    a. Pages should be linked to each other.
    b. A navigation menu should be created.
    c. Include an email link to your email (hint: "mailto:mail@maine.edu")
3. Use CSS to style the website, one .css file only, linked to all html files. Experiment with different colors, fonts, layouts, and backgrounds to make the website visually appealing.
    a. Pick a color scheme, BrightSpace has some links to color resources.
    b. Maintain ADA compliant design (high contrast, color blind aware, etc.)
    c. Use at least one font color change, one background color change, a width setting, float, and clear:both.
    d. Explore CSS documentation provided in BrightSpace and implement one CSS rule as a 'skills stretch' element. Be sure to comment where you referenced for this element.
4. Create basic website interactivity using JavaScript.
    a. Add functions like date, mouseover, mouseout, or other behavior as presented in class
    b. Use JS behaviors to enhance the user experience.
    c. Explore JS documentation provided in BrightSpace and implement one JS rule as a 'skills stretch' element. Be sure to comment where you referenced for this element.
    d. In JS, implement a required variable, loop, logical decision, and function
5. Include images that is aligned with the message of the site
    a. Use appropriate width, height, and alt attributes (ADA compliance).
    b. Resize images for use on websites – see Brightspace about Images for Web.
    c. All images should have the smallest file size that still creates a quality image. Thumbnails are usually under 80kilobites 80kb.
        i. Many sites have copies of an image for thumbnail, small, medium, large, and original size, saved on the computer for use as needed.
            1. For this project, submit only the images used for the project.
        ii. Save all images in the images folder of your root folder. Remember the img src needs to point to the correct file path (hint: src="images/picture.jpg").
6. Upload site to your server as a new directory.
    a. Test the website to ensure that it works properly and appears as intended on different devices and browsers.
    b. Identify the URL for this specific directory on your server.
7. Submit the compressed root folder and URL.

Check Submitting Guidelines (next page) and Grading Criteria (last page) before submitting.

Before submitting, check for:

- Project used root folder with a sub-folder for images.
    - No spaces in file and folder names, underscores and camelCase are best.
    - Use meaningful naming conventions.
    - First html page is always index.html.
    - All pages are linked correctly to relevant files (.html, .css, .js, images).
- Note all resources referenced for this project.
    - Create a citation .txt file in the root folder.
    - Use comments in html <!—comment -->, in css /* comment */, or in js // comment.
        - Note where you applied the knowledge from the reference sources.

- DO NOT move files into the server's main directory
    - use root folder to create a sub-directory in your web account
    - root folders go into the htdocs directory on infinityfreeapp.com or the hosting directory of your alternately chosen server.

> Example: if my main url is http://drlg.infinityfreeapp.com if I upload a root folder called lab1_drg, then my url for this assignment would be http://drlg.infinityfreeapp.com/lab1_drg
>
> Note: Your URL will have a different designation before the infinityfreeapp.com, where mine is drlg, yours will have something different.

Grading Criteria on next page.

Grading Criteria:

- Website Structure 20pts
  - Includes an index.html file as the homepage 4pts
  - Additional HTML files for each page 5pts
  - Valid HTML tags for content structure 8pts
  - All external resources (css, js) linked correctly 3pts
- CSS Styling 20pts
  - Consistent and visually appealing design 6pts
  - Effective use of colors, fonts, and layouts 6pts
  - Included all CSS elements indicated in assignment 4pts
  - Valid CSS syntax 4pts
- JavaScript Functionality 15pts
  - Proper implementation of JavaScript for interactivity 4pts
  - Included all JS elements indicated in assignment. 3pts
  - Valid JS 8pts
- Accessibility and Inclusive Design 10pts
  - Design implemented ADA compliance 5pts
  - Technical and presentation elements for inclusion & accessibility implemented. 5pts
- Content and Creativity 10pts
  - Relevant and engaging content 5pts
  - Clear purpose and theme 5pts
- Technical Accuracy 10pts
  - Proper use of coding conventions and syntax 5pts
  - No errors or bugs 5pts
- Additional Features 5pts
  - Implementation of extra features beyond the basics
  - Effective use of images, videos, and external links
- Overall Presentation 10pts
  - Neat and well-organized code 4pts
  - Consistent and professional visual presentation 4pts
  - Clear and concise documentation 2pts

Total /100