

Übungsblatt 2

6 Eine Webseite aktualisieren*

Laden Sie die Webseite auf <https://github.com/WebAppLecture/Exercises.Day2/blob/master/Templates/Aufgabe6.html> herunter. Modifizieren Sie diese so, dass die neue Seite den HTML5 Standards gerecht wird. Gehen Sie dazu folgendermaßen vor:

- Ersetzen Sie den Document Type durch `<!DOCTYPE html>`
- Ersetzen Sie die `<div>`s durch Ihre entsprechenden HTML5 Pendanten – z.B. `<div id=header>` durch `<header>` (Navigation sollte ebenfalls in den entsprechenden Tag)
- Kürzen Sie unnötigen Code heraus – so wird aus dem `<meta>`-Tag `<meta charset=utf-8>`
- Ändern Sie die `<div>`s mit den post Klassen in entsprechende `<article>` Tags um
- Den Inhalt des Style Tags (über CSS werden Sie später mehr lernen) können Sie durch die folgenden Modifikationen auf den neuesten Stand bringen:

```
1 | nav { float: left; width: 20%; }  
2 | article { float: right; width: 79%; }  
3 | footer { clear: both; }
```

7 Taschenrechner basteln*

Sie bauen nun einen Taschenrechner mit HTML und JavaScript. Denkbar wäre die Verwendung eines Formulars für die Eingabedaten. Es soll zwei Textfelder geben, die durch eine `<select>`-Box getrennt sind. Durch einen Button wird die Rechnung gestartet. Ein mögliches Design ist in Abb. 1 dargestellt.



The image shows a horizontal layout of a calculator interface. It consists of two text input fields, each containing the number '0'. Between these two fields is a small dropdown menu with a plus sign and a downward arrow. To the right of the second input field is an equals sign button, and to its right is a third text input field, also containing the number '0'.

Abbildung 1: Mögliches Design des Taschenrechners

Die Operation soll durch eine `switch-case`-Anweisung entschieden werden. Nach der Berechnung soll das Ergebnis entweder in einer dritten (`readonly` gesetzten) Textbox oder in einem `<output>`-Tag angezeigt werden.

8 Ein Formular erschaffen*

Das Formular verfügt über 8 Felder, von denen Sie so oft wie möglich ein HTML5 Feld verwenden sollen:

- Name*, hier soll nach dem Muster Name, Vorname eingegeben werden, z.B.:

```
1 | [a-zA-Z\ -]+, \s[a-zA-Z]+
```

- E-Mail*
- Telefonnummer, mit dem Format (für z.B. +49 (941) 1234):

```
1 | \+[0-9]{2}\s?\([0-9]{3}\)\s?[0-9]{4,}
```

- Geburtstag* (Datum¹)
- Anzahl der bisherigen Leistungspunkte (dies soll als Slider eingebaut werden von 0 bis 180)
- Url der Homepage
- Passwort*
- Passwortbestätigung*

Mit * versehene Punkte sollen Pflichtfelder darstellen. Beim Betreten der Seite muss das erste Feld (Name) automatisch fokussiert werden. Einen Button zum Versenden des Formulars können Sie z.B. über `<input type=submit value='Ich bin der Button! '>` erstellen.

9 Das Formular verbessern*

Sie verbessern nun das Formular aus Aufgabe 8 durch einige sehr nützliche JavaScripts:

- Das angegebene Alter (Geburtsdatum) soll mindestens 18 Jahre sein. Ansonsten: Fehler.
- Die beiden Passwörter müssen auf Gleichheit überprüft werden. Ansonsten: Fehler.

¹Geht nicht bei Firefox (Version 7) und IE (Version 9)

- Die Anzahl der ausgewählten Leistungspunkte soll neben dem Slider (range) angezeigt werden.
- Wenn die Anzahl der Leistungspunkte kleiner als 50 ist, so soll eine Warnung angezeigt werden.

In den beiden Fehlerfällen soll das Formular ebenfalls nicht validieren, d.h. wir lassen keinen Submit zu. Um den Submit noch weiter zu verbessern, schreiben Sie einen EventHandler für das `oninput` Ereignis des Formulars. Also: Deaktivieren Sie den Submit Button, wenn das Formular nicht validiert über JavaScript.

10 Eigener Dialog mit Rückgabewert

Beim Klicken eines Buttons soll ein Dialog (d.h. ein neues Browser-Fenster) über `window.open()` geöffnet werden, der eine TextBox mit der Aufforderung Text einzugeben beinhalten soll. In diesem Dialog soll bei Änderung des Eingabetextes immer, d.h. bei Eingabe, eine Auswertung dieses Textes erfolgen.

Am Besten implementieren Sie die für die Auswertung notwendigen Funktionen bereits am Anfang so, dass jede Funktion eine Variable (einen String) als Übergabewert erhält und einen Wert zurückgibt.

Die Auswertung soll direkt unter der TextBox angezeigt werden und folgende Daten umfassen:

- Länge des eingegebenen Textes
- Anzahl der unterschiedlichen Zeichen (ohne Leerzeichen)
- Anzahl der Wörter
- Anzahl der Sonderzeichen und Zahlen (bzw. Nichtbuchstaben (`a-z`, `A-Z`))

Im Hauptdialog soll ebenfalls eine kleine Zusammenfassung angezeigt werden:

- Alle gefundenen Wörter (jedes in eine neue Zeile)
- Summe aller gefundenen Zahlen (z.B. ist dies bei “ha22llo12” als Ergebnis 7)

Solang der Dialog geöffnet ist, soll kein weiterer Dialog aufgemacht werden können (Tipp: Hierfür die Eigenschaft `closed` des `window` Objektes verwenden!).

Weiterhin könnten Ihnen folgende Tipps sehr hilfreich sein:

- Das Ereignis `onkeyup` eignet sich besser zum Überprüfen auf Veränderung als `onchange`, da `onchange` erst beim Verlassen (`onblur` Ereignis) des Eingabefeldes ausgelöst wird.
- Um das neue Fenster als kleinen Dialog zu öffnen, sollte folgender Code verwendet werden:

```
1 | window.open('', '', 'width=300,height=500');
```

- Über `createElement()` kann man das Element direkt als Variable erhalten.
- Die Ausgabe kann man über die `innerHTML` Eigenschaft von Elementen setzen – damit man nicht lauter Strings verketteten muss, kann man ein Array von Strings namens `ausgabe` anlegen, welches man anschließend über `ausgabe.join('
')` so verbindet, dass man lauter einzelne Zeilen erhält.