

Michael Weigend



So einfach!

Programmieren lernen mit **Python**



Spielend lernen anhand von anschaulichen Bildern

Mit einfachen Schritt-für-Schritt-Anleitungen

Für Kinder und Erwachsene – ab 10 Jahre

Inhalt

Einleitung	8
------------	---


1 Willkommen in der Welt der Programmierung	10
---	----


1.1 Computerprogramme: Schlaue Texte für dumme Computer	12
1.2 Wahrscheinlich kannst du schon programmieren	16
1.3 Lösungen zu den Aufgaben im Text	22

2 Einstieg in Python	24
----------------------	----



2.1 Was ist Python?	26
2.2 Python als Taschenrechner – die IDLE-Shell	28
2.3 Funktionen aufrufen	33
2.4 Module	34
2.5 Variablen	38
2.6 Der Python-Interpreter	41

3 Die ersten Programme	42
------------------------	----




3.1 Der Python-Editor	44
 Projekt: Der Begrüßungsautomat	45
3.2 Ein Programm im Dateimanager starten	49
3.3 Strings – Experimente in der IDLE-Shell	51
3.4 Das EVA-Prinzip	52
3.5 Zahlen eingeben	52
3.6 So oder so? Programmverzweigung	56

3.7 Zeilen und Blöcke – Layout eines Python-Programms.....	59
3.8 Noch einmal bitte! Wiederholung	60
 Projekt: Zahlenraten.....	64
3.9 Wahr oder nicht wahr: Logische Aussagen.....	65
3.10 Logische Operatoren	66
3.11 Schwierige Entscheidungen	68
3.12 Datum und Zeit.....	72




4 Neue Möglichkeiten mit Funktionen 74

4.1 Wie definiert man eine Funktion?	77
4.2 Funktionen können Werte zurückgeben	83
 Projekt: Volumenassistent – ein Programm mit Auswahlmenü.....	86
 Projekt: Der Briefassistent – globale Variablen und Seiteneffekte	89
4.3 Bilder mit der Turtle-Grafik	92
4.4 Was ist Rekursion?	100
4.5 Rekursive Funktionen	101






5 Daten in Kollektionen zusammenfassen 104

5.1 Welche Arten von Kollektionen gibt es?.....	106
5.2 Suchen und aufzählen	110
5.3 Sequenzen verarbeiten	112
5.4 Listen können sich verändern.....	114
 Projekt: Planeten aufzählen	117
 Projekt: Digitales Telefonbuch	119
5.5 Dateien öffnen, speichern und bearbeiten.....	122
 Projekt: Eine Notiz schreiben und als Textdatei speichern.....	125
5.6 Laufzeitfehler abfangen.....	126




5.7	Kollektionen speichern und laden	128
	Projekt: Telefonnummern verwalten	129
	Projekt: Die Reimmaschine	131
	Projekt: Wie warm wird es? – Informationen aus dem Internet	134

6 Bunt und zum Anklicken: Grafische Benutzungsoberflächen 138

	Projekt: Digitaler Würfel	141
6.1	Widgets konfigurieren	143
	Projekt: Farbmischer	147
	Projekt: Dichten mit Goethe	150
	Projekt: StarQuest – Bilder anzeigen	154
	Projekt: Digitaluhr	157

7 Der nächste Schritt: Die große Welt der Informatik 162

7.1	Die Geschwindigkeit von Programmen	164
7.2	Objektorientierte Programmierung – eigene Klassen definieren	168
7.3	Agile Software-Entwicklung – arbeiten wie die Profis	171
	Projekt: Digitales Notizbuch – agil entwickelt	172
7.4	Webseiten programmieren mit JavaScript	180
7.5	Noch mehr Programmiersprachen	184
7.6	Programmierwettbewerbe	187

