edv-buchversand.de

JavaScript Schnelleinstieg

Dynamische Webseiten programmieren in 14 Tagen

» Hier geht's direkt zum Buch

DAS VORWORT

Dieses Vorwort wird Ihnen von www.edv-buchversand.de zur Verfügung gestellt.

Einleitung

Programmieren lernen in 14 Tagen

Mit diesem Buch haben Sie sich für einen einfachen, praktischen und fundierten Einstieg in JavaScript entschieden. Sie lernen ohne unnötigen Ballast alles, was Sie wissen müssen, um JavaScript effektiv für Projekte in Ihrem Berufs- und Interessensgebiet einzusetzen. Alles, was Sie dafür benötigen, sind grundlegende Kenntnisse in HTML und CSS.

Wenn Sie Zeit genug haben, können Sie jeden Tag ein neues Kapitel durcharbeiten und so innerhalb von zwei Wochen Programmieren lernen. Am besten lesen Sie das Buch neben der Computer-Tastatur und probieren die Programmbeispiele und Übungen gleich aus. Sie werden schnell erste Erfolge erzielen und Freude an der Programmierung mit JavaScript finden.

Der Sandkasten – Programmbeispiele zum Download

Sie müssen die Beispiele in diesem Buch nicht abtippen. Der Code der Beispiele und zu den Übungen steht Ihnen auf der Webseite des Verlags unter *www.mitp.de/0778* zur Verfügung.

Fast immer liegen die Beispiel-Lösungen in einer externen JavaScript-Datei, sodass Sie die Wahl haben, sich eine eigene Lösung zu erarbeiten oder den Lösungsweg zu studieren.

Fragen und Feedback

Unsere Verlagsprodukte werden mit großer Sorgfalt erstellt. Sollten Sie trotzdem einen Fehler bemerken oder eine andere Anmerkung zum Buch haben, freuen wir uns über eine direkte Rückmeldung an `lektorat@mitp.de`.

Falls es zu diesem Buch bereits eine Errata-Liste gibt, finden Sie diese unter *www.mitp.de/0778* im Reiter DOWNLOADS.

Wir wünschen Ihnen viel Erfolg und Spaß bei der Programmierung mit JavaScript!

Vorwort

Sofort ins kalte Wasser – ein fremdes Skript lesen und verstehen

Die ersten Gehversuche mit JavaScript starten die meisten Webdesigner mit einem fertigen Skript, das sie z.B. auf GitHub finden und vielleicht in einzelnen Anweisungen nach ihrer Vorstellung ändern.

Für den ersten Schritt in die Programmierung mit JavaScript reicht eine normale Webseite. Das `script`-Element am Ende des Dokuments (vor dem schließenden body-Tag) braucht keinerlei Attribute. Innerhalb der öffnenden und schließenden `script`-Tags wirken JavaScript-Anweisungen nur auf dieser Seite.

HTML:

```
<!DOCTYPE html>
<html>
<head>
   <title>Sofort ins kalte Wasser!</title>
</head>
<body>
   <h1>Sofort ins kalte Wasser!</h1>
   <blockquote>
      <p class="hallo">Hallo Welt!</p>
      <p>Die klassische Einführung in die Programmierung.</p>
      <cite>Obi Web Kenobi</cite>
   </blockquote>
<script>
document.querySelector (“body”).style = “color: green”;
</script>
</body>
</html>
```

Der Aufruf `document.querySelector` ist das JavaScript-Äquivalent zum CSS-Selektor. In den runden Klammern des Aufrufs kann ein beliebiger CSS-Selektor stehen. In diesem Beispiel ist es ein HTML-Tag-Selektor. Nach den

schließenden runden Klammer erzeugt `.style` ein style-Attribut im öffnenden body-Tag und `"color: green"` ist eine ganz normale CSS-Regel.

`document.querySelector` ist sozusagen das JavaScript-Äquivalent eines CSS-Selektors und erreicht die Elemente über einen CSS-Selektor. Nach demselben Muster würde ein CSS-`class`-Selektor in den runden Klammern wirken. Schreiben Sie `".hallo"` in die runden Klammern und wählen Sie eine andere CSS-Regel, z.B. die Schriftgröße und die Hintergrundfarbe, um gezielt den Text im `p`-Element mit der Klasse `hallo` zu stylen.

```
document.querySelector(".hallo").style =
   "font-size: 24px; background: yellow";
```

Der CSS-Selektor wird mit demselben Punkt vor dem Klassennamen angesprochen wie in der CSS-Datei, das Semikolon trennt die beiden Regeln voneinander.

