

# Python im Selbststudium

Die vorliegenden Lernpfade bieten eine umfassende Einführung in die Programmiersprache Python und richten sich an Personen, die Python von Grund auf erlernen möchten. Im Laufe von neun Wochen werden die Teilnehmer von den absoluten Grundlagen über fortgeschrittene Konzepte bis hin zur objektorientierten Programmierung und darüber hinausgeführt.

Die ersten Wochen konzentrieren sich auf die grundlegenden Aspekte von Python, einschließlich Syntax, Variablen, Datenstrukturen und Kontrollstrukturen. Die darauffolgenden Wochen vertiefen das Wissen in spezifischeren Bereichen wie Funktionen, Modulen und der Standardbibliothek.

Im weiteren Verlauf des Kurses wird ein besonderer Schwerpunkt auf die objektorientierte Programmierung gelegt, ein Konzept, das für die Strukturierung und Organisation komplexer Python-Projekte von entscheidender Bedeutung ist.

Weiterhin werden viele interessante Techniken im Bereich der Datenanalyse und Netzwerkprogrammierung angesprochen, mit denen während der täglichen Arbeit anspruchsvolle Lösungen erarbeitet werden können.

Jede Woche besteht aus Phasen des Selbststudiums, praktischen Aufgaben zur Vertiefung des gelernten Wissens, sowie Übungstagen, in denen das Erlernte wiederholt und gefestigt wird. Abschließende Tests dienen der Überprüfung des Lernerfolgs und der Identifizierung von Bereichen, in denen noch Nachholbedarf besteht.

Unser Ziel ist es, den Teilnehmern ein tiefgreifendes Verständnis der Python-Programmierung zu vermitteln und sie darauf vorzubereiten, selbstständig Python-Projekte durchführen zu können.

# Gesamtübersicht

## 1. Woche

- a. Installation
- b. Syntax
- c. Variablen
- d. Datentypen
- e. Einfache Ein-/Ausgabe
- f. Verzweigung
- g. Schleife

## 2. Woche

- a. Strings
- b. Listen
- c. Tupel
- d. Mengen
- e. Listenoperationen
- f. Dictionaries
- g. Dictionary-Operationen

## 3. Woche

- a. Funktionen
- b. Parameter und Rückgabewerte
- c. Module
- d. Modulimport
- e. Modulinstallation
- f. Wichtige Module der Standardbibliothek

## 4. Woche

- a. Konzepte der OOP
- b. Klassendefinition
- c. Objektanlage
- d. Objekte und Datenstrukturen
- e. Methodenanlage
- f. Attribute und der Zugriff darauf

## 5. Woche

- a. Vererbung und Erstellen von Unterklasse
- b. Aufrufe von geerbten Methoden
- c. Überschreiben von Methoden und Polymorphie
- d. Abstrakte Klassen und Methoden
- e. Schnittstellen und Protokolle
- f. Dunder-Methoden

## 6. Woche

- a. Reguläre Ausdrücke
- b. re Modul
- c. Testen von Anwendungen
- d. Unittest
- e. Pytest
- f. Dokumentation von Anwendungen

7. Woche

- a. Sockets
- b. RESTful APIs
- c. JSON und YAML
- d. Requests
- e. Beautiful Soup

8. Woche

- a. Datenstrukturen
- b. CSV-Dateien
- c. Relationale Datenbanken (SQLite)
- d. SQL-Syntax in Python
- e. CRUD-Operationen
- f. NoSQL Datenbanken (MongoDB)

9. Woche

- a. Datenanalyse
- b. Daten(vor)verarbeitung
- c. Datenvisualisierung
- d. NumPy
- e. Pandas
- f. matplotlib

## Literaturliste und andere Quellen

Es gibt eine unüberschaubare Menge an Büchern, Kursen, Webseiten und hot-tos zu Python. Alleine wenn Sie bei percipio (<https://dbsysteme1ms.percipio.com>) das Suchwort „Python“ eingeben, bekommen Sie aktuell 3636 Ergebnisse. Genauso ist es z.B. auf amazon, udemy oder ganz allgemein bei einer Google-Abfrage.

