

PYTHON

TIMO
BRUMM

ZEILE FÜR ZEILE ZUM
PYTHON PROGRAMMIERER

MIT VIELEN
PRAKTISCHEN
ÜBUNGEN SAMT
LÖSUNG

Python

Zeile für Zeile zum Python Programmierer
Timo Brumm

**PY
TH
on**

**TIMO
BRUMM**

© 2021 Timo Brumm

Alle Rechte vorbehalten.

ISBN: 979-8-544-57496-5

1. Auflage

Niklas Loos

An der Radrunde 119

90455 Nürnberg

Korrektorat: Christina Müller

Cover-Design: semnitz

Buchsatz: Kathrin Lembcke

Das Werk, einschließlich seiner Teile, ist urheberrechtlich geschützt. Jede Verwertung ist ohne Zustimmung des Verlages und des Autors unzulässig. Dies gilt insbesondere für die elektronische oder sonstige Vervielfältigung, Übersetzung, Verbreitung und öffentliche Zugänglichmachung.

Einleitung

Das Programmieren zu erlernen, bringt unzählige Vorteile mit sich: gute Karrierechancen, ein attraktives Gehalt, ausgezeichnete Möglichkeiten für die Arbeit als Freelancer, spannende berufliche Projekte, Mitwirkung an der Gestaltung neuer Technologien und viele mehr. Darüber hinaus stellt es für viele Menschen einfach eine spannende Aufgabe dar, ein Computerprogramm zu gestalten. Ganz egal, welches der wichtigste Grund für Sie ist: Es lohnt sich! Wenn Sie diesen Text lesen, dann haben Sie sich wahrscheinlich dazu entschlossen, diese Herausforderung in Angriff zu nehmen.

Hierbei ist es jedoch notwendig, sich für eine passende Programmiersprache zu entscheiden. Die Auswahl in diesem Bereich ist riesig. Es gibt mehrere Hundert verschiedene Möglichkeiten. Dieses Buch befasst sich mit der Programmiersprache Python. Diese ist nicht nur einfach zu erlernen. Darüber hinaus bietet sie sehr viele Anwendungsmöglichkeiten. Bevor wir mit den ersten Programmen beginnen, wollen wir die Eigenschaften dieser Programmiersprache kurz darstellen und dabei herausarbeiten, weshalb sie sich sehr gut dazu eignet, um das Programmieren zu erlernen.

Python – gut für Anfänger geeignet

Python ist zwar mittlerweile mehr als drei Jahrzehnte alt. Dennoch handelt es sich hierbei um eine moderne Programmiersprache auf dem aktuellen Stand der Technik. Die erste Ausführung entstand bereits 1991. Damit ist diese Sprache jedoch bereits deutlich aktueller als viele andere beliebte Programmiersprachen wie C oder C++. Außerdem kam es im Laufe der Zeit zu umfassenden Überarbeitungen – bei der Veröffentlichung von Python 2 im Jahr 2000 und nochmals 2008 mit der Einführung der Version 3. Die Änderungen, die dabei durchgeführt wurden, passten die Sprache an neuere Entwicklungen an.

Eines der wichtigsten Kennzeichen von Python besteht darin, dass es sich hierbei um eine objektorientierte Programmiersprache handelt. Dieser Ansatz ist in der modernen Informatik von großer Bedeutung. Für die ersten kleinen Programme spielt er aber noch keine Rolle. Wenn Sie später jedoch etwas fortgeschrittenere Programme erstellen – eventuell sogar in einem professionellen Umfeld –, ist es sehr wichtig, dass Sie dieses Konzept beherrschen. Aus diesem Grund ist es für Anfänger empfehlenswert, sich für eine objektorientierte Sprache zu entscheiden. Python erfüllt diesen Anspruch. Im weiteren Verlauf werden wir uns noch etwas intensiver mit diesem Thema befassen und selbst objektorientierte Programme gestalten.

Allerdings handelt es sich bei Python nicht um eine ausschließlich objektorientierte Sprache, sondern um eine sogenannte Multi-Paradigmen-Sprache. Das bedeutet, dass sie auch andere Ansätze unterstützt. Auf diese Weise können Sie sich in andere Techniken einarbeiten und dann genau die Vorgehensweise auswählen, die sich für Ihr Projekt am besten eignet.

