

Coding für Dummies

Alles-in-einem-Band

» Hier geht's
direkt
zum Buch

DIE LESEPROBE

Erkennen, was Code ist – und was er Ihnen ermöglicht

Den Code einer Web-App genauer betrachten

Programmiersprachen kennenlernen, mit denen Code geschrieben wird

Kapitel 1

Was ist Programmierung?

»Eine Million Dollar ist nicht cool, aber weißt du, was cool ist? Eine Milliarde Dollar.«

- SEAN PARKER, DAS SOZIALE NETZWERK

Beinahe wöchentlich berichten die Medien über Technologieunternehmen, die an die Börse gehen oder für Millionen von Dollar verkauft werden. Nicht selten geht es dabei um Milliarden Dollar, wie etwa bei Zoom, Coinbase oder Squarespace zu beobachten war. Vielleicht haben diese Artikel Ihre Neugierde geweckt, und Sie möchten erfahren, wie Code aussehen könnte, mit dem Apps geschrieben werden, die diese finanziellen Ergebnisse erzielen. Vielleicht hat Ihr Interesse aber auch mehr mit Ihrem Job zu tun. Vielleicht arbeiten Sie in einer Branche, die sich im Niedergang befindet, oder in einer Funktion, die sich technologisch rasch verändert. Ganz gleich, ob Sie über einen Berufswechsel nachdenken oder Ihre derzeitige Karriere verbessern wollen, das Verständnis der Programmierung kann Ihnen bei Ihrer beruflichen Entwicklung helfen. Vielleicht haben Sie aber auch ein persönliches Interesse – eine Idee, ein brennendes Verlangen, etwas zu schaffen, eine Website oder eine App, um ein Problem zu lösen, das Ihnen am Herzen liegt, und Sie ahnen, dass das Lesen und Schreiben von Code der erste Schritt zur Entwicklung Ihrer Lösung ist. Was auch immer Ihre Motivation ist, dieses Buch wird Licht auf die Programmierung und Programmierer werfen und Ihnen helfen, den Ansatz nicht als mysteriös und komplex zu betrachten, sondern als etwas, das Sie selbst nutzen können.

In diesem Kapitel erfahren Sie, was Code ist, für welche Branchen Computersoftware besonders wichtig ist, welche verschiedenen Arten von Programmiersprachen zum Schreiben von Code verwendet werden und wie eine Web-App mit Code erstellt wird.

Definieren, was Code ist

Computercode zu schreiben, ist keine kryptische Tätigkeit, die Genies und Orakeln vorbehalten ist. Tatsächlich werden Sie in wenigen Minuten selbst Computercode schreiben! Der meiste Computercode führt bestimmte Aufgaben in unserer Welt aus, die ganz alltäglich, aber auch völlig außergewöhnlich sein können. Code steuert unsere Ampeln und Fußgänger signale, die Aufzüge in unseren Gebäuden, die Handymasten, die unsere Telefonsignale übertragen, und die Raumschiffe, die ins Weltall fliegen. Wir haben es aber auch auf sehr persönlicher Ebene mit Code zu tun, wenn wir beispielsweise auf unseren Telefonen und Computern E-Mails oder das Wetter abrufen.

Anweisungen befolgen

Computercode besteht aus einer Reihe von Anweisungen, ganz ähnlich den Sätzen in unserer natürlichen Sprache. Jede Anweisung befiehlt dem Computer, einen einzelnen Schritt oder Befehl auszuführen. Jeder dieser Schritte ist sehr präzise formuliert und wird genau wie angegeben ausgeführt. Wenn Sie zum Beispiel in einem Restaurant einen Kellner bitten, Ihnen den Weg zur Toilette zu zeigen, könnte dieser sagen: »Gehen Sie nach hinten und versuchen Sie es mit der mittleren Tür«. Diese Anweisungen sind so vage, dass sie für einen Computer völlig unbrauchbar sind. Würde Ihnen der Kellner stattdessen Anweisungen geben, als wären Sie ein Computerprogramm, könnten diese lauten: »Gehen Sie von diesem Tisch aus 40 Schritte nach Nordosten. Dann drehen Sie sich um 90 Grad nach rechts, gehen 5 Schritte, drehen sich um 90 Grad nach links und gehen 5 Schritte. Öffnen Sie die Tür, der Sie jetzt gegenüberstehen, und betreten Sie die Toilette«. Abbildung 1.1 zeigt Codezeilen aus dem beliebten Spiel Pong. Machen Sie sich keine Gedanken darüber, was jede einzelne Zeile bewirkt, und lassen Sie sich nicht einschüchtern. Sie werden bald schon Ihren eigenen Code lesen und schreiben.

```

1- launchPong(function () {
2-   function colour_random() {
3-       var num = Math.floor(Math.random() * Math.pow(2, 24));
4-       return '#' + ('00000' + num.toString(16)).substr(-6);
5-   }
6-
7-
8-   pongSettings.ball.size = 15;
9-   pongSettings.ball.color = colour_random();
10-  pongSettings.ball.velocity[0] = 15;
11-  pongSettings.ball.velocity[1] = 15;
12-
13- });
14
15

```

Abbildung 1.1: Computercode aus dem Spiel Pong.

