**Schritt 5 von 6**

**Wörter abrufen und verbleibende Versuche**

Willkommen zu Teil 4 deines Projekts "Wort erraten"! In diesem Teil erstellst du eine Funktion, um die verbleibenden Versuche des Spielers zu zählen und zu überwachen. Die Funktion zeigt eine Meldung an, die die verbleibenden Versuche angibt, einschließlich "keine Versuche mehr", wenn dem Spieler die Versuche ausgehen!

Du erstellst auch eine asynchrone Funktion, um dein Spiel vom Anzeigen eines Wortes auf das Abrufen eines zufälligen Wortes aus einer Datei mit über 800 Wörtern zu aktualisieren. Die Datei, aus der du ziehst, ist eine Textdatei (.txt). Um mit Textdateien im Gegensatz zu JSON zu arbeiten, benötigst du ein paar weitere Tricks zur Auswahl eines Wortes.

Zuerst musst du die Wörter aus der Textdatei mit split() in ein Array aufteilen und die Zeilenumbrüche (AKA Zeilenumbrüche) zwischen den einzelnen Wörtern verwenden, um sie zu trennen. Um die Zeilenumbrüche zu verwenden, fügst du ein Trennzeichen - ein Zeichen zum Trennen von Wörtern - in deine split()-Methode ein. In JavaScript werden Zeilenumbrüche durch das Zeichen n dargestellt.

Du benötigst auch noch einen weiteren Trick, um mit dieser Textdatei zu arbeiten: trim(). Die trim()-Methode ermöglicht es dir, zusätzliche Leerzeichen vor oder nach jedem Wort zu entfernen, wie z. B. einen Tabulator oder ein Leerzeichen. Zusätzliche Leerzeichen sind bei der Arbeit mit Textdateien üblich. Du möchtest keine zusätzlichen Leerzeichen, da du sonst leere Zeichen im Wort erhalten könntest, und es gibt keine Möglichkeit, dass deine Spieler ein leeres Zeichen erraten könnten. Wir werden dich anleiten, wann du das Trennzeichen setzen und trim() in deinem Code verwenden musst!

Das Spiel zeigt nun an, wie viele Versuche der Spieler noch hat. Außerdem erhalten die Spieler eines von über 800 Wörtern zum Erraten!

Schließlich rufst du die asynchrone Funktion auf und testest das Spiel. Mit so vielen möglichen Wörtern kannst du jetzt auch am Spiel teilnehmen!

**Was zu tun ist:**

**Deklariere eine globale Variable für die Anzahl der Versuche**

**1.** Erstelle eine globale Variable namens remainingGuesses und setze sie auf den Wert 8. Der Wert 8 ist die maximale Anzahl von Versuchen, die der Spieler unternehmen kann. Du kannst diesen Wert verringern oder erhöhen, um das Spiel für den Spieler schwieriger oder einfacher zu machen! Hinweis: Der Wert der Variable remainingGuesses ändert sich im Laufe der Zeit.

**Erstelle eine Funktion zum Zählen der verbleibenden Versuche**

1. Erstelle eine neue Funktion und gib ihr einen Namen, die die Eingabe guess als Parameter akzeptiert. Platziere diese Funktion im Code vor der Funktion, die prüft, ob der Spieler gewonnen hat.

2. Nimm in der Funktion das Wort und mache es groß. Da die Vermutung des Spielers großgeschrieben ist, werden durch die Großschreibung des Wortes, das er errät, Buchstaben mit der gleichen Groß- und Kleinschreibung verglichen.

3. Finde heraus, ob das Wort die Vermutung enthält. Wenn es den Buchstaben aus der Vermutung nicht enthält, teile dem Spieler mit, dass das Wort den Buchstaben nicht enthält, und subtrahiere 1 von seinen verbleibenden Vermutungen. Wenn es einen Buchstaben enthält, teile dem Spieler mit, dass sich der Buchstabe im Wort befindet.