Der entscheidende Grund dafür, dass Python sehr gut für den Einstieg in die Programmierung geeignet ist, liegt jedoch darin, dass die Sprache genau für diesen Einsatzzweck entwickelt wurde. Guido van Rossum, der Erfinder von Python, arbeitete in den 1990er-Jahren an einem Forschungszentrum in Amsterdam. Zu seinen Aufgaben gehörte es, jungen Leuten das Programmieren beizubringen. Hierfür kam ursprünglich die Sprache ABC zum Einsatz. Diese wies jedoch erhebliche Mängel auf. Daher beschloss van Rossum, eine eigene Lehrsprache zu entwickeln. Diese sollte den Einstieg für Anfänger so leicht wie möglich gestalten.

Um dieses Ziel zu erreichen, setzte van Rossum auf eine möglichst einfache Syntax. Hierbei handelt es sich um die formalen Regeln einer Programmiersprache – vergleichbar mit der Grammatik in einer natürlichen Sprache.

Viele andere Programmiersprachen benötigen zahlreiche Klammern, Semikolons und weitere formale Elemente, um ein Programm zu gestalten. Diese stellen eine häufige Fehlerquelle dar. Eine vergessene Klammer oder ein nicht gesetztes Semikolon führt in vielen Sprachen dazu, dass das gesamte Programm nicht ausführbar ist. Python verzichtet jedoch soweit wie möglich auf diese Elemente. Das führt zu einem einfachen und leicht verständlichen Programmaufbau, sodass der Einstieg kein Problem darstellt. Außerdem werden dadurch häufige Fehlerquellen beseitigt.

Um dies zu verdeutlichen, wollen wir zwei kleine Beispielprogramme vorstellen. Beide erledigen genau die gleiche Aufgabe: Sie geben die Begrüßung "Hallo!" auf dem Bildschirm aus.

Das erste Programm ist in Java geschrieben:

```
public class Hallo {  
    public static void main(String[] args) {  
        System.out.print("Hallo!");  
    }  
}
```

Und jetzt das Ganze mit Python:

```
print("Hallo!")
```

Dabei ist es natürlich nicht notwendig, dass Sie die Codebeispiele verstehen. Es geht nur darum, die Unterschiede in der Komplexität der Programme aufzuzeigen. Sie sehen, dass das Python-Programm deutlich einfacher aufgebaut ist, obwohl die Funktion genau die gleiche ist.

Hinzu kommen einige weitere Eigenschaften, die die Programmgestaltung mit Python für Anfänger erleichtern. Beispielsweise setzt Python bei der Gestaltung der Befehle deutlich stärker auf Ausdrücke der natürlichen englischen Sprache als die meisten anderen Programmiersprachen. Das verhindert die Verwendung kryptischer Zeichenkombinationen, deren Bedeutung sich intuitiv nicht erschließen lässt. Außerdem weist Python eine recht kleine Standardbibliothek auf. Das bedeutet, dass die Sprache nur vergleichsweise wenige Befehle enthält. Auch das führt dazu, dass sie einfach zu erlernen ist. Für spezielle Anwendungen lassen sich die Funktionen jedoch einfach erweitern.

Wenn Sie beim einen oder anderen Begriff in diesem Kapitel noch nicht die genaue Bedeutung verstehen, stellt das kein Problem dar. Im weiteren Verlauf werden Sie dazu noch viele weitere Informationen erhalten. Wichtig ist es an dieser Stelle nur, dass Folgendes klar wird: Alle genannten Eigenschaften führen dazu, dass Python aktuelle Konzepte umsetzt und einfach zu erlernen ist. Deshalb ist die Programmiersprache gerade in der Lehre ausgesprochen beliebt. Viele Schulen und Universitäten setzen diese Programmiersprache ein, um die ersten Programmierkenntnisse zu vermitteln.

Python – eine der beliebtesten Programmiersprachen

Das Konzept von Python, das darauf ausgerichtet war, den Einstieg in die Programmierung zu erleichtern, führte zu einem weiteren Vorteil. Die einfachen Strukturen erleichtern und beschleunigen die Programmerstellung auch für erfahrene Programmierer. Das führt dazu, dass dieser Prozess sehr effizient abläuft. Das bringt bei der professionellen Softwareentwicklung einen erheblichen Kostenvorteil mit sich. Deshalb verwenden immer mehr Unternehmen Python für die Entwicklung ihrer Projekte.