Die Komplexität eines Programms kann annähernd gemessen werden, indem seine Anweisungen oder Codezeilen gezählt werden. Einfache Anwendungen wie das Pong-Spiel haben 5.000 Codezeilen, während komplexere Anwendungen wie Facebook derzeit über 10 Millionen Codezeilen haben. Unabhängig davon, ob es sich um wenige oder viele Codezeilen

handelt, befolgt der Computer jede Anweisung genau und mühelos und wird nie müde, so wie ein Kellner, der zum hundertsten Mal nach der Toilette gefragt wird.



Seien Sie vorsichtig, wenn Sie nur Codezeilen als Maß für die Komplexität eines Programms verwenden. 100 gut geschriebene Codezeilen können dieselbe Funktion erfüllen wie 1.000 schlecht geschriebene Codezeilen.

Code schreiben – mit ein paar Angry Birds

Wenn Sie noch nie Code geschrieben haben, haben Sie jetzt die Gelegenheit, es zu versuchen! Gehen Sie auf <https://hourofcode.com/us/learn> und scrollen Sie nach unten (oder verwenden Sie das Suchfeld), um den Link »Schreibe dein erstes Computerprogramm« mit dem Angry-Birds-Symbol zu finden, wie in Abbildung 1.2 gezeigt. Dieses Lernprogramm ist für all diejenigen gedacht, die noch keine Erfahrung mit der Computerprogrammierung haben. Es führt in die grundlegenden Bausteine ein, die von allen Computerprogrammen verwendet werden. Das Wichtigste, was Sie daraus mitnehmen können, ist zu verstehen, dass Computerprogramme Code verwenden, um dem Computer buchstäblich und genau zu sagen, dass er eine Reihe von Anweisungen ausführen soll.

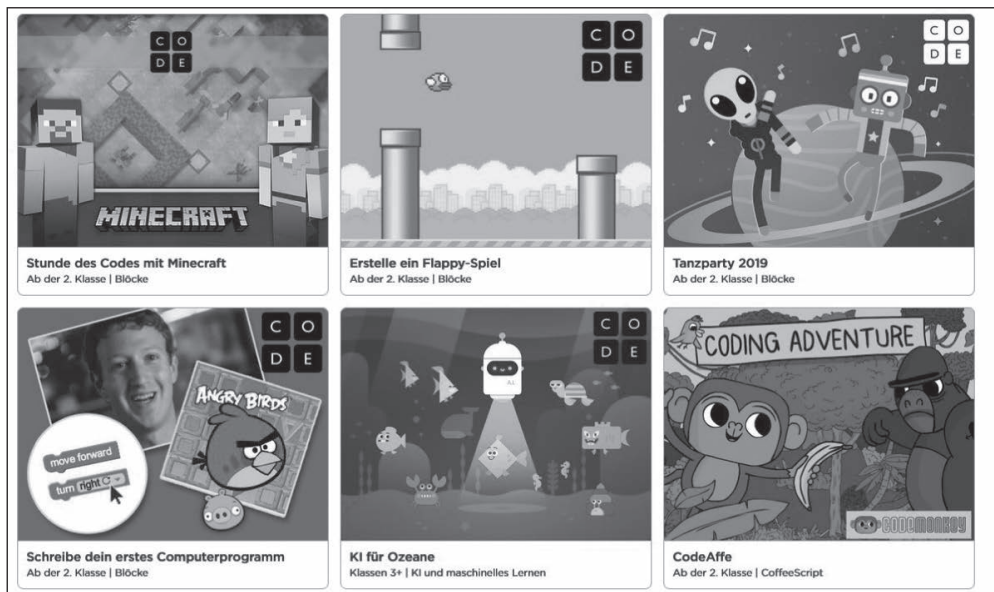


Abbildung 1.2: Schreiben Sie Ihr erstes Computerprogramm mit einem spieleähnlichen Lernprogramm unter Verwendung von Angry Birds.



Die »Stunde des Codes« ist eine jährliche Initiative, die während einer Woche im Dezember das Profil der Informatik schärfen soll. In der Vergangenheit haben Präsident Obama, Bill Gates, der Basketballspieler Stephen Curry, die Sängerin Shakira und sogar der bescheidene Autor dieses Buches Menschen aus den Vereinigten Staaten und der ganzen Welt unterstützt und zur Teilnahme ermutigt.

