

Übungsblatt 2: HTML-Formulare und JS-Kontrollflüsse

Vorstellung in den Tutorien am 8. - 11. November 2021

2.1 HTML: Formulare

In der Vorstellung wurde *YASN - Yet Another Social Network* vorgestellt. Implementieren Sie das Registrierungsformular für YASN mit den folgenden Elementen und nutzen Sie für jedes Eingabefeld den am besten passenden Feldtypen:

- Ein Eingabefeld für den Benutzernamen (Pflichtfeld).
- Ein Eingabefeld für die Telefonnummer (Pflichtfeld).
- Ein Eingabefeld für die E-Mail-Adresse (Pflichtfeld).
- Ein Eingabefeld für den Wohnort. Hier sollen dem Nutzer zur Autovervollständigung einige der meistgewählten Städte (Hamburg, Hannover, Bochum, Berlin und München) vorgeschlagen werden.
- Ein Eingabefeld für die E-Mail-Adressen von Freunden, die ebenfalls zu YASN eingeladen werden sollen.
- Ein Eingabefeld für die URL der Homepage des Nutzers.
- Ein Eingabefeld für das Geburtsdatum. Beschränken Sie die Auswahlmöglichkeiten auf volljährige Nutzer, die im 20. und 21. Jahrhundert geboren wurden.
- Ein Eingabefeld für den Geburtsort. Machen Sie durch einen Platzhalter deutlich, dass eine Stadt angegeben werden soll.
- Ein Eingabefeld für die gewünschte Hintergrundfarbe des Nutzerprofils.
- Ein Button zum Absenden der eingegebenen Daten.

2.2 JS: Kontrollflüsse

1. Schreiben Sie in JavaScript einen exemplarischen Kontrollfluss bestehend aus einem if-Zweig, einem else-if-Zweig und einem else-Zweig.
2. Implementieren Sie für den Kontrollfluss aus der vorherigen Aufgabe einen äquivalenten switch-Kontrollfluss. Wann sind die break-Statements zwingend notwendig?
3. Implementieren Sie exemplarisch eine for-Schleife, eine while-Schleife und eine do-while Schleife. Diskutieren Sie die jeweiligen Unterschiede der Schleifen.

2.3 JS: Obfuscation

Da JavaScript-Code (im Browser ausgeführt) immer für den Nutzer sichtbar ist, gibt es heute viele Wege, diesen Code zu verschleiern bzw. zu verstecken. Diese Praxis nennt man „Obfuscation“ (zu Deutsch: Verschleierung). Dazu wird der Code mittels unterschiedlicher Methoden unkenntlich gemacht. Eine Möglichkeit ist z.B. das Entfernen aller Leerzeichen und Zeilenumbrüche, um die Lesbarkeit zu mindern. Dieser Ansatz ist allerdings wenig effektiv, da es ebenfalls Tools gibt, um JavaScript-Code automatisch zu strukturieren. Ein anderer (effektiverer) Ansatz ist es, jedes einzelne Zeichen des Codes (Variablennamen, Funktionsnamen, usw.) durch eine Hexadezimalzahl darzustellen. Siehe folgendes Beispiel:

```
var myVariable = „test“;
```

```
var myVariableEncoded = „0x740x650x730x74“;
```

1. Schreiben Sie ein JavaScript-Programm, das jedes Zeichen einer Variablen `message` mit dem Wert „message“ durch eine Hexadezimalzahl ersetzt und das Ergebnis in einer Variable `encoded` speichert.
2. Schreiben sie ein JavaScript-Programm, welches diesen Prozess rückgängig macht, also Hexadezimaldarstellung in lesbaren Text übersetzt.

Hinweis: Ihnen ist eine Vorlage gegeben, welche die Konvertierung eines Zeichens für Sie übernimmt. Sie müssen lediglich den richtigen Kontrollfluss implementieren.