Aus diesen Gründen hat sich Python zu einer der beliebtesten Programmiersprachen entwickelt. Hierzu genaue Nutzungszahlen anzugeben, ist aber ausgesprochen schwierig, da es keine umfassenden Erhebungen zu diesem Thema gibt. Verschiedene Institutionen, die sich der Frage

widmen, welches die beliebteste Programmiersprache ist, sehen Python jedoch in der Regel sehr weit vorne. Das RedMonk Programming Language Ranking¹ sieht Python beispielsweise auf dem zweiten Platz. Im TIOBE-Index² steht Python an dritter Stelle. Im PYPL-Index³ ist Python sogar an erster Stelle aufgelistet. All diese Angaben beziehen sich auf die aktuellsten Veröffentlichungen zum Zeitpunkt der Erstellung dieses Textes (April 2021). Zwar kann es hierbei im Laufe der Zeit zu einigen Veränderungen kommen. Dennoch wird deutlich, dass es sich bei Python um eine der am häufigsten benutzten Programmiersprachen handelt.

Die hohe Beliebtheit dieser Sprache führt dazu, dass Python-Programmierer sehr gefragt sind. Das hat zur Folge, dass sich das Erlernen dieser Programmiersprache besonders positiv auf die Karriereaussichten auswirkt.

Von vielfältigen Anwendungsmöglichkeiten profitieren

Python zeichnet sich außerdem durch sehr vielfältige Anwendungsmöglichkeiten aus. Beispielsweise eignet sich diese Programmiersprache hervorragend, um Desktop-Programme zu schreiben, die sich auf einem gewöhnlichen Computer ausführen lassen. Hierbei bietet Python vielfältige Möglichkeiten, um ansprechende Programme mit grafischen Benutzeroberflächen zu gestalten. Darüber hinaus kommt Python sehr häufig zum Einsatz, um Server-Prozesse zu steuern. Diese laufen bei zahlreichen Internet- oder Businessanwendungen im Hintergrund ab.

Sogar für die Gestaltung von Internetseiten lässt sich Python einsetzen. Das stellt zwar nicht den eigentlichen Anwendungszweck dieser Programmiersprache dar. Allerdings gibt es Frameworks wie beispielsweise Django, die es erlauben, auch hierfür Python zu verwenden.

Bei Frameworks und Programmbibliotheken handelt es sich um Erweiterungen, die eigentlich nicht zum Grundwortschatz von Python gehören. Für die Entwicklung sind unterschiedliche unabhängige Gruppen oder Unternehmen verantwortlich. Diese erweitern die Anwendungsmöglichkeiten der Programmiersprache deutlich. Die entsprechenden Funktionen lassen sich problemlos in ein Python-Programm einbinden. Auf diese Weise lässt sich die Programmiersprache für viele weitere Zwecke einsetzen – von naturwissenschaftlichen Anwendungen bis hin zur Entwicklung von Spielen. Ein großer Vorteil der Programmiersprache Python besteht darin, dass hierfür ein ausgesprochen vielfältiges Angebot an Frameworks und Programmbibliotheken besteht. Daher lässt sie sich für viele verschiedene Zwecke einsetzen.

Hierbei ist noch eine weitere Anwendungsmöglichkeit zu nennen, die immer stärker an Bedeutung gewinnt: die künstliche Intelligenz (KI). Diese kommt bereits in vielen Unternehmen zum Einsatz und spielt eine immer wichtigere wirtschaftliche Rolle. Python stellt in diesem Bereich eine der beliebtesten Programmiersprachen dar. Auch hierfür sind verschiedene Frameworks verfügbar. Von besonderer Bedeutung ist dabei TensorFlow. Dieses Python-Framework hat Google für seine eigenen KI-Anwendungen entwickelt. Mittlerweile steht es jedoch öffentlich zur Verfügung. Damit ist es recht einfach, KI-Funktionen in die eigenen Programme einzubinden. Darüber hinaus gibt es noch einige weitere beliebte Frameworks und Bibliotheken für die Gestaltung von KI-Anwendungen mit Python– wie etwa Theano oder Keras. Diese vielfältigen Möglichkeiten führen dazu, dass Python im Bereich der künstlichen Intelligenz eine bedeutende Stellung einnimmt. Wenn Sie sich in diese Zukunftstechnologie einarbeiten wollen, stellt es daher ebenfalls einen guten ersten Schritt dar, Python zu lernen.

