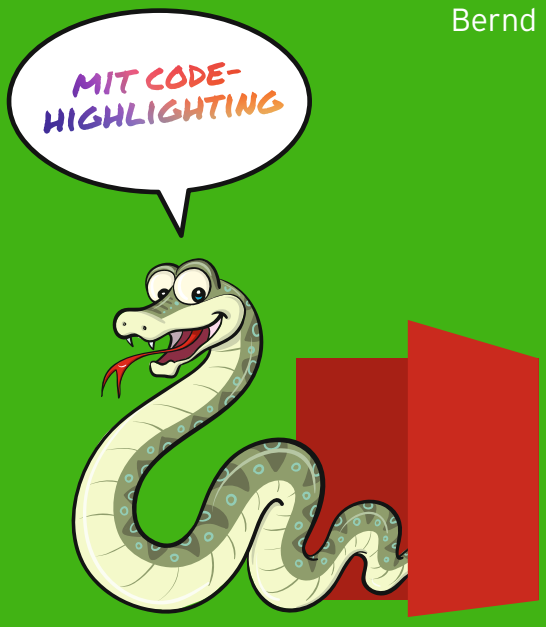
Bernd KLEIN



EINFÜHRUNG IN PYTHON 3

FÜR EIN- UND UMSTEIGER

4. Auflage

Im Internet:
Musterlösungen zu den Übungen

HANSER

Klein

Einführung in Python 3

Bernd Klein

Einführung in Python 3

Für Ein- und Umsteiger

4., vollständig überarbeitete Auflage

HANSER

Der Autor:
Bernd Klein, bernd@python-kurs.eu

Bibliografische Information der Deutschen Nationalbibliothek:

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

Lektorat: Brigitte Bauer-Schiewek
Copy editing: Jürgen Dubau, Freiburg/Elbe
Layout: le-tex publishing services, Leipzig
Umschlagdesign: Marc Müller-Bremer, www.rebranding.de, München
Umschlagrealisation: Max Kostopoulos
Titelmotiv: © istockphoto.com/zaricm
Druck und Bindung: Eberl & Kœsel GmbH & Co. KG, Krugzell
Ausstattung patentrechtlich geschützt. Kösel FD 351, Patent-Nr. 0748702
Printed in Germany

Print-ISBN: 978-3-446-46379-0
E-Book-ISBN: 978-3-446-46556-5
E-Pub-ISBN: 978-3-446-46467-4

Inhalt

Vorwort

Ist es wirklich so, dass Vorworte – ähnlich wie Bedienungsanleitungen – meistens nicht gelesen werden? Auch wenn dies sicherlich für viele zutreffen mag, so gibt es Situationen, in denen gerade ein Vorwort wertvolle Dienste leisten kann. Zum Beispiel, um die Kaufentscheidung für ein Buch zu erleichtern. So stehen auch Sie jetzt vielleicht am Regal einer guten Buchhandlung und werden möglicherweise von zwei Fragen bewegt: Sie brauchen noch eine Bestätigung, dass Python die richtige Programmiersprache für Sie ist, und möchten wissen, wie Ihnen dieses Buch helfen wird, die Sprache schnell und effizient zu erlernen.

Für Python spricht der traumhafte Anstieg seiner Bedeutung in Wissenschaft, Forschung und Wirtschaft in den letzten Jahren. Dies spiegelt sich auch in den Rankings von Programmiersprachen wider, die von verschiedenen Stellen durchgeführt werden. PYPL zählt wohl zu den bekanntesten und anerkanntesten unter diesen Webrankings. Python führt im Dezember 2020 mit 30,34 % den Index von PYPL an, gefolgt von Java mit nur 17,23 %.[1]

Weitere wichtige Gründe, Python zu lernen, sind: Python ist mit seiner einfachen natürlichen Syntax sehr einfach zu lernen. Python läuft auf allen Plattformen und erfreut sich auch beim Raspberry Pi größter Beliebtheit. Außerdem ist Python eine universell einsetzbare Programmiersprache, die sich bei den Aufgaben der Systemadministration ebenso effektiv einsetzen lässt wie im Maschinenbau, der Linguistik, der Pharmaindustrie, in der Banken- und Finanzwelt, der Physik, der Psychologie und vielen anderen Bereichen. Weil bedeutende Firmen wie Google, Facebook, Instagram, Spotify, Quora, Netflix und Dropbox Python benutzen und unterstützen, wird Python auch kontinuierlich weiterentwickelt.