Verstehen, wie Programmierung Sie unterstützen kann

Mit Hilfe der Programmierung können Aufgaben und Probleme gelöst werden, die Ihnen jeden Tag begegnen. Die Anzahl der »alltäglichen« Situationen, in denen Programme oder Apps Sie unterstützen können, wächst exponentiell – was nicht immer der Fall war. Mit dem Aufkommen von Web-Apps, Internetverbindungen und Mobiltelefonen haben Softwareprogramme Einzug in das tägliche Leben gehalten und die Hürde für Sie gesenkt, selbst zum Programmierer zu werden und persönliche und berufliche Probleme mit Hilfe von Code zu lösen.

Software übernimmt die Welt

2011 erklärte Marc Andreessen, Erfinder eines der ersten Webbrowser, Netscape Navigator, und heute Risikokapitalgeber: »Software wird die Welt übernehmen«. Er sagte voraus, dass Softwareunternehmen bestehende Unternehmen in rasantem Tempo verdrängen würden. Früher brauchte man Software, um Programme auf Desktops und Laptops auszuführen. Die Software musste zunächst installiert werden, und dann musste man dem Programm Daten zur Verfügung stellen. Mittlerweile haben drei Trends die Verwendung von Code im täglichen Leben drastisch ausgeweitet:

- ✓ **Webbasierte Software:** Diese Software wird innerhalb des Browsers ausgeführt, ohne dass eine Installation erforderlich ist. Wenn Sie früher zum Beispiel Ihre E-Mails abrufen wollten, mussten Sie einen E-Mail-Client installieren, entweder durch Herunterladen der Software oder von einer CD-ROM. Manchmal traten Probleme auf, wenn die Software nicht für Ihr Betriebssystem verfügbar war oder mit Ihrer Betriebssystemversion in Konflikt geriet. Hotmail, ein webbasierter E-Mail-Client, wurde unter anderem deshalb so beliebt, weil er es den Benutzern ermöglichte, ihre E-Mails auf `www.hotmail.com` sofort abzurufen, ohne sich Gedanken über die Installation oder Softwarekompatibilität machen zu müssen. Web-Apps steigerten den Appetit der Verbraucher, weitere Apps auszuprobieren, was wiederum für die Entwickler einen Anreiz darstellte, weitere Apps zu entwickeln.
- ✓ **Internet-Breitbandanschlüsse:** Die Zahl der Breitbandanschlüsse hat zugenommen, sodass in den letzten Jahren immer mehr Menschen einen schnellen Internetzugang erhalten haben. Heute können mehr als zwei Milliarden Menschen auf webbasierte Software zugreifen, vor zehn Jahren waren es nur etwa 50 Millionen.
- ✓ **Handys:** Mit den heutigen Smartphones kann man Programme überallhin mitnehmen und die Programme mit Daten versorgen. Viele Softwareprogramme wurden durch den Zugriff von unterwegs aus sehr viel nützlicher, weil sie nicht mehr auf einen statischen Desktop-Computer beschränkt waren. So hat beispielsweise die Nutzung von Kartenanwendungen dank der Mobiltelefone stark zugenommen, da die Benutzer am meisten nach Wegbeschreibungen suchen, wenn sie sich verirrt haben, und nicht nur, wenn sie zu Hause am Computer eine Reise planen. Darüber hinaus sind

Mobiltelefone mit Sensoren ausgestattet, die Daten wie Richtung, Beschleunigung und aktuelle Position über GPS messen und an Programme weitergeben. Statt alle Daten selbst in Programme eingeben zu müssen, können mobile Geräte dabei helfen. Bei einer Fitnessanwendung wie Runkeeper müssen Sie zum Beispiel keine Start- und Endzeiten eingeben, um Ihre Läufe zu verfolgen. Sie drücken zu Beginn Ihres Laufs auf »Start«, und das Telefon zeichnet automatisch Ihre Strecke, Geschwindigkeit und Zeit auf.

Die Kombination dieser Trends hat Softwareunternehmen hervorgebracht, die etablierte Unternehmen in fast jeder Branche auf den Kopf gestellt haben, insbesondere in Bereichen, die ursprünglich immun gegen Technologie waren. Einige bemerkenswerte Beispiele sind:

- ✓ **Airbnb:** Airbnb ist ein Peer-to-Peer-Unternehmensunternehmen, das keine eigenen Zimmer besitzt, aber mehr Übernachtungen bucht als Hilton und Intercontinental, die größten Hotelketten der Welt (siehe Abbildung 1.3).
- ✓ **Uber:** Uber ist ein Auto-Transportunternehmen, das keine eigenen Fahrzeuge besitzt, mehr Fahrten bucht und mehr Fahrer in den 200 größten Städten hat als jeder andere Auto- oder Taxidienst.
- ✓ **Groupon:** Groupon, das Unternehmen für täglich neue Angebote, hat in nur zwei Jahren fast 1 Milliarde Dollar erwirtschaftet und ist damit schneller gewachsen als jedes andere Unternehmen in der Geschichte, ganz zu schweigen von jedem anderen traditionellen Direktmarketing-Unternehmen.

