**Step 2 of 6**

**Select Elements & Add Placeholders**

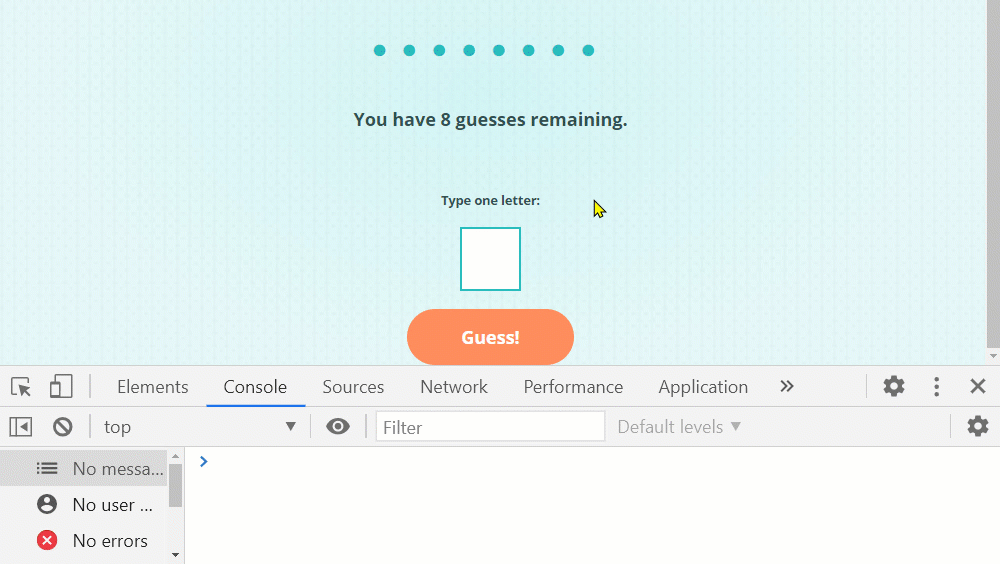
Welcome to the first part of your Guess the Word game project!

Below, you’ll download the starter files for the project. You’ll notice the starter code has a .gitignore file. As you may remember, the [.gitignore file](https://learn.skillcrush.com/module-10/contributing-to-team-projects/#gitignore) prevents Git from tracking specific files, like Thumbs.db (PC) or .DS\_Store (Mac), that you don’t want under version control.

Sometimes in the instructions, we’ll recommend variable and function names. However, other times you’ll be in charge of naming the variables and functions, so you get practice choosing descriptive names (be sure to avoid using [reserved keywords(opens in a new tab)](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Lexical_grammar#keywords) in your names). Our solution code will look a little different from yours because of the different variable and function names.

At the end of this step, you’ll have selected all the HTML elements you’ll need to target using querySelector(). Hint: Add comments above each element you select to help you remember what you’re targeting!

You’ll also program placeholders to represent each letter in the word and add an event listener for the Guess button. Letters guessed by the player won’t appear on the screen yet, but you’ll be able to check the input is working with the console.

[](https://s3.amazonaws.com/media.skillcrush.com/skillcrush/wp-content/uploads/2023/03/102-L12-challenge1-letter-console-2.gif)

*When the player enters and submits a letter, the letter will appear in the console (for now).*

**Pro-Tip!** Make sure to have your [JavaScript Arrays Cheatsheet](https://learn.skillcrush.com/module-7/javascript-arrays-cheatsheet/) handy! You’ll be referencing it throughout the project.

**What to Do:**

**Create Global Variables**

1. Download the [starter code for the project(opens in a new tab)](https://github.com/skillcrush/guess-the-word/archive/refs/heads/main.zip).
2. Open the directory in your text editor. Then, navigate to the script.js file.
3. Create global variables to select the following elements:
   1. The unordered list where the player’s guessed letters will appear.
   2. The button with the text “Guess!” in it.
   3. The text input where the player will guess a letter.
   4. The empty paragraph where the word in progress will appear.
   5. The paragraph where the remaining guesses will display.
   6. The span inside the paragraph where the remaining guesses will display.
   7. The empty paragraph where messages will appear when the player guesses a letter.
   8. The hidden button that will appear prompting the player to play again.
4. Create another global variable called word and give it the value of "magnolia". Magnolia is your starting word to test out the game until you fetch words from a hosted file in a later step.

**Write a Function to Add Placeholders for Each Letter**

1. Create and name a function to update the paragraph’s innerText for the “words-in-progress” element with circle symbols (●) to represent each letter in the word. The symbols will stay on the screen until the correct letter is guessed (in a future step). Hint: Copy and paste the ● symbol into your code!
2. Call the function and pass it the word variable as the argument. You should see 8 circle symbols on the screen, one for each letter in the word “magnolia.” Hint: You’ll need to use an array and then join it back to a string using the .join("") method.

**Add an Event Listener for the Button**

1. Add an event listener for when a player clicks the Guess button. In the callback function, add a parameter for the event: e.
2. Because you’re working with a form, you want to prevent the default behavior of clicking a button, the form submitting, and then reloading the page. To prevent this reloading behavior, add this line of code at the top of the callback function: e.preventDefault();.
3. Create and name a variable to capture the value of the input. Log out the value of the variable capturing the input. Then, empty the value of the input. You should see the letter you enter into the input field in the console when the Guess button is clicked.
4. In the command line, add and commit your changes. Push the changes up to GitHub. Copy the link to your repo and submit it below. Part 1 of the project is done!

Solution Code: <https://github.com/skillcrush/guess-the-word/tree/v01>