¹ vgl. <https://redmonk.com/sograzy/2021/03/01/language-rankings-1-21>

2 vgl. <https://www.tiobe.com/tiobe-index>

3 vgl. <https://pypl.github.io/PYPL.html>

Das Konzept für dieses Buch

Bereits in der Einleitung haben Sie viele Details zur Programmiersprache Python erfahren – was diese auszeichnet und weshalb sie insbesondere für Anfänger gut geeignet ist. Dabei haben Sie aber sicher festgestellt, dass der eine oder andere Begriff schwer zu verstehen ist und für Verwirrung sorgt. Das ist aber für Anfänger ganz normal. Allerdings führt das häufig zu Frustration – und manchmal sogar zum Abbruch. Um dies zu vermeiden, haben wir für dieses Buch ein Konzept entwickelt, das sich von den meisten anderen Werken in diesem Bereich etwas abhebt. Damit Sie wissen, was auf Sie zukommt und wie Sie die Inhalte bearbeiten müssen, wollen wir Ihnen dieses zunächst kurz vorstellen.

Jedes neue Thema beginnen wir mit einer etwas umfassenderen Aufgabe. Diese führt stets mehrere neue Programmiertechniken oder Befehle ein. Im ersten Schritt entwickeln wir das Programm für dieses Projekt. Dieses sollten Sie unbedingt auf Ihrem eigenen Rechner nachstellen – auch wenn Sie zunächst noch nicht alle Details verstehen. Wenn Sie das Programmieren lernen möchten, ist die praktische Arbeit unverzichtbar. Sollten Sie dieses Buch einfach nur durchlesen, ohne die Programme selbst zu erstellen, ist der Effekt nur sehr gering. Wenn Sie dann ein eigenes Programm schreiben möchten, ist es sehr schwer, dies umzusetzen. Daher ist es wichtig, dass Sie alle Projekte, die wir hier vorstellen, selbst abtippen und ausführen. Nur so werden Sie lernen, eigenständig zu programmieren. Nachdem das Programm fertiggestellt ist, können Sie es ausprobieren und versuchen herauszufinden, welche Aufgaben die einzelnen Befehle erfüllen. Auf diese Weise lernen Sie gleich zu Beginn, Programme mit interessanten Funktionen zu erstellen und erhalten einen Eindruck davon, wie Sie die entsprechenden Befehle und Funktionen anwenden können.

Erst im nächsten Schritt folgen die Erklärungen. Wenn Sie bei der Ausarbeitung des Projekts das eine oder andere Detail zunächst nicht verstanden haben, stellt das daher kein Problem dar. Nun erhalten Sie Antworten auf alle Fragen. Dabei stellen wir bei vielen der neu vorgestellten Befehle und Programmiertechniken auch verschiedene Varianten und Erweiterungsmöglichkeiten vor. Auch hierbei ist es sehr zu empfehlen, die kleinen Beispielprogramme, die wir hierbei anführen, selbst auszuprobieren.

Am Ende der einzelnen Projekte entdecken Sie einige Übungsaufgaben. Diese wiederholen die erlernten Themen und vertiefen die Inhalte. Diese Aufgaben sollten Sie wenn möglich selbstständig beantworten. Dabei können Sie selbstverständlich auch einige Seiten zurückblättern, um sich die Details aus den Erklärungen nochmals anzuschauen. Es ist ganz normal, dass Sie nach dem ersten Durchlesen noch nicht alle Befehle beherrschen. Ganz zum Schluss ist noch eine Musterlösung angefügt. So können Sie Ihre Ergebnisse mit der Lösung vergleichen. Außerdem dient sie als Hilfestellung, wenn Sie einmal alleine nicht weiterkommen.