Dieses Buch erscheint nun in der vierten, völlig überarbeiteten Ausgabe. Es eignet sich ebenso für Programmieranfänger als auch für Umsteiger von anderen Programmiersprachen. Behandelt werden nicht nur alle grundlegenden Sprachelemente von Python, sondern auch weiterführende Themen wie Generatoren, Dekorateure, Systemprogrammierung, Threads, Forks, Ausnahmebehandlungen und Modultests. Auf über hundert Seiten wird anschaulich und mit zahlreichen Beispielen auf die vielfältigen Aspekte der Objektorientierung eingegangen.

Brigitte Bauer-Schiewek, Lektorin

[1] Im Vorwort zur 3. Auflage stand hier: Im renommierten PYPL-Index stand Python im Juli 2017 auf dem zweiten Platz hinter Java. Während Java im Vergleich zum Vorjahr jedoch -1,1 % Anteil verloren hatte, gewann Python +4,0 % hinzu.

Danksagung

Zum Schreiben eines Buches benötigt es neben der nötigen Erfahrung und Kompetenz im Fachgebiet vor allem viel Zeit. Zeit außerhalb des üblichen Rahmens. Zeit, die vor allem die Familie mitzutragen hat. Deshalb gilt mein besonderer Dank, wie bereits bei allen vorigen Auflagen auch, meiner Frau Karola, die mich während dieser Zeit tatkräftig unterstützt hat.

Außerdem danke ich den zahlreichen Teilnehmerinnen und Teilnehmern an meinen Python-Kursen, die mir geholfen haben, meine didaktischen und fachlichen Kenntnisse kontinuierlich zu verbessern. Ebenso möchte ich den Besucherinnen und Besuchern meiner Online-Tutorials unter www.python-kurs.eu und www.python-course.eu danken, vor allem jenen, die sich mit konstruktiven Anmerkungen bei mir gemeldet haben. Wertvolle Anregungen erhielt ich auch von denen, die das Buch vorab Korrektur gelesen hatten: Stefan Günther für die Erstauflage des Buches. Für die Hilfe zur zweiten – weitestgehend überarbeiteten – Auflage möchte ich im besonderen Maße Herrn Jan Lendertse und Herrn Vincent Bermel Dank sagen. Für die dritte – wiederum stark veränderte – Auflage danke ich Herrn Wilhelm Wall für seine wertvolle Hilfe, insbesondere Tests unter macOS.

Mein besonderer Dank für die vierte Auflage gilt zum einen Herrn Tobias Habermann und zum anderen Herrn Dr. Konrad Wienands. Herr Habermann hat dafür gesorgt, dass alle Python-Beispiele dieser Auflage automatisch mittels „Pythontex" getestet und ausgeführt werden. Herr Dr. Wienands ist mit großem Sachverstand in die Rolle eines potenziellen Anfängers geschlüpft und hat mir viele wertvolle Hinweise gegeben, wo es möglicherweise offene Fragen oder Probleme bei Neulingen geben könnte. Außerdem hat er auch kleine Unstimmigkeiten gefunden.

Vergessen darf ich auch nicht all diejenigen, die die Exemplare der vorigen Auflagen gekauft und damit erst den Erfolg des Buches ermöglicht hatten. Ebenso danke ich all denjenigen, die mich in E-Mails auf kleine Fehler oder Unstimmigkeiten hingewiesen haben, die ich aber hier leider nicht alle namentlich erwähnen kann.

Zuletzt danke ich auch ganz herzlich dem Hanser Verlag, der dieses Buch – nun auch in der vierten Auflage – ermöglicht. Vor allem danke ich Frau Brigitte Bauer-Schiewek und Frau Kristin Rothe, Programmplanung Computerbuch, für die kontinuierliche ausgezeichnete Unterstützung. Für die technische Unterstützung bei LaTeX-Problemen danke ich Herrn Stephan Korell und Frau Irene Weilhart. Herrn Jürgen Dubau danke ich fürs Lektorat.

