



1

Was ist JavaScript?

Computer sind enorm leistungsfähige Maschinen. Sie können die erstaunlichsten Dinge, zum Beispiel Schach spielen, Tausende von Webseiten zur Verfügung stellen oder Millionen komplexer Berechnungen in ein paar Sekunden ausführen. Aber tief drinnen sind Computer, ehrlich gesagt, ziemlich einfältig. Sie können nur genau das, was wir Menschen ihnen befehlen. Was ein Computer tun soll, sagen wir ihm mithilfe eines Computerprogramms. Das ist einfach eine Abfolge von Anweisungen, die der Computer abarbeitet. Ohne Programme könnten Computer gar nichts!

1.1 Eine Begegnung mit JavaScript

Es kommt sogar noch schlimmer: Computer verstehen weder Deutsch noch Englisch noch irgendeine andere gesprochene Sprache. Computerprogramme sind in *Programmiersprachen* wie etwa JavaScript geschrieben. Auch wenn du noch nie etwas von JavaScript gehört hast, hast du es ganz sicher schon benutzt. Mit der Programmiersprache JavaScript werden Programme geschrieben, die auf Webseiten laufen. JavaScript kann kontrollieren, wie eine Webseite aussieht oder wie sie reagiert, sobald der Benutzer auf einen Button klickt oder die Maus bewegt.

Sites wie Gmail, Facebook und Twitter ermöglichen mit JavaScript den Versand von E-Mails, das Posten von Kommentaren oder die Suche nach Websites. Wenn du beispielsweise auf Twitter Tweets liest, dann herunterscrollst und am Ende der Seite weitere Tweets angezeigt bekommst – dann siehst du JavaScript in Aktion.



Du brauchst nur ein paar Websites zu besuchen, damit dir klar wird, warum JavaScript so spannend ist.

- Mit JavaScript kannst du Musik abspielen und fantastische visuelle Effekte erzeugen. Zum Beispiel fliegst du auf <http://lights.helloenjoy.com> durch ein interaktives Musikvideo, das HelloEnjoy für Ellie Gouldings Song »Lights« programmiert hat (siehe Abb. 1–1).
- Mit JavaScript kannst du Werkzeuge erstellen, mit denen andere Anwender ihre eigenen Kunstwerke gestalten können. Patatap ist eine Art virtuelle »Drum machine« (<http://www.patatap.com>), die alle Arten von tollen Sounds erzeugt – und dabei auch gleich coole Animationen. Du siehst sie in Abbildung 1–2.

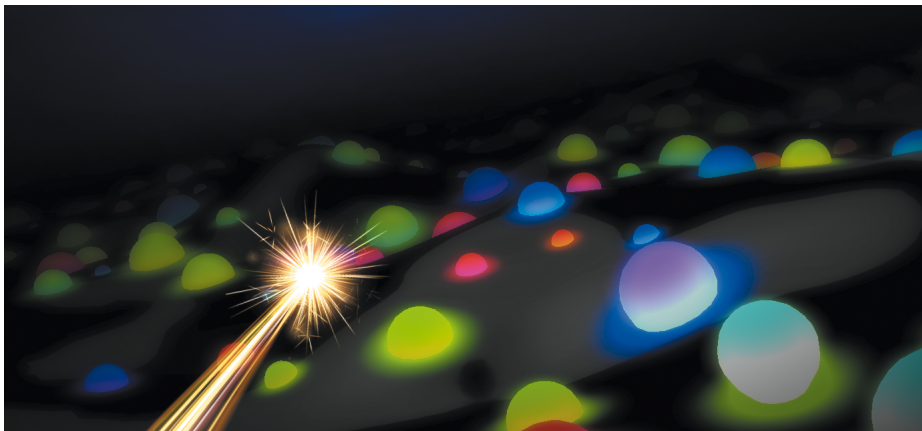


Abb. 1–1 Im Musikvideo »Lights« von HelloEnjoy kontrollierst du den funkelnden Cursor.

