

PYTHON

PROGRAMMIEREN LERNEN MIT

CHATGPT

Als Einsteiger 5-mal schneller
professionelle Anwendungen programmieren
mit künstlicher Intelligenz (KI)

2. Auflage 2024

AES Verlag

Inhaltsverzeichnis

1. Warum Sie programmieren lernen sollten	5
1.1. Warum Python der perfekte Einstieg ins Programmieren ist	5
1.2. Die Geschichte von Python.....	7
1.3. Was ist an Python so besonders?	8
2. So legen Sie richtig mit Python los.....	10
2.1. Python.....	11
2.2. Die passende IDE.....	14
3. „Hello World“ – Der Einstieg in die Programmierwelt	16
3.1. Der Programmcode	16
3.2. Ausführen eines Programms.....	22
4. Künstliche Intelligenz.....	25
4.1. Generative AI.....	26
4.2. Large Language Models (LLMs)	27
4.3. ChatGPT	31
5. Variablen.....	33
5.1. Strings	35
5.1.1. Aneinanderhängen von Variablen	36
5.1.2. Sonderzeichen	37
5.1.3. Nützliche Funktionen	39
5.2. Datenstrukturen	40
5.2.1. Listen	41

5.2.2. Tupel.....	44
5.2.3. Sets.....	45
5.2.4. Dictionaries.....	46
5.3. Mathematische Operatoren.....	48
5.4. Logische Operatoren.....	49
6. Kontrollstrukturen.....	51
6.1. If-Abfragen.....	51
6.2. While-Schleifen.....	56
6.3. For-Schleifen.....	59
6.4. Abbruch von Schleifen.....	64
6.5. Übungen zu Schleifen.....	65
<i>Übung1 – Würfeln.....</i>	<i>65</i>
<i>Übung2 – Ratespiel.....</i>	<i>67</i>
<i>Übung3 – Duplikate Eliminieren.....</i>	<i>68</i>
<i>Lösung zu Übung1.....</i>	<i>69</i>
<i>Lösung zu Übung2.....</i>	<i>70</i>
<i>Lösung zu Übung3.....</i>	<i>71</i>
7. Funktionen.....	73
8. Objektorientierte Programmierung.....	77
8.1. Klassen und Objekte.....	77
8.2. Vererbung.....	84
8.3. Übungen zur objektorientierten Programmierung.....	88
<i>Übung4 – Würfeln.....</i>	<i>88</i>
<i>Übung5 – Personalverwaltung.....</i>	<i>89</i>
<i>Lösung zu Übung4.....</i>	<i>92</i>
<i>Lösung zu Übung5.....</i>	<i>94</i>

9. Exception Handling	98
9.1. Fehlerquellen erkennen.....	98
9.2. Fehler abfangen und verarbeiten.....	101
10. Was Sie als Python-Programmierer wissen sollten	104
10.1. Vergleich von Objekten und Variablen.....	104
10.2. Python 2 und Python 3	108
11. Programmieren mit ChatGPT	109
11.1. Codegenerierung	110
11.2. Fehlersuche	116
11.3. Codeoptimierung	121
11.4. Beispiele und Übungen.....	123
11.5. Nachteile.....	124
12. Bonus: Fortgeschrittene Übungen	125
12.1. Übung 6 – Temperaturumrechner.....	125
<i>Lösung zu Übung 6</i>	<i>127</i>
12.2. Übung 7 – Textverarbeitung mit Dateien.....	131
<i>Lösung Übung 7</i>	<i>133</i>
12.3. Übung 8 – Rekursive Programmierung.....	135
<i>Lösung zu Übung 8</i>	<i>137</i>
Schlusswort	140
Bevor Sie das Buch schließen	141