Um Ihnen bei der Auswahl geeigneter Quellen zu helfen, folgen ein paar Buchtitel und Kurse, von denen wir meinen, dass sie Sie gut unterstützen können. In den einzelnen Lernpfaden wird dann genauer aufgezeigt, welche Kapitel und Inhalte passen.

Darüber hinaus wollen wir Ihnen auch noch ein paar interessante Quellen im Internet zeigen, die Sie beim Erlernen von Python unterstützen können.

### Literatur

Die folgenden für diesen Kurs ausgesuchten Titel stehen Ihnen online zur Verfügung. Wir werden in den einzelnen Wochen auf die Kapitel verweisen, die behandelt werden und die Sie durcharbeiten sollten.

### Grundlagen und Fortgeschritten

- **Python Crash Course**, Eric Matthes – dies ist unsere Hauptreferenz. Bitte schauen Sie sich auch [https://ehmatthes.github.io/pcc\\_2e/cheat\\_sheets/cheat\\_sheets/](https://ehmatthes.github.io/pcc_2e/cheat_sheets/cheat_sheets/) an, wo Sie „cheat sheets“ herunterladen können. Auf github gibt es ein Repository mit den Aufgaben aus dem Buch.
- **Automate the boring stuff with Python**, Al Sweigart – hier finden Sie viele kleine Projekte, die mit Python realisiert werden und wo Sie „so ganz nebenbei“ auch noch Python erlernen können.
- **Serious Python**, Julien Danjou – hier werden fortgeschrittene Themen abgehandelt, mit denen Sie Ihr Verständnis von Python noch weiter vertiefen können.

### Einzelthemen

- **Network Programming in Python**, John Galbraith – dieses Buch soll die 8. Woche begleiten und handelt darüber hinaus noch sehr viel mehr Themen im Bereich der Netzwerk-Programmierung ab.
- **Getting Structured Data from the Internet**, Jay M. Patel – in diesem Buch geht es um Webscraping, also das Durchforsten von Webseiten und die Verarbeitung der gefundenen Daten. Außerdem findet sich ein Kapitel zum Thema Datenbanken.

- **Hands-on Data Analysis and Visualization with Pandas**, Purna Chander Rao. Kathula – Eine sehr gute Einführung in die Techniken der Datenanalyse mit Python und daher auch die Basis für die 9. Woche.

## Programmiertechniken

- **Clean Python**, Sunil Kapil – Nachdem Sie die Grundlagen beherrschen und auch fortgeschrittene Konzepte kennen, geht die Reise weiter. Und da geht es vor allem darum „schönen“ Code zu schreiben. Diesem Thema widmet sich dieses Buch.
- **The Pythonic Way**, Sonal Raj – Auch dieses Buch widmet sich dem guten Stil in Python und gibt Ihnen sicher einige weitere Anregungen, wie sie gut strukturierte und lesbare Python-Programme schreiben können.
- **Practices of the Python Pro**, Dane Hillard – Ein sehr schönes Buch zum Thema „wirklich“ fortgeschrittenes Programmieren mit Python, mit einer Vielzahl von praktischen Betrachtungen zu den bei der professionellen Programmentwicklung auftretenden Fragen.
- **Practical Python Design Patterns**, Wessel Badenhorst – Design Pattern, also Entwurfsmuster, sind gängige Lösungsansätze für typische, immer wiederkehrende Programmieraufgaben. Wer mit diesen Pattern vertraut ist, kann schneller saubere Lösungen für seine Programmieraufgaben entwerfen.
- **Python Algorithms**, Magnus Lie Hetland – Dieses Buch ist relativ alt, aber immer noch gut. Denn es geht hier um grundlegende Fragestellung der Algorithmik, also des Programmentwurfs. Und diese Fragestellungen sind auch genauso heute noch wichtig.
- **Python Data Structures**, Oswald Campesato – Und auch dieses Buch widmet sich einem wichtigen Thema der täglichen Arbeit mit Python, nämlich den Datenstrukturen. Lassen Sie sich überraschen, was man alles mit Listen, Dictionaries und Strings bauen kann, wenn man sie clever kombiniert.

