

Aufgabenblatt 5

Webtechnologien
Wirtschaftsinformatik
WS 2020/21

Lernziele

Es wird folgendes gelernt:

- Formularprüfung in JavaScript
- Mustererkennung in JavaScript
- Reguläre Ausdrücke
- Layout mit `<div>`
- Ausgabe von Text im DOM
- Anzeigen und Löschen von DOM-Knoten

JavaScript, Formulare und `<div>` - 3 Punkte

In diesem Aufgabenblatt wird ein Klassiker gelernt: Zugriff auf Formularelemente sowie deren Prüfung in JavaScript. Alle guten Websites prüfen die Formulare vor dem Abschicken zum Server. Allerdings müssen dieselben Werte in jedem Fall auf dem Server (in PHP dann) nochmals geprüft werden, da ein HTTP-Request auch ohne ein Browser samt JavaScript an den Server geschickt werden kann. Das erste Prüfen in JavaScript führt daher nur zu einer guten Mensch-Maschine-Schnittstelle, nicht zu einem sicheren System.

Beispiele zur Manipulation des DOMs befinden sich in den Foliensätzen über JavaScript.

Und nun die einzelnen Schritte:

1. Dazu realisieren Sie ein Formular, das auf einer eCommerce-Site vorkommen könnte: Name (Vorname und Nachname zusammen), Postleitzahl, Stadt, Straße, Hausnummer, IBAN und Geldbetrag.
2. Alle Formularfelder erhalten ein Label (`<label>`), das ganze Formular wird in ein Layer (`<div>..</div>`) mit einem eindeutigen Namen (`"id=..."`) gebracht. Die Koordinaten sowie die Größe des Formulars stehen in einer externen CSS-Definition, d.h. Größe und Lage sind nicht in der Seite enthalten, wohl aber in der extern dazu gebundenen CSS-

Datei. Die einzelnen Elemente der Formulare werden durch weitere, geschachtelte Layer jeweils mit "id" benannt gekapselt und damit auch per CSS positioniert. Diese Layer enthalten jeweils das Label und das Formularelement. Die Benutzung von Tabellen zum Layout ist nicht zulässig. Am Ende des Formulars wird mit <button> ein Knopf eingefügt.

Dies ist auf den ersten Blick umständlich; es hat aber den großen Vorteil, dass Form und Inhalt streng getrennt sind. Dies führt zu einer besseren Wartbarkeit und höheren Wiederverwendbarkeit.

3. Für jeden Datentyp des Formulars schreiben Sie eine eigene JavaScript-Routine, die ein True oder ein False liefert, wenn der Inhalt in Ordnung bzw. nicht in Ordnung ist. Dabei schränken Sie die Verwendung von Zeichen *sinnvoll* so eng wie möglich ein. Orientieren Sie sich hierbei an den Gepflogenheiten in Europa (Namen und Straßen werden groß geschrieben etc.). Es wird darüber hinaus immer die maximale Länge geprüft. Gehen Sie davon aus, dass alle Formularelemente ausgefüllt werden müssen. Beispiele:

Name: A-Za-z und - erlaubt

Nachname: A-Za-z und - und ' erlaubt, denken Sie an "O'Connor"

IBAN: [A-Z][A-Z] gefolgt von 20 Ziffern

Für jeden Typ definieren Sie einen regulären Ausdruck für JavaScript.

Sie können auch als vor- und nachlaufende Zeichen Blanks erlauben. Dann wäre es aber besser vor dem Pattern-Matching diese Blanks zu entfernen (dazu gibt es die trim()-Routine).

Wenn Sie es richtig schön machen wollen, dann entfernen Sie vor der Prüfung auch im String stehende Blanks, die bei der IBAN zur besseren Lesbarkeit gerne eingefügt werden, z.B. DE47 3050 1000 1001 10...

4. Jedes Formularelement hat einen eindeutigen Namen bekommen, über den es mit JavaScript angesprochen werden kann. Diesen Namen übergeben Sie den Prüfroutinen als Parameter. Diese holen sich den Formularinhalt und prüfen die Korrektheit anhand des regulären Ausdrucks.

