



Webentwicklung Grundkurs Teil 5

Description

[Im letzten Teil](#) haben wir den ersten Schritt in die JavaScript-Welt getan. Nun geben wir etwas Gas und programmieren einen Schachturnier Manager. Der Sprung erscheint auf den ersten Blick als groß, aber im Kern sind die meisten Aufgaben lediglich Erweiterungen von Grundlagen.

Tabelle

Spieler	S	U	N	Punkte
Gerald	0	1	9	0.5
Heinz	4	5	1	6.5
Klaus	4	3	3	5.5
Sven	8	2	0	9
Thomas	2	2	6	3
Volker	5	1	4	5.5

Der Schachturnier Manager sollte folgende Kriterien erfüllen:

1. Man soll die Anzahl der Spieler festlegen können.
2. Man soll die Namen der Spieler eingeben können.
3. Die Paarungen sollten angegeben werden können.
4. Die Ergebnisse sollen eingegeben werden.
5. Die Tabelle soll erzeugt werden.
6. Die Seite soll nicht laufend neu geladen werden müssen.

Neben den wesentlichen Zielen gibt es noch sekundäre. Zum Beispiel die Überprüfung, dass die Anzahl der Spieler eine gerade Zahl sein muss. Bei den Ergebnissen ist es wichtig, dass, wie im Schach üblich, nur die Zahlen 0, 0,5 und 1 möglich sind.

Was wir nicht machen werden: Es gibt keinen Spielplan-Generator. Das hätte den Rahmen gesprengt. Ich liefere dazu unten weitere Informationen, falls sich jemand dafür interessiert. Die Tabelle wird am Ende nur Alphabetisch sortiert, aber auch dazu gibt es ein paar Hinweise.

[Hier kann das Ergebnis angeschaut und getestet werden.](#)

index.html

Wie in den vergangenen Tutorials auch, brauchen wir eine **index.html**. Außerdem lediglich eine **main.js**, die wir im Ordner *js* anlegen.

```
<!DOCTYPE html>
<html lang="de">
  <head>
    <meta charset="utf-8">
    <title>Schachturnier Manager</title>
    <script src="js/main.js"></script>
  </head>
  <body>
    <h1>Schachturnier Manager</h1>
    <div id="spielerZahl">
      <p>Geben Sie die Anzahl der Spieler ein. Für das Turnier werden <b>nur gera
      <form id="form1" action="javascript:;" onsubmit="step1(document.getElementB
        <input type="number" id="zahl" min="2" step="2" value="6" required>
        <button type="submit">Abschicken</button>
      </form>
    </div>
    <div id="spielerNamen">
      <form id="form2" action="javascript:;" onsubmit="step2()">
      </form>
    </div>
    <div id="ergebnisse">
      <form id="form3" action="javascript:;" onsubmit="step3()">
      </form>
    </div>
    <div id="tabelle">
    </div>
  </body>
</html>
```

Wer sich das fertige Programm angeschaut hat, wird vielleicht ein wenig staunen, dass wir so wenig in der index.html stehen haben. Nahezu alles, was wir brauchen, ist in der JavaScript Datei und wird nach Bedarf in die Divs gelegt, aber dazu gleich mehr.

Über den Header muss ich nicht mehr reden. Dazu habe ich schon genug Worte in den vorherigen Tutorials verloren. Wir konzentrieren uns also auf den body-Bereich. Neben der Überschrift (h1) finden

wir nur vier Div-Bereiche. In drei davon finden wir noch Formulare.

<div> steht für Division, man kann es auch als Sektion bezeichnen. Innerhalb einer Sektion kann man alles mögliche anlegen. Sektionen können Klassen zugeordnet werden, was später in Bezug auf CSS interessant sein wird. Momentan ist für uns die Möglichkeit spannend, dass wir IDs vergeben können. Das kann man nicht nur bei Divs so machen, sondern, wie man sehen kann, auch bei Formularen. Nachher werden wir sehen, dass das auch bei Eingabefeldern funktioniert.

