



Webentwicklung Grundkurs Teil 4

Description

Das heutige Internet ist ohne JavaScript kaum vorstellbar. Ab diesen Teil schauen wir uns genauer an, wie diese Skriptsprache funktioniert und welche Möglichkeiten sie bietet.

Vorwort

Viele Tutorials und vor allem Bücher über JavaScript versuchen einen umfassenden Einstieg zu bieten oder ein sehr spezielles Thema zu behandeln. Im Rahmen dieses Grundkurses ist dies nicht möglich, da ich sonst über JS, PHP, CSS3 und HTML mehrere Bücher füllen müsste. Es geht viel mehr darum, ein grundlegendes Verständnis zu vermitteln. Ein Teil davon betrifft Themen, die in Tutorials und Büchern oft zu kurz kommen, ein anderer Teil würde sich dem Inhalt anschließen.

Das bedeutet: Ich werde hier nicht auf die Grundlagen der Programmierung eingehen und alles von A über B nach C erklären. Wer sich dafür interessiert, wie Schleifen, Abfragen, Arrays etc. funktionieren, findet auf dieser Seite bereits einige Tutorials. Die sind dann meist in GML, aber bis auf ein paar Kleinigkeiten in der Syntax ist das alles mehr oder weniger gleich.

Ich hoffe, dass ich euch dennoch für das Thema begeistern kann und euch die Serie dazu inspiriert, euch tiefer mit dem Thema zu befassen.

Der Anwendungsfall

Wie bereits in der Serie erwähnt, ist JavaScript eine Skriptsprache, die im Browser ausgeführt wird. Das bedeutet, dass der Seitenbesucher den ganzen Code herunterladen muss und dieser lokal ausgeführt wird. Für Entwickler und User hat das mehrere Vorteile. Der User muss nicht laufend neue Daten laden. So ist es bspw. möglich, eine komplette Anwendung im Browser lokal zu speichern. Das schont auch den Server etwas, weil nicht laufend darauf zugegriffen werden muss. Außerdem kann man mit JavaScript bereits viele Dinge prüfen, bevor zum Beispiel eingaben an den Server geschickt werden.

Kurz: JavaScript kann die Daten verarbeiten, bevor sie an den Server gelangen.

Im heutigen Beispiel wird es genau darum gehen. Ein simples Beispiel, bei dem ein User seinen Namen eingeben kann und er mit Klick auf den Button eine Begrüßung erhält. Ja, dass ist wirklich nicht die große Kunst, soll aber eine sehr wichtige Aufgabe von JavaScript zeigen.

Das Beispiel kann man sich [hier anschauen](#).

Dateistruktur

Wir brauchen lediglich einen Ordner, den wir *js* nennen. Darin legen wir eine Datei mit dem Namen **main.js** an. Im Hauptordner legen wir noch eine Datei mit **index.html** an. Das war es schon.

index.html

```
<!DOCTYPE html>
<html lang="de">
  <head>
    <!-- Ein Codekommentar in HTML -->
    <meta charset="utf-8">
    <title>Grundkurs Webentwicklung</title>
  </head>
  <body>
    <!-- Hier landet der Seiteninhalt -->
    <h1>Java-Script Test</h1>
    <input type="text" id="inputName" placeholder="Vorname">
    <button type="button" id="hello">Begrüßung</button>
    <script src="js/main.js"></script>
  </body>
</html>
```

Die **index.html** ist ähnlich aufgebaut wie sie [im zweiten Teil beschrieben wird](#). Im header- und im body-Bereich habe ich jeweils noch einen Kommentar eingebaut, wie er in HTML üblich ist. Ein Kommentar in HTML wird mit `<!--` gestartet und mit `-->` beendet. In JavaScript reicht ein `//` aus, um eine Zeile zu kommentieren. Das werden wir auch gleich sehen.

Im Header definieren wir lediglich den Zeichensatz (charset) und den Titel der Seite. Der Titel wird im Tab des Browsers angezeigt.

`<h1>` ist unsere Überschrift. Durch das Beenden der Überschrift mit `</h1>` erfolgt der Zeilenumbruch automatisch. Nun kommt ein Eingabefeld vom Typ Text. Wichtig für JavaScript ist hier die **id**. Als Platzhalter hab ich *Vorname* hingeschrieben, damit der User auch weiß, was er eingeben muss.