Bernd Klein, Singen

Teil I

Einleitung

1 Einleitung

1.1 Einfach und schnell zu lernen

Python ist eine Programmiersprache, die in vielerlei Hinsicht begeistert. Ein riesiger Vorteil im Vergleich zu anderen Sprachen liegt in der leichten Erlernbarkeit der Sprache. In der ersten Auflage führte das Buch den Untertitel „In einer Woche programmieren lernen". Kann man wirklich in so kurzer Zeit programmieren lernen? Das ist möglich! Vor allem mit Python. Wir erfahren es mehrmals im Monat in unseren meist fünftägigen Python-Kursen. Sie werden zum einen von Leuten, die noch keinerlei Programmiererfahrung haben, und zum anderen von Programmierenden mit Erfahrungen in C, C++, Java, Perl und anderen Programmiersprachen besucht. Manche haben auch schon vorher Erfahrungen in Python gesammelt. Aber eines ist in allen Gruppen gleich: Wir haben es immer geschafft, dass jeder programmieren gelernt hat, und vor allen Dingen – was uns am wichtigsten ist – konnten wir immer die Begeisterung für die Sprache Python entfachen.

So ist es auch eines der wichtigsten Ziele dieses Buches, die Leserinnen und Leser möglichst schnell und mit viel Freude zum selbständigen Programmieren zu bringen. Komplexe Sachverhalte werden in einfachen, leicht verständlichen Diagrammen veranschaulicht und an kleinen Beispielen eingeübt, die sich im Laufe vieler Schulungen herausgebildet und bewährt haben.

1.2 Geschichte von Python

Was hat Python mit dem lateinischen Alphabet zu tun? Beide beginnen mit ABC. Im Falle von Python ist es die Programmiersprache ABC. Die Sprache wurde Anfang der 1990er Jahre von Guido van Rossum am Zentrum für Mathematik (Centrum voor Wiskunde en Informatica) in Amsterdam entwickelt. Ursprünglich war Python als Nachfolger für die Lehrsprache ABC entwickelt worden und sollte auf dem verteilten Betriebssystem Amoeba laufen. Guido van Rossum hatte auch an der Entwicklung der Sprache ABC mitgewirkt. In einem Interview mit Bill Venners sagte er: „In den frühen 1980er Jahren arbeitete ich als Implementierer in einem Team, das eine Sprache namens ABC am Centrum voor Wiskunde en Informatica (CWI) aufbaute. Ich weiß nicht, wie gut die Leute den Einfluss von ABC auf Python kennen. Ich versuche, den Einfluss von ABC zu erwähnen, weil ich alles, was

ich während dieses Projekts gelernt habe, und die Leute, die daran gearbeitet haben, zu verdanken habe."[1]

Auch wenn sich das offizielle Logo von Python aus zwei abstrahierten Python-Schlangen zusammensetzt, hat der Name der Programmiersprache Python herkunftsmäßig nichts mit Schlangen zu tun. Für Guido van Rossum stand vielmehr die britische Komikertruppe „Monty Python" mit ihrem legendären „Flying Circus" Pate für den Namen. Guido van Rossum schrieb dazu: „Vor über sechs Jahren, im Dezember 1989, suchte ich nach einem ‚Hobby'-Programmierprojekt, mit dem ich mich während der Weihnachtswoche beschäftigen könnte. Mein Büro (ein staatliches Forschungslabor in Amsterdam) würde geschlossen sein, aber ich hatte einen Computer und nicht viel anderes vor. Ich beschloss, einen Interpreter für die neue Skriptsprache zu schreiben, über die ich in letzter Zeit nachgedacht hatte: einen Nachkommen von ABC, der Unix-/C-Hacker ansprechen würde. Ich habe Python als Arbeitstitel für das Projekt gewählt, da ich in einer etwas respektlosen Stimmung bin (und ein großer Fan von ‚Monty Python's Flying Circus')."[2]

Dennoch sind Assoziationen mit Schlangen möglich und sinnvoll: Man denke nur an das Python-Toolkit „Boa" oder die Programmiersprache Cobra. Außerdem ist eine Schlange, wie schon erwähnt, im Python-Logo.