Beachten Sie dabei, dass es nicht notwendig ist, dass Ihr Programm bis ins kleinste Detail der Musterlösung entspricht, um die Aufgabe korrekt zu erfüllen. In der Informatik führen häufig mehrere Wege ans Ziel. Wenn Ihr Programm alle Vorgaben aus der Aufgabenstellung beachtet, handelt es sich dabei ebenfalls um eine korrekte Lösung – auch wenn es von der Musterlösung abweicht. Es ist empfehlenswert, für jede Aufgabe eine eigene Datei zu erstellen – so wie Sie dies im ersten Projekt lernen werden. So können Sie das Programm ausführlich ausprobieren und überprüfen, ob es alle Anforderungen erfüllt.

Um den Einstieg zu erleichtern, werden wir in diesem Buch im Programmcode so weit wie möglich Ausdrücke in deutscher Sprache verwenden. Die Befehle in Python entstammen selbstverständlich der englischen Sprache und müssen genau in dieser Weise in das Programm übernommen werden.

Darüber hinaus ist es jedoch notwendig, für Variablen, Funktionen und viele weitere Bestandteile des Programms eigene Namen zu vergeben. Dabei werden wir mit deutschsprachigen Begriffen arbeiten. Es ist uns jedoch bewusst, dass es in einem professionellen Umfeld üblich ist, hierfür englischsprachige Bezeichner zu verwenden. Das bedeutet, dass später eine Umstellung notwendig sein wird, falls Sie Ihre Programmierkenntnisse auch für die Berufsausübung einsetzen wollen. Für viele Anfänger stellt es allerdings eine zusätzliche Herausforderung dar, die Ausdrücke in englischer Sprache zu verstehen. Um den Einstieg so einfach wie möglich zu gestalten, nutzen wir daher deutschsprachige Begriffe.

Voraussetzungen für das Programmieren mit Python

Nachdem das Konzept dieses Buchs bekannt ist, wollen wir mit der praktischen Arbeit beginnen. Dafür müssen Sie Ihren Computer jedoch zunächst vorbereiten. Um ein Python-Programm zu erstellen und auszuführen, benötigen Sie einige Zusatzprogramme. Diese müssen Sie daher vor dem Erstellen des ersten Programms installieren. Befolgen Sie hierfür die Anweisungen in den folgenden Abschnitten.

Den Python-Interpreter installieren

Um ein Python-Programm auszuführen, benötigen Sie ein Programm, das als Python-Interpreter bezeichnet wird. Wenn Sie das Betriebssystem Windows verwenden, ist der Python-Interpreter nicht vorinstalliert, sodass Sie sich selbst darum kümmern müssen. Das Programm ist unter dem folgenden Link kostenfrei erhältlich:

<https://www.python.org/downloads>



Abb.1: Bei der Installation ist es wichtig, die Checkbox "Add to PATH" auszuwählen

Auf dieser Seite wird ein Download-Link angezeigt, der normalerweise direkt zur passenden Version für Ihr Betriebssystem führt. Wenn Sie diesen anklicken, wird der Installationsassistent heruntergeladen. Dieser führt Sie automatisch durch den Installationsprozess. Dabei ist es ganz am Anfang wichtig, darauf zu achten, dass die Checkbox "Add to PATH" ausgewählt ist – so wie Sie dies in Abbildung 1 sehen.

Wenn Sie nun auf "Install Now" klicken, beginnt der Installationsprozess, der selbstständig abläuft. Um zu kontrollieren, ob die Installation korrekt abgelaufen ist, ist es sinnvoll, einen Kommandozeileninterpreter aufzurufen. Ein solches Programm nimmt schriftliche Befehle für die Steuerung des Computers entgegen. Hierfür bietet sich das Programm

cmd.exe an, das Sie schnell und einfach über die Windows-Suchfunktion erreichen. Öffnen Sie dieses und geben Sie den folgenden Befehl ein:

```
python --version
```

Wenn wie in Abbildung 2 bei der Eingabe des Befehls eine Versionsnummer erscheint, die mit der Zahl 3 beginnt, ist die Installation des Python-Interpreters korrekt abgeschlossen.



Abb. 2: Die Abfrage der Versionsnummer als Kontrolle für die korrekte Installation

Hinweis: Wenn Sie den Kommandozeileninterpreter bereits vor der Installation des Python-Interpreters geöffnet haben, müssen Sie ihn zunächst schließen, bevor die Änderungen wirksam werden.