1. Warum Sie programmieren lernen sollten

1.1. Warum Python der perfekte Einstieg ins Programmieren ist

Python wird oft als Programmiersprache angesehen, die sowohl vielseitig einsetzbar, als auch einfach zu erlernen ist. Damit eignet sich diese Sprache ideal für Programmieranfänger. Obwohl Python bereits Anfang der 90er entwickelt wurde, ist die Programmiersprache auch heute noch, immerhin 30 Jahre später, noch immer für viele Anwendungsbereiche extrem relevant. Dabei weist Python einige Besonderheiten auf, die die Sprache von anderen Sprachen aus dieser Zeit abhebt. Dazu gehört zum Beispiel die Tatsache, dass Python auch interaktiv genutzt werden kann. Es ist also möglich, Zeile um Zeile einzugeben und jeweils sofort eine entsprechende Ausgabe zu erhalten. Dazu aber später mehr.

Wegen der vielfältigen Einsatzmöglichkeiten bietet Python eine hervorragende Gelegenheit für Anfänger in einem Anwendungsgebiet ihrer Wahl erste Erfahrungen zu sammeln. Zudem eröffnet das Beherrschen einer Programmiersprache eine Vielzahl beruflicher Perspektiven. Im digitalen Zeitalter ist es also fast schon ein Muss sich

wenigstens die Grundlagen des Programmierens anzueignen. Gerade in der IT-Branche gibt es hervorragende Karrieremöglichkeiten. Mit Python sind Sie dafür bestens gerüstet.

Python dient ebenfalls als Türöffner um weitere Programmiersprachen, wie Java, C#, C++, JavaScript oder Swift einfacher und schneller zu erlernen. Die Grundlagen aus diesem Buch werden Ihnen ebenfalls bei diesen Programmiersprachen behilflich sein.

Des Weiteren bietet dieses Buch nicht nur einen Einblick in Python als Programmiersprache, sondern auch in die faszinierende Welt der künstlichen Intelligenz. In Kapitel 4 werde ich Sie mitnehmen auf einen kleinen Exkurs in deren Theorie und im Laufe des Buches werden Sie lernen, wie Sie KI-Anwendungen, wie ChatGPT, gewinnbringend einsetzen können.

Somit haben Sie sich genau für die richtige Sprache und natürlich die passende Anleitung entschieden, um die Grundlagen Pythons und des Programmierens ganz allgemein zu lernen. Glückwunsch dazu! Nun trennt Sie nur noch das Durchlesen des Buches und natürlich – ganz wichtig – das Durchführen der praktischen Übungen vom Erfolg mit Python.

Ich wünsche Ihnen viel Spaß und Erfolg bei der Umsetzung!

1.2. Die Geschichte von Python

Wie eingangs bereits erwähnt, begann der Entwicklungsprozess bereits in den frühen 90er Jahren. Python wurde von dem Niederländer Guido van Rossum als Skriptsprache am Zentrum für Mathematik in Amsterdam entwickelt. Ursprünglich war die Sprache als Nachfolger für die Programmiersprache „ABC“ gedacht und sollte auf dem Betriebssystem Amoeba eingesetzt werden.

1994 erschien die erste Vollversion Python 1.0. Sechs Jahre später folgte dann Python 2.0. Python 2 wird in der Version 2.7.2 bis heute verwendet, obwohl Python 3 bereits 2008 erschienen ist. Da die dritte Vollversion allerdings nicht abwärtskompatibel zu Version 2 ist, existieren beide Versionen parallel. Weiterentwickelt wird indessen nur Version 3. Aus diesem Grund wird in diesem Buch ausschließlich mit Version 3 gearbeitet. Inwiefern sich die beiden Versionen im Einzelnen unterscheiden, wird in einem späteren Kapitel nochmals näher betrachtet werden.

Der Name „Python“ leitet sich nicht wie allgemein angenommen von der gleichnamigen Schlangen-Familie ab, sondern stammt von der englischen Comedy-Gruppe „Monty Python“.