Die ID ist wichtig, damit wir die Positionen später finden. Hier werden Werte entweder ausgelesen oder Inhalte platziert bzw. auch gelöscht.

Im ersten div haben wir ein kleines Formular mit einem Button. Damit etwas passiert, wenn wir den Knopf rücken, haben wir das Formular wie folgt definiert:

```
<form id="form1" action="javascript:;" onsubmit="step1(document.getElementById("form1").value)">
```

Mit **action** und **onsubmit** sorgen wir dafür, dass die Funktion *step1* in der JS-Datei ausgeführt wird. Mit *document.getElementById('zahl').value* übergeben wir einen Parameter an die Funktion *step1*. Der Parameter ist die Anzahl der Spieler.

Wer kontrolliert wen?

Besonders bei Formularen kommt man nicht umhin, die Eingaben zu kontrollieren. Früher war das eine große Aufgabe von JavaScript. Da der HTML-Standard aber irgendwann erweitert wurde, können wir viele Dinge bereits von HTML prüfen lassen.

Das ist unser Eingabefeld vom Typ *number*. Man kann nur Zahlen, keine Buchstaben eingeben. Die Zahl muss mindestens 2 sein (*min="2"*), muss gerade sein, was wir durch *step="2"* erreichen. Der Standardwert steht bei 6 (*value="6"*). Der Zusatz *required* sagt aus, dass eine Eingabe erfolgen muss.

Die Kontrolle der Eingabe übernimmt der HTML-Part und wir können uns das in JavaScript

JavaScript-Header

Was jetzt folgt, landet alles in der Datei **main.js**. Es gibt einen kleinen „header“, drei Funktionen und eine Bonusfunktion. Doch

Das erste Formular haben wir in HTML erstellt. Ab jetzt übernimmt JavaScript.

```
"use strict";
```

```
let schachturnier = new Object();
schachturnier.spielerZahl = 0;
schachturnier.spielLos = 0;
schachturnier.spielerNamen = [];
schachturnier.spielerPunkte = [];
schachturnier.spielerSiege = [];
schachturnier.spielerUnentschieden = [];
schachturnier.spielerNiederlagen = [];
```

Wir erstellen ein Objekt mit dem Namen **schachturnier**

. In dieses Objekt speichern wir einige Variablen und Arrays. Das System lässt

- spielerZahl -> Anzahl Spieler
- spielLos -> Das sind die Select-Felder bei den Spielernamen im Spielplan
- spielerNamen -> Ein Array mit den Namen der Spieler
- spielerPunkte, spielerSiege, spielerUnentschieden und spielerNiederlagen zählt später die entsprechenden Resultate pro Spieler auf

function step1(zahl)

```
// Spielernamen
function step1(zahl)
{
  if ((zahl%2)===0)
  {
    document.getElementById('spielerZahl').innerHTML = „"; // Wir löschen den In

    schachturnier.spielerZahl = zahl;

    let formular = '<br><p>Geben Sie die Namen der Spieler ein:</p>';
    let counter = 1;

    for (let i=0; i<zahl; i++)
    {
      formular += '<input type=\"text\" id=\"s' + counter + '\" placeholder=\"Spieler ' +
        counter++ + '\">';
    }

    formular += '<br><button type=\"submit\">Abschicken</button>';
    document.getElementById('form2').innerHTML = formular;

  } else {
    let output = zahl + ' ist keine gerade Zahl!';
    alert(output);
  }
  return false;
}
```

Was diese Funktion macht: Sie prüft, ob die Zahl der Spieler gerade ist. Wenn ja, wird das alte

Formular gelöscht und ein neues Formular erstellt. Wenn nein, gibt sie eine Fehlermeldung aus.

