

# Sprechen Sie mit Ihrem Computer!



## In diesem Kapitel

- ▶ Sprechen Sie mit Ihrem Computer
- ▶ Erstellen Sie Programme, um mit Ihrem Computer zu sprechen
- ▶ Werden Sie verstehen, was ein Programm so macht und warum Sie es erstellen möchten
- ▶ Denken Sie darüber nach, warum Sie Python als Programmiersprache verwenden wollen

---

**E**ine Unterhaltung mit einem Computer führen? Das hört sich eher an wie das Drehbuch zu einem Science-Fiction-Film. Die Besatzung der *Enterprise* in *Star Trek* hat zum Beispiel regelmäßig mit ihrem Computer gesprochen. Und der Computer hat auch oft geantwortet. Seit dem Aufkommen von Apples Siri ([www.apple.com/de/ios/siri](http://www.apple.com/de/ios/siri)) und anderer interaktiver Software jedoch scheint eine solche Unterhaltung schon nicht mehr so unglaublich.



Den Computer um Informationen zu bitten, ist eine Sache, ihn mit Befehlen zu füttern, eine andere. Dieses Kapitel diskutiert, warum Sie Ihrem Computer überhaupt etwas befehlen wollen und was Sie davon haben. Sie werden auch feststellen, dass Sie für diese spezielle Kommunikationsform eine spezielle Sprache benötigen und warum Sie dafür Python benutzen sollten. Das hauptsächliche Fazit, das Sie aus diesem Kapitel ziehen sollten, ist, dass Programmieren einfach nur eine Art der Kommunikation ist, die anderen Formen der Kommunikation, die Sie schon mit Ihrem Computer pflegen, ähnelt.

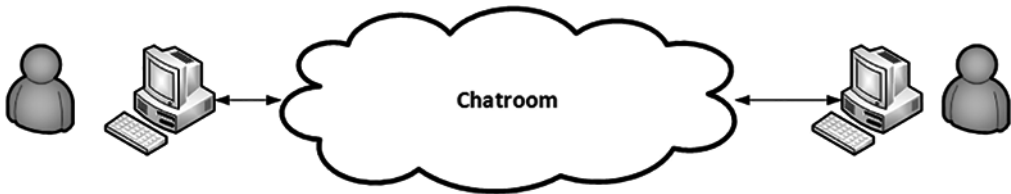
## Warum wollen Sie sich mit Ihrem Computer unterhalten?

Sich mit einer Maschine zu unterhalten, scheint zunächst einmal ziemlich langweilig. Es ist aber notwendig, da ein Computer ja nicht Gedanken lesen kann – noch nicht. Auch wenn der Computer tatsächlich Ihre Gedanken lesen könnte, würde er ja trotzdem mit Ihnen kommunizieren. Ohne den Austausch von Informationen zwischen der Maschine und Ihnen kann nichts vonstattengehen. Aktivitäten wie

- ✓ das Lesen Ihrer E-Mails
- ✓ etwas über Ihren Urlaub schreiben
- ✓ das beste Geschenk der Welt finden

sind alles Beispiele für Kommunikation, die zwischen Ihrem Computer und Ihnen passiert. Die grundlegende Idee ist, dass Kommunikation notwendig ist, um Ergebnisse zu produzieren – das trifft natürlich auch auf Kommunikation zu, die zwischen Ihrem Computer und anderen Rechnern oder Menschen stattfindet.

Solange Sie nicht genauer darüber nachdenken, ist die Kommunikation in den meisten Fällen für Sie transparent. Wenn Sie beispielsweise online einen Chatroom betreten, haben Sie den Eindruck, dass Sie direkt mit einer anderen Person kommunizieren. Tatsächlich kommunizieren Sie aber mit Ihrem Computer, Ihr Computer spricht über den Chatroom mit dem Computer der anderen Person und dieser Computer kommuniziert wiederum mit seinem Menschen. Abbildung 1.1 zeigt Ihnen, was tatsächlich im Hintergrund passiert.



*Abbildung 1.1: Die Kommunikation mit Ihrem Computer ist für Sie transparent, solange Sie nicht genauer darüber nachdenken.*

Schauen Sie sich die Wolke in der Mitte von Abbildung 1.1 an. Die Wolke könnte alles Mögliche enthalten. Wie Sie aber natürlich wissen, enthält sie auf jeden Fall andere Computer, die weitere Anwendungen ausführen. Diese Computer ermöglichen es Ihnen und Ihren Freunden, miteinander zu chatten. Nun denken Sie einmal daran, wie einfach der gesamte Vorgang scheint, wenn Sie ein Chatprogramm verwenden. Obwohl all diese komplexen Dinge im Hintergrund passieren, kommt es Ihnen doch so vor, als wenn Sie sich einfach nur mit Ihrem Freund unterhalten, aber der Vorgang an sich ist für Sie transparent.