Lösungen 188

Stichwortverzeichnis 205

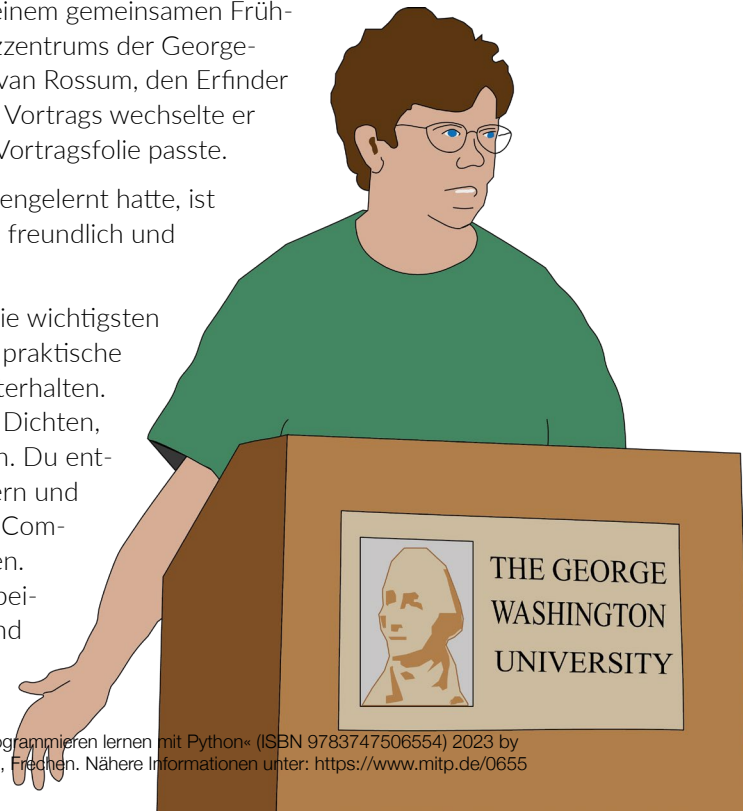
Einleitung

Python ist eine Programmiersprache, die sehr leicht zu erlernen ist und die sich deshalb perfekt für deinen Einstieg in die Programmierung eignet.

Ich selbst habe Python im Jahr 2000 kennengelernt. Damals war ich an der Technischen Universität Dortmund und betreute ein Team, das E-Learning-Material zur Einführung in die Informatik entwickelte. Ein Kapitel widmete sich dem Thema »Objektorientierte Programmierung« mit der bis dahin ziemlich unbekannten Programmiersprache Python. Als ich die ersten Python-Skripte sah, war ich sofort begeistert. Kurze Texte, deren Bedeutung man rasch erfassen konnte. Ideal zum Lernen. Zwei Jahre später schrieb ich für den mitp-Verlag mein erstes Python-Buch. Im Frühjahr 2003 besuchte ich zum ersten Mal die PyCon, die internationale Konferenz der Python-Community, in Washington DC (USA). Hier trafen sich Wissenschaftlerinnen und Wissenschaftler aus der ganzen Welt und berichteten in Vorträgen, wie sie Python in ihren Fachgebieten einsetzten. Die Konferenztage begannen immer mit einem gemeinsamen Frühstück an großen Tischen im Ballsaal des Konferenzzentrums der George-Washington-Universität. Hier traf ich auch Guido van Rossum, den Erfinder der Programmiersprache Python. Während seines Vortrags wechselte er mehrmals sein T-Shirt, damit die Farbe immer zur Vortragsfolie passte.

So wie die Menschen, die ich auf der PyCon kennengelernt hatte, ist auch die Programmiersprache selbst: Sie ist offen, freundlich und lädt zum Mitmachen ein.

Freue dich auf spannende Projekte, in denen du die wichtigsten Konzepte von Python kennenlernst. Du schreibst praktische Rechenprogramme, die sich mit dem Benutzer unterhalten. Du gestaltest Spiele und digitale Assistenten zum Dichten, Briefeschreiben und zum Bestimmen von Pflanzen. Du entwickelst Benutzungsoberflächen mit Eingabefeldern und Schaltflächen. In kreativen Projekten lässt du den Computer Bilder zeichnen und Geschenkpapier drucken. Du lernst, wie man Daten aus dem Internet verarbeiten kann und wie man eine App zum Speichern und Verwalten von Notizen entwickelt.





Die Programmbeispiele sind kurz und können von dir weiter ausgebaut werden. Alle wichtigen Codezeilen werden durch Texte und Bilder erklärt.

Du kannst die Programmbeispiele von der Website des mitp-Verlags herunterladen. Der URL ist:

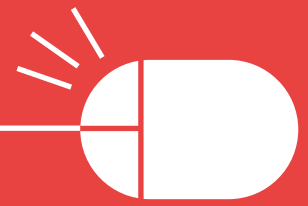
<https://www.mitp.de/0655>

Wähle die Registerkarte **Downloads** und klicke auf den Link **Programmbeispiele**. Nach dem Entpacken hast du für jedes Kapitel ein Verzeichnis mit den Programmen, die im Buch erklärt werden.

Viel Spaß mit Python!

