

LERNEN LEICHTER GEMACHT



2. Auflage

# Python programmieren lernen

für  
**dummies**<sup>®</sup>



Verstehen, warum  
es beim Programmieren  
(in Python) geht

Typische Herausforderungen  
beim Programmieren  
bewältigen

Mit Anaconda arbeiten  
und magische Methoden  
nutzen

**John Paul Mueller**

Abbildung 13.1: Eine Liste ist einfach eine Sequenz von Elementen.

Abbildung 13.2: Den zu einer Liste hinzugefügten Elementen wird jeweils der nächste freie Platz im Hauptspeicher zugeordnet.

Abbildung 13.3: Python gibt den Inhalt von `Liste1` aus.

Abbildung 13.4: Python zeigt Ihnen eine Liste aller Aktionen an, die Sie für Listen ausführen können.

Abbildung 13.5: Sie müssen immer den korrekten Index eingeben.

Abbildung 13.6: Mit Bereichsangaben lassen sich mehrere Werte zurückgeben.

Abbildung 13.7: Lässt man den Endindex eines Wertebereichs leer, wird der Listenrest ausgegeben.

Abbildung 13.8: Lässt man den Startindex leer, werden alle Elemente ab Element 0 ausgegeben.

Abbildung 13.9: Mit einer Schleife können Sie sehr einfach auf eine Kopie jedes Elements zugreifen und sie bearbeiten.

Abbildung 13.10: Sie können in Ihrer Anwendung bei Bedarf prüfen, ob eine Liste leer ist.

Abbildung 13.11: Das Anhängen eines Elements speichert den Wert am Ende der Liste und verändert die Länge der Liste.

Abbildung 13.12: Mit der Funktion `insert()` können Sie ein Element an einer beliebigen Stelle einfügen.

Abbildung 13.13: Mit den Funktionen `copy()` und `extend()` können Sie schnell viele Daten kopieren und einfügen.

Abbildung 13.14: Mit `pop()` können Sie Elemente am Ende der Liste entfernen.

Abbildung 13.15: Existiert eine Farbe in der Liste, wird eine Erfolgsmeldung ausgegeben.

Abbildung 13.16: Gibt man eine Farbe ein, die in der Liste nicht existiert, wird eine Fehlermeldung ausgegeben.

Abbildung 13.17: Mit der Funktion `sort()` ist das Sortieren einer Liste ganz einfach.

Abbildung 13.18: Durch die Nutzung des Splat-Operators wird Ihr Code deutlich kürzer.

Abbildung 13.19: Mit String-Funktionen können Sie Ihre Ausgabe speziell formatieren.

Abbildung 13.20: Verwenden Sie die Funktion »format()«, um bestimmte Arten von Ausgaben mit Ihren Anwendungen zu erhalten.

Abbildung 13.21: Das Counter-Objekt ist praktisch, wenn Statistiken zu sehr langen Listen benötigt werden.

## Kapitel 14

Abbildung 14.1: Tupel werden in runden statt eckigen Klammern angegeben.

Abbildung 14.2: Für Tupel scheint es weniger Funktionen zu geben.

Abbildung 14.3: Die neue Kopie von MeinTupel enthält einen zusätzlichen Eintrag.

Abbildung 14.4: Verwenden Sie Indexe, um auf einzelne Tupeleinträge zugreifen zu können.

Abbildung 14.5: Ein Dictionary speichert die Einträge in sortierter Reihenfolge.

Abbildung 14.6: Mit Dictionarys wird der Zugriff auf Einträge über deren Wert leicht und selbstbeschreibend.

Abbildung 14.7: Sie können sich bei Dictionarys eine Liste der Schlüssel ausgeben lassen.

Abbildung 14.8: Aussagekräftige Schlüssel können die Datenausgabe erleichtern.

Abbildung 14.9: Dictionarys können sehr einfach geändert werden.

Abbildung 14.10: Die Anwendung zeigt zu Beginn das Auswahlmenü an.

Abbildung 14.11: Die Anwendung zeigt das Menü wieder an, nachdem sie Ihre Auswahl ausgegeben hat.

Abbildung 14.12: Neue Werte werden oben auf dem Stapel abgelegt.

Abbildung 14.13: Ist der Stapel voll, kann man darauf keinen neuen Eintrag speichern.

Abbildung 14.14: Die Funktion pop() entfernt einen Eintrag von der Oberseite des Stapels.

Abbildung 14.15: Sorgen Sie dafür, dass Ihre Anwendung Über- und Unterläufe erkennen kann.

Abbildung 14.16: Speichert die Anwendung neue Einträge in der Warteschlange, gibt diese nicht mehr an, dass sie leer ist.

Abbildung 14.17: Die Eigenschaften einer Warteschlange muss man sehr häufig prüfen.

Abbildung 14.18: Eine Deque hat zwei Enden und die entsprechenden

Funktionen, um diese zu verwalten.

## Kapitel 15

Abbildung 15.1: Die Instanz enthält die benötigte Variable.

Abbildung 15.2: Der Klassenname stimmt auch. Also wissen Sie nun, dass die Instanz mit `MeineKlasse` erzeugt wurde.

Abbildung 15.3: Verwenden Sie die Funktion `dir()`, um herauszufinden, welche integrierten Attribute es gibt.

Abbildung 15.4: Python stellt Ihnen zu jedem Attribut, das es Ihrer Klasse hinzufügt, einen Hilfeintrag zur Verfügung.

Abbildung 15.5: Die Klassenmethode gibt eine einfache Nachricht aus.

Abbildung 15.6: Die Instanzmethode wird als Teil des Objekts aufgerufen und gibt eine einfache Nachricht aus.

Abbildung 15.7: Die erste Version des Konstruktors stellt einen Standardwert für den Namen zur Verfügung.