## **Eine Anwendung – nur eine Form der Kommunikation**

Die Kommunikation mit dem Computer erfolgt durch die Verwendung von Anwendungen. Sie verwenden zum Beispiel eine Anwendung, um Ihre E-Mails zu beantworten, eine andere, um Sachen zu kaufen, und wieder eine andere, um eine Präsentation zu erstellen. Eine *Anwendung* (*Programm* oder auch manchmal *App* genannt) ist eine Methode, um dem Computer menschliche Ideen so zu präsentieren, dass er sie verstehen kann. Außerdem werden durch eine Anwendung die Werkzeuge festgelegt, die Daten spezifisch für die Kommunikation aufbereiten. Daten, die den Inhalt einer Präsentation repräsentieren, unterscheiden sich von Daten, die benötigt werden, um ein Geschenk für Ihre Mutter zu kaufen. Die Art und Weise, wie Sie die Daten anschauen, verwenden und verstehen, ist für jede Aufgabe verschieden. Also müssen Sie auch verschiedene Anwendungen benutzen, um mit den Daten so zu interagieren, dass der Computer und Sie diese verstehen können.

Sie können quasi für jeden Bedarf, den Sie sich heutzutage vorstellen können, eine Anwendung schreiben. In der Tat haben Sie wahrscheinlich Zugriff auf Anwendungen, von denen Sie

noch nicht einmal wissen, wofür man sie braucht. Programmierer erstellen seit vielen Jahren fleißig Millionen von Anwendungen. Daher ist es vielleicht erst einmal nicht leicht nachzuvollziehen, warum Sie eine neue Anwendung schreiben sollten, um mit Ihrem Computer zu kommunizieren. Letzten Endes finden Sie die Antwort, wenn Sie darüber nachdenken, welche Daten Sie haben und wie Sie damit interagieren wollen. Manche Daten kommen vielleicht nicht so häufig vor, sodass es Programmierern bisher nicht in den Sinn kam, dafür eine Anwendung zu schreiben, oder die Daten liegen in einer Form vor, die keine Anwendung aktuell unterstützt. Also haben Sie keine Möglichkeit, Ihrem Computer die Daten zu präsentieren, ohne nicht selbst eine eigene Anwendung dafür zu schreiben.

Der folgende Abschnitt beschreibt Anwendungen mit dem Hintergedanken, dass Sie mit ganz spezifischen Daten auf eine spezielle Art und Weise arbeiten. Zum Beispiel könnten Sie Zugriff auf eine Datenbank für eine Videobibliothek haben, aber keine Möglichkeit, auf eine für Sie sinnvolle Weise damit zu arbeiten. Sowohl die Daten als auch Ihre Zugriffsanforderungen sind sehr speziell – also keimt in Ihnen der Gedanke, eine Anwendung zu schreiben, die sowohl den Daten als auch Ihren Bedürfnissen gerecht wird.

### ***Das alltägliche Prozedere***

Ein *Arbeitsablauf* oder *Prozess* ist einfach eine Menge aus Schritten, denen Sie folgen, um eine Aufgabe auszuführen. Wenn Sie sich zum Beispiel einen Toast machen, führen Sie vielleicht den folgenden Ablauf aus:

1. Sie holen das Brot und die Butter aus dem Kühlschrank.
2. Sie öffnen die Brottüte und holen zwei Scheiben Toast heraus.
3. Sie stecken das Kabel des Toasters in die Steckdose.
4. Sie positionieren je eine Scheibe Brot in einen Schlitz.
5. Sie drücken den Toasterhebel herunter, um den Toastvorgang zu starten.
6. Sie warten, bis der Toastvorgang abgeschlossen ist.
7. Sie holen den Toast aus dem Toaster.
8. Sie legen die Toastscheiben auf einen Teller.
9. Sie bestreichen den Toast mit Butter.

Ihre Vorgehensweise unterscheidet sich vielleicht von der hier vorgestellten, aber es ist sehr unwahrscheinlich, dass Sie den Toast mit Butter bestreichen, bevor Sie ihn toasten. Natürlich müssen Sie vorher den Toast aus der Verpackung holen, bevor Sie ihn toasten (den Toast mit samt Verpackung und allem Drum und Dran in den Toaster zu stecken, könnte zu unerwünschten Ergebnissen führen ...). Die meisten Leute denken gar nicht über das Toastmachen nach. Aber trotzdem verwenden sie einen solchen Arbeitsablauf, auch wenn sie nicht darüber nachdenken.



Ohne einen Ablaufplan können Computer Aufgaben nicht ausführen. Sie müssen dem Computer sagen, welche Schritte er ausführen soll, in welcher Reihenfolge er sie ausführen soll, und Sie müssen ihn auf alle Ausnahmen, die auftreten und zu einem Fehler führen können, vorbereiten. All diese Informationen (und noch mehr) sind in einer Anwendung beschrieben. Kurz gefasst ist eine Anwendung einfach ein aufgeschriebener Arbeitsablauf, den Sie verwenden, um dem Computer zu sagen, was er machen soll, wann er es machen soll und wie er es machen soll. Da Sie Arbeitsabläufe ja schon Ihr Leben lang verwenden, müssen Sie dieses Wissen nur noch auf die Dinge anwenden, die ein Computer über bestimmte Aufgaben wissen muss.