1.3 Zen von Python

Vom Softwareentwickler Tim Peters stammt das „Zen of Python", welches er 1999 veröffentlichte. Dabei handelt es sich um „Leitprinzipien" zum Schreiben guter Python-Programme. Prinzipien, die sich sowohl an Programmierende als auch an die Designer der Sprache richten und die Sprache Python auch entscheidend prägten:

- Schön ist besser als hässlich.
- Explizit ist besser als implizit.
- Einfach ist besser als komplex.
- Komplex ist besser als kompliziert.
- Flach ist besser als verschachtelt.
- Spärlich ist besser als dicht.
- Lesbarkeit zählt.
- Sonderfälle sind nicht speziell genug, um gegen die Regeln zu verstoßen.
- Obwohl Praktikabilität die Reinheit übertrifft,
- Fehler sollten niemals stillschweigend ablaufen.
- Sofern nicht ausdrücklich zum Schweigen gebracht.
- Widerstehe im Angesicht von Zweideutigkeiten der Versuchung zu raten.

[1] Bill Venners: The Making of Python, A Conversation with Guido van Rossum, Part I, January 13, 2003, https://www.artima.com/intv/python.html

[2] Guido van Rossum schrieb dies in einem Vorwort zur ersten Auflage des Buches „Programming Python" von Marc Lutz.

- Es sollte einen – und vorzugsweise nur einen – offensichtlichen Weg geben, es zu tun.
- Obwohl dieser Weg zunächst vielleicht nicht offensichtlich ist, es sei denn, Sie sind Niederländer.
- Jetzt ist besser als nie.
- Obwohl nie oft besser ist als gerade jetzt.
- Wenn die Implementierung schwer zu erklären ist, ist es eine schlechte Idee.
- Wenn die Implementierung leicht zu erklären ist, kann dies eine gute Idee sein.
- Namespaces sind eine großartige Idee – lasst uns mehr davon machen!

1.4 Zielgruppe des Buches

Beim Schreiben eines Buches stellt man sich ein Gegenüber vor, eine Person, die das Geschriebene lesen, mögen und verstehen soll. Diese Person kann ein beliebiges Geschlecht haben, deswegen haben wir uns im Buch um geschlechtsneutrale Formulierungen bemüht. Begriffe wie „der Nutzer" oder im Plural „die Nutzer" kommen nicht mehr im Buch vor. Allerdings haben wir diese auch nicht durch Formulierungen wie „die NutzerInnen" bzw. „Nutzer*innen" ersetzt, weil diese Formulierungen von einigen abgelehnt werden. Wir denken, dass es uns gelungen ist, geschlechtsneutral zu formulieren, ohne dass der Lesefluss gestört wird oder die Gefühle von manchen verletzt werden.

Natürlich hatten wir beim Schreiben aber nicht nur eine Person im Blickfeld, sondern eine ganze Schar von Lesenden. Da sind zum einen diejenigen, die noch nie programmiert haben und Sachverhalte erklärt haben wollen, die Leute mit Programmiererfahrung in anderen Sprachen vielleicht als „trivial" oder „selbstverständlich" bezeichnen würden. Aber hier ist ein Buch wohl dem Präsenzunterricht, also wenn Lehrende und Lernende am gleichen Ort zusammen sind, deutlich überlegen: Ist einem der Stoff eines Abschnitts oder sogar Kapitels bereits vertraut, kann man es einfach überspringen, bevor man sich zu langweilen beginnt. Ebenso können, falls keine Kenntnisse in anderen Programmiersprachen vorhanden sind, Abschnitte übersprungen werden, in denen wir Ähnlichkeiten, aber auch generelle Unterschiede in der Vorgehensweise von Python im Vergleich zu anderen Sprachen herausarbeiten.

In der obigen Aufzählung fehlt aber noch eine wichtige Gruppe, nämlich diejenigen, die schon Erfahrungen mit Python haben. Dies ist eine Gruppe mit einer breiten Streuung: angefangen bei denjenigen, die bereits ein wenig reingeschnuppert haben, gefolgt von solchen, die bereits kleine oder auch größere Programme geschrieben haben, bis hin zu jenen, die sich als Experten bezeichnen. Unsere Erfahrungen zeigen, dass auch einige aus der zuletzt genannten Gruppe neue Einblicke und Sachverhalte in diesem Buch finden werden, die sich positiv auf deren zukünftige Programmierung mit Python auswirken.

Ansonsten kann dieses Buch auch bestens als Nachschlagewerk benutzt werden. Der umfangreiche Index in diesem Buch macht das Auffinden besonders einfach und erlaubt es damit, dieses Buch außerdem als Referenz zu verwenden, auch wenn es keinesfalls unter diesem Aspekt geschrieben worden ist.

