

19.6	Zeichenwerkzeuge	354
19.7	Speichern und Laden	356
19.8	Der Undo-Verlauf	359
19.9	Die Anwendung einrichten	360
19.10	Warum ist das so schwer?	361
19.11	Übungen	362

Teil III Node.js **365**

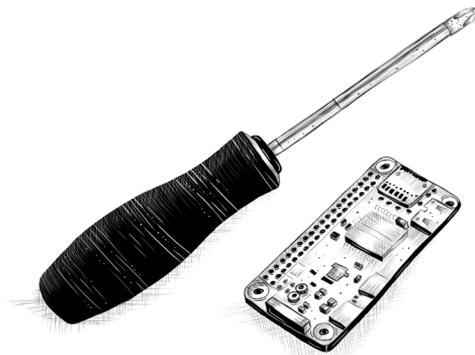
20	Einführung in Node.js	367
20.1	Hintergrund	367
20.2	Der Befehl node	368
20.3	Module	369
20.4	Installation mit NPM	370
20.5	Das Dateisystemmodul	372
20.6	Das HTTP-Modul	374
20.7	Streams	376
20.8	Ein Dateiserver	377
20.9	Zusammenfassung	382
20.10	Übungen	383
21	Projekt: Eine Website zur Wissensvermittlung	387
21.1	Design	388
21.2	Long Polling	389
21.3	Die HTTP-Schnittstelle	389
21.4	Der Server	391
21.5	Der Client	398
21.6	Übungen	404
22	Leistung	407
22.1	Stufenweise Kompilierung	408
22.2	Graphzeichnen	408
22.3	Definition eines Graphen	410
22.4	Kräftebasiertes Graphzeichnen	411
22.5	Arbeit vermeiden	413
22.6	Profiling	415
22.7	Inline-Ersetzung	417

22.8	Weniger überflüssige Objekte erzeugen	418
22.9	Garbage Collection	419
22.10	Dynamische Typen	420
22.11	Zusammenfassung	421
22.12	Übungen	422
	Hinweise zu den Übungen	425
	Stichwortverzeichnis	449

Für Lotte und Jan

»Wir glauben, dass wir das System zu unserem eigenen Nutzen erstellen. Wir glauben, dass wir es nach unserem eigenen Bilde erschaffen. [...] Aber der Computer ist nicht wie wir. Er ist eine Projektion eines sehr kleinen Teils von uns: des Teils, der Logik, Ordnung, Regeln und Klarheit zugetan ist.«

– Ellen Ullman, *Close to the Machine: Technophilia and its Discontents* (auf Deutsch erschienen als *Close to the Machine: Mein Leben mit dem Computer*)



Einleitung

In diesem Buch geht es darum, Computern Anweisungen zu geben. Computer sind heutzutage so alltäglich geworden wie Schraubenzieher, allerdings deutlich komplizierter. Deshalb ist es nicht immer einfach, sie auch wirklich das tun zu lassen, was man will.

Wenn Sie Ihren Computer für eine übliche und klar umrissene Aufgabe einsetzen möchten, z. B. um E-Mails anzuzeigen oder Berechnungen wie mit einem Taschenrechner durchzuführen, können Sie einfach die entsprechende Anwendung öffnen und loslegen. Für besondere Aufgaben gibt es dagegen möglicherweise noch keine Anwendung.

An dieser Stelle kommt die *Programmierung* ins Spiel. Dabei handelt es sich um den Vorgang, ein *Programm* zu erstellen, also eine Folge genauer Anweisungen, die dem Computer sagen, was er tun soll. Da Computer stumpfsinzig und pedantisch sind, ist Programmierung im Grunde zunächst mühselig und frustrierend.

Wenn Sie jedoch darüber hinwegkommen und vielleicht sogar Freude an einer Denkweise in strengen Bahnen finden, die auch eine stumpfe Maschine versteht, kann Programmierung lohnenswert sein. Denn damit lassen sich in Sekunden Dinge erledigen, die sonst ewig dauern würden. Sie bietet eine Möglichkeit, Ihr Werkzeug, den Computer, Aufgaben ausführen zu lassen, die er zuvor nicht beherrschte. Und außerdem ist es eine hervorragende Übung für abstraktes Denken.

Programmierung erfolgt mithilfe einer *Programmiersprache*. Dabei handelt es sich um eine künstliche Sprache, die dazu dient, Computern Anweisungen zu erteilen. Es ist schon bemerkenswert, dass sich die effektivste Möglichkeit zur Kommunikation mit Computern, die wir erfunden haben, so stark an die Art und Weise anlehnt, wie wir miteinander kommunizieren. Ebenso wie in menschlichen Sprachen können auch in Computersprachen Wörter und Ausdrücke kombiniert werden, um Ideen auszudrücken.

Textschnittstellen wie die BASIC- und DOS-Eingabeaufforderungen der 80er und 90er bildeten einst die Hauptmethode für die Kommunikation mit Computern. Mittlerweile wurden sie größtenteils durch grafische Schnittstellen ersetzt,