

# Science Camp Informatik - Tag 2

## Javascript

### Was ist Javascript überhaupt?

Javascript ist wie der Name schon sagt eine Skriptsprache, das heißt in etwa eine Sprache die für kurze kleine Programme gedacht ist. Javascript ist insbesondere dafür gedacht in Browsern eingesetzt zu werden, um Benutzerinteraktionen auszuwerten, Inhalte zu verändern, nachzuladen oder zu generieren und so die Möglichkeiten von HTML und CSS zu erweitern. Kurz gesagt um Internetseiten interaktiver zu machen.

Javascript-Dateien enden mit dem Dateiformat `.js`.

### Was ist wichtig?

#### Wie binde ich meine Javascript-Datei in mein HTML ein?

Du schreibst oben in die head tags den Pfad zu deiner Javascript-Datei als source in script tags. Das sieht dann zum Beispiel so aus:

```
1
2 <script src="js/bilder.js"></script>
3
```

#### Variablen:

Variablen sind sowas wie Behälter für Werte, zum Beispiel für Zahlen oder Wörter. Durch das Schlüsselwort `var` legt man die folgende Zeichenkette als Variable fest. Zum Beispiel wenn man das Wort "zahl" als Variable haben möchte sieht das so aus:

```
var zahl;
```

Variablen kann man natürlich auch Werte zuweisen, das funktioniert mit dem `=` Operator. Der Term:

```
var zahl = 7;
```

bedeutet also, dass zahl eine Variable mit dem Wert 7 ist. Werte von Variablen sind aber im Allgemeinen nicht endgültig man kann also `zahl` in der nächsten Zeile auch den Wert 8 zuweisen, mit dem Ausdruck:

```
zahl = 8;
```

Beachtet, dass man hier das Schlüsselwort `var` nicht mehr braucht.

#### If-Abfragen

If-Abfragen führen einen Teil Code aus falls eine Bedingung zutrifft.

```
if(zahl>7){...}
```

führt also den Code in den Klammern genau dann aus, wenn die Variable number größer als 7 ist.

If-Abfragen lassen sich auch mit einem Else- Block verbinden. Dieser Else Block wird dann ausgeführt, wenn die Bedingung bei der If-Abfrage nicht erfüllt wurde. Zum Beispiel:

```
if(zahl>7){...}else{...}
```

#### Schleifen

Schleifen benutzt man dann, wenn man ein Stück Code mehrmals ausführen möchte. Es gibt verschiedene Arten von Schleifen aber für die meisten Fälle reicht eine for-Schleife. Die Schleife hat eine Laufvariable, die meistens i genannt wird. In jedem Schleifendurchlauf wird i erhöht, erreicht i einen bestimmten Wert bricht die Schleife ab und der Code der hinter der Schleife steht wird ausgeführt. Die Syntax für eine solche for-Schleifen, die abbricht wenn i=10 ist, ist:

```
for(var i = 0; i<10; i++){...}
```

## Funktionen

Funktionen sind Codeabschnitte, das wesentliche an ihnen ist, dass man ihnen Parameter übergeben kann und sie ein Ergebnis zurückgeben können. Eine einfache Funktion wäre zum Beispiel:

```
1 function add(num1, num2){  
2     return num1 + num2;  
3 }
```

Diese Funktion add übernimmt zwei Variablen num1 und num2 als übergabeparameter und gibt ihre Summe zurück.

## Funktionsaufrufe

Die Funktionen die wir gerade kennengelernt haben muss man auch aufrufen. Zum einen kann man natürlich Funktionen direkt im Javascript aufrufen. Dazu schreibt man einfach den Funktionsnamen und in Klammern dahinter die Parameter, die man übergeben möchte. Aber Achtung die Reihenfolge der Parameter ist wichtig und immer so, wie man sie in der Methodendefinition geschrieben hat. Ein solcher Aufruf würde dann so aussehen:

```
add(7,9);
```

Hier würde 7 also in num1 und 9 in num2 gespeichert werden.

Man kann Javascript Funktionen auch im HTML-Code aufrufen, meistens tut man das mit sogenannten Events. Hat man einen Button designed und möchte man das beim klicken auf diesen Button die Funktion `add()` aufgerufen wird, so sieht der zugehörige Code wie Folgt aus:

```
<div onclick="add(7,9)">Addiere 7 und 9!</div>
```

Hier verwendet man das onclick-Event, dieses findet immer dann statt wenn auf das zugehörige HTML-Element geklickt wurde.

Möchte man dass eine Funktion direkt beim laden der Seite aufgerufen wird, so kann man sie auch in script tags an die gewünschte Stelle schreiben. Zum Beispiel so:

```
<script>add(7,9);</script>
```

Diese script tags geben an das zwischen ihnen Javascript-Code steht. Man könnte also auch den Code der Funktion, die man aufrufen möchte zwischen die tags schreiben. Aber das wird sehr schnell unübersichtlich und macht es schwer Funktionen wiederzuverwenden.

## Arrays

Arrays sind eine Art Tabelle in der viele Daten liegen, also zum Beispiel viele Zahlen oder Wörter. Anlegen lassen sich Arrays ähnlich zu anderen Variablen:

```
var zahlen = [7,2,4,5];
```

Hier wird ein Array mit 4 Zahlen angelegt. Solange man die einzelnen Elemente im Array nicht verändert bleiben sie und ihre Reihenfolge gleich. Auf ein Element innerhalb eines Arrays lässt sich wie folgt zugreifen:

```
zahlen[0] = 1;
```

Dieser Ausdruck setzt den ersten Eintrag des Arrays `zahlen` auf 1. Es sieht nun also so aus: `[1, 2, 4, 5]`. Dabei muss man beachten, dass man bei Arrays mit 0 anfängt zu zählen, das erste Element hat also den Index 0.

Man kann auch die Länge eines Arrays mit folgendem Befehl abfragen:

```
var laenge = zahlen.length;
```

Dieser Befehl würde in unserem Beispiel in der Variablen `laenge` also 4 gespeichert, da vier Elemente im Array liegen.