1.3. Was ist an Python so besonders?

Im Gegensatz zu den meisten anderen Programmiersprachen ist die Formatierung des Codes in Python ein funktionaler Bestandteil der Programmiersprache. Diese Eigenschaft ist eine der größten Stärken Pythons. Zum einen benötigen logische Blöcke dadurch keine geschweiften Klammern, was die Lesbarkeit des Codes fördert. Zum anderen werden dadurch viele Fehlerquellen vermieden, da syntaktische Elemente, wie beispielsweise Semikolons überflüssig sind und vom Programmierer daher auch nicht vergessen werden können.

Eine weitere Stärke Pythons ist, dass die Sprache durch die relativ geringe Anzahl an Schlüsselwörtern als leicht zu erlernen gilt. Mit Python lassen sich sowohl große Anwendungen implementieren, aber auch kleine Skripte schreiben.

Ähnlich wie Java, läuft Python dabei auf fast allen gängigen Betriebssystemen und bietet von Haus aus wichtige Features, wie zum Beispiel eine Garbage Collection an. Diese übernimmt die Bereinigung des Speichers, so dass nicht mehr genutzte Variablen und Objekte nicht manuell beseitigt werden müssen.

Im Gegensatz zu Java, C oder C++ ist Python allerdings keine kompilierte Sprache. Bei kompilierten Sprachen wird der vom Entwickler geschriebene Quellcode in eine Form von für den Computer verständlichen Code umgewandelt – dieser Vorgang nennt sich Kompilieren. Python hingegen nutzt einen sogenannten Interpreter. Dieser verarbeitet den Quellcode eines Programms zur Laufzeit. Es wird also Zeile für

Zeile verarbeitet bis das Programm beendet wurde oder ein Fehler auftritt. Die Ausführung eines Programms mithilfe eines Interpreters ist normalerweise langsamer, als die Ausführung mittels eines Compilers. Dies liegt auch daran, dass man ein kompiliertes Programm nicht bei jedem Durchlauf neu kompilieren muss. Der bereits kompilierte Code kann einfach erneut ausgeführt werden. Ein Interpreter hingegen muss eben jedes Mal aufs Neue den Code „live“ interpretieren.

2. So legen Sie richtig mit Python los

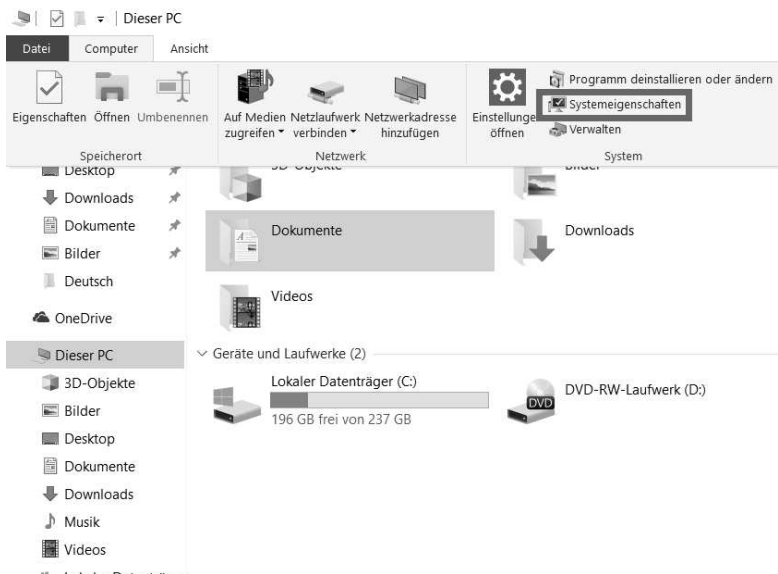
Das Schreiben eines lauffähigen Python-Programms ist nur der erste Schritt. Um Programme in Python ausführen zu können, benötigen Sie den Python-Interpreter. Außerdem ist es sinnvoll auch eine passende Entwicklungsumgebung zu nutzen. Denn obwohl Python als anfängerfreundliche Sprache gilt, kann etwas Hilfe beim Entwicklungsprozess nicht schaden. Bevor Sie mit dem Programmieren loslegen, sollten wir also zuallererst darüber nachdenken, welche Softwarekomponenten benötigt werden. Das Wichtigste ist natürlich der Python-Interpreter. Dieser ist zusammen mit den Python-Standardbibliotheken kostenlos im Internet erhältlich. Außerdem benötigen Sie, wie bereits erwähnt, eine entsprechende Entwicklungsumgebung, die Ihnen dabei hilft, Fehler im Code zu vermeiden und eine bessere Übersicht über Ihre Projekte zu behalten. Wo Sie diese Softwarekomponenten beziehen können, werden Sie in diesem Kapitel erfahren. Des Weiteren beinhaltet dieses Kapitel eine Anleitung aller wichtigen Installationschritte, sodass Sie so schnell wie möglich mit dem eigentlichen Programmieren loslegen können.