### ***Arbeitsabläufe aufschreiben***

Als ich in der Grundschule war, gab uns unsere Lehrerin die Aufgabe, etwas über das Toastmachen zu schreiben. Nachdem wir alle unsere Aufsätze abgegeben hatten, holte sie einen Toaster und ein paar Brote hervor. Jeder Aufsatz wurde dann vorgelesen und demonstriert. Keine der von uns beschriebenen Vorgehensweisen funktionierte wie vorgesehen, produzierte aber sehr lustige Ergebnisse. In meinem Fall vergaß ich, die Lehrerin zu bitten, die Verpackung zu entfernen. Also versuchte sie pflichtbewusst, das Brot mitsamt der Verpackung in den Toaster zu stecken. Diese Lektion ist bei mir hängen geblieben. Prozeduren aufzuschreiben, kann ziemlich schwer sein, da wir zwar genau wissen, was wir machen wollen, aber oft Schritte auslassen – wir nehmen einfach an, dass die andere Person auch genau weiß, was wir machen wollen.

Viele alltägliche Lebenssituationen drehen sich um Arbeitsabläufe. Denken Sie nur an die Checkliste, die Piloten vor einem Start verwenden. Ohne einen standardisierten Prozess könnte das Flugzeug abstürzen. Das Aufschreiben eines Arbeitsablaufs braucht seine Zeit, ist aber machbar. Vielleicht müssen Sie mehrere Anläufe starten, um einen voll funktionsfähigen Arbeitsablauf aufzuschreiben, aber wahrscheinlich werden Sie eine solche erstellen können. Es reicht aber nicht nur, den Arbeitsablauf aufzuschreiben – Sie müssen den Arbeitsablauf auch von jemandem testen lassen, der mit der zu bewältigenden Aufgabe nicht vertraut ist. Arbeitet man mit Computern, ist der Computer das perfekte Testobjekt.

### ***Anwendungen als gewöhnliche Arbeitsabläufe verstehen***

Ein Computer verhält sich genauso wie die Grundschullehrerin in meinem Beispiel aus dem vorherigen Abschnitt. Wenn Sie eine Anwendung schreiben, schreiben Sie einen Prozess auf, der eine vom Computer auszuführende Schrittfolge definiert, um irgendeine von Ihnen vorgegebene Aufgabe zu erfüllen. Lassen Sie einen Schritt aus, werden die Ergebnisse nicht wie erwartet. Der Computer versteht nicht, was Sie meinen, oder, dass Sie dachten, dass er manche Sachen automatisch macht. Der Computer weiß nur, dass Sie ihm einen Prozess vorgegeben haben und dass er diesen Prozess ausführen soll.

## ***Computer nehmen alles wörtlich***

Die Leute gewöhnen sich wahrscheinlich an die Arbeitsabläufe, die Sie für sie erstellen. Sie kompensieren automatisch Mängel in Ihrem Ablauf oder machen sich zu den Dingen, die Sie vergessen haben, Notizen. Mit anderen Worten: Die Leute kompensieren Probleme der Prozesse, die Sie erstellt haben.



Wenn Sie mit der Programmierung beginnen, werden Sie frustriert sein, da Computer Aufgaben präzise ausführen und Ihre Anweisungen wörtlich nehmen. Sagen Sie beispielsweise dem Computer, dass ein bestimmter Wert gleich 5 sein soll, wird der Computer nach einem Wert suchen, der exakt 5 ist. Ein Mensch könnte eine 4,9 sehen und wissen, dass das doch gut genug ist. Aber Computer denken nicht so. Ein Computer sieht den Wert 4,9 und entscheidet, dass dieser nicht genau gleich 5 ist. Kurz gefasst sind Computer unflexibel, nicht intuitiv und einfallslos. Wenn Sie einen Arbeitsablauf für einen Computer definieren, wird der Computer jedes Mal präzise das tun, was Sie von ihm verlangen, und Ihren Prozess nie ändern oder entscheiden, dass Sie ihm ja eigentlich etwas anderes aufgetragen hatten.

## ***Was ist eine Anwendung eigentlich***

Wie schon erwähnt, bieten Anwendungen die Möglichkeit, menschliche Ideen so auszudrücken, dass ein Computer sie verstehen kann. Um dieses Ziel zu erreichen, benötigt die Anwendung einen oder mehrere Arbeitsabläufe, die dem Computer sagen, wie Aufgaben zur Verarbeitung und Anzeige von Daten auszuführen sind. Sie sehen am Bildschirm zum Beispiel Text in Ihrem Textverarbeitungsprogramm. Aber damit Sie diese Informationen sehen können, braucht der Computer Vorgaben, wie er Daten von der Festplatte lesen, diese in eine Form bringen, die Sie verstehen können, und schließlich am Bildschirm anzeigen kann. Die folgenden Abschnitte beschreiben die Eigenschaften einer Anwendung detaillierter.