Schauen wir es im Detail an.

```
document.getElementById('spielerZahl').innerHTML = „“;
```

Im Kommentar steht zwar, dass wir den Inhalt löschen, aber so ganz richtig ist es nicht. Der Inhalt wird nicht gelöscht, er wird lediglich gegen nichts ("") ersetzt. Das ist ein feiner Unterschied. Konkret wird auch nicht das Formular, sondern der ganze Inhalt von `<div id="spielerZahl">` ersetzt. Für den Anwender verschwindet er einfach.

```
let formular = '<br><p>Geben Sie die Namen der Spieler ein:</p>';
```

Die Variable *formular* wird später der neue Inhalt. Zunächst weisen wir ihr die erste Zeile zu. Es ist leicht zu erkennen, dass wir innerhalb der einfachen Anführungszeichen HTML-Code haben. `<br>` ist der Zeilenumbruch. `<p>` ist der Paragraph mit dem entsprechenden Text darin. Man muss es ein bisschen auf sich wirken lassen, aber dann merkt man, dass sich dadurch viele coole Möglichkeiten ergeben.

```
let counter = 1;
```

```
for (let i=0; i<zahl; i++)
{
  formular += '<input type="text" id="s' + counter + '" placeholder="Spiel'
  counter++;
}
```

```
formular += '<br><button type="submit">Abschicken</button>';
```

Mit der for-Schleife erzeugen wir die Eingabefelder für die Namen. Den counter nehmen wir, damit die Felder und der Platzhalter bei 1 und nicht bei 0 beginnen.

Wichtig ist immer das *formular +=* ! Damit fügen wir der Variable *formular* neuen Content hinzu.

```
// Partien
function step2()
{
  let zahl = schachturnier.spielerZahl;

  // Wir ermitteln das aktuelle Datum
  let today = new Date();
  let dd = String(today.getDate()).padStart(2, '0');
  let mm = String(today.getMonth() + 1).padStart(2, '0');
  let yyyy = today.getFullYear();

  today = dd + '.' + mm + '.' + yyyy; // heute ist...

  let formular = '<br><p>Geben Sie die Ergebnisse der Spiele für den ' + today

  // Wir lesen die Namen aus dem vorherigen Formular und speichern sie im Objekt
  let counter = 1;
  for (let i=0; i<zahl; i++)
  {
```

```
    schachturnier.spielerNamen[i] = document.getElementById('s'+counter).value;
    counter++;
}

document.getElementById('spielerNamen').innerHTML = „"; // Wir löschen den In

let anzSpiele = Math.pow(zahl, 2)-zahl; // Wir berechnen die Anzahl der Spiel

formular += '<table>';
formular += '<th>Paarung</th><th>Resultat</th>';

counter = 1;

for (let x = 0; x<anzSpiele; x++)
{
    formular += '<tr><td>';
    // Auswahlfelder für die Spieler
    formular += '<select name="player" id="pf' + schachturnier.spielLos + '">';
    formular += '<option value="-1" selected=\\"selected\\">Spieler wählen</option>';
    for (let y = 0; y<zahl; y++)
    {
        formular += '<option value="' + y + '">' + schachturnier.spielerNamen[y] + '
    }
    formular += '</select>'

    schachturnier.spielLos++;
    formular += ' - '

    formular += '<select name="player" id="pf' + schachturnier.spielLos + '">';
    formular += '<option value="-1" selected=\\"selected\\">Spieler wählen</option>';
    for (let y = 0; y<zahl; y++)
    {
        formular += '<option value="' + y + '">' + schachturnier.spielerNamen[y] + '
    }
    formular += '</select>'
    schachturnier.spielLos++;
    formular += '</td><td>';
    formular += ' <input type=\\"number\\" id=\\"g' + counter + '\\"size=\\"4\\" min=
    counter++;
    formular += '<input type=\\"number\\" id=\\"g' + counter + '\\"size=\\"4\\" min=
    formular += '</td></tr>';
    counter++;
}

formular += '</table>';
formular += '<br><button type=\\"submit\\">Abschicken</button>';

document.getElementById('form3').innerHTML = formular;

return false;
}
```

Viele Prinzipien haben wir uns bereits in der vorherigen Funktion angeschaut, weshalb ich nur auf das Wesentliche eingehe.

```
// Wir lesen die Namen aus dem vorherigen Formular und speichern sie im Objekt
let counter = 1;
for (let i=0; i<zahl; i++)
{
    schachturnier.spielerNamen[i] = document.getElementById('s'+counter).value;
    counter++;
}
```

```
document.getElementById('spielerNamen').innerHTML = „“; // Wir löschen den Inhalt
```

