

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

## **Hinweis des Verlages zum Urheberrecht und Digitalen Rechtemanagement (DRM)**

Liebe Leserinnen und Leser,

dieses E-Book, einschließlich aller seiner Teile, ist urheberrechtlich geschützt. Mit dem Kauf räumen wir Ihnen das Recht ein, die Inhalte im Rahmen des geltenden Urheberrechts zu nutzen. Jede Verwertung außerhalb dieser Grenzen ist ohne unsere Zustimmung unzulässig und strafbar. Das gilt besonders für Vervielfältigungen, Übersetzungen sowie Einspeicherung und Verarbeitung in elektronischen Systemen.

Je nachdem wo Sie Ihr E-Book gekauft haben, kann dieser Shop das E-Book vor Missbrauch durch ein digitales Rechtemanagement schützen. Häufig erfolgt dies in Form eines nicht sichtbaren digitalen Wasserzeichens, das dann individuell pro Nutzer signiert ist. Angaben zu diesem DRM finden Sie auf den Seiten der jeweiligen Anbieter.

Beim Kauf des E-Books in unserem Verlagsshop ist Ihr E-Book DRM-frei.

Viele Grüße und viel Spaß beim Lesen,

*Ihr mitp-Verlagsteam*



# Programmieren lernen mit Python

*Für Carolin Dorothea*

Michael Weigend

**So einfach!**

# Programmieren lernen mit Python



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.

ISBN 978-3-7475-0656-1

1. Auflage 2023

[www.mitp.de](http://www.mitp.de)

E-Mail: [mitp-verlag@sigloch.de](mailto:mitp-verlag@sigloch.de)

Telefon: +49 7953 / 7189 - 079

Telefax: +49 7953 / 7189 - 082

© 2023 mitp Verlags GmbH & Co. KG, Frechen

Dieses Werk, einschließlich aller seiner Teile, ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlages unzulässig und strafbar. Dies gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen. Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Lektorat: Janina Bahlmann

Sprachkorrektur: Jürgen Benvenuti

Covergestaltung & Coverbild: Christian Kalkert

Illustrationen: Michael Weigend

Satz: Ulrich Borstelmann

Layoutgestaltung: Kai Neugebauer

# Inhalt

<b>Einleitung</b>	<b>8</b>
<b>1 Willkommen in der Welt der Programmierung</b>	<b>10</b>
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
<b>2 Einstieg in Python</b>	<b>24</b>
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
<b>3 Die ersten Programme</b>	<b>42</b>
3.1 Der Python-Editor	44
 <b>Projekt:</b> 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
 <b>Projekt:</b> 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
 <b>Projekt:</b> Volumenassistent – ein Programm mit Auswahlmenü. ....	86
 <b>Projekt:</b> 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
 <b>Projekt:</b> Planeten aufzählen .....	117
 <b>Projekt:</b> Digitales Telefonbuch .....	119
5.5 Dateien öffnen, speichern und bearbeiten. ....	122
 <b>Projekt:</b> Eine Notiz schreiben und als Textdatei speichern. ....	125
5.6 Laufzeitfehler abfangen. ....	126



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

## **6 Bunt und zum Anklicken: Grafische Benutzungsoberflächen** **138**

 <b>Projekt:</b> Digitaler Würfel .....	141
6.1 Widgets konfigurieren .....	143
 <b>Projekt:</b> Farbmischer .....	147
 <b>Projekt:</b> Dichten mit Goethe .....	150
 <b>Projekt:</b> StarQuest – Bilder anzeigen .....	154
 <b>Projekt:</b> 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
 <b>Projekt:</b> 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 unbekannteren 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 kennlernst. 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.

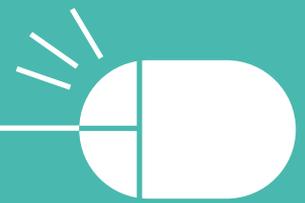
Viel Spaß mit Python!

Michael Weigend

1

Willkommen  
in der Welt der  
Programmierung

---



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

## 1.1 Computerprogramme: Schlaue Texte für dumme Computer

Programmieren heißt ein Programm schreiben. Statt Programm sagt man auch *App* oder *Software*. Ein Programm ist ein Text, eine Schritt-für-Schritt-Anleitung für einen Computer. Es besteht aus Anweisungen, die von einem Computer verstanden und ausgeführt werden können.

So ganz von allein macht ein Computer nichts. Programme sagen ihm, was er tun soll. Damit der Computer deine Anweisungen versteht, musst du sie in einer Programmiersprache schreiben. Python ist eine solche Programmiersprache. Das Schöne ist: Python-Programme können nicht nur von einem Computer verstanden werden, sie sind auch für Menschen gut lesbar.

Programmierer nennen einen Programmtext oft auch *Code* und das Programmieren *Coding*.



### Die universelle Maschine

Ein Computer ist eine universelle Maschine. Das heißt, er hat nicht nur einen einzigen Verwendungszweck, sondern man kann ihn für alle möglichen Aufgaben einsetzen.

Früher gab es für jeden Verwendungszweck eine spezielle Maschine, zum Beispiel ein Telefon zum Telefonieren, einen Kompass, der die Himmelsrichtung anzeigt, und so weiter.

All das und noch viel mehr kann heute ein einziger Computer, zum Beispiel dein Smartphone. Mit deinem Smartphone kannst du telefonieren, spielen, deinen Weg finden und vieles mehr. Für jeden Verwendungszweck gibt es eine App.