## ***Computer haben ihre eigene Sprache***

Die menschliche Sprache ist komplex und schwierig zu verstehen. Sogar Anwendungen wie Siri haben nur sehr begrenzte Möglichkeiten zu verstehen, was Sie sagen. Mit der Zeit haben Computer gelernt, menschliche Sprache als Eingabe zu verarbeiten und bestimmte gesprochene Wörter als Befehle zu interpretieren. Aber Computer verstehen die menschliche Sprache im Wesentlichen immer noch nicht so richtig. Die Schwierigkeit der menschlichen Sprache kann man gut am Beispiel der Arbeit von Anwälten darstellen. Wenn Sie sich ein Dokument in Juristendeutsch anschauen, liest sich das wie das reinste Kauderwelsch. Ziel dieser Sprache ist es jedoch, keinerlei Spielraum für Interpretationen zu lassen. Anwälte haben trotzdem selten Erfolg, dieses Ziel vollständig zu erreichen, da die menschliche Sprache einfach zu ungenau ist.

Wenn man sich nun anschaut, was Sie aus den vorigen Abschnitten dieses Kapitels wissen, könnten Computer sich niemals auf die menschliche Sprache verlassen, um die von Ihnen

definierten Prozesse zu verstehen. Computer nehmen alles wörtlich, sodass Sie völlig unvorhersehbare Ergebnisse bekommen würden, wenn Sie Anwendungen in menschlicher Sprache schreiben würden. Darum verwenden Menschen spezielle Sprachen, *Programmiersprachen* genannt, um mit Computern zu kommunizieren. Mit diesen speziellen Sprachen können Sie Prozesse definieren, die sowohl präzise als auch durch Computer und Menschen gleichermaßen zu verstehen sind.



Computer sprechen eigentlich gar keine Sprache. Sie verwenden Binärcodes, um Schalter intern umzulegen und mathematische Berechnungen durchzuführen. Computer verstehen noch nicht einmal Buchstaben – sie verstehen nur Zahlen. Eine spezielle Anwendung macht aus der computerspezifischen Sprache, die Sie zum Definieren von Arbeitsabläufen benötigen, Binärcodes. Aber für die Zwecke dieses Buches brauchen Sie sich nicht allzu viel Gedanken zu machen über die internen Details, wie Computer auf der Binärebene funktionieren. Trotzdem ist es interessant zu wissen, dass Computer in Form von Mathematik und Zahlen kommunizieren, und nicht wirklich mit einer Sprache.

### ***Den Menschen helfen, mit dem Computer zu sprechen***

Es ist wichtig, dass Sie sich immer des Zwecks einer Anwendung bewusst sind, während Sie sie schreiben. Eine Anwendung hilft den Menschen, sich mit dem Computer auf bestimmte Weise zu unterhalten. Jede Anwendung benutzt irgendwelche Daten, die der Anwendung als Eingabe übergeben, gespeichert, verändert und ausgegeben werden, sodass die Menschen, die die Anwendung benutzen, ein gewünschtes Ergebnis bekommen. Ob es sich nun um ein Spiel oder eine Tabellenkalkulation handelt, die Grundidee ist dieselbe. Computer arbeiten mit Daten, die ihnen von Menschen zur Verfügung gestellt werden, um ein gewünschtes Ergebnis zu erhalten.

Wenn Sie eine Anwendung programmieren, entwickeln Sie eine neue Methode, wie Menschen mit Computern kommunizieren können. Der von Ihnen neu erstellte Ansatz ermöglicht es anderen Menschen, Daten auf neue Art zu betrachten. Die Kommunikation zwischen Mensch und Computer sollte so einfach sein, dass das Programm eigentlich in den Hintergrund tritt. Denken Sie an Programme, die Sie in der Vergangenheit verwendet haben. Die besten Anwendungen helfen Ihnen dabei, sich auf die Daten zu konzentrieren, mit denen Sie arbeiten. Zum Beispiel fesselt Sie ein Spiel nur dann, wenn Sie sich darauf konzentrieren können, einen Planeten zu retten oder ein Raumschiff zu fliegen, und nicht auf die Anwendung, die Ihnen das Spielen ermöglicht.



Einer der besten Wege, an die Entwicklung einer Anwendung heranzugehen, ist, sich anzuschauen, wie andere Leute Programme schreiben. Aufzuschreiben, was Sie an anderen Programmen mögen oder nicht so gut finden, hilft Ihnen herauszufinden, wie Ihre Programme aussehen und funktionieren sollen. Folgende Dinge können Sie sich fragen, wenn Sie mit diesen Anwendungen arbeiten:

- ✓ Was finde ich an der Anwendung verwirrend?
- ✓ Welche Funktionen waren einfach zu bedienen?

- ✓ Welche Funktionen waren schwierig zu verwenden?
- ✓ Wie hat mir die Anwendung geholfen, mit meinen Daten zu arbeiten?
- ✓ Wie würde ich die Arbeit mit den Daten einfacher gestalten?
- ✓ Was möchte ich mit meiner Anwendung besser machen, was das vorhandene Programm nicht kann?

