

## Beschreibung des Ergebnisses

### Was war unser Ziel

Wir hatten uns vorgenommen einen Einheiten-Umrechner zu programmieren. Die Bereiche in denen umgerechnet werden soll haben wir auf Länge, Fläche, Volumen, Temperatur, Masse und Zeit festgelegt. Der Umrechner soll als Internet-Seite funktionieren, sozusagen als Webapp. Desweiteren hatten wir uns vorgenommen die Umrechnungsformeln für jede mögliche Umrechnung auszugeben und Fakten zu den einzelnen Bereichen auszugeben. Wir hatten uns unter anderem auch vorgenommen die Möglichkeit zu programmieren, zwischen Deutsch und Englisch zu wechseln. Das hatten wir aber von vornherein lediglich als „nice-to-have“ erwähnt und nicht umgesetzt bekommen. Es sollte beim ersten Mal starten der Seite eine Nachricht mit der Beschreibung der Funktionen auftauchen. Diesen Gedanken haben wir schnell wieder verworfen, da die Seite selbsterklärend ist. Für die Verwaltung der Fakten und Formeln hatten wir vor eine Sqlite3-Datenbank zu verwenden.

Für die einzelnen Seiten zu den Bereichen wollten wir je eine „embedded Javascript (ejs)“-Datei erstellen und mit einer gemeinsamen CSS-Datei gestalten. Die Verwaltung der Seiten wollten wir über eine Javascript-Datei, auf der auch die Module ihre Anwendung finden, erfolgen lassen. Folgende Module wollten wir verwenden: embedded Javascript (ejs), body-parser, express und sqlite3.

### Was waren unsere Problemstellen

Unser erstes Problem entstand beim auslesen aus der Datenbank, da wir nicht wussten wie wir die einzelnen Komponenten einer ausgelesenen Zeile zurückgeben können. Gelöst haben wir das Problem durch herumprobieren und informieren auf diversen Foren und Seiten bei Google.

Das nächste Problem erfolgte gleich im Anschluss bei der Ausgabe des richtigen Ergebnisses bei der Umrechnung. Hier wussten wir nicht wie man „Strings“ als Rechnung benutzen kann. Die Lösung hierfür war die „eval()“-Funktion.

Ein weiteres Problem hatte mit dem asynchronen Aufbau von Javascript-Funktionen zu tun. Die von uns programmierten Funktionen wurden zu spät ausgeführt. Die meist genutzte Lösung für dieses Problem wären Callback-Funktionen gewesen, allerdings haben wir es damit nicht gelöst bekommen. Die Alternative war der Verzicht auf Funktionen.

Unser letztes größeres Problem war das die Drop-Down Werte durch das neu laden der Seite auf ihre Standardwerte zurückgesetzt wurden. Gelöst haben wir dieses Problem durch eine if-Abfrage in dem jeweiligen „option-tag“. Diese fragt ab welche Einheit ausgewählt ist und setzt an dieser Stelle dann ein „selected“.

### Was ist unser Endergebnis

Unser Endergebnis kommt dem was wir uns vorgenommen haben schon sehr nahe, bis auf die oben erwähnten Ausnahmen. Desweiteren haben wir noch eine Funktion hinzugefügt, mit der die Farbe der Navigationsleiste und der Überschriften gewechselt werden kann. Allerdings haben wir auch hier das Problem mit dem neu laden der Seite. Hierfür haben wir noch keine Lösung. Den größten Aufwand hatten wir beim abtippen aller Formeln und Fakten in die Datenbank.