4. Bestimme in der Funktion und unterhalb der bedingten Anweisung, ob die remainingGuesses einen Wert von 0 haben. Wenn keine Versuche mehr übrig sind, aktualisiere die Meldung, um zu sagen, dass das Spiel vorbei ist und welches Wort es ist. Wenn er noch 1 Versuch hat, aktualisiere den Span innerhalb des Absatzes, in dem die verbleibenden Versuche angezeigt werden, um dem Spieler mitzuteilen, dass er noch einen Versuch hat. Wenn er mehr als einen Versuch hat, aktualisiere dasselbe Span-Element, um ihm die Anzahl der verbleibenden Versuche mitzuteilen.

5. Rufe im Else-Zweig deiner MakeGuess-Funktion, vor dem Aufruf der Funktion, die das Wort im Fortschritt aktualisiert, deine neue Funktion auf, um die verbleibenden Versuche zu aktualisieren, und übergib ihr den Buchstaben, den der Spieler als Argument erraten hat.

6. Spiele das Spiel für ein paar Versuche. Du solltest sehen, dass sich die Anzahl der verbleibenden Versuche auf der Seite aktualisiert. Denke daran, dass sich die Anzahl der Versuche nur aktualisiert, wenn du eine falsche Vermutung abgibst.

**Füge eine Async-Funktion hinzu**

1. Füge am oberen Rand deiner Datei, unter den globalen Variablen word, guessedLetters und remainingGuesses, eine Async-Funktion namens getWord() hinzu, um Daten aus einer Datei unter dieser Adresse abzurufen: "<https://gist.githubusercontent.com/skillcrush-curriculum/7061f1d4d3d5bfe47efbfbcfe42bf57e/raw/5ffc447694486e7dea686f34a6c085ae371b43fe/words.txt>". Hinweis: Du hast auch Daten aus einer Datei in der Schulausflug-Übung in einer vorherigen Lektion abgerufen. Der Unterschied hier ist, dass du Daten aus einer Textdatei anstelle einer JSON-Datei abrufst. Verwende in der zweiten Await-Anweisung .text() anstelle von .json().

2. Gib das Ergebnis der zweiten Await-Anweisung aus, um zu sehen, welche Daten du abgerufen hast! Vergiss nicht, dass du getWord() aufrufen musst, um das Ergebnis in der Konsole anzuzeigen.

3. Du weißt, wie man ein zufälliges Element aus einem Array abruft, jetzt rufst du ein zufälliges Wort ab. Um ein zufälliges Wort auszuwählen, musst du zuerst die abgerufenen Daten in ein Array umwandeln. Jedes Wort ist durch einen Zeilenumbruch (Zeilenumbruch) getrennt, daher ist dies das Trennzeichen, das du zum Erstellen des Arrays verwendest: const wordArray = words.split("\n");. Gib dein wordArray aus, um die Daten anzuzeigen.

4. Um ein zufälliges Wort aus der Datei abzurufen, erstelle eine Variable, um einen zufälligen Index aus dem wordArray abzurufen. Hinweis: Du hast ähnlichen Code geschrieben, als du in einer vorherigen Aufgabe ein zufälliges Bild abgerufen hast.

5. Rufe immer noch in der Funktion ein zufälliges Wort aus dem Array ab und entferne alle zusätzlichen Leerzeichen um das Wort herum mit der Trim()-Methode. Weisen Sie der vorhandenen globalen Variablen word diesen neuen zufälligen Wert zu. Das bedeutet, dass Sie die globale Variable word jetzt auch mit let anstelle von const deklarieren sollten.

6. Rufe die Platzhalterfunktion, die du zuvor erstellt hast, am Ende der Funktion auf. Gib die Variable ein, die dein zufälliges (und frisch getrimmtes) Wort enthält!

**Rufe die neue Funktion auf und teste das Spiel**  
1. Nimm placeholder(word) aus dem globalen Bereich deines Codes und platziere sie am Ende von getWord(). An der Stelle, an der sich der Aufruf von placeholder(word) befand, rufe stattdessen getWord() auf.

2. Teste das Spiel, um sicherzustellen, dass ein neues, zufälliges Wort auf dem Bildschirm angezeigt wird. Jetzt ist dies ein ECHTES Spiel, das du auch spielen kannst!

3. Wenn du mit dem Spielen fertig bist, füge deine Änderungen in der Befehlszeile hinzu und bestätige sie. Der Hauptteil des Projekts ist abgeschlossen!

Challenge-Lösung(<https://github.com/skillcrush/guess-the-word/tree/v04>)