Abbildung 15.8: Übergibt man dem Konstruktor einen Namen, wird die Ausgabe entsprechend angepasst.

Abbildung 15.9: Sie können den Wert von `Gruss` ändern.

Abbildung 15.10: Die Änderung von `Gruss` überträgt sich in die Instanz.

Abbildung 15.11: Als Ausgabe erhält man einfach die Summe zweier Zahlen.

Abbildung 15.12: Der Code kann eine beliebige Anzahl an Listeneinträgen verarbeiten.

Abbildung 15.13: Wenn man zwei Objekte der Klasse `MeineKlasse` miteinander addiert, erhält man als Ergebnis ein Objekt desselben Typs.

Abbildung 15.14: Die Ausgabe zeigt, dass die Klasse voll funktionsfähig ist.

Abbildung 15.15: Hildegard hat Geburtstag und sagt ein paar Worte.

## Kapitel 16

Abbildung 16.1: Das Beispiel zeigt, wie die Daten im CSV-Format aussehen.

Abbildung 16.2: Die Ausgabe der Anwendung in Excel

Abbildung 16.3: Die ausgegebene `.csv`-Datei in WordPad

Abbildung 16.4: Die Daten, nachdem sie durch die Anwendung verarbeitet wurden

Abbildung 16.5: Die Anwendung zeigt die einzelnen Änderungen direkt an.

Abbildung 16.6: Die geänderten Daten erscheinen in der Datei `GeänderteDatei.csv` wie erwartet.

## Kapitel 17

Abbildung 17.1: Die `localhost`-Adresse bezieht sich auf Ihren Computer.

Abbildung 17.2: Adressen, die Sie zum Versenden von E-Mails verwenden, sind über das ganze Internet hinweg eindeutig.

Abbildung 17.3: Der `localhost` hat sowohl eine IPv6-Adresse als auch eine IPv4-Adresse.

Abbildung 17.4: Viele Webserver stellen nur eine IPv4-Adresse bereit.

Abbildung 17.5: Standardisierte Ports bieten bestimmte Dienste auf jedem Server an.

Abbildung 17.6: Manchmal müssen Sie den Namen des lokalen Rechners kennen.

Abbildung 17.7: Vermeiden Sie möglichst die Verwendung von fest codierten Werten für das lokale System.

Abbildung 17.8: Python fügt Ihrer Nachricht ein paar zusätzliche Daten hinzu, die benötigt werden, damit sie funktioniert.

Abbildung 17.9: Die Testnachricht in der Statusleiste eines Android-Tablets

Abbildung 17.10: Die HTML-Ausgabe enthält wie erwartet eine Überschrift und einen Textabschnitt.

# Einleitung

---

Python ist ein Beispiel für eine Sprache, die in ihrem speziellen Bereich alles richtig macht. Das ist nicht nur meine Meinung: Die Programmierer haben durch die Nutzung der Sprache Python abgestimmt und sie mittlerweile weltweit zur Nummer 4 unter den Computersprachen gemacht (eine Rangliste dazu finden sie unter <https://www.tiobe.com/tiobe-index/>). Das Tolle an Python ist, dass Sie tatsächlich Anwendungen auf einem System schreiben und sie dann auf jeder anderen Zielplattform verwenden können. Im Gegensatz zu anderen Programmiersprachen, die Plattformunabhängigkeit versprechen, bietet Python diese Unabhängigkeit tatsächlich. Python hat in dieser Hinsicht sein Versprechen gehalten.

Python legt den Schwerpunkt auf die Lesbarkeit des Codes und eine sparsame Syntax, mit der Sie Anwendungen mit weniger Codezeilen als in anderen Sprachen schreiben können. Sie können auch einen Programmierstil verwenden, der Ihren Anforderungen gerecht wird, zumindest sofern Python diese funktionalen, imperativen, objektorientierten und prozeduralen Programmierstile auch unterstützt (mehr dazu erfahren Sie in [Kapitel 3](#)). Außerdem ist Python aufgrund seiner Eigenschaften in sehr vielen Bereichen vertreten, in denen auch Quereinsteiger und nicht nur reine Programmierer tätig sind. Die vorliegende zweite Auflage von *Python programmieren lernen für Dummies* wurde geschrieben, um allen – Programmierern und Nichtprogrammierern – zu helfen, Python schnell einsetzen zu können.

Mancher sieht Python als Skriptsprache, aber eigentlich ist es sehr viel mehr als das. ([Kapitel 19](#) vermittelt Ihnen einen Eindruck davon, in welchen Bereichen Python genutzt wird.) Wie dem auch sei, Python stellt seine Fähigkeiten im Bildungssektor und in anderen Bereichen unter Beweis, für deren Anforderungen andere Programmiersprachen möglicherweise nicht geeignet sind. Tatsächlich wird für die Programmierbeispiele in diesem Buch *Jupyter Notebook* verwendet, das sich das vom Stanforder Computerwissenschaftler Donald Knuth propagierte Programmierparadigma der guten Lesbarkeit auf die Fahnen schreibt (mehr dazu in [Kapitel 4](#)). Auch Ihre Programme werden am Ende wie sehr gut lesbare Berichte aussehen, die beinahe jedermann leicht verstehen kann.

## Über dieses Buch

In *Python programmieren lernen für Dummies* dreht sich alles um den schnellen Einstieg in Python. Sie möchten die Sprache sicherlich schnell erlernen, um Ihre eigentliche Arbeit zu erledigen, egal worum es sich dabei handelt. Anders als andere Bücher zeigt Ihnen dieses Buch von Anfang an, wie sich Python von anderen Sprachen unterscheidet und wie es Ihnen dabei helfen kann, als Nicht-Programmierer gute Arbeit