Michael Weigend

3 Die ersten Programme



3.1 Der Python-Editor

🚩 Projekt: Der Begrüßungsautomat

3.2 Ein Programm im Dateimanager starten

3.3 Strings – Experimente in der IDLE-Shell

3.4 Das EVA-Prinzip

3.5 Zahlen eingeben

3.6 So oder so? Programmverzweigung

3.7 Zeilen und Blöcke – Layout eines Python-Programms

3.8 Noch einmal bitte! Wiederholung

🚩 Projekt: Zahlenraten

3.9 Wahr oder nicht wahr: Logische Aussagen

3.10 Logische Operatoren

3.11 Schwierige Entscheidungen

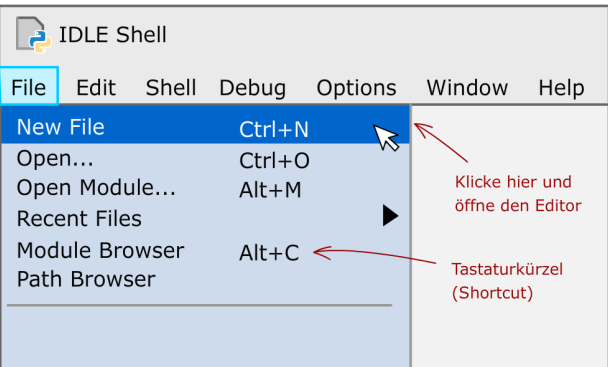
3.12 Datum und Zeit

Bisher hast du nur einzelne Befehle in der Python-Shell ausprobiert. In diesem Kapitel verwenden wir den Editor der Programmierumgebung *IDLE*, um Programme mit mehreren Anweisungen zu schreiben.

3.1 Der Python-Editor

Den Editor starten

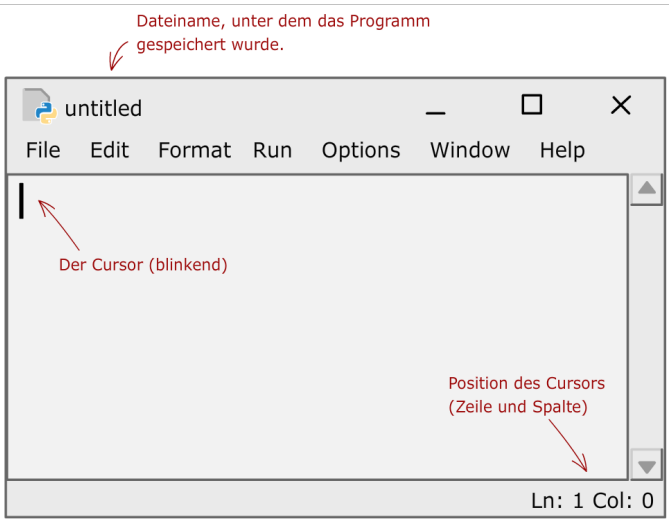
Ein Editor ist – ganz allgemein – eine Software, mit der man Texte oder Bilder bearbeiten kann. IDLE enthält einen speziellen Editor für Python-Programme. Starte IDLE und klicke dann im Menü **File** (Datei) auf den Befehl **New File** (Neue Datei). Dann öffnet sich ein neues Editor-Fenster. Oben im Rahmen trägt es den Titel `untitled` (ohne Titel).



Das Editor-Fenster

Im Editor-Fenster von IDLE kannst du einen Programmtext eingeben.

Es ist ein Texteditor mit einigen Besonderheiten, die die Programmierung erleichtern. Wie in der Python-Shell werden Textpassagen je nach Bedeutung automatisch in unterschiedlichen Farben angezeigt. Das nennt man auch *Syntax-Highlighting* (sprich: Süntax-Hailaiting). Zum Beispiel sind die Namen von Standardfunktionen lila.



Shortcuts

Menüs mit Befehlen zum Anklicken sind ja schön und gut, aber die Handhabung kostet Zeit. Für Editor-Befehle, die du häufig verwendest, solltest du dir Tastenkürzel merken.

Taste	Wirkung
<code>[Strg] + [N]</code>	Neues Editor-Fenster
<code>[Strg] + [S]</code>	Speichern
<code>[F5]</code>	Programm starten
<code>[Alt] + [G]</code>	Gehe zu Zeilennummer ...
<code>[Strg] + [C]</code>	Kopieren
<code>[Strg] + [V]</code>	Einfügen



Projekt Der Begrüßungsautomat

Die Idee

Wenn Menschen sich treffen, begrüßen sie sich. Programmiere einen Begrüßungsautomat. Der Benutzer wird nach seinem Namen gefragt. Dann wird er freundlich begrüßt.



Programmierung

1 Programmtext eingeben

- Starte die Programmierungsumgebung IDLE.
- Wähle den Menübefehl **File|New File** oder drücke das Tastenkürzel `Strg` + `N`.
- Schreibe in das Editorfenster das Python-Programm.