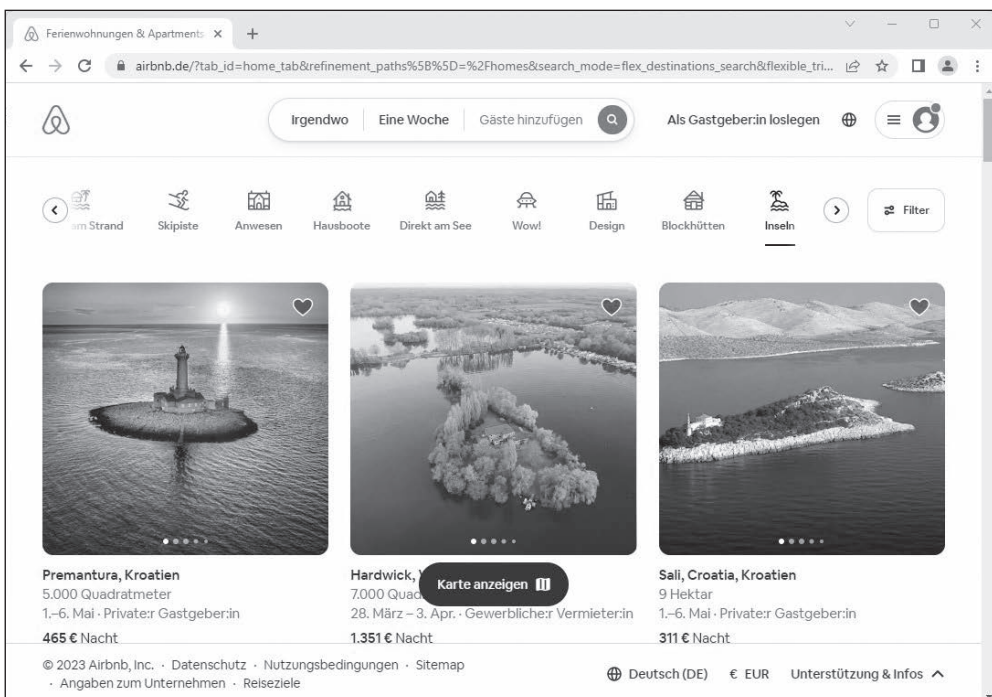


Abbildung 1.3: Airbnb buchte 5 Millionen Übernachtungen nach 3,5 Jahren – und die nächsten 5 Millionen Übernachtungen 6 Monate später.

Programmierung bei der Arbeit

Auch am Arbeitsplatz kann Programmierung nützlich sein. Außerhalb des Technologiesektors ist die Programmierung am Arbeitsplatz in einigen Berufen üblich, wie beispielsweise bei Finanzhändlern, Wirtschaftlern oder Wissenschaftlern. Für die meisten Berufe außerhalb des Technologiesektors beginnt die Programmierung jedoch gerade erst, den Arbeitsplatz einzunehmen, und gewinnt allmählich an Bedeutung. Hier Bereiche, in denen die Programmierung eine größere Rolle am Arbeitsplatz spielt:

- ✓ **Werbung:** Unternehmen verlagern ihre Ausgaben von Print- und TV-Werbung auf digitale Kampagnen, und die Suchmaschinenwerbung und -optimierung stützt sich auf Schlüsselwörter, um Besucher auf Websites zu bringen. Werbetreibende, die Code verstehen, erkennen erfolgreiche Schlüsselwörter, die von Wettbewerbern verwendet werden, und nutzen diese Daten, um effektivere eigene Kampagnen zu erstellen.
- ✓ **Marketing:** Bei der Werbung für Produkte ist die Personalisierung der Kommunikation eine Strategie, die oft zu besseren Ergebnissen führt. Vermarkter, die programmieren, können Kundendatenbanken abfragen und personalisierte Mitteilungen erstellen, die Kundennamen und auf bestimmte Interessen zugeschnittene Produkte enthalten.
- ✓ **Vertrieb:** Der Verkaufsprozess beginnt immer mit Kontakten. Vertriebsmitarbeiter, die programmieren, holen sich ihre eigenen Kontakte von Webseiten und Verzeichnissen und sortieren und qualifizieren diese Kontakte anschließend.



