

Assignment 1 – Environment Setup

Setup

Each week you will be asked to create a new folder under web-231 following a naming convention of “week-<number>.” If we are on week two, the folder name should be “week-2.” All files associated with the weekly assignment will be added to the appropriate weekly folder. All programs must be linked in the index.html landing page under the “Weekly Assignments” section. Projects will be linked under the “Projects” section of the index.html landing page. To be clear, all of the JavaScript, HTML, images, and CSS files associated with a weekly assignment must be placed under the appropriate weekly folder. The page title for all HTML files in this course must say “WEB 231 – Enterprise JavaScript I.” And, all HTML and CSS files must be valid HTML/CSS, tested through the WC3 validator. The links were provided during WEB 200 and were added to the index.html landing page. Also, the blue border around the provided images is to show they are images and should not be included in your submission. In other words, do not add a blue border around your work, unless the instructions explicitly ask for it.

User interface styling and formatting requirements are located in the HTML, CSS, and JavaScript Requirements document.

HTML: **index.html**

CSS: **site.css**

Grading Reminders

- A. (50%-points) All code sources (.html, .css, .js) must be cited in the opening programmers’ comments, following the format specified in the code attribution document.
- B. (25%-points) All code sources (.html, .css, .js) must show evidence of code comments. This means each section of the program (.html, .css, .js) must include code comments that explains what the block of codes purpose is, what the required parameters are (data type, if any), and what the expected output is.

- C. (rubric) All code sources (.html, .css, .js) are measured against
1. Code functionality: Does it work? Does it meet requirements?
 2. Adherence to standards and conventions. Are you using the appropriate data types, including proper indentation, are variables named appropriate (variable x is an example of poor naming conventions), is there an appropriate use of whitespace, is the code organized, and are semicolons being used to terminate code sentences?
 3. Efficiency: Use of language features. Are you practicing DRY (Don't-Repeat-Yourself?), are you leveraging built-in language features where appropriate, and are you using classes/functions to reduce code clutter?
 4. Documentation: Self-documenting, naming conventions, code is maintainable by others. Is the code your write easy to read and maintainable by others?
 5. Error trapping/handling. Are there errors in the program? Is there evidence of coding best practices to reduce user errors?
 6. Assignment Specific Compliance. Does the delivered solution follow the instructions, as they are written? Does the output match what was provided in the screenshots (including spaces, styling, etc.)?

Required Modifications

- Cite any sources in your opening programmer's comment

Exhibit A. User Interface (final solution)

Professor Krasso's Landing Page

WEB 231 Enterprise JavaScript I

Weekly Assignments

- Exercise 2.2
- Assignment 2.3
- Assignment 3.2
- Assignment 4.2
- Assignment 5.2
- Assignment 6.2
- Assignment 7.2
- Assignment 8.2

Projects

- Final: Card Game
- bioSite Project Home Page

Important Links

- W3C HTML Validator
- W3C CSS Validator
- HTML Tutorial
- JavaScript Best Practices
- WEB 231's GitHub Repository
- BRUIN Connect

1. Add a new file under web-231 and name it index.html

HTML Requirements

- a. Give the HTML document a title of “WEB 231 – Enterprise JavaScript I”
- b. Add a link to site.css in the <header> section.
- c. Add the courses Google font kit

Google Font Kit Requirements

All assignments in this course will be using the Google font kit: Oswald, Verdana, Arial, sans-serif. The process for creating Google fonts was covered in WEB 200. This is the only approved font type for WEB 231 – Enterprise JavaScript I.

- d. Add a main div to the body of the HTML page and give it a CSS id of **container**.

Additional HTML Requirements

- 1) Add an h1 tag and give it a value of “<yourFirstName> <yourLastName>’s Landing Page”
 - 2) Add an h2 tag and give it a value of “WEB 231 Enterprise JavaScript I”
 - 3) Add an HTML horizontal line.
 - 4) Add an h3 tag and give it a text value of “Weekly Assignments.”
 - 5) Underneath the h3 tag add an unordered list with list items for the assignments shown in Exhibit A.
 - 6) Underneath the assignments section add another h3 tag and give it a text value of “Projects.”
 - 7) Underneath the projects section add another unordered list with list items for the projects shown in Exhibit A.
 - 8) Underneath the projects section add an h4 tag with a text value of “Important Links.”
 - 9) Underneath the important links section add links for
 - i. W3C HTML Validator: <https://validator.w3.org/>
 - ii. W3C CSS Validator: <http://jigsaw.w3.org/css-validator/>
 - iii. HTML Tutorial: <https://www.w3schools.com/html/>
 - iv. JavaScript Best Practices: https://www.w3schools.com/js/js_best_practices.asp
 - v. WEB 231’s GitHub Repository: <https://github.com/buwebdev/web-231>
 - vi. BRUIN Connect: <https://bruinconnect.bellevue.edu/>
-
2. Add a new file under web-231 and name it site.css

Additional Styling Requirements

- a. Set the bodies font family to the courses Google font kit.

- b. Set the container to a margin top of 50 pixels, a margin left of 100 pixels, and a margin right of 100 pixels.
 - c. Set the h1, h2, h3, and h4 tags to a background color of 3C275A, a font color of white, a border of a 1 solid pixel and a color of D6A800. Finally, set the padding to 10 pixels.
 - d. Set the anchor tag to a color of 453674 and remove the underline.
 - e. Set the anchor tag's hover to a background color of D6A800 and a font color of 4F3674.
 - f. Set the anchor tag's visited font color to 3C275A.
-
- 3. Validate the HTML and CSS by using W3C's HTML and CSS validators. Take screen captures of each passing test (index.html and site.css). There should not be any errors in the results printout. If errors exist, correct them and rerun the tests.
 - 4. Combine all images in a single Word document include your name, date, and assignment number.
 - 5. Configure the repository for GitHub pages. Refer to the provided resources on how to configure the repository for GitHub pages.