## Fragen, Quizzes und Projekte

- **Test Your Skills in Python**, Dr. Shivani Goel - Ein Selbsttest-Buch, kombiniert mit einer sehr kurzen und knappen Einführung in die Python-Themen. NumPy und Datenbanken werden auch angesprochen.
- **Python Challenges**, Michael Inden – Lernen durch Üben. 100 Aufgaben verschiedener Schweregrade und Lösungsvorschläge.
- **Python Interview Questions**, Swati Saxena – Nicht nur für das Job-Interviews interessant, sondern auch als weitere Quelle für Selbsttests.
- **Tiny Python Projects**, Ken Youens-Clark – Einige interessant Übungsprojekte, wobei auch auf das Thema „Testing“ Wert gelegt wird.

## Sonstiges

- **Visual Studio Code for Python Programmers**, April Speight - Da wir Visual Studio Code für den Kurs und die tägliche Arbeit empfehlen, ist das hier eine gute Einführung und Übersicht, wie Sie einrichten und nutzen können.
- **Impractical Python Projects**: Playful Programming Activities to Make You Smarter, Lee Vaughan – Spass am Programmieren mit mehr oder minder „seltsamen“ Projekten. Einfach mal reinschauen, unser persönlicher Favorit ist Kapitel 10: „Sind wir alleine im Universum“

Je nach Kapitelinhalt werden wir Ihnen noch weiterführende Titel nennen, mit denen die Inhalte vertieft werden können.

Darüber hinaus möchten wir hier noch einige Titel nennen, die aktuell nicht direkt in percipio zur Verfügung stehen, die aber auch für Sie von Interesse sein könnten.

- Python auf Deutsch, sehr umfassend. Zwar für eine etwas ältere Python-Version, aber immer noch ein guter Einstieg: <https://openbook.rheinwerk-verlag.de/python/>

## Videokurse

Wenn Sie eher eine Videopräsentation der Inhalte bevorzugen, werden wir uns auf die folgenden Inhalte konzentrieren. Auch hier werden für die jeweiligen Wochen die empfohlenen Kapitel angegeben werden. Hier erst einmal die Liste der Videokurse, die dafür ausgewählt wurden:

- Codecademy: Python for Programmers – Ein erster Einstieg
- Codecademy: Learn Python 3 – Ein umfassender Kurs mit 27h Dauer
- Codecademy: Learn Intermediate Python 3 – Fortgeschrittene Programmier Techniken
- Codecademy: Learn Advanced Python 3 – Sehr fortgeschrittene Programmier Techniken mit Python
- Codecademy: Learn Advanced Python 3: Database Operations – Datenbankzugriffe
- Codecademy: Getting Started with Python for Data Science – Einstieg in die Datenanalyse
- Codecademy: Learn Data Visualization with Python – Matplotlib im Einsatz
- Codecademy: Learn Intermediate Python 3 - Exceptions and Unit Testing – Fehlerbehandlung und Testen
- Python Requests - HTTP Requests mit Python

## Übungen und Labs

Es finden sich auf percipio auch Übungskurse, Workshops und Projekte. Eine Auswahl davon wurde getroffen und soll Ihnen hier präsentiert werden. Vielleicht findet sich ja etwas für Sie, mit dem Sie

Ihre Praxiserfahrung aufbauen können. Denn eines ist für diesen Kurs wichtig: coden, coden, coden – dann kommt der Rest von ganz alleine:

- Lab: Python Novice – Einsteiger
- Lab: Python Apprentice – Schwerpunkt OOP
- Lab: Python for Developers – Grundlegendes Python inklusive Datenstrukturen
- Lab: Pythonista – Sehr fortgeschrittene Techniken
- Lab: Dynamic Data Handling with Python - Datenbanken (MySQL) und CSV
- Lab: Python Journeyman – PyTest, Flask, requests

## Sonstige Quellen

Im Internet finden sich ebenfalls hilfreiche Seiten, um Python zu erlernen und damit zu üben. Hier eine kleine Liste von Links, die vielleicht interessant sind, einfach mal reinschauen.