Das Abrufen von Informationen durch Kopieren von Text auf Webseiten und in Verzeichnissen wird als *Scraping* bezeichnet.

- ✓ **Design:** Nach dem Erstellen einer Webseite oder eines digitalen Entwurfs müssen Designer andere Designer und schließlich Entwickler davon überzeugen, ihre Zeichnungen in die Programmierung für das Produkt zu übernehmen. Designer, die programmieren können, können ihre Entwürfe leichter zum Leben erwecken und effektiver für bestimmte Entwürfe eintreten, indem sie funktionierende Prototypen erstellen, mit denen andere interagieren können.
- ✓ **PR:** Unternehmen messen ständig, wie Kunden und die Öffentlichkeit auf Ankündigungen und Nachrichten reagieren. Wenn zum Beispiel ein prominenter Botschafter eines Unternehmens etwas Fragwürdiges tut oder sagt, sollte das Unternehmen dann die Zusammenarbeit mit dem Prominenten beenden?
PR-Mitarbeiter, die programmieren können, fragen soziale Netzwerke wie Twitter oder Facebook ab und analysieren Hunderttausende einzelner Nachrichten, um die Stimmung auf dem Markt zu verstehen.
- ✓ **Operatives Geschäft:** Zusätzliche Gewinne können zum Teil durch die Analyse der Kosten eines Unternehmens erzielt werden. Programmierer schreiben Programme, mit denen sie Millionen von Kombinationen ausprobieren, um Verpackungsmethoden, Verladeroutinen und Lieferwege zu optimieren.

Sich selbst auf die Sprünge helfen (und reich und berühmt werden)

Bei der Verwendung von Code, der von anderen erstellt wurde, denken Sie vielleicht irgendwann über Probleme nach, die Sie mit eigenem Code lösen könnten. Vielleicht haben Sie eine Idee für eine Website für ein soziales Netzwerk, eine bessere Fitness-App oder etwas völlig Neues. Der Weg von einer Idee zu einem funktionierenden Prototyp, der von anderen genutzt wird, ist mit viel Zeit und Arbeit verbunden, aber womöglich leichter zu bewältigen, als Sie denken. Nehmen Sie zum Beispiel Coffitivity, eine Website, die Umgebungsgeräusche aus einem Café streamt, um eine produktive Arbeitsumgebung zu erzeugen. Die Website wurde von zwei Personen erstellt, die erst ein paar Monate zuvor das Programmieren gelernt hatten. Kurz nach dem Start von Coffitivity wurde die Website vom *TIME*-Magazin zu einer der 50 besten Websites des Jahres 2013 gekürt, und auch das *Wall Street Journal* hat die Website bewertet. Auch wenn nicht jedes Start-up oder jede App anfangs so viel Aufmerksamkeit in den Medien erhält, kann es hilfreich sein zu wissen, was möglich ist, wenn eine Lösung ein Problem wirklich löst.

Ein Ziel zu haben, zum Beispiel eine Website oder eine App zu erstellen, die ein bestimmtes Problem löst, ist eine der besten Möglichkeiten, das Programmieren zu lernen. Wenn Sie mit einer schwierigen Aufgabe oder einem komplizierten Konzept konfrontiert sind, wird die Idee, eine Website zum Leben zu erwecken, Ihnen die nötige Motivation geben. Genauso wichtig ist, dass Sie nicht nur deshalb Programmieren lernen wollen, um reich und berühmt zu werden, denn die Wahrscheinlichkeit, dass Ihre Website oder App erfolgreich wird, hängt größtenteils von Faktoren ab, auf die Sie keinen Einfluss haben.



Merkmale, die eine Website oder eine App suchterzeugend machen, werden anhand des Hook-Modells unter <https://www.nirandfar.com/how-to-manufacture-desire/> beschrieben. Produkte werden in der Regel von Unternehmen hergestellt, und die Merkmale eines dauerhaft erfolgreichen Unternehmens werden auf <https://articles.sequoiacap.com/elements-of-enduring-companies> beschrieben, basierend auf einer Untersuchung von Unternehmen, die von Sequoia finanziert werden, einer der erfolgreichsten Risikokapitalfirmen der Welt und frühem Investor in Apple, Google und PayPal.

Ein Überblick über die verschiedenen Arten von Programmiersprachen

Code gibt es in verschiedenen Geschmacksrichtungen, den so genannten *Programmiersprachen*. Man schätzt, dass es etwa 9.000 verschiedene Programmiersprachen gibt, deshalb können wir natürlich nicht über alle sprechen. Einige der beliebtesten sind JavaScript, Python, Java, Rust, Ruby, Go, Kotlin, R, PHP, C, C++, Dart und Swift.