2.1. Python

Die neuste Python-Version kann von der offiziellen Python-Webseite bezogen werden. In diesem Fall wäre dies Version 3.12.3 (Stand Mai 2024).

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

Üblicherweise erkennt die Webseite automatisch, welches System man verwendet und bietet direkt die korrekte Version für das eigene Betriebssystem an – in diesem Beispiel Windows 64bit. Sollten Sie sich unsicher sein, welche Version die richtige für Sie ist, so können Sie das im Explorer unter *Computer > Systemeigenschaften* einsehen.



Screenshot 1: Schaltfläche für Systemeigenschaften

Alternativ kann diese Information auch über die Konsole mit dem Befehl `systeminfo` erlangt werden, wie in Screenshot 2 unter *Systemtyp* ganz unten im Bild zu sehen ist. Die Windows-Konsole können Sie über die Tastenkombination *Windowstaste + r* gefolgt von der Eingabe „cmd“ + *Enter* öffnen.



```

Eingabeaufforderung
(c) 2019 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\simon>systeminfo

Hostname:                DESKTOP-07JS4JV
Betriebssystemname:       Microsoft Windows 10 Home
Betriebssystemversion:    10.0.18363 Nicht zutreffend Build 18363
Betriebssystemhersteller: Microsoft Corporation
Betriebssystemkonfiguration: Eigenständige Arbeitsstation
Betriebssystem-Buildtyp:  Multiprocessor Free
Registrierter Benutzer:  simon
Registrierte Organisation:
Produkt-ID:
Ursprüngliches Installationsdatum: 18.02.2020, 11:00:36
Systemstartzeit:         15.06.2020, 13:25:56
Systemhersteller:        HP
Systemmodell:             HP 250 G5 Notebook PC
Systemtyp:                x64-based PC

```

Screenshot 2: systeminfo

Nachdem Python heruntergeladen wurde, kann die Installation gestartet werden. Hier sollten Sie den Haken bei „Add Python 3.12.3 to PATH“ setzen. Anschließend können Sie entweder die Option „Install Now“ wählen oder aber die Installation nach Ihren Wünschen anpassen.

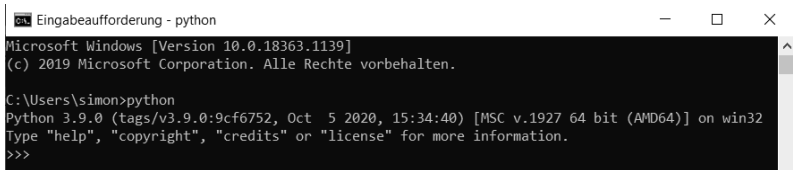


Screenshot 3: Python Installation

Um zu überprüfen, ob Python richtig installiert wurde, reicht es die Kommandozeile (Windows-Konsole) zu öffnen und den Befehl

`python`

einzugeben. Hat bei der Installation alles geklappt, so sollte sich die Python-Konsole öffnen – siehe Screenshot 4. Hier können Sie auch sehen, welche Python-Version bei Ihnen installiert ist.