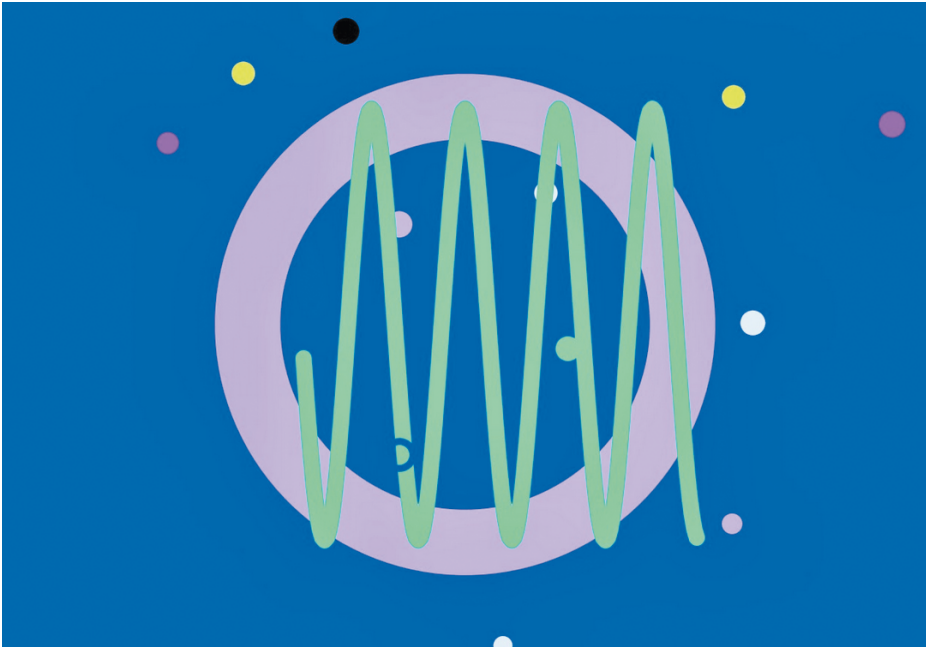


Abb. 1–2 Drücke auf Patatap einfach verschiedene Tasten, um unterschiedliche Geräusche hervorzubringen!

- Mit JavaScript kannst du witzige Spiele spielen. CubeSlam ist eine 3D-Variante des Spieleklassikers Pong (<https://www.cubeslam.com/>). Spiele gegen einen Freund oder gegen einen computergenerierten Bären (siehe Abb. 1–3)!



Abb. 1–3 Das Spiel CubeSlam ist komplett in JavaScript programmiert.

1.2 Warum solltest du gerade JavaScript lernen?

JavaScript ist nicht die einzige Programmiersprache – in Wirklichkeit gibt es Hunderte von Programmiersprachen. Es gibt aber viele Gründe, gerade JavaScript zu lernen. Zum einen ist es viel einfacher (und unterhaltsamer) zu erlernen als viele andere Programmiersprachen. Das Beste ist aber vielleicht, dass du nur einen Webbrowser wie den Internet Explorer, Mozilla Firefox oder Google Chrome brauchst, um JavaScript-Programme zu schreiben und auszuführen. Jeder Webbrowser hat einen JavaScript-*Interpreter*, der JavaScript-Programme lesen kann.

Sobald du ein JavaScript-Programm geschrieben hast, kannst du deinen Freunden einen Link dazu schicken, sodass sie es ebenfalls im Webbrowser auf ihrem eigenen Computer ausführen können (siehe Abschnitt »Deinen Code per JSFiddle teilen« auf Seite 271).

1.3 JavaScript-Code schreiben

Schreiben wir einfach einmal in Google Chrome (<http://www.google.com/chrome/>) ein wenig einfachen JavaScript-Code. Falls du diesen Browser noch nicht hast, installierst du ihn auf deinem Computer. Öffne ihn dann und gib `about:blank` in die Adresszeile ein. Wenn du jetzt Enter drückst, siehst du eine leere Seite (wie in Abb. 1–4).

Zunächst programmieren wir in der JavaScript-Konsole von Chrome. Das ist eine versteckte Möglichkeit, mit der Programmierer kurze JavaScript-Programme testen können. Unter Microsoft Windows oder Linux hältst du die Tasten Strg und Umschalt gedrückt und drückst J. Am Mac hältst du die Befehls- und die Options-Taste gedrückt und drückst ebenfalls J.

Wenn du alles richtig gemacht hast, siehst du jetzt eine leere Webseite mit einem blinkenden Cursor (|) neben einer spitzen Klammer (>), wie in Abbildung 1–4 gezeigt. Nun kannst du JavaScript schreiben!