Man kann sich Programmiersprachen wie gesprochene Sprachen vorstellen, denn sie haben viele Gemeinsamkeiten, wie zum Beispiel:

- ✓ **Sprachübergreifende Funktionalität:** Programmiersprachen können alle die gleiche Funktionalität erzeugen, ähnlich wie gesprochene Sprachen alle die gleichen Objekte, Sätze und Gefühle ausdrücken können.
- ✓ **Syntax und Struktur:** Befehle in Programmiersprachen können sich überschneiden, so wie sich Wörter in gesprochenen Sprachen überschneiden. Um in Python oder Ruby Text auf dem Bildschirm auszugeben, verwenden Sie den Befehl `print`, so wie *imprimer* und *imprimir* die Verben für »drucken« im Französischen und Spanischen sind.
- ✓ **Natürliche Lebensspanne:** Programmiersprachen entstehen, wenn ein Programmierer eine neue oder einfachere Möglichkeit findet, ein Konzept auszudrücken, um mit dem Computer ein Problem zu lösen. Wenn andere Programmierer dem zustimmen, übernehmen sie die Sprache für ihre eigenen Programme und die Programmiersprache verbreitet sich. Wenn die Programmiersprache jedoch, genau wie Latein oder Aramäisch bei den natürlichen Sprachen, nicht von weiteren Programmierern übernommen wird oder eine bessere Sprache auftaucht, dann stirbt die Programmiersprache langsam aus Mangel an Verwendung aus.

Trotz dieser Ähnlichkeiten unterscheiden sich Programmiersprachen in einigen wichtigen Punkten von gesprochenen Sprachen:

- ✓ **Ein Urheber:** Im Gegensatz zu gesprochenen Sprachen können Programmiersprachen von einer Person in kurzer Zeit, manchmal in nur wenigen Tagen, geschaffen werden. Beliebte Sprachen mit einem einzigen Urheber sind JavaScript (Brendan Eich), Python (Guido van Rossum) und Ruby (Yukihiro Matsumoto).
- ✓ **Englische Begriffe:** Im Gegensatz zu gesprochenen Sprachen (außer natürlich Englisch) sind fast alle Programmiersprachen auf Englisch geschrieben. Ob sie nun in HTML, JavaScript, Python oder Ruby programmieren – brasilianische, französische oder chinesische Programmierer verwenden alle dieselben englischen Schlüsselwörter und Syntax in ihrem Code. Es gibt zwar einige nicht-englische Programmiersprachen, wie zum Beispiel in Hindi oder Arabisch, aber keine dieser Sprachen ist weit verbreitet oder hat sich durchgesetzt.

Vergleich von Low-Level- und High-Level-Programmiersprachen

Eine Möglichkeit, Programmiersprachen zu klassifizieren, ist die Einteilung in Low-Level-Sprachen und High-Level-Sprachen. Low-Level-Sprachen interagieren direkt mit dem Computerprozessor oder der CPU, sind in der Lage, sehr einfache Befehle auszuführen und sind im Allgemeinen schwer zu lesen. Maschinencode, ein Beispiel für eine Low-Level-Sprache, verwendet einen Code, der nur aus zwei Ziffern besteht – 0 und 1. Abbildung 1.4 zeigt ein Beispiel für Maschinencode. Assembler, eine weitere Low-Level-Sprache, verwendet Schlüsselwörter, um grundlegende Befehle wie das Lesen, Verschieben und Speichern von Daten auszuführen.

```

011010100110101000101101100100101100101010101001010101
01111000101011110001101110111000101010010101001101010100
01010100010010010110101000101001011100011001010100100110
00110101010111101011011110100100100010110101010100000101
00110101001101010001011011001001010110010101010100101010
10111100010101111000110111011100010101001010100110101010
00101010001001001011010100010100101110001100101010010011
00011010101011110101101111010010010001011010101010000010
00110101001101010001011011001001010110010101010100101010
10111100010101111000110111011100010101001010100110101010
00101010001001001011010100010100101110001100101010010011
00011010101011110101101111010010010001011010101010000010
00110101001101010001011011001001010110010101010100101010
10111100010101111000110111011100010101001010100110101010
00101010001001001011010100010100101110001100101010010011
00011010101011110101101111010010010001011010101010000010
00110101001101010001011011001001010110010101010100101010
10111100010101111000110111011100010101001010100110101010
00101010001001001011010100010100101110001100101010010011
00011010101011110101101111010010010001011010101010000010
000110101011110101101111010010010001011010101010000010
00110101001101010001011011001001010110010101010100101010
10111100010101111000110111011100010101001010100110101010
00101010001001001011010100010100101110001100101010010011

```

Abbildung 1.4: Maschinencode besteht aus 0en und 1en.