Sollten Sie das Betriebssystem Linux oder macOS verwenden, ist der Python-Interpreter wahrscheinlich bereits auf Ihrem Rechner installiert – allerdings eventuell in der falschen Version. Um dies zu überprüfen, ist es sinnvoll, genau wie soeben beschrieben die Python-Version abzufragen. Der Kommandozeileninterpreter trägt hier allerdings die Bezeichnung Terminal. Sie erreichen dieses ebenfalls über die Suchfunktion.

Wenn bei dieser Abfrage eine Version erscheint, die mit der Zahl 3 beginnt, sind keine weiteren Aktionen notwendig. Sollte hier hingegen eine Version erscheinen, die mit der Zahl 2 beginnt, bedeutet das, dass das System Python 2 – die Vorgängerversion – verwendet. Das ist häufig bei älteren Linux-Distributionen und bei macOS der Fall. Python 2 ist jedoch nicht zu Python 3 kompatibel und eignet sich daher nicht für die Bearbeitung dieses Buchs.

Dennoch ist es möglich, dass auch Python 3 bereits installiert ist. Das können Sie kontrollieren, indem Sie `python3 --version` eingeben. Sollte hierbei eine passende Versionsnummer erscheinen, sind keine weiteren Maßnahmen notwendig. Wird hingegen eine Fehlermeldung angezeigt, ist es erforderlich, das Programm zu installieren.

Auf dem Mac läuft die Installation praktisch auf die gleiche Weise wie unter Windows ab. Dabei müssen Sie lediglich beim angegebenen Download-Link kontrollieren, ob Sie zur passenden Version für macOS

weitergeleitet wurden. Ist dies nicht der Fall, können Sie diese auch manuell auswählen. Danach müssen Sie nur noch den Installationsassistenten ausführen, um das Programm zu installieren.

Falls Sie hingegen mit Linux arbeiten, ist es nicht zu empfehlen, den oben verwendeten Link zu verwenden. Dieser ermöglicht es lediglich, den Quellcode herunterzuladen. Diesen müssen Sie dann selbst installieren. Deutlich einfacher ist es, für diese Aufgabe den Paketmanager zu verwenden. Bei Linux-Distributionen mit dem apt-Paketmanager benötigen Sie hierfür den folgenden Befehl:

```
sudo apt install python3
```

Verwendet Ihr System hingegen den dnf-Paketmanager, ist das folgende Kommando erforderlich:

```
sudo dnf install python3
```

PyCharm installieren

Neben dem Python-Interpreter verwenden wir noch ein weiteres Programm: PyCharm. Auch dieses können Sie kostenlos herunterladen. Es steht unter dem folgenden Link zur Verfügung:

<https://www.jetbrains.com/de-de/pycharm/download>

In Abbildung 3 sehen Sie, dass hierfür zwei verschiedene Versionen zur Auswahl stehen: Professional und Community. Während die Community-Edition kostenlos erhältlich ist, fallen für die Professional-Edition Gebühren an. Für die Bearbeitung dieses Buchs ist die Community-Ausführung jedoch vollkommen ausreichend. Daher ist es zu empfehlen, die kostenfreie Version zu wählen.

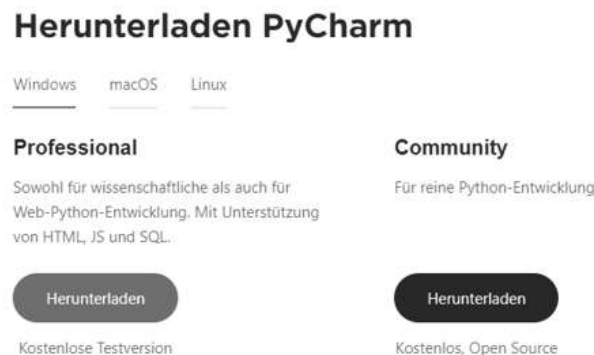


Abb. 3: Die beiden Editionen von PyCharm

Darüber hinaus ist in der Abbildung zu sehen, dass PyCharm sowohl für Windows als auch für macOS und Linux verfügbar ist. Achten Sie darauf, die passende Ausführung für Ihr Betriebssystem auszuwählen.