Neben dem Eingabefeld kommt ein Button mit der **id** *hello*. Auch diese id ist sehr wichtig. Am Ende des body-Bereichs wird noch das JavaScript geladen.

main.js

```
"use strict";
function welcome()
{
  let person = document.getElementById('inputName'); // Wir lesen den Namen aus
  let output = „Hallo „ + person.value + „!"; // Wir formulieren die Begrüßung
  alert(output); // Alert gibt die Begrüßung aus.
  inputName.value = „"; // Wir löschen den Inhalt des Textfelds.
}
```

```
let hello = document.getElementById('hello'); // Wir identifizieren den Butt
hello.addEventListener('click', welcome, true); // Wir rufen bei Klick die F
```

Im Prinzip sagt der Code-Kommentar schon sehr viel darüber aus, was wir machen. Ich möchte dennoch ein bisschen darauf eingehen.

Was ist eigentlich *“use strict”*? Gute Frage! Es zwingt den Browser, ECMAScript in der Version 5 zu verwenden bzw. darüber. Das „zwingt“ uns sauberer und nach den neusten JS-Standards zu programmieren. So kann man bspw. keine Variablen benutzen, die vorher nicht deklariert wurden.

Die Funktion *welcome* ist ein in sich abgeschlossener Code. Funktionen gibt es in vielen Programmiersprachen und sie werden vor allem geschrieben, wenn eine bestimmte Reihenfolge von Befehlen bzw. allgemein eine Aufgabe immer wieder ausgeführt wird. Auch in GML gibt es mittlerweile Funktionen, man kann es dort aber am ehesten mit Skripten vergleichen. Funktionen können auch mehrere Argumente aufnehmen und verarbeiten. In diesem Beispiel brauchen wir das aber nicht.

Mit *let* wird die Variable deklariert. Es geht auch *var*, wird aber heute in JavaScript nicht mehr oder nur selten benutzt. Das hat u. a. den Grund, dass *let* nur innerhalb der Funktion gültig ist, *var* hingegen gilt auch außerhalb einer Funktion. Man könnte *let* somit als „lokale Variable“ ansehen.

Schauen wir uns die erste Zeile der Funktion genauer an:

```
let output = „Hallo „ + person.value + „!";
```

Davor haben wir die Variable *person* definiert, nun verwenden wir den Inhalt der Variable (*person.value*) um die Begrüßung zu schreiben.

In der dritten Zeile geben wir die Begrüßung per *alert(output)*; aus.

Debugging

Wenn wir schon dabei sind, könnten wir auch statt *alert(output)*; den Befehl *console.log(output)*; verwenden. Das wird der User zwar nicht sehen, aber die Ausgabe der Konsole können wir Werte speichern oder – und das wird noch mehr verwendet – zur Kontrolle Zwischenwerte ausgeben. Es ist somit eine ganz gute Debug-Hilfe. In die Konsole kommt man in den meisten Browsern mit F12.

In der letzten Zeile löschen wir den Inhalt des Eingabefeldes.

Nach der Funktion definieren wir unseren Button und sagen, was zu tun ist, sobald jemand den Knopf drückt. Wir können hier somit eine beliebige Funktion definieren.

Mehr Nutzen

Das sieht alles recht primitiv aus, aber es sind elementare Grundlagen. Mit dem bisher gezeigten Wissen kann man schon recht viel machen. Zum Beispiel aus Vor- und Nachnamen eine Begrüßung basteln. Der Sprung, um mehrere Zahlen mathematisch zu verarbeiten, ist minimal. Man kann damit auch Eingaben auf Plausibilität prüfen, wie etwa eine Mailadresse.

Ausblick

Ab dem nächsten Teil erhöhen wir die Komplexität. Dies betrifft die Ausgabe Texten/Zahlen wie auch die Verarbeitung von Daten.

Überblick Webdev-Serie

[Webentwicklung Grundkurs Teil 1 – Einstieg](#)

[Webentwicklung Grundkurs Teil 2 – Aufbau von Webseiten](#)

[Webentwicklung Grundkurs Teil 3 – Datei- und Ordnerstrukturen](#)

[Webentwicklung Grundkurs Teil 4 – Einstieg in JavaScript](#)

[Webentwicklung Grundkurs Teil 5 – Datenverarbeitung und Formulare mit JavaScript](#)

Überblick Interviews

[Interview mit Magnus Reiß – Webgamers](#)

[Interview mit Wolfgang Scheidle – Tischtennis Manager](#)

[Interview mit Warg – Drifting Souls II](#)

Date Created

29. November 2019

Author

sven