

Modify the wordle.html webpage to produce a working (primitive) version of the game Wordle.

Note: your grade will be based on how well you carry out the specific instructions below, so make sure to follow these carefully.

1. At the top of your HTML file, include an HTML comment that includes your name, description and due date [3 points]
2. Generate a description of the game on the main page. Your description must include the following:
 - a. One or more words in bold. [4 points]
 - b. A working link to the original Wordle game: [4 points]
<https://www.powerlanguage.co.uk/wordle/>
 - c. An unordered list containing a description *and* demonstration of the color-coding results. To color-code the answers, you should create 3 CSS classes, one for correct letters, one for letters not in the right position, and one for letters that are not in the word (I will refer to these as *correct*, *almost*, and *wrong*). You may use colors/styles of your choosing, but if you want to match the “official” colors, you should use hexadecimal values #6aaa64, #c9b458, and #787c7e which correspond to correct, almost, and wrong guesses. The official style also uses a border of 2px of the same color. [12 points]

Directions: Guess the 5 letter word. After submitting a guess, the characters will be color-coded as follows:

- the letter is in the correct position
- the letter is in the wrong position
- the letter is not in the word

Note that your guesses do not have to be real words. For more fun, play Wordle at <https://www.powerlanguage.co.uk/wordle/>

3. Give your page the title “Wordle”, using the title tag. [3 points]
4. Use the appropriate CSS pseudo-class so that hovering over a link will change its font size to 110%. [3 points]

5. Add a navbar heading with two links, *Wordle* and *Dark Mode* (you may use a different name/theme if you'd like). Use **jQuery** so that clicking on "Dark Mode" does the following: [15 points]
- Changes the background color of the page to a color of your choice (such as "black" for a dark theme), and also changes the text color of your description.
 - Changes the "Dark Mode" text to say "Light Mode" (or an appropriate name). Clicking on "Light Mode" will return to the original "Dark Mode" style.
 - You may change other styles as well, but note that all text must be readable in either mode.
6. Inside of the form, add 5 textboxes, a submit button, and an input button. The 5 textboxes should occupy a single row, with appropriate space between them. The Submit and Reset buttons should be in a second row. There are several ways to do this (including some, such as using *grid templates*, which we have not discussed). You may use any approach that works for you. Note also that the form's *onsubmit* is set to "return false;", which stops the form submission; otherwise, the values will be submitted and the form will be reset, which is not what we want here. The textboxes should only allow the user to enter a single character. [12 points]



The image shows a visual representation of the form structure required in question 6. It consists of a single row of five empty text input boxes, each with a thin border. Below this row, there are two buttons: a 'Submit' button and a 'Reset' button, both with a light blue background and rounded corners.

7. The textboxes should always display the character entered in upper case. This can be accomplished by using appropriate CSS to change how the characters are *displayed*, or by using JavaScript to convert a lowercase character to an uppercase character. [3 points]
8. When the user clicks the Submit button, the text boxes and "keyboard" buttons are colored appropriately, by using JavaScript to add/remove the appropriate CSS classes (developed in question 1), based on whether the character is *correct*, *almost*, or *wrong*. Note: if you are using CSS to display the value of a text input in upper case, the *value* of the text input can still be in lowercase, and will need to be converted. **Hints:** If a textbox does not have a *value*, it will be *undefined*, which your logic should account for (otherwise you will get an error). Inside of a *for* loop, you can use the *continue* statement to advance to the next loop iteration. **Note regarding doubles:** if the user guesses the same character twice, and only one of them is correct, you may mark the other character as *almost* instead of as *wrong*. Ideally, the other character would be marked *wrong*, but this is more difficult, so will be optional. [12 points]

F

L

A

N

K

Submit

Reset

Q W E R T Y U I O P

A S D F G H J K L

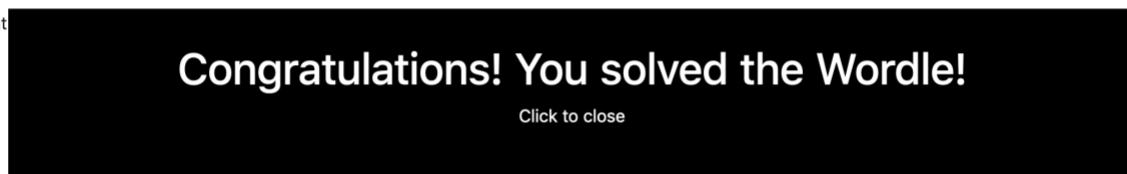
Z X C V B N M

9. When the input of a textbox changes, clear the color coding of that text box. Note that this can be done by listening to the *onchange* event (which is only triggered if the input changes and loses focus), or by listening to the *oninput* event (which is triggered as soon as the input is changed). [4 points]
10. Create a *div* that has a congratulatory message that will be shown if the user guesses the word. By default, this div should be hidden. If the user's guess is correct, the div should be displayed in a *fixed* position somewhere on the page. The user removes the div by clicking on it. Note: if you are unable to determine whether or not the user's guess is correct, then add a button on the page that will display this div when clicked. [12 points]

reactions: Guess the 5 letter word. After submitting a guess, the characters will be color-coded as follows:

- the letter is in the **correct position**
- the letter is in the **wrong position**
- the letter is **not in the word**

ote that



11. When the user clicks the *reset* button, the following happens (using **jQuery**): [8 points]
 - a. the text boxes are cleared (this should happen automatically as long as you are using a *reset* input inside of the form)
 - b. the color coding for *correct*, *almost*, and *wrong* answers are cleared from both the textboxes and the keyboard. (if you are unable to get the color codes for the textboxes and keyboards, you may clear the color codes from your description).

12. Include an “I give up” button somewhere on the page. Clicking on the button will display an alert with the answer. [5 points]

Extra Credit

Require that the user enters the characters in order, by disabling and enabling the text boxes appropriately. Also allow the user to select characters by clicking the buttons on the keyboard.