Professionelle Entwickler stellen noch viele andere Fragen bei der Erstellung einer Anwendung, aber diese Fragen sind für den Anfang gut, da sie Ihnen einen Denkanstoß geben sollen, dass Anwendungen ein Hilfsmittel für Menschen sind, mit dem Computer zu sprechen. Wenn Sie sich jemals über ein Programm geärgert haben, wissen Sie schon, wie sich andere Leute fühlen werden, wenn Sie nicht die richtigen Fragen vor der Programmierung Ihrer Anwendung stellen. Kommunikation ist der wichtigste Baustein jeder Anwendung, die Sie erstellen.

Sie können auch schon mal damit beginnen, über die Art und Weise, wie Sie arbeiten wollen, nachzudenken. Schreiben Sie Prozesse für die Dinge auf, die Sie so tun. Es ist sinnvoll, den jeweiligen Prozess in Schritte einzuteilen und zu jedem Schritt alles aufzuschreiben, was Ihnen dazu einfällt. Wenn Sie damit fertig sind, bitten Sie jemand anders, den Prozess auszuprobieren, damit Sie sehen, ob er wirklich funktioniert. Es wird Sie sicherlich überraschen, dass Sie trotz intensiver Bemühungen sehr schnell Schritte vergessen können.



Das schlechteste Programm der Welt beginnt normalerweise mit einem Programmierer, der nicht weiß, was die Anwendung tun soll, warum sie etwas Besonderes ist, welche Aufgabe sie erfüllen soll oder für wen sie ist. Wenn Sie eine Anwendung schreiben, versichern Sie sich, dass Sie wissen, warum Sie sie erstellen und was Sie damit erreichen wollen. Einen Plan in der Tasche zu haben, hilft dabei, den Spaß am Programmieren nicht zu verlieren. Während Sie an Ihrem neuen Programm arbeiten, wird sich ein Ziel nach dem anderen erfüllen, bis die Anwendung fertig ist. Diese können Sie dann Ihren Freunden zeigen (die natürlich alle denken, wie cool es ist, dass Sie dieses Programm geschrieben haben).

## ***Warum Python so cool ist***

Heutzutage gibt es viele Programmiersprachen. Tatsächlich kann ein Student an der Universität ein ganzes Semester lang Computersprachen studieren und kennt sie dann immer noch nicht alle. (Genau das habe ich während meiner Studienzzeit getan.) Eigentlich könnte man meinen, dass Programmierer mit all diesen Programmiersprachen zufrieden sein könnten und einfach eine auswählen, um mit dem Computer zu sprechen. Aber trotzdem denken sie sich immer wieder neue aus.



Programmierer denken sich aus gutem Grund neue Sprachen aus. Jede Sprache hat etwas Besonderes zu bieten – etwas, das sie außergewöhnlich gut kann. Außerdem entwickeln sich Programmiersprachen auch weiter, um mit dem Fortschritt der Computertechnologie Schritt zu halten. Da das Schreiben einer Anwendung sich hauptsächlich um effiziente Kommunikation dreht, können die meisten Programmierer mehrere Sprachen, sodass sie sich die richtige Sprache

für eine bestimmte Aufgabe herausuchen können. Die eine Sprache ist vielleicht besser geeignet, Daten aus einer Datenbank abzurufen, während Sie mit einer anderen besonders gut Benutzeroberflächen erstellen können.

Wie bei jeder anderen Programmiersprache auch macht Python einige Dinge außergewöhnlich gut und Sie sollten wissen, welche Dinge das sind, bevor Sie Python einsetzen. Sie werden vielleicht verblüfft sein, welche wirklich coolen Sachen Sie mit Python machen können. Bei der Verwendung einer Programmiersprache hilft es, ihre Stärken und Schwächen zu kennen. Außerdem können Sie sich Frust ersparen, wenn Sie die Sprache nicht für Dinge verwenden, die man damit nicht so gut machen kann. Die folgenden Abschnitte helfen Ihnen dabei, solche Entscheidungen für Python zu treffen.

### *Warum man Python verwenden sollte*

Die meisten Programmiersprachen werden mit bestimmten Zielen im Hinterkopf erfunden. Diese Ziele helfen dabei, die Charakteristika der Sprache zu definieren und zu bestimmen, was man mit der Sprache tun können soll. Es ist nicht wirklich möglich, eine Programmiersprache zu erfinden, die alles kann, da Programmierer miteinander in Konflikt stehende Ziele und Anforderungen bei der Programmierung haben. Bei Python war das Hauptziel, eine Programmiersprache zu erfinden, die Programmierer effizient und produktiv macht. Mit diesem Grundgedanken im Hinterkopf zähle ich im Folgenden ein paar Gründe auf, warum Sie Python zur Anwendungsentwicklung benutzen sollten:

- ✓ **Kürzere Entwicklungszeiten:** Python-Code ist normalerweise zwei bis zehn Mal kürzer als vergleichbarer Code, der mit Sprachen wie C/C++ und Java geschrieben wurde. Das bedeutet, dass Sie weniger Zeit für die Erstellung Ihres Programms brauchen, aber mehr Zeit dafür haben, es zu benutzen.
- ✓ **Einfach zu lesen:** Eine Programmiersprache ist wie jede andere Sprache – Sie müssen sie lesen können, damit Sie verstehen, was sie macht. Python-Code ist tendenziell einfacher zu lesen als mit anderen Sprachen geschriebener Code. Das bedeutet, dass Sie weniger Zeit brauchen, den Code zu interpretieren, und mehr Zeit dafür haben, wichtige Änderungen vorzunehmen.
- ✓ **Schneller zu erlernen:** Die Erfinder von Python wollten eine einfacher zu erlernende Programmiersprache mit weniger sonderbaren Regeln erschaffen. Im Grunde wollen Programmierer ja Anwendungen schreiben und nicht obskure und schwierige Sprachen lernen.



Sie sollten wissen, dass Python zwar eine beliebte Sprache ist, aber längst nicht die beliebteste. Tatsächlich belegt sie nur Rang acht auf Seiten wie TIOBE ([www.tiobe.com/index.php/content/paperinfo/tpci/index.html](http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html)), eine Organisation, die (unter anderem) Nutzungsstatistiken verfolgt. Wenn Sie eine Sprache erlernen wollen, nur um einen Job zu bekommen, ist Python eine gute Wahl, aber C/C++, Java oder Visual Basic sind da die besseren Alternativen. Wählen Sie eine Sprache aus, die Sie mögen und die Ihren Anwendungsentwicklungsanforderungen entspricht, aber wählen Sie sie auch danach aus, was Sie damit erreichen wollen. Python war 2007 und 2010 die Sprache des Jahres und 2011 die



viertbeliebteste Sprache. Also ist sie gar nicht schlecht, wenn Sie einen Job suchen, aber nicht unbedingt die beste Wahl. Es wird Sie vielleicht überraschen, dass viele Universitäten in den USA Python verwenden, um den Studenten Programmieren beizubringen, und dass Python dafür die beliebteste Sprache geworden ist. Wenn Sie mehr darüber erfahren wollen, lesen Sie am besten meinen (englischen) Blogbeitrag <http://blog.johnmuelเลอร์books.com/2014/07/14/python-as-a-learning-tool>.

### ***Wie Sie persönlich von Python profitieren können***

Letztendlich können Sie jede Programmiersprache dazu verwenden, jede beliebige Anwendung zu schreiben. Setzen Sie aber die falsche Programmiersprache für eine Aufgabe ein, werden Sie die Arbeit daran hassen und das Ergebnis wird langsam und fehleranfällig sein – aber Sie würden die Aufgabe irgendwie lösen können. Natürlich wollen die meisten von uns schreckliche und qualvolle Erfahrungen vermeiden. Also sollte man wissen, welche Anwendungen die Leute normalerweise mit Python schreiben (auch wenn es Leute gibt, die Python auch für andere Zwecke einsetzen):

- ✓ **Anwendungsbeispiele erstellen:** Entwickler müssen oft einen *Prototyp*, ein erstes Beispiel für eine Anwendung, erstellen, bevor sie die Ressourcen aufwenden, das eigentliche Programm zu schreiben. Da in Python Produktivität besonders groß geschrieben wird, können Sie es gut verwenden, um schnell Prototypen für eine Anwendung zu erstellen.
- ✓ **Browser-basierte Skripte programmieren:** Auch wenn JavaScript wahrscheinlich die beliebteste Sprache für Browser-basierte Anwendungsskripte ist, folgt ihr Python doch dicht auf dem zweiten Platz. Python bietet Funktionen an, die JavaScript nicht hat (für weitere Informationen finden Sie hier einen Vergleich: <https://blog.glyphobet.net/essay/2557>) und durch seine hohe Effizienz können Sie Browser-basierte Programme schneller erstellen (ein klarer Vorteil in der heutigen schnelllebigen Welt).
- ✓ **Mathematische, naturwissenschaftliche und ingenieurstechnische Anwendungen entwerfen:** Besonders interessant ist, dass Python Ihnen Zugriff auf ein paar wirklich coole Bibliotheken bietet, um mathematische, naturwissenschaftliche und ingenieurstechnische Programme einfacher zu schreiben. Die zwei beliebtesten Bibliotheken sind NumPy ([www.numpy.org](http://www.numpy.org)) und SciPy ([www.scipy.org](http://www.scipy.org)). Mit diesen Bibliotheken können Sie solche Anwendungen wesentlich schneller schreiben.
- ✓ **Mit XML arbeiten:** Die eXtensible Markup Language (XML) ist die Grundlage für die meisten Datenverwaltungslösungen vieler Internet- und Desktopanwendungen. Bei anderen Sprachen wurde XML eher dazugebastelt, während es bei Python einen Platz in der ersten Reihe hat. Wenn Sie beispielsweise mit einem Webdienst arbeiten müssen, der wichtigsten Methode, Informationen über das Internet auszutauschen (oder irgendeiner anderen XML-intensiven Anwendung), ist Python eine gute Wahl.
- ✓ **Mit Datenbanken interagieren:** Die Industrie ist sehr von Datenbanken abhängig. Python ist nicht wirklich eine Abfragesprache wie die Structured Query Language (SQL) oder Language Integrated Query (LINQ), aber man kann mit ihr sehr gut mit Datenbanken arbeiten. Sie macht das Herstellen von Verbindungen zur Datenbank und die Arbeit mit den Daten ziemlich einfach.