Das aufgeschobene Vorwort

Einsatz von JavaScript

In den frühen Zeiten des Internets gab es nur statische Seiten auf der Basis von HTML. Erst CSS, JavaScript und PHP haben das Internet populär gemacht. In den Anfängen brachte JavaScript Effekte und kleine Animationen für die Webseite und übernahm dann mehr die Rolle des Vermittlers zwischen der Anwendung auf dem Server und dem Benutzer. Heute gestaltet JavaScript nicht nur Webseiten, sondern sitzt auch hinter PDF-Dateien und in Photoshop, in den Anwendungen für mobile Geräte und ist der Kern von Node.js, einem weiteren Webserver neben Apache.

Dieses Buch konzentriert sich auf die Grundlagen von JavaScript für Webseiten, auf praktische Anwendungen von einfachen Effekten bis hin zur Prüfung und Übermittlung von Formulardaten.

- Der erste Teil widmet sich den einfachen Vokabeln und der Grammatik von JavaScript: Datentypen wie Zahlen und Zeichenketten, Objekten und Arrays, Kontrollstrukturen und Funktionen.
- Der zweite Teil dreht sich um die Anwendung dieser Grundlagen in Webseiten: Wie ändert man Inhalte, erkennt Ereignisse wie einen Mausklick, erzeugt HTML-Elemente on the fly und weist ihnen CSS zu?

- Die Reaktion auf Aktionen des Benutzers eröffnet schließlich den Weg in die Kommunikation mit dem Server: Wie funktioniert die Übergabe der Benutzereingaben aus dem Formular an die Anwendung auf dem Server?

Programmiersprachen vs. natürliche Sprachen

Programmiersprachen haben viel mit natürlichen Sprachen gemeinsam: Sie haben Vokabeln und eine Grammatik – ihre `Syntax`. So wie man eine natürliche Sprache nicht allein aus dem Lehrbuch lernt, brauchen die Vokabeln der Programmiersprache und ihre Anwendung praktische Übung. Es dauert, bis Sie eine Diskussion in italienischer Sprache führen können, aber für den Anfang ist es super, wenn wir den Fremdenführer gut verstehen. Das Verständnis für ein fremdes Skript ist ein großer Schritt, um eigene Programme zu entwickeln.

Bis Sie als Einsteiger den ersten Algorithmus für eine Aufgabe formulieren können, passiert dasselbe wie bei den ersten Sätzen einer frisch erlernten Sprache: Da stottert man schon mal, aber mit jedem Gespräch wird es flüssiger.

Zwei Schritte vor, ein Schritt zurück

Bei den Grundlagen von JavaScript stellen sich Einsteiger die Frage: »Was mache ich mit dieser ganzen Theorie überhaupt?«. Dabei geht es weder um Vollständigkeit noch um technische Akkuratesse, sondern um das Verständnis für die Konzepte. Erst wenn es nach der Vorstellung von einfachen Datentypen, Arrays, Objekten und Funktionen an das Ändern des HTML-Dokuments (das `Document Object Model`) geht, offenbaren sich viele Konzepte von JavaScript. Das vorliegende Buch greift darum an vielen Stellen zwei Schritte vor und verweist andererseits aus den Kapiteln zur Anwendung von JavaScript in Webseiten zurück auf die Grundlagen. Keine Sorge: Der praktische Umgang wirft ein neues Licht auf die Konzepte und bringt das tiefe Verständnis.

JavaScript-Versionen und Browser-Versionen

ECMAScript und JavaScript

JavaScript wurde 1995 von Netscape entwickelt und von Microsoft als JScript adaptiert. Ein Standard wurde aber erst durch die ECMA (European Computer Manufacturers Association) erreicht.

Die Grundlagen von JavaScript werden von der ECMA standardisiert, das W3C steuert alles rund um das Document Object Model zu, also den Teil von JavaScript, der sich speziell auf Webseiten bezieht. Auf jeden Fall aber werden JavaScript-Versionen als ECMAScript veröffentlicht. ECMAScript 6 war ein großer Sprung im Jahr 2015 und ist eine stabile Version für alle modernen Browser. Heute nutzt die ECMA Jahreszahlen anstelle von Versionsnummern und beschert uns jedes Jahr mit einer neuen Versionsnummer.

Das läuft nicht anders als bei neuen Elementen in HTML oder neuen Eigenschaften für CSS: Bis neue Versionen für die Programmierung von Webseiten genutzt werden können, muss sicher sein, dass die Browser die Neuheiten auf einer breiten Basis unterstützen. Darum gilt die Version 6 von JavaScript bis heute als Eckpfeiler für Online-Anwendungen.

Browser-Unterstützung

Das Erscheinen einer neuen Version bedeutet natürlich nicht, dass alle Browser direkt auf den Zug aufspringen. Als Entwickler fiebert man dem einen oder anderen lang ersehnten Feature entgegen, aber wir können es in der Regel erst nutzen, wenn die Mainstream-Browser die neuen Funktionen in ihre Versionen eingebaut haben und der größte Teil der potenziellen Benutzer wiederum auf die aktuellen Versionen der Browser umgestiegen ist.