Klingt selbstredend, aber dennoch: Erst lesen wir die Namen aus dem vorherigen Formular aus, dann „löschen“ wir es. Dabei werden die Namen in *schachturnier.spielerNamen* geschrieben.

```
for (let x = 0; x<anzSpiele; x++)
```

Für jedes Spiel erstellen wir eine Zeile in der Tabelle. In der Variable *schachturnier.spielLos* zählen wir hoch, wie viele Eingabefelder wir haben. Theoretisch hätten wir auch einfach *anzSpiele * 2* nehmen können, aber da wir die Select-Felder durchnummerieren und wir ohnehin einen Counter brauchen (pro Zeile kommen wir auf zwei Felder) kann man das Problem gleich so lösen.

Was wir erzeugen, kann in etwa so aussehen:

Paarung		Resultat	
Volker ▼	- Sven ▼	0	- 1
Sven ▼	- Volker ▼	1	- 0
Volker ▼	- Klaus ▼	1	- 0
Klaus ▼	- Volker ▼	1	- 0
Volker ▼	- Thomas ▼	1	- 0
Thomas ▼	- Volker ▼	0	- 1
Volker ▼	- Heinz ▼	0,5	- 0,5
Heinz ▼	- Volker ▼	1	- 0
Volker ▼	- Gerald ▼	1	- 0
Gerald ▼	- Volker ▼	0	- 1
Sven ▼	- Klaus ▼	1	- 0
Klaus ▼	- Sven ▼	0,5	- 0,5
Sven ▼	- Thomas ▼	1	- 0
Thomas ▼	- Sven ▼	0	- 1
Sven ▼	- Heinz ▼	1	- 0
Heinz ▼	- Sven ▼	0,5	- 0,5
Sven ▼	- Gerald ▼	1	- 0
Gerald ▼	- Sven ▼	0	- 1
Klaus ▼	- Thomas ▼	1	- 0
Thomas ▼	- Klaus ▼	0	- 1
Klaus ▼	- Heinz ▼	0,5	- 0,5
Heinz ▼	- Klaus ▼	1	- 0
Klaus ▼	- Gerald ▼	0,5	- 0,5
Gerald ▼	- Klaus ▼	0	- 1
Thomas ▼	- Heinz ▼	0,5	- 0,5
Heinz ▼	- Thomas ▼	0,5	- 0,5
Thomas ▼	- Gerald ▼	1	- 0
Gerald ▼	- Thomas ▼	0	- 1
Heinz ▼	- Gerald ▼	1	- 0
Gerald ▼	- Heinz ▼	0	- 1

Abschicken

Man muss wissen, dass in die Auswahlfelder die IDs der Spieler als Value angegeben werden. Deshalb ist default auch -1. Theoretisch könnte man dies noch abfragen, bevor es weiter geht. Wichtig ist momentan nur, dass wir den Wert auslesen und damit dem Spieler zuordnen können.

function step3()