Hinweis

Die Chrome-Konsole gibt deinem Code verschiedene Farben. Zum Beispiel erscheinen Texteingaben blau, die Ausgabe erhält je nach Typ unterschiedliche Farben. Wenn wir in diesem Buch die Konsole verwenden, siehst du im Codetext ähnliche Farben.

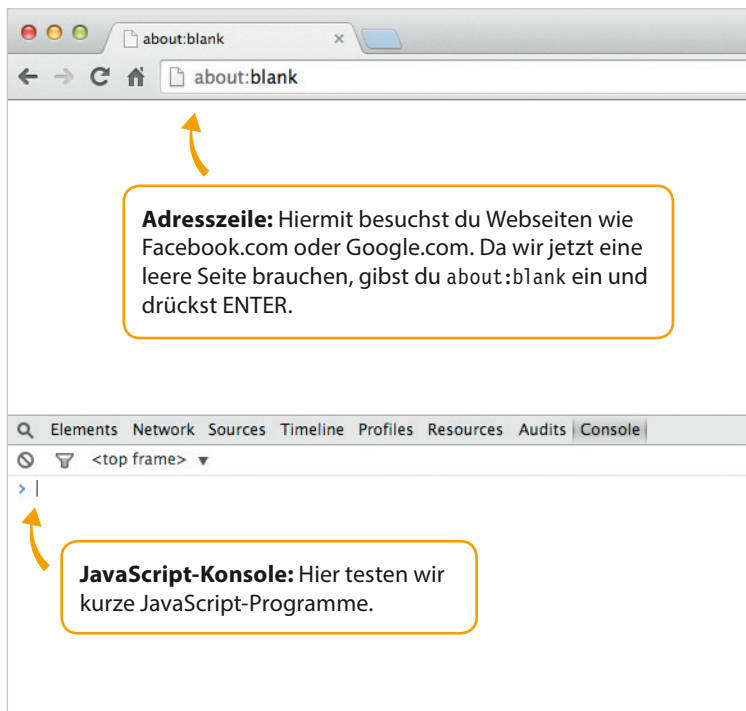


Abb. 1–4 Die JavaScript-Konsole von Google Chrome

Wenn du an der Cursorposition Code eintippst und dann die Enter-Taste drückst, führt JavaScript deinen Code aus und gibt das Ergebnis (falls es ein solches gibt) in der nächsten Zeile aus. Gib beispielsweise Folgendes in die Konsole ein:

```
3 + 4;
```

Jetzt drückst du Enter. JavaScript gibt die Antwort (7) auf diese einfache Addition in der folgenden Zeile aus:

```
3 + 4;
7
```

Das war ziemlich einfach. JavaScript ist aber mehr als ein besserer Taschenrechner. Wir probieren deshalb etwas anderes.

1.4 Die Struktur eines JavaScript-Programms

Jetzt erzeugen wir etwas Lustigeres: ein JavaScript-Programm, das eine Reihe von Katzengesichtern ausgibt, wie dieses:

```
=^.^=
```

Im Gegensatz zu unserem Additionsprogramm besteht dieses JavaScript-Programm aus mehreren Zeilen. Um das Programm in die Konsole einzugeben, musst du neue Zeilen hinzufügen, indem du am Ende jeder Zeile Umschalt+Enter drückst. (Wenn du nur Enter drücken würdest, würde Chrome versuchen, das Geschriebene gleich auszuführen und das Programm würde nicht wie erwartet funktionieren. Ich habe dich gewarnt: Computer sind dumm!)



Gib Folgendes in die Browserkonsole ein:

```
// Zeichne so viele Katzen, wie du willst!
var katzenZeichnen = function (wieViele) {
  for (var i = 0; i < wieViele; i++) {
    console.log(i + " =^.^=");
  }
};

katzenZeichnen(10); // Hier kannst du statt 10 jede beliebige Zahl
// einfügen.
```

Am Ende drückst du nicht Umschalt+Enter, sondern Enter. Dann solltest du die folgende Ausgabe sehen:

```
0 =^.^=
1 =^.^=
2 =^.^=
3 =^.^=
4 =^.^=
5 =^.^=
6 =^.^=
7 =^.^=
8 =^.^=
9 =^.^=
```

Wenn du dich vertippt hast, sieht deine Ausgabe möglicherweise ganz anders aus oder du bekommst sogar eine Fehlermeldung. Das meine ich damit, dass Computer dumm sind – selbst ein ganz einfacher Code muss perfekt sein, damit der Computer versteht, was er machen soll!