- ✓ **Benutzeroberflächen erstellen:** Bei Python haben Sie nicht so wie bei anderen Sprachen wie C# einen integrierten Designer, wo Sie Elemente aus einer Palette mit Werkzeugen auf eine Benutzeroberfläche ziehen können. Python bietet jedoch eine große Auswahl an Frameworks für grafische Benutzeroberflächen (englisch: Graphical User Interfaces, GUIs) – Erweiterungen, die die Erstellung von Benutzeroberflächen viel einfacher machen (mehr Informationen finden Sie unter <https://wiki.python.org/moin/GuiProgramming>). Manche dieser Frameworks liefern Designer mit, die die Erstellung von Benutzeroberflächen vereinfachen. Der Punkt ist, dass Python nicht nur eine Methode zur Erstellung einer Benutzeroberfläche anbietet – Sie können sich die Methode aussuchen, die Ihren Bedürfnissen am besten gerecht wird.

### ***Welche Organisationen verwenden Python?***

Python erfüllt die Aufgaben, für die es konzipiert wurde, ziemlich gut. Und das ist auch der Grund, warum viele große Firmen und Organisationen Python zur Programmierung von Anwendungen (allgemein zur Entwicklung) verwenden. Ein Grund mehr, Python zu nutzen, da diese Organisationen die Sprache unterstützen und Geld dafür ausgeben, sie zu verbessern. Im Folgenden finden Sie eine Liste von Organisationen, die Python häufig einsetzen und auch, wofür sie sie einsetzen:

- ✓ Alice Educational Software – Carnegie Mellon University ([www.cmu.edu/corporate/news/2007/features/alice.shtml](http://www.cmu.edu/corporate/news/2007/features/alice.shtml)): Lernsoftware
- ✓ Fermilab ([www.fnal.gov](http://www.fnal.gov)): Wissenschaftliche Anwendungen
- ✓ Go.com (<http://go.com>): Browser-basierte Anwendungen
- ✓ Google ([www.google.de](http://www.google.de)): Suchmaschine
- ✓ Industrial Light & Magic ([www.ilm.com](http://www.ilm.com), ja, genau, die Firma, die die Special Effects für Star Wars gemacht hat!): Alles, was so an Programmierung anfällt.
- ✓ Lawrence Livermore National Library ([www.llnl.gov](http://www.llnl.gov)): Wissenschaftliche Anwendungen
- ✓ National Space and Aeronautics Administration (NASA) ([www.nasa.gov](http://www.nasa.gov)): Wissenschaftliche Anwendungen
- ✓ New York Stock Exchange (<https://nyse.nyx.com>): Browser-basierte Anwendungen
- ✓ Redhat ([www.redhat.com](http://www.redhat.com)): Installationswerkzeuge für Linux
- ✓ Yahoo! ([www.yahoo.com](http://www.yahoo.com)): Teile von Yahoo!-Mail
- ✓ YouTube ([www.youtube.com](http://www.youtube.com)): Grafikengine
- ✓ Zope ([www.zope.com](http://www.zope.com)): Plattform zur Erstellung von Webanwendungen (Content Management Systeme)



Dies sind nur ein paar der vielen Organisationen, die Python intensiv nutzen. Eine Liste mit weiteren Firmen finden Sie hier: [www.python.org/about/success](http://www.python.org/about/success). Die Anzahl der Erfolgsgeschichten ist so riesig geworden, dass diese Liste wahrscheinlich auch nicht vollständig ist und man sie in Kategorien einteilen musste, damit sie übersichtlicher wird.

## ***Nützliche Python-Programme finden***

Vielleicht haben Sie auf Ihrem Computer schon eine Anwendung, die in Python geschrieben wurde, und wissen es noch nicht einmal. Python wird in der Industrie in sehr vielen Anwendungsbereichen eingesetzt. Die Programme reichen von kleinen Konsolenanwendungen bis hin zu ausgewachsenen CAD/CAM-Plattformen. Manche Programme laufen auf mobilen Geräten, während andere auf großen Firmenservern ausgeführt werden. Zusammengefasst gibt es nichts, was Sie mit Python nicht tun können, aber es hilft, sich anzuschauen, was andere so damit gemacht haben. Es gibt viele Seiten im Internet, wo Sie das nachschauen können, aber am besten schauen Sie hier nach: <https://wiki.python.org/moin/Applications>.