Screenshot 4: Python-Konsole

Um diese Konsole wieder zu verlassen und zur ursprünglichen Windows-Konsole zurückzukehren, genügt die Tastenkombination Strg + Z gefolgt von der Eingabetaste.

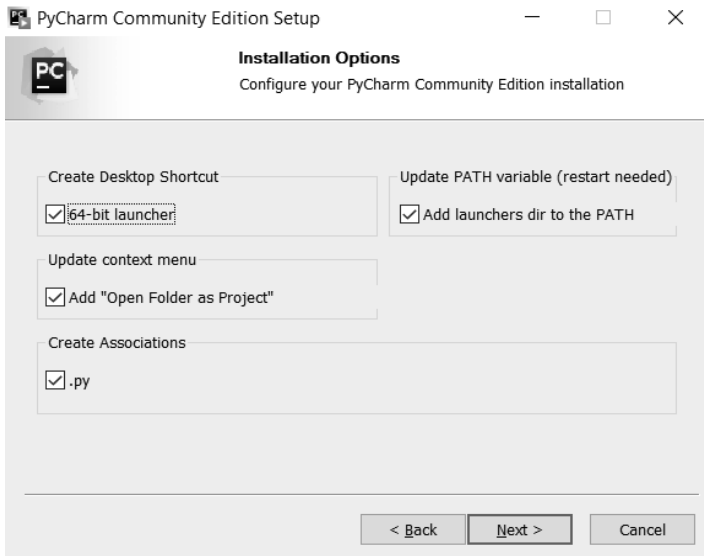
2.2. Die passende IDE

IDE ist die Abkürzung für eine Integrierte Entwicklungsumgebung (im Englischen: integrated development environment). Diese Programme unterstützen den Entwickler beim Programmieren, da sie zum einen Fehler in der Syntax bereits während des Programmierens hervorheben, und zum anderen das Ausführen des geschriebenen Codes vereinfachen.

Entwicklungsumgebungen gibt es in Hülle und Fülle. Oft sind Sie auf eine Programmiersprache spezialisiert. Somit gibt es für fast jede Sprache eine große Auswahl an kostenlosen und kostenpflichtigen Entwicklungsumgebungen verschiedener Anbieter. Kostenpflichtige Varianten bieten in der Regel einen höheren Funktionsumfang. Für Anfänger sind allerdings auch die meisten kostenlosen Varianten für den Einstieg in das Programmieren vollkommen ausreichend. Daher wird in diesem Buch die Community-Edition von PyCharm verwendet. PyCharm ist eine IDE speziell für Python und wird von der Firma JetBrains vertrieben. Die Community-Edition ist komplett kostenfrei erhältlich. Sie können sich PyCharm ganz einfach von der offiziellen Webseite herunterladen:

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

Achten Sie auch hier darauf die jeweils richtige Version für Ihr Betriebssystem herunterzuladen. Nach Ausführen des heruntergeladenen Programms können Sie die Installation nach Ihren Wünschen anpassen. Beim in Screenshot 5 abgebildeten Installationsschritt, empfehle ich Ihnen, die Haken bei „Create Associations“ und „Update PATH variable“ zu setzen. Die anderen beiden Optionen können Sie nach Belieben auswählen, wobei die Option „Open Folder as Project“ später nützlich sein könnte.



Screenshot 5: Installation PyCharm

Klicken Sie anschließend auf „Weiter“ und dann auf „Install“. Um die Installation abzuschließen müssen Sie Ihr System neu starten.

Nun sind Sie bereit mit dem Programmieren loszulegen. Sie haben alle erforderlichen Vorbereitungsschritte erfolgreich absolviert. Viel Spaß mit Kapitel 3!