Im Gegensatz dazu verwenden High-Level-Sprachen eine Art natürlicher Sprache, die für den Menschen leichter zu lesen und zu schreiben ist. Wenn Code in einer High-Level-Sprache wie C++, Python oder Ruby geschrieben ist, übersetzt ein Interpreter oder Compiler diese High-Level-Sprache in einen Code auf niedriger Ebene, den ein Computer verstehen kann.

Vergleich von kompiliertem und interpretiertem Code

High-Level-Programmiersprachen müssen je nach Sprache mit Hilfe eines Interpreters oder Compilers in Low-Level-Programmiersprachen umgewandelt werden. Interpretierte Sprachen gelten als portabler als kompilierte Sprachen, während kompilierte Sprachen schneller ausgeführt werden als interpretierte Sprachen. Der Geschwindigkeitsvorteil, den kompilierte Sprachen haben, verliert jedoch allmählich an Bedeutung, da die Leistungsunterschiede zwischen interpretierten und kompilierten Sprachen aufgrund steigender Prozessorgeschwindigkeiten vernachlässigbar werden.

High-Level-Programmiersprachen wie JavaScript, Python und Ruby werden interpretiert. Bei diesen Sprachen führt der Interpreter das Programm direkt aus und übersetzt jede Anweisung *Zeile für Zeile* in Maschinencode. High-Level-Programmiersprachen wie C++, COBOL und Visual Basic werden kompiliert. Bei diesen Sprachen übersetzt ein Compiler nach dem Schreiben des Codes *den gesamten* Code in Maschinencode, und es wird eine ausführbare Datei erstellt. Diese ausführbare Datei wird dann über eine Website oder einen App-Store verbreitet und ausgeführt. Software, die Sie auf Ihrem Computer, Smartphone oder Tablet installieren, wie Microsoft Office oder Adobe Photoshop, wird in kompilierten Sprachen programmiert.

Programmierung für das Web

Software, die über Websites zugänglich ist, beginnt allmählich, die installierte Software zu verdrängen. Denken Sie an das letzte Mal, als Sie Software für Ihren Computer heruntergeladen und installiert haben – vielleicht erinnern Sie sich gar nicht mehr daran! Installierte

Software wie Windows Media Player und Winamp, die Musik und Filme abspielen, wurden durch Websites wie YouTube und Netflix ersetzt. Traditionelle installierte Textverarbeitungs- und Tabellenkalkulationsprogramme wie Microsoft Word und Excel konkurrieren mit Web-Software wie Google Docs und Sheets. Das Google Chromebook enthält keine installierte Software (außer dem Chrome-Browser von Google) und verlässt sich stattdessen ausschließlich auf Web-Software, um Funktionen bereitzustellen.

Ein Großteil dieses Buches befasst sich mit der Entwicklung und Erstellung von Web-Software, nicht nur, weil der Bedarf an Web-Software schnell zunimmt, sondern auch, weil Programme für das Web einfacher zu erlernen und zu starten sind als herkömmliche installierte Software.

Eine mit Code erstellte Web-App

Nach all diesen Informationen über die Programmierung wollen wir jetzt eine mit Code erstellte Web-App ansehen. Yelp.com ist eine Website, auf der Sie Bewertungen von Menschen für lokale Unternehmen wie Restaurants, Nachtleben und Einkaufsmöglichkeiten suchen und finden können. Wie in Abbildung 1.5 zu erkennen ist, sah Yelp nicht immer so ausgefeilt aus wie heute, aber sein Zweck ist im Laufe der Jahre relativ konstant geblieben.

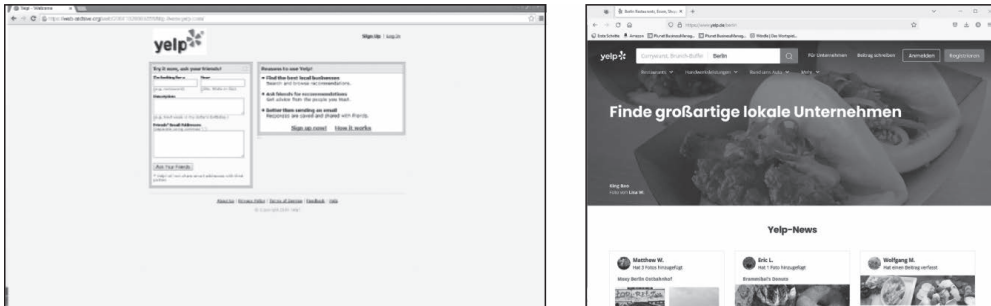


Abbildung 1.5: Die Website von Yelp im Jahr 2004 und im Jahr 2022.