Als Python-Programmierer wollen Sie natürlich auch wissen, welche Entwicklungswerkzeuge es für Python gibt, die Ihnen die Arbeit erleichtern. Ein *Entwicklungswerkzeug* bietet Ihnen einen gewissen Automatisierungsgrad beim Schreiben von Programmen, die dem Computer sagen, was er zu tun hat. Mit einem oder mehreren Entwicklungswerkzeugen müssen Sie weniger tun, um eine funktionierende Anwendung zu erstellen. Zwar geben Entwickler gerne die Liste ihrer Lieblingswerkzeuge preis, aber eine richtig umfangreiche Liste mit Werkzeugen unterteilt in Kategorien finden Sie hier: [www.python.org/about/apps](http://www.python.org/about/apps).



Natürlich beschreibt dieses Kapitel auch ein paar Werkzeuge, wie NumPy und SciPy (zwei naturwissenschaftliche Bibliotheken). Im restlichen Buch werden auch noch ein paar andere Tools erwähnt. Merken Sie sich am besten Ihre Lieblingswerkzeuge für den späteren Einsatz.

## ***Python mit anderen Sprachen vergleichen***

Es ist gefährlich, eine Sprache mit einer anderen zu vergleichen, da die Auswahl einer Sprache Geschmackssache und persönliche Vorliebe und keine exakte Wissenschaft ist. Also, bevor ich von den vehementen Verteidigern der anderen Sprachen attackiert werde, sollten Sie wissen, dass auch ich eine Vielzahl Sprachen verwende und es bei allen eine gewisse Überschneidung gibt. Die beste Sprache der Welt gibt es nicht, nur die beste Sprache für eine bestimmte Anwendung. Denken Sie daran, wenn Sie die folgenden Abschnitte lesen. Diese bieten einen Überblicksvergleich zwischen Python und anderen Sprachen. (Vergleiche mit weiteren Sprachen finden Sie unter <https://wiki.python.org/moin/LanguageComparisons>).

### **C#**

Viele Leute behaupten, dass Microsoft einfach Java abgekupfert hat, um C# zu erstellen. Dennoch hat C# ein paar Vorteile gegenüber Java (und auch ein paar Nachteile). Die (unangefochtene) Hauptabsicht, C# zu entwickeln, war es, eine bessere C/C++-Sprache zu schaffen – eine, die leichter zu erlernen und zu verwenden ist. Aber wir wollten ja hier über C# und Python diskutieren. Vergleicht man Python mit C#, bietet Python die folgenden Vorteile:

- ✓ Deutlich einfacher zu erlernen
- ✓ Der Code ist schlanker (prägnanter)
- ✓ Vollständig quelloffen
- ✓ Bessere Plattformunabhängigkeit

- ✓ Verwendung verschiedener Entwicklungsumgebungen sehr einfach möglich
- ✓ Einfacher mit Java und C/C++ zu erweitern
- ✓ Verstärkte Unterstützung für wissenschaftliche Projekte und Ingenieuraufgaben

## *Java*

Lange Jahre suchten Programmierer nach einer Sprache, mit der sie ein Programm nur einmal schreiben müssen und dann überall ausführen können. Java ist dafür gedacht, auf jedem Betriebssystem gut zu funktionieren. Um dieses Wunder möglich zu machen, bedient es sich einiger Tricks, über die Sie weiter hinten in diesem Buch noch etwas erfahren werden. Was Sie für den Moment nur wissen müssen, ist, dass Java dermaßen erfolgreich damit war, überall gut zu funktionieren, dass andere Sprachen ihm nacheifern wollten (mit unterschiedlichem Erfolg). Trotzdem hat Python gegenüber Java ein paar wichtige Vorteile:

- ✓ Deutlich einfacher zu erlernen
- ✓ Der Code ist schlanker (prägnanter)
- ✓ Erweitertes Variablenmodell (Variablen sind temporäre Aufbewahrungsorte für Daten im Computerspeicher). Basierend auf den Anwendungsbedürfnissen können die Variablen verschiedene Arten von Daten während der Laufzeit enthalten (auch *Dynamische Typisierung* genannt).
- ✓ Kürzere Entwicklungszeiten

## *Perl*

Perl war ursprünglich ein Akronym für Practical Extraction and Report Language (was so viel heißt wie zweckmäßige Extraktions- und Berichtssprache). Heutzutage nennen die Leute sie einfach nur noch Perl – Punkt. Die Wurzeln von Perl erkennt man heute noch daran, dass es eine überragende Vorstellung bei der Abfrage von Daten aus einer Datenbank und deren Präsentation in Berichtsform abgibt. Natürlich wurde Perl so erweitert, dass es sehr viel mehr als das kann – Sie können damit alle möglichen Anwendungen schreiben. (Ich habe es zum Beispiel zum Schreiben einer Webdienst-Anwendung verwendet.) Im Vergleich mit Python werden Sie feststellen, dass Python in folgenden Dingen besser ist:

- ✓ Einfacher zu erlernen
- ✓ Einfacher zu lesen
- ✓ Besserer Schutz von Daten
- ✓ Bessere Java-Integration
- ✓ Weniger plattformspezifische Eigenheiten