

Assignment 2 – Palindrome App

Setup

Each week you will be asked to create a new folder under web-330 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 folder. All programs must be linked in the appropriate landing page. Projects will be linked under the “Projects” section of the index.html landing page.

The document title of all HTML files in this course must say “WEB 330 – Enterprise JavaScript II.” And, all HTML and CSS files must be valid HTML/CSS, tested through the WC3 validator. The links were provided during WEB 200 and 231. As part of your submission, be sure to include screenshots of the results from the validation tests (HTML and CSS validators).

User interface styling and formatting requirements are located in the Web 330 HTML, CSS, and JavaScript Requirements document.

HTML: **<yourLastName>-palindrome.html**

CSS:

Grading Reminders

- A. (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: Code is maintainable by others
 - i. Code comments are present in all blocks of code, written as full sentences, free of grammatical errors, and function/class parameters and data types have been identified.
 - ii. Code attribution is present in all files and authorship is clearly annotated.
- 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
- Link the appropriate CSS, JavaScript, and Google fonts in the document's header section

Exhibit A. User Interface (final solution)

Welcome to the Palindrome App! 1

Palindrome Checker 2

3
Phrase value: 4

5

[Return](#)

Palindrome Results 6

7
Date: 2/16/2021
Original Phrase: racecar
Reversed Phrase: racecar
Phrase Length: 7

8
racecar is a palindrome!

1. h1 with the CSS class app-header and a text value of “Welcome to the Palindrome App!”
2. card-title with a text value of “Palindrome Checker”
3. form-field with a text value of “Phrase value:”
4. input field with an id txtPhrase
5. form-field button with an id of btnCheckPhrase
6. card title with a text value of “Palindrome Results”

Additional JavaScript requirements

- a. Create a function named getTodaysDate() and return a new JavaScript date with the parameter “en-US”

- b. Create a function named `getLength()` with a single parameter for a string value. In the body of the function, return the string's length.
- c. Create a function named `reverse()` with a single parameter for a string value. In the body of the function, split the string to a character array, reverse the string (use JavaScript's built-in `reverse()` function), and then convert the character array back to a string value (use JavaScript's `join()` function). Finally, return the string.
- d. Create a function named `isPalindrome()` with a single parameter for a string value. In the body of the function, call the `reverse()` function (this was created in c) and return true if the results from the `reverse()` function call equal the string parameter. If they do not match, return false.
- e. Register an onclick event using `document.getElementById("id").onclick` function

Additional Requirements

- 1) Capture the inputted phrase and assign it to a variable named `txtPhrase`
- 2) Assign the "assign-results" div to a variable named `assignResults`
- 3) Create a variable named `today` and assign it the results from the `getTodaysDate()` function call
- 4) Create a variable named `len` and assign it the results from the `getLength()` function call.
- 5) Create a variable named `reversedPhrase` and assign it the results from the `reverse()` function call.
- 6) Create a string variable named `header` and build a string that matches the output in Exhibit A, item 7 (hint: use the variables you created in items 3-5 to build the string)
- 7) Assign the `header` variable to the `assign-results-header` div using the `innerHTML` property. See Exhibit B.

Exhibit B. assign-results-header

Date: 2/16/2021

Original Phrase: racecar


Reversed Phrase: racecar

Phrase Length: 7

- 8) Using an if...else statement, call the isPalindrome() function by passing-in the txtPhrase variable. Make sure you append the txtPhrase variable with JavaScripts toLowerCase() function. If true, assign the string message in Exhibit C. to the assignResults div's innerHTML.

Exhibit C. Matching palindrome


racecar is a palindrome!



- 9) For false, assign the string message in Exhibit D. to the assignResults div's innerHTML.

Exhibit D. Non-matching palindrome

richard is not a palindrome!



7. div with an id of assign-results-header
8. div with an id of assign-results and a CSS class of assign-results-text