Auch hierbei führt Sie ein Assistent durch den Installationsprozess. Hierbei können Sie verschiedene Konfigurationen vornehmen. Es ist jedoch empfehlenswert, dabei einfach die Standard-Einstellungen zu übernehmen.

Sobald der Installationsvorgang abgeschlossen ist, können Sie PyCharm öffnen. Wenn dabei keine Probleme auftreten, ist das Programm korrekt installiert.

Vom Code zum Programm: der Python-Interpreter

Nachdem die beiden Programme korrekt installiert sind, erklären wir noch, wofür diese notwendig sind. Dabei beginnen wir mit dem Python-Interpreter.

Wenn Sie ein Computer-Programm schreiben, besteht dieses aus bestimmten Befehlen, Variablen und einigen weiteren Elementen. Diese sind alle in Schriftform in Ihrem Programm festgehalten. Der Computer arbeitet hingegen mit einem ganz anderen System, das als Maschinensprache bezeichnet wird. Hierbei werden elektrische Impulse übertragen. In der Informatik werden diese häufig mit den Zahlen 0 und 1 symbolisiert: Die Zahl 0 sagt aus, dass kein Impuls vorhanden ist, die Zahl 1 bringt zum Ausdruck, dass ein solcher Impuls ausgeführt wurde. Man spricht dabei auch von Binärcode. Die Befehle, die der Prozessor verarbeiten kann, bestehen aus einer bestimmten Folge aus elektrischen Impulsen.

Die genauen Details dieser Technik sind sehr kompliziert und es ist nicht notwendig, dass Sie diese verstehen. Wichtig ist es lediglich, dass klar wird, dass es sich bei der Textform, aus der Ihre Programme bestehen und der Maschinensprache, die der Computer verwendet, um zwei unterschiedliche Systeme handelt, die nicht direkt kompatibel zueinander sind. Um Ihr Programm auszuführen, ist daher ein Übersetzungsprozess notwendig. Dabei werden die Befehle in Ihrem Programm unter Berücksichtigung der

Regeln der verwendeten Programmiersprache in Binärcode übertragen. Diese Aufgabe übernimmt der Python-Interpreter.

Für diesen Übersetzungsprozess bestehen verschiedene Möglichkeiten. Programmiersprachen wie C oder C++ verwenden hierfür einen Compiler. Dieser führt den Übersetzungsprozess einmal durch und erzeugt dabei ein ausführbares Programm. Dieses können Sie dann immer wieder ausführen, ohne dass eine erneute Übersetzung notwendig ist.

Interpretierte Programmiersprachen wie Python – aber auch Perl, JavaScript, PHP und viele weitere – verwenden hingegen einen Interpreter. Dieser übernimmt die Übersetzung während der Ausführung jedes Mal aufs Neue. Dabei entsteht kein ausführbares Programm.

Beide Alternativen bieten gewisse Vor- und Nachteile. Kompilierte Programme sind besonders effizient und lassen sich daher schnell ausführen. Interpretierte Programme sind zwar etwas langsamer. Allerdings sind sie problemlos portierbar. Das bedeutet, dass sie sich auf unterschiedlichen Betriebssystemen ausführen lassen. Bei einem kompilierten Programm ist dies hingegen nicht möglich.

Darüber hinaus gibt es auch Programmiersprachen wie Java, die hierbei eine Zwischenlösung wählen, indem sie das Programm zunächst in einen Zwischencode kompilieren. Dieser ist jedoch noch nicht ausführbar und muss anschließend noch interpretiert werden. Das Ziel besteht hierbei darin, von den Vorteilen beider Alternativen zu profitieren.

Der Python-Interpreter übernimmt den beschriebenen Übersetzungsprozess für die Programmiersprache Python. Das bedeutet, dass er nur Python-Befehle verarbeitet. Außerdem kontrolliert er, ob Sie alle Regeln befolgt und das Programm korrekt erstellt haben. Ohne den Python-Interpreter lässt sich kein Python-Programm ausführen.

IDE oder Texteditor: unverzichtbar für die Codeerstellung

Darüber hinaus ist ein Programm notwendig, um den Quellcode zu verfassen. Textverarbeitungsprogramme wie Word sind hierfür jedoch ungeeignet. Diese speichern neben dem Text noch viele weitere Details zur Gestaltung ab – wie etwa die Schriftart und die Schriftgröße. Ein