Schritt 3 von 6  
**Spielereingaben akzeptieren & validieren**  
In diesem Teil des Projekts schreiben Sie zwei neue Funktionen und validieren die Eingabe des Spielers, um sicherzustellen, dass er nur einen einzelnen Buchstaben eingibt.

Die erste Funktion, die Sie erstellen, prüft, ob die Eingabe des Spielers ein Buchstabe ist. Dazu verwenden Sie einen regulären Ausdruck. Ein regulärer Ausdruck ermöglicht es Ihnen, Text zu finden, der einem bestimmten Muster entspricht, wie z. B. nur alphabetischer Text (keine Zahl oder Symbole). Da dies Ihr erstes Mal ist, dass Sie mit einem regulären Ausdruck arbeiten, geben wir Ihnen die Codezeile, die Sie benötigen, um sicherzustellen, dass Ihr Spieler nur Buchstaben eingibt. Wenn Sie mehr über reguläre Ausdrücke lesen möchten, besuchen Sie die Seite Reguläre Ausdrücke von MDN (<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular_expressions>).

Es gibt noch ein weiteres neues Code-Element, mit dem Sie arbeiten werden: match(). Die Methode match() arbeitet mit dem regulären Ausdruck zusammen, um die Zeichenketten (d. h. den Buchstaben, den der Spieler eingibt) zu durchsuchen, um sie mit dem regulären Ausdruck abzugleichen. Wir geben Ihnen einen Hinweis, wo Sie diese Methode in Ihrem Projekt verwenden können.

Sie kehren dann zu dem Klick-Ereignis zurück, das Sie im letzten Schritt erstellt haben, um die Eingabe des Spielers zu validieren, um sicherzustellen, dass er nur einen einzigen Buchstaben hinzugefügt hat. Schließlich schreiben Sie eine zweite Funktion, die die Vermutung des Spielers erfasst, um zu sehen, ob er diesen Buchstaben bereits erraten hat. Wenn nicht, schiebt die Funktion den Buchstaben in ein Array von erratenen Buchstaben.  
Wenn Sie fertig sind, testen Sie Ihr Spiel, um sicherzustellen, dass das Spiel nur einzelne Buchstaben als Vermutung akzeptiert. Sie stellen auch sicher, dass eine Meldung auf dem Bildschirm erscheint, wenn der Benutzer denselben Buchstaben mehr als einmal errät.

Wenn ein Spieler denselben Buchstaben zweimal errät oder ein nicht-alphabetisches Zeichen eingibt, erscheint eine Meldung, die ihn darüber informiert, dass er den Buchstaben bereits erraten hat oder einen Buchstaben von A bis Z eingeben soll.

Was zu tun ist:

**Erstellen Sie eine Funktion, um die Eingabe des Spielers zu überprüfen**

Erstellen und benennen Sie eine Funktion, die den Eingabewert als Parameter akzeptiert. Der Zweck dieser Funktion ist es, die Eingabe des Spielers zu validieren.

Erstellen Sie innerhalb der Funktion eine Variable für die akzeptierte Buchstabenfolge: const acceptedLetter = /[a-zA-Z]/. Jetzt verwendet Ihr Code einen regulären Ausdruck, um sicherzustellen, dass der Spieler einen Buchstaben eingibt!

Verwenden Sie immer noch innerhalb der Funktion einen bedingten Block, um verschiedene Szenarien zu überprüfen. Überprüfen Sie zuerst, ob die Eingabe leer ist. Überprüfen Sie dann, ob der Spieler mehr als einen Buchstaben eingegeben hat. Überprüfen Sie schließlich, ob er ein Zeichen eingegeben hat, das nicht mit dem regulären Ausdrucksmuster übereinstimmt. Hinweis: Sie benötigen hier die Methode .match(). Jede Bedingung sollte eine Meldung enthalten, die den Spieler anweist, was er eingeben soll.

Wenn alle anderen Bedingungen nicht erfüllt sind, ist die Eingabe ein Buchstabe, was Sie suchen! Geben Sie die Eingabe zurück.

**Validieren Sie die Eingabe im Button Event Handler**

Leeren Sie innerhalb der Event-Handler-Funktion für den Guess-Button den Text des Nachrichtenelements.

Rufen Sie am Ende des Event Handlers die Funktion auf, die Sie erstellt haben, um die Eingabe zu überprüfen, und übergeben Sie ihr den Eingabewert als Argument. Speichern Sie das Ergebnis dieses Funktionsaufrufs in einer Variablen und protokollieren Sie es in der Konsole.

Verwenden Sie die Konsole, um die Eingabe zu überprüfen. Geben Sie ein anderes Zeichen als einen Buchstaben in die Eingabe ein. Beachten Sie, wie sich die Meldung auf dem Bildschirm ändert!

**Fügen Sie eine neue globale Variable für Spielerratungen hinzu**

Erstellen Sie unter der Variablen word, nahe dem Anfang des Codes, eine weitere globale Variable namens guessedLetters mit einem leeren Array. Dieses Array enthält alle Buchstaben, die der Spieler errät.

**Erstellen Sie eine Funktion zum Erfassen von Eingaben**

Erstellen Sie unterhalb der Funktion, die Eingaben überprüft, eine neue Funktion namens makeGuess, die einen Buchstaben als Parameter akzeptiert.

JavaScript unterscheidet zwischen Groß- und Kleinschreibung, sodass Groß- und Kleinbuchstaben als unterschiedliche Zeichen angesehen werden. Der einfachste Weg, mit der Unterscheidung zwischen Groß- und Kleinschreibung umzugehen, ist, alle Buchstaben in eine einzige Schreibweise umzuwandeln. Wir empfehlen, Ihren Buchstabenparameter in Großbuchstaben umzuwandeln. Sobald der Buchstabe in Großbuchstaben umgewandelt wurde, überprüfen Sie, ob Ihr guessedLetters-Array diesen Buchstaben bereits enthält.

Wenn der Spieler bereits denselben Buchstaben erraten hat, aktualisieren Sie die Meldung, um den Spieler darüber zu informieren, dass er diesen Buchstaben bereits erraten hat und es erneut versuchen soll. Wenn er diesen Buchstaben noch nicht zuvor erraten hat, fügen Sie den Buchstaben dem guessedLetters-Array hinzu.

Protokollieren Sie das guessedLetters-Array in der Konsole.

Kehren Sie zum Event Handler für den Guess-Button zurück. Stellen Sie sicher, dass die Variable, die dem Ergebnis der Funktion zugeordnet ist, überprüft, ob die Eingabe des Spielers einen Buchstaben zurückgibt (im Gegensatz zu "undefiniert"). Wenn es einen Buchstaben zurückgibt, übergeben Sie ihn als Argument an Ihre makeGuess-Funktion.

Versuchen Sie ein paar Buchstabenvermutungen im Browserfenster. Stellen Sie sicher, dass Sie sehen, wie sich der Inhalt des guessedLetters-Arrays ändert, wenn Sie neue Buchstaben eingeben und auf den Button klicken.

Fügen Sie Ihre Änderungen mit der Befehlszeile hinzu und übernehmen Sie sie. Pushen Sie die Änderungen auf GitHub. Kopieren Sie den Link zu Ihrem Repo und senden Sie ihn unten ein. Teil 2 Ihres Projekts ist fertig!

Challenge-Lösung (<https://github.com/skillcrush/guess-the-word/tree/v02>)