Frontend Web Development   Lab 2: HTML, CSS, JS Pretests
CSCI 6333 - Section 30

HTML challenge adapted from:
https://www.codecademy.com/resources/blog/html-and-css-code-challenges-for-beginners/
JS challenge adapted from: https://leetcode.com/problems/to-be-or-not-to-be/
CSS challenge adapted from: https://css-challenges.com/black-or-white/

Overall Instructions

1. The following code challenges are just to give me an idea about your grasp of JS. Don't feel anxious if you can't complete the challenge. **You will receive full credit for the assignment as long as you put in effort.** Future assignments are graded on accuracy. This one is graded on trying hard and being creative. Give it a crack and show me that you know how to *think* in code. Whether you get the result we are looking for or not this will help me understand where you are as a student.

2. The three challenges below each have their own instructions. Be sure to read those instructions carefully.

Part A - HTML

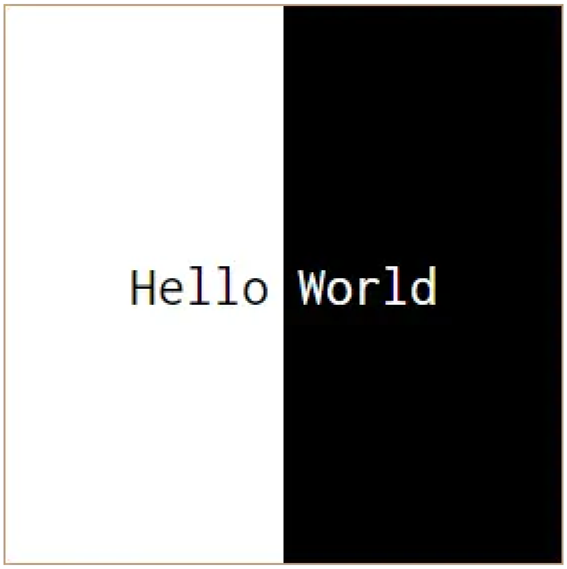
Instructions

1. Create a tribute page for a historical figure
2. You can use any IDE or text editor you want. The only constraint is that it should be runnable via "Go Live" in VS Code
3. Choose a historical figure who is meaningful to you and create a webpage dedicated to them. The webpage should include:
   a. A title or heading with the person's name
   b. An image of the person
   c. A caption for the image
   d. Alt text for the image
   e. A timeline of the person's life in the form of a list
   f. Some <body> text that describes why this person is important to you
4. Upload the HTML file to the D2L dropbox

Part B - CSS

Instructions

1. Using the following code and images, recreate the CSS.
2. Your finished page should load in browser like this:



Hello World

3. No HTML Edits: Here is your baseline HTML:

```
<html>
        <body>
                <div class="box"></div>
        </body>
</html>
```

4. No SVG: Embedded SVGs are not allowed via the url() tag inside background-image.
5. No JavaScript: Don't try to add any scripts!
6. No Image: I think this one is trivial. Don't take a screenshot of the solution and use it!\
7. Here is your starter CSS:

```
.box {
                width:280px;
                height:280px;
                display:flex;
                align-items:center;
                justify-content:center;
                font-family: "Inconsolata", monospace;
                font-size: 28px;
                border:1px solid #c39f76;
                box-sizing:border-box;
        }
```

8. Upload your completed .css and .html files to the D2L dropbox. **Note: The HTML file should ONLY INCLUDE THE LINES ABOVE.**

Part C - Javascript

Instructions

1. The following code challenge is just to give me an idea about your grasp of JS. Don't feel anxious if you can't complete the challenge. Give it a crack and show me that you know how to *think* in code. Whether you get the result we are looking for or not this will help me understand where you are as a student.

2. Use whatever IDE you like. If you don't want to set up a project, you can also jsfiddle to complete this project.

3. Write a function called expect that helps developers test their code. It should take in any value val and return an object with the following two functions:
    a. toBe(val) accepts another value and returns true if the two values === each other. If they are not equal, it should throw an error "Not Equal".
    b. notToBe(val) accepts another value and returns true if the two values !== each other. If they are equal, it should throw an error "Equal".

4. Review the code examples before starting.

5. Start by manually typing the starter code below into your chosen IDE.

6. Turn your JS file into the D2L dropbox

Code Examples:

Example 1:

Input:func = () => expect(5).toBe(5)
Output: {"value": true}
Explanation: 5 === 5 so this expression returns true.

Example 2:

Input: func = () => expect(5).toBe(null)
Output: {"error": "Not Equal"}
Explanation: 5 !== null so this expression throw the error "Not Equal".

Example 3:

Input: func = () => expect(5).notToBe(null)
Output: {"value": true}
Explanation: 5 !== null so this expression returns true.

Starter Code:

I would type this into your IDE. The backticks to make the formatting show code could break any copy/paste attempt. Note: Be sure to uncomment before proceeding.

```
/**
* @param {string} val
* @return {Object}
*/
var expect = function(val) {

};

/**
 * expect(5).toBe(5); // true
 * expect(5).notToBe(5); // throws "Equal"
 */
```