1.5 Aufbau des Buches

Dieses Buch besteht aus sechs Teilen:

Teil I: Wir haben uns entschieden, die Einleitung, in der Sie sich gerade befinden, als eigenen Teil des Buches aufzunehmen. Schließlich enthält sie wichtige Informationen und sollte nach Möglichkeit auch gelesen werden.

Teil II: Im zweiten Teil behandeln wir die Grundlagen der Sprache. Wir lernen Variablen und deren Besonderheit in Python kennen. In diesem Zusammenhang lernen wir auch die Datenstrukturen wie ganze Zahlen (Integers), Fließkommazahlen (Floats), Zeichenketten (Strings), Listen, Tupel, Dictionaries und Mengen kennen. Wir beschäftigen uns eingehend mit allen Anweisungsarten von Python, d.h. Zuweisungen, Kontrollstrukturen wie bedingten Anweisungen und Schleifen sowie Möglichkeiten der Ein- und Ausgabe. Außerdem behandeln wir eingehend Funktionen, Definitionen, einfache und auch komplexere Beispiele, Parameterübergabe sowie Gültigkeitsbereiche von Variablen, und in einem umfangreichen Kapitel gehen wir auch auf rekursive Funktionen ein. Die Modularisierung in Modulen und Paketen ist ein weiterer Themenschwerpunkt.

Teil III: Den dritten Teil des Buches haben wir – entsprechend ihrer Bedeutung – ganz der Objektorientierung gewidmet. In anschaulichen Beispielen führen wir sanft in das objektorientierte Denken und Programmieren ein und zeigen die Besonderheiten von Python deutlich auf. Danach fahren wir mit den fortgeschrittenen Themen der Objektorientierung fort und behandeln neben Vererbung, Mehrfachvererbung und Slots auch extrem fortgeschrittene Themen wie Metaklassen und abstrakte Klassen.

Teil IV: Viele lieben Python, weil es neben der objektorientierten Programmierung auch wesentliche Konzepte der funktionalen Programmierung unterstützt. Deshalb widmen wir diesem Thema den vierten Hauptteil des Buches. Dieser Teil des Buches wendet sich eher an Personen mit Programmiererfahrung, aber es lohnt, sich die behandelten Konzepte anzueignen. Sie sind notwendig, um Python richtig zu verstehen. Die Listen-Abstraktionen bieten ebenso wie die `lambda`-, `map`-, `filter`- und `reduce`-Funktionen eine faszinierende Möglichkeit, die Programmierung auf ein abstrakteres Level zu bringen. Dadurch kann man komplexere Probleme mit geringerem Programmieraufwand lösen, was außerdem zu einer besseren Verständlichkeit und einfacheren Wartbarkeit der Programme führt. Dennoch kann man die gleichen Programme auch ohne diese Techniken schreiben. Eingehend befassen wir uns auch mit dem Themenkomplex „Generatoren und Iteratoren" sowie „Dekoration und Dekorateure"!

Teil V: Im fünften Teil mit dem Titel „Weiterführende Themen " verlassen wir das „eigentliche" Python und wenden uns weiterführenden Themen der Programmierung zu. Diesen Teil hätten wir auch mit „Anwendungsprogrammierung" überschreiben können. Testverfahren und Debugging gehört zu den Fragestellungen, die alle Programmiererinnen und Programmierer bewegen sollten. Im Prinzip hätten wir dieses Kapitel ebenso gut in den ersten Teil nehmen können. Außerdem behandeln wir in einem Kapitel „Reguläre Ausdrücke", die nahezu unerlässlich sind, wenn man Textverarbeitung effizient betreiben will. Man braucht sie beispiels-

weise, wenn man aus Log- oder Parameterdateien bestimmte Informationen herausfiltern will. Den ab Python 3.5 eingeführten Typanmerkungen (*type annotations*) haben wir auch ein eigenes Kapitel gewidmet. In einem weiteren Kapitel zeigen wir außerdem, wie man Daten strukturerhaltend übers Programmende hinaus konservieren kann. Das Kapitel zur Systemprogrammierung dürfte von besonderer Bedeutung für Leute sein, die gerne mit dem Betriebssystem agieren möchten, damit sie ihre Systemprogramme zukünftig unter Python und nicht mehr mit Shell-Skripting verfassen können.