In dem Begrüßungsprogramm spielen Texte eine wichtige Rolle. Ein Text wird in der Programmierung *Zeichenkette* oder *String* (engl. für Kette) genannt. Ein String steht immer in Anführungszeichen, z.B. 'Wie heißt du?'.

```

untitled
File Edit Format Run Options

print('Wie heißt du?')
name = input('Name: ')
gruß = 'Hallo ' + name + '!'
print(gruß)
  
```

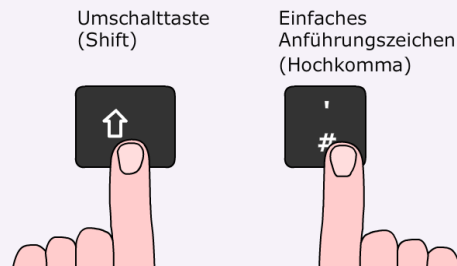
Standardfunktionen sind lila.

Strings sind grün.

PRAXISTIPP

Anführungszeichen

Das einfache Anführungszeichen ' findest du auf der Taste mit dem Hashtag-Zeichen #. Verwechsele es nicht mit einem Apostroph '.



PRAXISTIPP

Achte auf die Textfarbe!

Das Syntax-Highlighting hilft dir, Schreibfehler zu erkennen. Zum Beispiel sind Strings im IDLE-Editor grün.

Wenn du bei einem String wie 'Hallo' das zweite Anführungszeichen vergessen hast, fällt dir das so leicht auf.

```
gruß = 'Hallo + name + '!''
```

Hier wurde das zweite Anführungszeichen vergessen.

Hoppla, das sollte eigentlich nicht grün sein.

2 Speichern

Speichere dein Programm in deinem Projektordner unter dem Namen `gruss.py` ab. Das geht so:

- ➔ Wähle im Menü **File** den Befehl **save as** (Speichern unter).
- ➔ Gehe in deinen Projektordner.
- ➔ Gib den Dateinamen `gruss.py` ein. Achte darauf, dass du die Datei-Endung `.py` verwendest.

Übrigens: Im Menü findest du auch die Shortcuts der Befehle (Tastenkürzel).

2: Nach dem ersten Speichern steht hier der Dateiname.

File	Edit	Format	Run	Options	Window
New File			Ctrl+N		
Open...			Ctrl+O		
Open Module...			Alt+M		
Recent Files					
Module Browser			Alt+C		
Path Browser					
Save			Ctrl+S		
Save As...			Ctrl+Shift+S		
Save Copy As			Alt+Shift+S		

Shortcut *Strg+N*

3: Wenn das Programm schon einen Dateinamen hat, speicherst du mit diesem Befehl.

1: Zum Speichern beim ersten Mal hier klicken!

Warum soll ich denn die Datei `gruss.py` nennen? Warum schreibe ich nicht `gruß.py`?

In Dateinamen verwendest du besser kein ß und keine Umlaute wie ä, ö, ü. Denn manche Betriebssysteme kommen damit nicht zurecht.

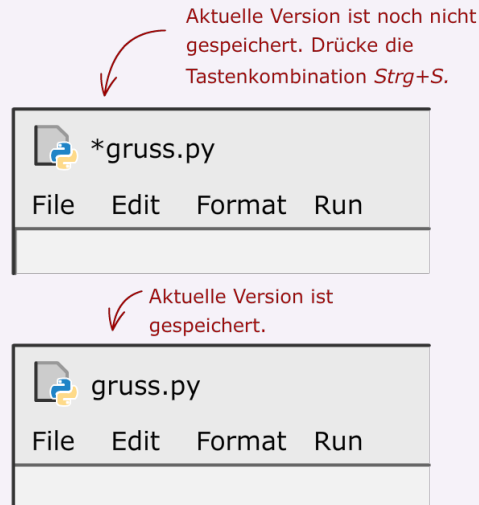
Und dein Programm soll doch auf allen Computern laufen, oder?



PRAXISTIPP

Das Geheimnis des Sternchens

Während du dein Programm entwickelst, solltest du dein Projekt immer wieder zwischenspeichern. Das geht am schnellsten, indem du die Tastenkombination `Strg+S` drückst. Achte auf den Dateinamen am oberen Rand des Editor-Fensters! Wenn vor dem Dateinamen ein Sternchen steht, ist die aktuelle Version deines Programms noch nicht gespeichert.



3 Das Programm mit IDLE starten

So startest du dein Programm: Klicke im Menü **Run** (Ausführen) auf den Befehl **Run module** (Programm-Modul ausführen) oder drücke die Taste `F5`.

Es öffnet sich die IDLE-Shell mit der Meldung **RESTART** und das Programm läuft. Du kannst mit dem Computer kommunizieren.

```
>>> = RESTART:
      wie heißt du?
      Name: Tina
      Hallo Tina!
>>>
```

Wenn die Ausführung des Programms beendet ist, erscheint wieder der Prompt der IDLE-Shell.