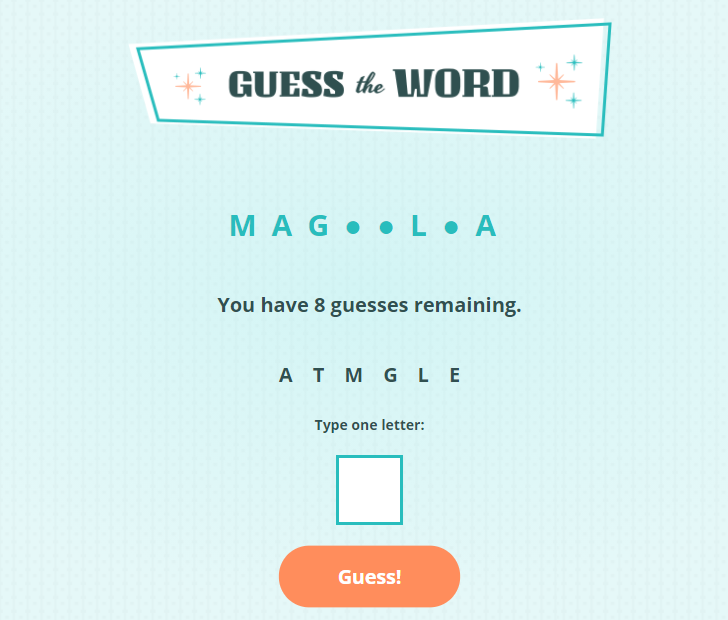
**Step 4 of 6**

**Display Word & Guessed Letters**

So far, you’ve been testing the letter input in your console. Now it’s time to display those letters on the screen! You’ll start by creating a function to display the guessed letters.

Then, you’ll create another function to update the word in progress when the player guesses correctly. This function will use two new-to-you methods: split() and join(). You’ll use the **split() method** to split the string (i.e., the letters the player inputs) so that it can return as an array. You’ll also use the **join() method** to join all the array elements into a string. Because these are new methods for you, we’ll let you know the best time to use them in your code!

Finally, you’ll create a function to check if the player successfully guessed the word and won the game!



*The letters the player guesses will now display on the screen instead of just the console!*

**What to Do:**

**Create a Function to Show the Guessed Letters**

1. Create and name a function to update the page with the letters the player guesses (see screenshot above).
2. Empty the innerHTML of the unordered list where the player’s guessed letters will display.
3. Create a new list item for each letter inside your guessedLetters array (i.e., the global variable) and add it to the unordered list.
4. Call the function inside the else statement of the makeGuess function so the letter displays when it hasn’t been guessed before.
5. Test it out! You should see each unique guessed letter show up on the screen when you hit the Guess button.

**Create a Function to Update the Word in Progress**

1. Create and name a function to update the word in progress that accepts the guessedLetters array as a parameter. This function will replace the circle symbols with the correct letters guessed.
2. Create a variable called wordUpper to change the word variable to uppercase. On the following line, create a variable to split the word string into an array so that the letter can appear in the guessedLetters array: const wordArray = wordUpper.split("");. Then, log out wordArray to see what this does!
3. Check if the wordArray contains any letters from the guessedLetters array. If it does contain any of the letters, update the circle symbol with the correct letter. Hint: You’ll want to create a new array with the updated characters and then use join() to update the empty paragraph where the word in progress will appear.
4. Call your new shiny new function at the bottom of the else statement inside the makeGuess function and pass it guessedLetters as an argument.
5. Give it a go! Try guessing a few of the correct letters from your test word “magnolia.” You should see the letters show up instead of the circle (●) symbol.

**Create a Function to Check If the Player Won**

1. Create and name a function to check if the player successfully guessed the word and won the game. Begin by verifying if their word in progress matches the word they should guess.
2. If the player has won, add the “win” class to the empty paragraph where messages appear when they guess the letter. Also, update the paragraph’s contents to: <p class="highlight">You guessed correct the word! Congrats!</p>.
3. At the bottom of the function that updates the word in progress, call this function to check if the player has won.
4. Play the game to make sure the guessed letters are displaying on the screen. When all the corrected letters are guessed, you should see the congratulatory message.
5. 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. Excellent job, Skillcrusher!

[Challenge Solution(https://github.com/skillcrush/guess-the-word/tree/v03)](https://github.com/skillcrush/guess-the-word/tree/v03)