Diese Prüfung aller Forminhalte wird durch Aufrufen der vielen Prüfroutinen in einer einzigen Routine zusammen gefasst, die an das Klick-Ereignis des Knopfes gebunden wird. Wenn also auf diesen Knopf geklickt wird, wird diese Routine gestartet und prüft alle Formularfelder, indem sie die Prüfroutinen zu den Feldern aufruft.

Wenn ein Fehler festgestellt wird, dann wird eine sinnvolle Meldung mit alert() ausgegeben; dies erfolgt in jeder Prüfroutine, d.h. es können mehrere alert()-Aufrufe erfolgen. Wenn keine Fehler festgestellt wurden, wird mittels alert() "Alles in Ordnung" ausgegeben.

In dieser Aufgabe soll nur die Prüfung realisiert werden, daher ist es völlig in Ordnung, dass der Formularinhalt nicht an den Server gesendet wird. Formular und Prüfung laufen rein lokal ab. In der Praxis wird natürlich die Prüfung der Formularfelder an den Submit-Knopf gebunden.

5. [Abgabe] Nun realisieren Sie eine Routine, die dynamisch folgendes macht: unterhalb des Formulars wird ein weiterer Layer (`<div>`) eingefügt, der einen roten Rahmen erhält (auch mit CSS und id). In diesen Layer werden die Meldungen der Prüfroutinen bzw. "Alles in Ordnung" geschrieben.

Am besten ist es, wenn Sie sich dazu eine Routine Namens `writeMsg(Message)` zum Schreiben sowie eine Routine `clearMsg()` zum Löschen des Textes samt Layer schreiben. Die erste Routine wird im Fehlerfall bzw. zur Ausgabe aufgerufen und ersetzt `alert()`. Bitte beachten Sie, dass es mehrere Meldungen geben kann; diese werden dann hintereinander in den Bereich geschrieben; es kann also sein, dass `writeMsg()` mehrfach aufgerufen wird.

Beachten Sie, dass der Layer am Anfang erzeugt und später benutzt werden muss. Der Layer mit den Meldungen wird von `clearMsg()` gelöscht.

Bei einer erneuten Prüfung – also nach einem weiteren Klick – werden natürlich durch Aufruf von `clearMsg()` eventuell schon vorhandene Meldungen gelöscht. In einem Teil der Folien wird der Source-Code zur Ausgabe von Text mit JavaScript in DOM sowie die Erzeugung von Layern beschrieben.

Das testen Sie nun, indem Sie die `alert()`-Aufrufe durch `writeMsg()` ersetzen. Dann brechen Sie nicht nach dem ersten Fehler ab, sondern analysieren immer alle Felder und geben daher durchaus mehrere Meldungen aus.

Damit haben Sie eine einfache Prüfung von Formularinhalten realisiert.

Links

- http://www.w3schools.com/tags/tag_button.asp
- http://www.w3schools.com/tags/tag_label.asp
- <https://de.wikipedia.org/wiki/IBAN>
- http://www.w3schools.com/cssref/pr_class_visibility.asp
- <http://php.net/manual/de/function.trim.php>

Abnahme und Bewertung

Das Ergebnis des Schrittes 5 wird per Email abgegeben. Dabei muss die gesamte Site als zip-Datei so wie sie unter htdocs liegt bzw. das netbeans-Projekt abgegeben werden. Bei Abgaben per Email muss die Lösung auf den Rechnern der Dozenten im ersten Anlauf laufen.

Es sind max. 3 Punkte zu erreichen; fehlt etwas oder ist es nicht in Ordnung, so werden Punkte abgezogen bzw. die Lösung nicht anerkannt. Die Verwendung regulärer Ausdrücke in JavaScript ist ein Muss bei dieser Aufgabe. Auch ist es nicht erlaubt, Frameworks wie z.B. jQuery, zu benutzen.

Es müssen auch die in den vorherigen Aufgabenblättern vorgestellten Regeln über CSS, Layout und HTML4/5 eingehalten werden.