Am Ende werden die Partien ausgewertet und in einer Tabelle dargestellt.

```
// Tabelle
function step3()
{
    let zahl = schachturnier.spielerZahl;
    let los = schachturnier.spielLos;
    let player;
    let score;

    let counter = 1;

    let formular = '<h2>Tabelle</h2>';

    // Auswertung der Spiele
    for (let i=0; i<los; i++)
    {
        // Spieler ID aus dem Auswahlfeld
        let e = document.getElementById('pf'+i);
        player = e.options[e.selectedIndex].value;

        // Wir schauen, ob die Variablen bereits definiert wurden
        if (typeof schachturnier.spielerPunkte[player] === „undefined“)
        {
            schachturnier.spielerPunkte[player] = 0;
            schachturnier.spielerSiege[player] = 0;
            schachturnier.spielerUnentschieden[player] = 0;
            schachturnier.spielerNiederlagen[player] = 0;
        }

        score = document.getElementById('g'+counter).value;

        schachturnier.spielerPunkte[player] += +score;

        switch (score)
        {
            case '0':
                schachturnier.spielerNiederlagen[player] += 1;
                break;
            case '0.5':
                schachturnier.spielerUnentschieden[player] += 1;
                break;
            case '1':
                schachturnier.spielerSiege[player] += 1;
                break;
        }
    }
}
```

```

    counter++;
}

document.getElementById('ergebnisse').innerHTML = „"; // Wir löschen den Inhalt

formular += '<table border="1" id="Tabelle">';
formular += '<th>Spieler</th><th>S</th><th>U</th><th>N</th><th>Punkte</th>';

counter = 1;

for (let x=0; x<zahl; x++)
{
    formular += '<tr><td>';
    formular += schachturnier.spielerNamen[x];
    formular += '</td><td>';
    formular += schachturnier.spielerSiege[x];
    formular += '</td><td>';
    formular += schachturnier.spielerUnentschieden[x];
    formular += '</td><td>';
    formular += schachturnier.spielerNiederlagen[x];
    formular += '</td><td>';
    formular += schachturnier.spielerPunkte[x];
    formular += '</td></tr>';
    counter++;
}

formular += '</table>';

document.getElementById('tabelle').innerHTML = formular;
sortTable();
}

```

Nachdem wir am Anfang die Variablen definiert und die Überschrift festgelegt haben, geht es bereits mit der ersten Schleife los.

```
for (let i=0; i<los; i++)
```

Das vorherige Formular wird noch nicht gelöscht. Wir gehen alle Felder durch und weisen die Resultate den entsprechenden Spielern zu.

```

// Wir schauen, ob die Variablen bereits definiert wurden
if (typeof schachturnier.spielerPunkte[player] === „undefined“)
{
    schachturnier.spielerPunkte[player] = 0;
    schachturnier.spielerSiege[player] = 0;
    schachturnier.spielerUnentschieden[player] = 0;
    schachturnier.spielerNiederlagen[player] = 0;
}

```

Das sieht komisch aus. Wir müssen schauen, ob die Variable definiert ist. Wenn nicht, weisen wir den vier Variablen, die alle zu einem Spieler gehören, den Wert 0 hinzu. Anschließend können wir damit arbeiten.

```
schachturnier.spielerPunkte[player] += +score;
```

Nun addieren wir den Wert dem Player hinzu. Wichtig: Das `+= +score` sieht komisch aus, muss aber so sein. `+=` addiert dem vorhandenen Wert den neuen Wert hinzu. Das `+score` muss man so schreiben, damit JavaScript weiß, dass wir zwei Zahlen addieren. So wird aus `0 + 1 = 1` statt `0 + 1 = 01`.

```
sortTable();
```

Am Ende rufen wir noch die Funktion `sortTable()`; auf, um die Tabelle zu sortieren. Das Skript kommt aber nicht von mir sondern von der w3school. Hier kann es [angeschaut und heruntergeladen](#) werden.

Spielplan Generator

Wie bereits zweimal gesagt, wäre das etwas zu viel gewesen. Der Sprung jetzt war bereits ohne den Generator recht groß. Wer Interesse hat, kann das Skript noch erweitern, etwa indem er die Rückrunde optional macht, zählt, wie viele Punkte die einzelnen Farben geholt haben etc.

Wer sich mit dem Generator befassen will, sollte sich folgende Seite anschauen: [Kantenfärbungs Algorithmus](#)

Einen Code, allerdings in PHP, [findet man auf GitHub](#).

Ausblick

In der nächsten Folge bleiben wir bei JavaScript und wollen ein bisschen was mit Grafiken machen. Ich denke, von Daten hin und her schaufeln hat jetzt jeder genug. ?

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](#)

[Webentwicklung Grundkurs Teil 6 – Canvas kann was](#)

Überblick Interviews

[Interview mit Magnus Reiß – Webgamers](#)

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

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

Date Created

13. Dezember 2019

Author

sven