**Step 5 of 6**

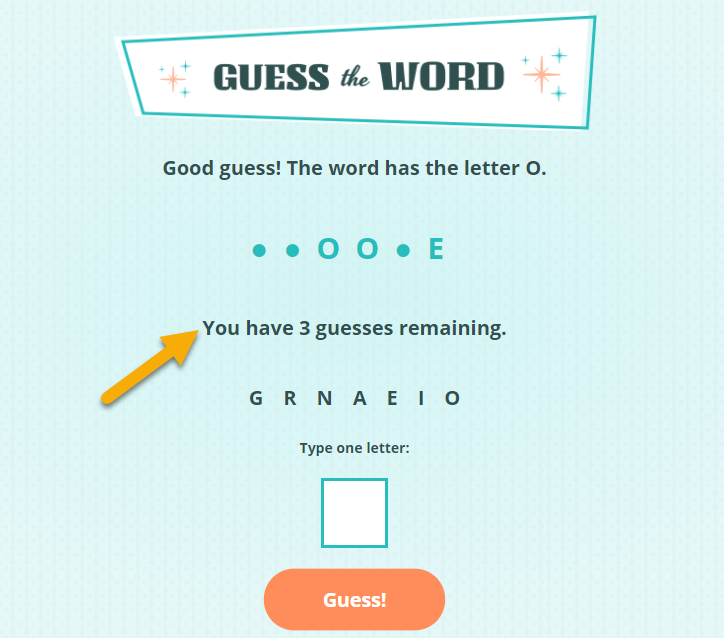
**Fetch Words & Remaining Guesses**

Welcome to part 4 of your Guess the Word project! In this part, you’ll create a function to count and monitor the player’s remaining guesses. The function will display a message indicating the remaining guesses, including no more guesses if the player runs out!

You’ll also create an async function to upgrade your game from displaying one word to fetching a random word from a file of over 800 words. The file you’re pulling from is a text (.txt) file. To work with text files, as opposed to JSON, you’ll need a few more tricks for choosing a word.

First, you’ll need to split the words from the text file into an array with split() and use the newlines (AKA line breaks) between each word to separate them. To use the newlines, you’ll include a **delimiter**—a character to separate words—in your split() method. In JavaScript, newlines are represented by the n character.

You’ll also need one more trick for working with this text file: trim(). The **trim() method** allows you to remove any extra whitespace before or after each word, like a tab or space. Extra whitespace is common when working with text files. You don’t want extra spaces because you could end up with empty characters in the word, and there’s no way your players would be able to guess an empty character.  We’ll guide you on when to place the delimiter and use trim() in your code!



*The game now displays how many remaining guesses the player has. Plus, players will receive one of over 800 words to guess!*

Finally, you’ll call the async function and test out the game. With so many possible words, you’ll be able to play along with the game now too!

**What to Do:**

**Declare a Global Variable for the Number of Guesses**

1. Create a global variable called remainingGuesses and set it to a value of 8. The value 8 is the maximum number of guesses the player can make. You can decrease or increase this value to make the game harder or easier for the player! Hint: The value of the remainingGuesses variable will change over time.

**Create a Function to Count Guesses Remaining**

1. Create and name a new function that will accept the guess input as a parameter. In the code, place this function before the function that checks if the player won.
2. In the function, grab the word and make it uppercase. Because the player’s guess is uppercase, making the word they’re guessing uppercase will compare letters with the same casing.
3. Find out if the word contains the guess. If it doesn’t include the letter from guess, let the player know that the word doesn’t contain the letter and subtract 1 from their remainingGuesses. If it does contain a letter, let the player know the letter is in the word.
4. Still in the function and below the conditional statement, determine if the remainingGuesses is a value of 0. If they have no guesses remaining, update the message to say the game is over and what the word is. If they have 1 guess, update the span inside the paragraph where the remaining guesses will display to tell the player they have one guess remaining. If they have more than one guess, update the same span element to tell them the number of guesses remaining.
5. In the else clause of your makeGuess function, before the call to the function that will update the word in progress, call your new function to update the remaining guesses and pass it the letter that the player guessed as an argument.
6. Play the game for a few guesses. You should see the number of remaining guesses update on the page. Remember, the number of guesses will only update when you make an incorrect guess.

**Add an Async Function**

1. Near the top of your file, under the word, guessedLetters, and remainingGuesses global variables, add an async function called getWord() to fetch data from a file at this address: “https://gist.githubusercontent.com/skillcrush-curriculum/7061f1d4d3d5bfe47efbfbcfe42bf57e/raw/5ffc447694486e7dea686f34a6c085ae371b43fe/words.txt”. Hint: You also retrieved data from a file in the [school field trip exercise](https://learn.skillcrush.com/module-7/practice-exercises-intro-to-apis/#exercise3) in a previous lesson. The difference here is that you’re fetching data from a text file instead of a JSON file. In the second await statement, use .text() instead of .json().
2. Log out the result of the second await statement to see what data you retrieved! Don’t forget you’ll need to call getWord() in order to view the result in the console.
3. You know how to grab a random element from an array, now you’ll grab a random word. To select a random word, you’ll need first to transform the data you fetched into an array. Each word is separated by a newline (line break), so this is the delimiter you’ll use to create the array: const wordArray = words.split("\n");. Log out your wordArray to see the data.
4. To grab a random word from the file, create a variable to pull a random index from the wordArray. Hint: You wrote similar code when you [pulled a random image](https://learn.skillcrush.com/module-7/reveal-a-random-image/#random-index) in a previous challenge.
5. Still in the function, pull out a random word from the array and remove any extra whitespace around the word using the trim() method. Reassign the value of the existing word global variable to this new random word. This means you should also now declare the global word variable with let instead of const.
6. Call the placeholder function you created previously at the bottom of the function. Pass it in the variable holding your random (and freshly trimmed) word!

**Call the New Function & Test the Game**

1. Take placeholder(word) from your code’s global space and place it at the bottom of getWord(). In the location the call to placeholder(word) used to be, call getWord() instead.
2. Test out the game to ensure a new, random word is displaying on the screen. Now this a REAL game that you can play too!
3. When done playing the game, add and commit your changes in the command line. The major part of the project is done!

[Challenge Solution(https://github.com/skillcrush/guess-the-word/tree/v04)](https://github.com/skillcrush/guess-the-word/tree/v04)