**Step 6 of 6**

**Play It Again!**

You made it to the final part of your project. Woo hoo!

In this part, you’ll create a function to hide elements, like the Guess button, when the game ends. You’ll also use the function to show the Play Again button.

You’ll then write a click event for the Play Again button to display a new word and restart the guesses back to 8 (or whichever number you decided on).



*When the game ends, the Play Again button prompts them to play again. The number of previous guesses and letters guessed disappears.*

Finally, you’ll host your completed game on GitHub Pages so you can share your project with the world! We’ll provide steps for hosting on GH Pages, but feel free to give yourself a refresh by visiting the [GitHub Pages](https://learn.skillcrush.com/module-11/hosting-with-github-pages/) step from your Working with Git & GitHub class.

**What to Do:**

**Create a Function to Hide and Show Elements**

1. At the bottom of the script.js file, create a function called startOver to hide:
   1. the Guess button.
   2. the paragraph where the remaining guesses will display.
   3. the unordered list where the guessed letters appear.
2. Use the startOver function to show the button to play again.
3. Call the startOver function when the game is over whether the player wins or loses. Test the game to be sure the Guess button, the paragraph with remaining guesses, and the guessed letters disappear when the player wins or loses. Also, check that the Play Again button appears so that players give the game another try. The Play Again button won’t do anything quite yet, but you’ll get to that next!

**Add a Click Event to the Play Again Button**

1. Add a click event listener for the Play Again button. Remove the class of “win” applied to the message element. Empty the message text and the unordered list where the guessed letters appear.
2. Set the remaining guess back to 8 or whichever number of guesses you decided on.  Set your guessedLetter global variable back to an empty array. Populate the text of the span inside the paragraph where the remaining guesses display with the new amount of guesses.
3. Show the Guess button, the paragraph with remaining guesses, and the guessed letters once more. Hide the Play Again button.
4. Call the getWord() async function that pulls the new word so the player can play again!
5. Test out the game to make sure the click event is working. Congratulations, you’ve programmed a computer game!  Go ahead and add and commit the changes to your repo.

**Host Your Project with GitHub Pages**

1. Now that your game is all finished, it’s time to host this baby! Start by creating a branch called gh-pages:  git checkout -b gh-pages. You must call the branch gh-pages so GitHub will automatically turn on the GitHub Pages for this branch.
2. Push your gh-pages branch to GitHub using the git push origin gh-pages command.
3. Go to GitHub and visit the repo you set up in the last step.
4. In the top menu of your repo, click the Settings button, which has a gear icon.
5. On the Settings page, scroll down to the GitHub Pages section. You should see a green section with a message that says, “Your site is published at https://username.github.io/guess-the-word/.” If you named your repository something else, then substitute the “guess-the-word” portion of the URL with the name of your repo.
6. Copy and paste the link to your game on GitHub Pages below. Yay, you finished your project!  Look at you go!

[Challenge Solution(https://github.com/skillcrush/guess-the-word/tree/v05)](https://github.com/skillcrush/guess-the-word/tree/v05)