Ich erkläre im Moment noch nicht genau, wie der Code funktioniert (in Kap. 8 kehren wir zu diesem Programm zurück). Aber schauen wir uns mal einige Merkmale dieses speziellen Programms und von JavaScript-Programmen ganz allgemein an.



Syntax

Unser Programm enthält zahlreiche Symbole, zum Beispiel runde Klammern `()`, Strichpunkte `;`, geschweifte Klammern `{}`, Pluszeichen `+` sowie ein paar Wörter, die auf den ersten Blick merkwürdig erscheinen (wie etwa `var` und `console.log`). All das sind Teile der JavaScript-Syntax – der JavaScript-Regeln, mit denen man Symbole und Wörter zu funktionierenden Programmen zusammenfügt.

Wenn du eine neue Programmiersprache erlernst, ist es besonders schwierig, sich an die Regeln zu gewöhnen, wie man unterschiedliche Anweisungen für den Computer schreibt. Am Anfang vergisst man leicht, wann Klammern gesetzt werden müssen, und man bringt die Reihenfolge durcheinander, in der bestimmte Werte hinzugefügt werden müssen. Mit der Zeit bekommst du den Dreh aber heraus.

In diesem Buch arbeiten wir uns langsam und stetig voran. Neue Syntaxregeln werden Schritt für Schritt vorgestellt, sodass du immer leistungsfähigere Programme entwickeln kannst.

Kommentare

Die erste Zeile in unserem Katzenprogramm lautet:

```
// Zeichne so viele Katzen, wie du willst!
```

Das ist ein *Kommentar*. Programmierer verwenden Kommentare, damit andere Programmierer ihren Code leichter lesen und verstehen können. Der Computer hingegen ignoriert Kommentare vollständig. In JavaScript beginnen Kommentare mit zwei Schrägstrichen (`//`). Alles, was den Schrägstrichen in derselben Zeile folgt, wird vom JavaScript-Interpreter ignoriert. Die Kommentare haben deshalb keine Auswirkungen darauf, wie das Programm ausgeführt wird – sie haben nur eine beschreibende Funktion.

In diesem Buch siehst du im Code Kommentare, die dir beschreiben, was passiert. Wenn du deinen eigenen Code erzeugst, fügst du deine eigenen Kommentare hinzu. Siehst du dir deinen Code später an, erinnern dich deine Kommentare daran, wie er funktioniert und wozu jeder Schritt gut ist.

In der letzten Zeile unseres Programms siehst du einen etwas andersartigen Kommentar. Denke daran, dass alles nach dem doppelten Schrägstrich `//` vom Computer nicht ausgeführt wird!

```
katzenZeichnen(10); // Hier kannst du statt 10 jede beliebige Zahl
                      // einfügen.
```

Codekommentare können sich in einer eigenen Zeile befinden oder direkt auf den Code folgen. Wenn du an den Anfang die doppelten Schrägstriche setzt, etwa so, ...

```
// katzenZeichnen(10);
```

... dann passiert gar nichts! Chrome betrachtet die gesamte Zeile als Kommentar, auch wenn es sich dabei in Wirklichkeit um JavaScript-Code handelt.

Sobald du dich mit JavaScript-Code anderer Programme beschäftigst, wirst du auch Kommentare finden, die so aussehen:

```
/*
Zeichne so viele Katzen,
wie du willst!
*/
```

Dies ist wieder ein anderer Kommentarstil. Er wird normalerweise für Kommentare verwendet, die länger sind als eine Zeile. Das Ergebnis ist aber dasselbe: Alles zwischen `/*` und `*/` ist ein Kommentar, den der Computer nicht ausführt.

1.5 Das hast du gelernt

In diesem Kapitel hast du ein wenig gelernt, was JavaScript ist und wofür du es verwenden kannst. Du hast auch erfahren, wie du JavaScript mithilfe des Browsers Google Chrome ausführst, und ein Beispielprogramm ausprobierst. Für den gesamten Code in diesem Buch – außer wenn ich etwas anderes sage – kannst (und solltest!) du die Chrome-JavaScript-Konsole verwenden. Lies den Code nicht nur einfach – sondern versuche, ihn einzugeben! Nur so kannst du programmieren lernen.

Im nächsten Kapitel lernst du die Grundlagen von JavaScript kennen. Wir beginnen mit den drei grundsätzlichen Informationstypen: Zahlen, Strings und booleschen Werten.