Teil VI: Programmieren lernen ist vor allen Dingen eine aktive Tätigkeit. Nur ein Buch zu lesen und Beispiele nachzuvollziehen, genügt nicht. Deshalb finden Sie zu den meisten Kapiteln interessante und lehrreiche Übungsaufgaben, die den Stoff vertiefen. Aufgaben ohne Lösungen wären aber wenig sinnvoll, weshalb sich im letzten Teil dieses Buches ausführliche Musterlösungen mit Erläuterungen zu den Aufgaben befinden.

Teil VII: Auch wenn es nicht als eigener Teil aufgeführt ist, wollen wir hier noch auf das Stichwortverzeichnis eingehen. Erst ein umfangreiches Stichwortverzeichnis macht aus einem guten Buch ein nützliches und brauchbares Buch!

1.6 Programmieren lernen „interaktiv”

Wie bereits erwähnt, floss in dieses Buch die jahrelange Erfahrung sowohl in der Theorie und Praxis des Programmierens allgemein, aber vor allem auch die Vermittlung des Stoffes in zahlreichen kleinen und großen Kursen mit unterschiedlichsten Besuchertypen ein. Aber ein Buch zu schreiben, stellt dennoch eine neue Herausforderung dar. Beim Buch fehlt leider die direkte Interaktion zwischen dem, der das Wissen vermittelt, und dem Lernenden. Vor Ort kann man sofort sehen, wenn sich bei einem Teilnehmer die Stirn runzelt, große Fragezeichen erscheinen oder wenn sich jemand aus Stress das Ohrläppchen zu reiben beginnt. Dann weiß man als erfahrener Dozent, dass es höchste Zeit ist, ein paar zusätzliche Beispiele und Analogien zu verwenden, oder dass man den Stoff nochmals in anderen – möglicherweise auch einfacheren – Worten erklären sollte. Beim Schreiben eines Buches muss man diese möglichen Klippen vorhersehen und die nötigen zusätzlichen Übungen und Beispiele an den entsprechenden Stellen bereitstellen. Aber was bei vielen Lesern hilft, diese Klippen zu umgehen, führt bei anderen nun vielleicht zu Langeweile und Ungeduld, denn sie empfinden diese zusätzlichen Erklärungen, Übungen oder Beispiele möglicherweise als Zeit- oder Platzvergeudung.

Das Grundproblem ist, dass es sich bei einem Buch nicht um ein interaktives Medium handelt. Aber dank des Internets können wir Ihnen diese Interaktivität dennoch bieten. Im nächsten Abschnitt finden Sie die Adressen, wo Sie Hilfe und zusätzliche Informationen zum Buch finden.

1.7 Download der Beispiele und Hilfe

Alle im Buch verwendeten Beispiele finden Sie zum Download unter

http://www.python-kurs.eu/buch/beispiele/

Auch wenn wir das Buch so geschrieben haben, dass Sie ohne zusätzliche Hilfe auskommen sollten, wird es dennoch hier und da mal ein Problem geben, wo Sie sich vielleicht festgebissen haben. Wenn das Glück es so will, dass Sie in einer Umgebung arbeiten, in der es andere Python-Programmierende gibt, haben Sie es natürlich gut. Aber viele Leserinnen und Leser genießen diesen Vorteil nicht. Dann könnte sich ein Besuch unserer Website besonders lohnen:

http://www.python-kurs.eu/buch/

Dort finden Sie ein Korrekturverzeichnis, zusätzliche bzw. aktualisierte Übungen und sonstige Hilfestellungen. Ansonsten bleibt natürlich immer noch der Einsatz einer Suchmaschine!

1.8 Anregungen und Kritik

Falls Sie glauben, eine Ungenauigkeit oder einen Fehler im Buch gefunden zu haben, können Sie auch gerne eine E-Mail direkt an den Autor schicken: klein@python-kurs.eu.

Natürlich gilt dies auch, wenn Sie Anregungen oder Wünsche zum Buch geben wollen. Leider können wir jedoch – so gerne wir es auch tun würden – keine individuellen Hilfen zu speziellen Problemen geben.

Wir werden versuchen, Fehler und Anmerkungen in kommenden Auflagen zu berücksichtigen. Selbstverständlich aktualisieren wir damit auch unsere Informationen unter

http://www.python-kurs.eu/buch/