## Installation und Dokumentation

- Die Originalquellen und Dokumentation: [www.python.org](http://www.python.org)
- Die alternative Distribution, die u.a. jupyter bereitstellt: [www.anaconda.com](http://www.anaconda.com)
- Das Repository für python-Module (via pip-Installation): [pypi.org](http://pypi.org)

## Online Playgrounds

Wenn Sie Python nicht installieren wollen, aber ein wenig damit programmieren, gibt es eine Vielzahl von Seiten, auf denen das möglich ist. Um eine Auflistung zu bekommen, geben Sie einmal „python playground“ ein. Einige Vertreter seien hier genannt:

<https://programiz.pro/learn/python/online-compiler>

<https://repl.it> – hier braucht man ein Konto, dafür ist die Umgebung auch ausgefeilter. Darüber hinaus gibt es auch kostenpflichtige Angebote, aber zu Beginn sind diese nicht nötig.

## Online-Foren

Manchmal weiß man nicht weiter, versteht etwas nicht oder hat keine Idee, wo man starten soll. Dann sind Online-Communities eine gute Quelle, um eine Lösung zu finden. Und seien wir ehrlich, „gut geklaut“ ist oftmals besser als lange rumprobiert. Wichtig ist dabei, dass man mit einem Kommentar auf die Fundstelle hinweist, um später nachvollziehen zu können, wo die Lösung herkam. Und natürlich auch, dass man verstanden hat, was da übernommen wurde.

Die wichtigste Community ist sicherlich [stackoverflow.com](https://stackoverflow.com). Einfach die Problembeschreibung bei Google eingeben und dann aus den Ergebnissen das Forum auswählen, in dem es behandelt wird. Stackoverflow wird sicher mit dabei sein.

## Andere nützliche Links

Sie wollen verstehen, was passiert, wenn ein Programm ausgeführt wird? Dann schauen Sie einmal bei **pythontutor.com** vorbei. Zuerst einmal ist es eine einfache Online-Umgebung, um Programme schrittweise auszuprobieren, also ein weiterer Online-Playground. Und es wird auch nur Python 3.6 unterstützt (aktuell sind wir schon bei 3.11).

Aber trotzdem kann man sehr viel damit lernen, denn während der Programmausführung wird das Geschehen im Speicher mit angezeigt, also wo Variablen liegen, wie sie miteinander im Zusammenhang stehen und sich ändern. Damit können Sie sehr schnell ein Verständnis für die inneren Abläufe bei der Programmausführung entwickeln.

Sie wollen **reguläre Ausdrücke** entwerfen oder analysieren? Auch hierfür gibt es viel Unterstützung im Netz. Eine sehr schöne Seite ist **regex101.com**. Hier geben Sie einen Beispielttext ein, entwerfen dann den regulären Ausdruck, während gleichzeitig analysiert wird, was gerade passiert und wie die Ergebnisse aussehen werden. Zum Schluss lassen Sie sich den Code erzeugen und können ihn dann in Ihrem Programm nutzen. Einfacher geht es nicht mehr.

Und es gibt soviel mehr, was man finden kann, hier ein paar Suchbegriffe:

- python online programming practice
- python examples and solutions

Und für die Hardwarebastler:

- micropython (ESP32, Raspberry Pico)

Scheuen Sie sich auch nicht, gute Newsletter zu abonnieren. Hier ein paar Beispiele:

- PyCoders Weekly – ein Nachrichten-Aggregator: <https://pycoders.com/>
- PythonSpeed – viele interessante Artikel um Python und andere Themen: <https://pythonspeed.com>
- RealPython – eine sehr gute Seite mit vielen spannenden Artikeln: <https://realpython.com>
- PlanetPython: <https://www.planetpython.org>