UDACITY

‹ Return to Classroom

Personal Blog Website

REVIEW

CODE REVIEW 10

HISTORY

Requires Changes

3 specifications require changes

- Keep up the Awesome job👏👏
- However, reviewers are ruled by the Rubric and there are still main requirements that have not been met.
- Thank you for all the hard work and dedication you have put into this project. it will make you an exceptional front-end developer in the future. Just a few more tiny adjustments and you will pass this submission. I have added detailed instructions on how to fix it in both the code and project review sections. If you have any questions please don't hesitate to ask on the knowledge platform.
- Your code quality is `top-notch` 👌 Well Done 👏🎉👏🎉
- I wish you Good Luck on your Journey Ahead, Stay Udacious and Happy Learning!

#PRO-CODER

Structure

1. Portfolio completely separates structure from design/style.
2. There are no attributes present in the body of the document.
3. There are no elements in the document.

👏👏

There should be at least 3 imported files in the main CSS file, but the student is welcome to break it down even further if that makes it easier for them.

👌

Files are organized with a directory structure that separates files based on page and functionality.

👏👏

There is an intentional user flow on each page with appropriate links as needed.

- The flow is correct👏👏

Design

Custom images, layout, and styling.

- Great job designing your webpage without using any external frameworks.👏👏

- Typography: Custom design for typography with at least 3 unique properties for each typography selector
  1. Headers (h1 to h3 at minimum) `<h1>`
  2. Paragraph Text `<p>`
     a. Bold
     b. Italic
     c. Underlined
  3. Links `<a>`
  4. Quotes
- Colors: At least 3 colors are used.

- You have to include `<b>` for `bold`, `<em>` for `italic`, and `<u>` for `underline` properties inside your paragraph tag and also include <blockquote> in your project.

The following are used in the webpage:

- Image(s)
- Image caption
- Buttons
- Card

- You have to include image caption in your project.

The general pattern of the webpage should be as follows:

- Blog Homepage
  - Navbar
  - Blog Post Card
    - Card
    - Image
    - Buttons
  - Footer
- Blog Post
  - Header
  - Author Info
    - Name
    - Title
    - Current Company/School
    - Short Bio
    - Avatar
  - Social Sharing
    - Twitter
    - Facebook
    - LinkedIn

- Your website follows the above pattern👏👏

The following pages should be present:

1. Blog Homepage
2. Blog Post

- Both pages are present.

Layout

Uses Flexbox at least twice throughout the project to create a component or pattern

👏👏

Uses CSS Grid at least twice throughout the project to create the layout for pages

👏👏

Responsiveness

Pages are mobile-friendly and display correctly on all display sizes (mobile, tablet, desktop).

Note: You can simulate Mobile Devices with Device Mode in Chrome DevTools.

- Responsiveness should be checked after making final changes.

Quality

- HTML5 semantic tags such as `<header>`, `<footer>`, `<article>`, `<section>` , etc. are used to add meaning to the code.
- No div or section selectors are without a CSS class or id.

- Nice use of semantic tags.

- All code is lowercase
- The code does not have trailing white spaces.
- Indentation is consistent (either all tabs or all 2 spaces or all 4 spaces etc).
- Code uses a new line for every block, list or table element and indent every such child element (it is acceptable to put all elements in one line).
- When quoting attribute values, code uses consistent quotation marks (single vs. double).

- Nice indentations.

- HTML documents HTML5 `<!doctype html>`
- Code omits type attributes for style sheets and scripts.
- [Optional] Code does not use entity references unless necessary e.g. characters with special meaning (like `<` and `&`) as well as control or “invisible” characters (like no-break spaces).

👏👏

- The code does not have trailing white spaces.
- Indentation is consistent (either all tabs or all 2 spaces or all 4 spaces etc).
- Code indents all content, that is rules within rules as well as declarations to reflect hierarchy and improve understanding.
- The code uses a semicolon after every declaration for consistency and extensibility reasons.
- Code always uses a space after a property name's colon, but no space between property and colon, for consistency reasons.
- Code always uses a single space between the last selector and the opening brace that begins the declaration block. Code always start a new line for each selector and declaration.
- Code always put a blank line (two line breaks) between rules.
- Code uses consistent quotation marks for attribute selectors or property values (single vs. double).

Code uses consistent quotation marks for attribute selectors or property values (single or double).

- Your CSS indentations are clean and consistent ✅👌

- The code uses meaningful or generic ID and class names that are as short as possible, but as long as necessary.
- The code does not use element names in conjunction with IDs or classes.
- The code uses shorthand properties where possible.
- [Optional] Code omits unit specification after 0 values.
- [Optional] Code includes leading 0s in decimal values for readability.
- [Optional] Code uses 3-character hexadecimal notation where possible.
- [Optional] Code separates words in ID and class names by a hyphen.

👏👏

☑ RESUBMIT

DOWNLOAD PROJECT

10 CODE REVIEW COMMENTS ›

Learn the best practices for revising and resubmitting your project.

Rate this review

START