Zweck und Umfang der Anwendung festlegen

Sobald Sie den Zweck einer App verstanden haben, können Sie Aufgaben identifizieren, die ein Benutzer ausführen können sollte, um diesen Zweck zu erreichen. Unabhängig vom Design hat die Yelp-Website den Nutzern schon immer ermöglicht:

- ✓ Lokale Angebote nach Art und Ort des Veranstaltungsortes zu suchen.
- ✓ Die Ergebnisse nach Adresse, Öffnungszeiten, Bewertungen, Fotos und Standort auf einer Karte zu durchsuchen.

Erfolgreiche Web-Apps unterstützen in der Regel nur einige wenige wichtige Aufgaben bei der Nutzung der App. Zu viele Funktionen in einer App verwässern die Stärke der wichtigsten Funktionen, deshalb vermeiden die meisten Entwickler, immer wieder neue Funktionen

hinzuzufügen. So brauchte Yelp, das über 30.000 Restaurantbewertungen verfügt, genau ein Jahrzehnt nach seiner Gründung, um seinen Benutzern die Möglichkeit zu bieten, in diesen Restaurants direkt auf der Website Reservierungen vorzunehmen. Unabhängig davon, ob Sie eine App verwenden oder entwickeln, sollten Sie sich über den Zweck der App im Klaren sein.

Die Größe von Giganten nutzen

Entwickler treffen strategische Entscheidungen, welche Teile der App sie selbst programmieren und welche Teile der App von anderen erstellt werden sollen. Für Funktionen, die entweder nicht zum Kerngeschäft gehören oder nicht zu ihren Stärken zählen, wenden sich die Entwickler häufig an Drittanbieter. Auf diese Weise profitieren die Apps von Anbietern, die bereits vor ihnen da waren und schwierige Probleme gelöst haben.

Yelp zum Beispiel zeigt Bewertungen für lokale Einträge an und platziert jeden Eintrag auf einer Karte. Während Yelp die Bewertungen einholt und den Code zur Anzeige der grundlegenden Daten für die verschiedenen Einträge schreibt, entwickelt Google, wie in Abbildung 1.6 dargestellt, die auf der Yelp-Website verwendeten Karten. Dank der Kartenanwendung von Google konnte die erste Version der App mit weniger Entwicklern erstellt werden, als andernfalls erforderlich gewesen wären.

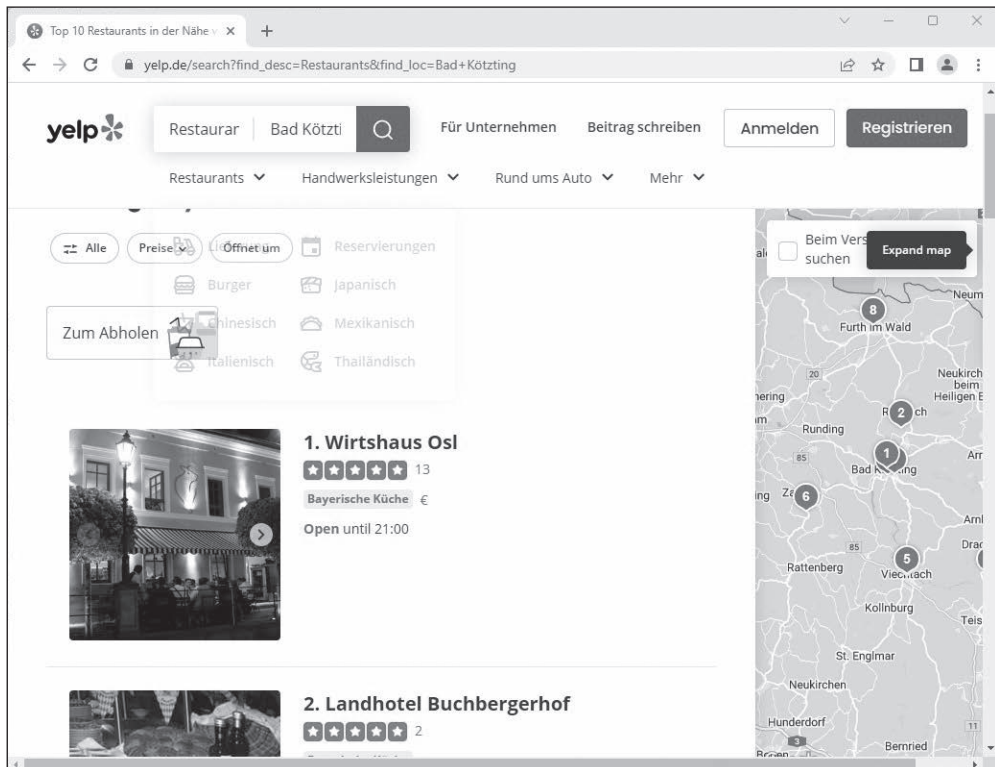


Abbildung 1.6: Die Web-App Yelp verwendet Google Maps.