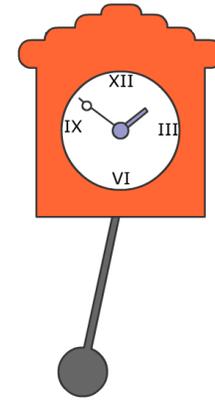


## Analog

In einer Sanduhr rieseln winzige Glaskugeln (kein Sand) durch eine enge Öffnung. Die Menge der durchgerieselten Glaskugeln entspricht der verstrichenen Zeit. Das nennt man *analog* (von altgriechisch *analogos* für »entsprechend«).

Auch eine mechanische Uhr mit Zeigern arbeitet analog. Der Winkel, um den sich ein Zeiger gedreht hat, entspricht der Zeit, die vergangen ist.

Heute sind viele analoge Geräte durch Computerprogramme ersetzt worden. Das nennt man *Digitalisierung*.



Es ist  
sechs Uhr  
fünfundvierzig.



## Digital

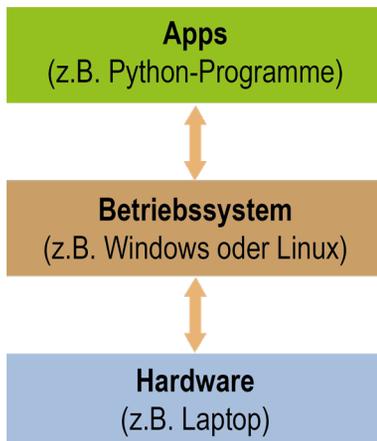
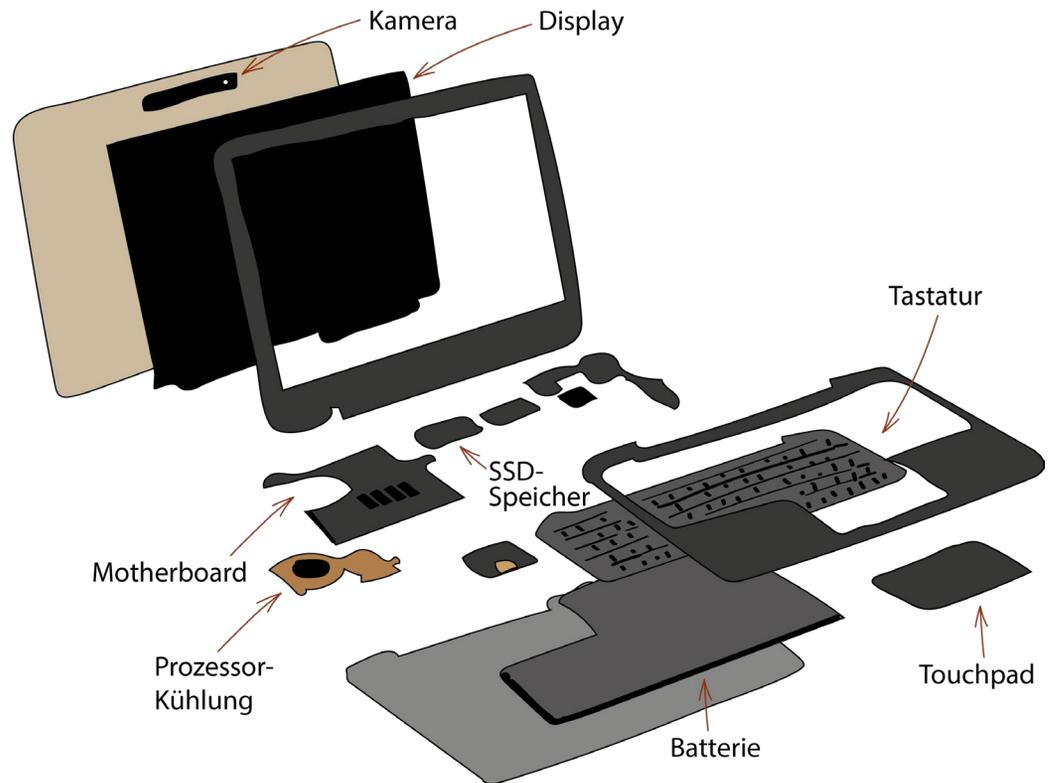
Eine Digitaluhr enthält einen Computer, der für die Anzeige der genauen Uhrzeit sorgt. Weil man Programme schneller und einfacher entwickeln kann als mechanische Geräte mit Zeigern und Zahnrädern, gibt es heute viele unterschiedliche Arten von digitalen Uhren.

Später in diesem Buch erfährst du, wie du mit Python Uhren programmieren kannst (siehe Kapitel 3 und 6).

## Hardware und Software

Ein Computer besteht aus Hardware und Software. Die Hardware umfasst alle Teile, die man anfassen kann: das Gehäuse, das Display, Lautsprecher, Kamera, die Chips auf dem Motherboard im Inneren, Batterie, Kabel und so weiter.

Die Software ist unsichtbar. Es sind alle Programme und Daten, die sich im Speicher befinden.



## Betriebssystem

Eine besonders wichtige Software ist das Betriebssystem (zum Beispiel Windows, Linux oder macOS). Das Betriebssystem sorgt dafür, dass die Apps auf die Hardware zugreifen können.



## Updates

Die Hardware eines modernen Computers ist sehr stabil und geht selten kaputt. Die Software jedoch bekommt häufig ein Update, das heißt, sie wird durch eine neue, bessere Version ersetzt. Updates sind vor allem wichtig, wenn man Sicherheitslücken feststellt.

Wenn man ein Handy oder einen PC nicht mehr weiterverwenden kann, liegt das meist daran, dass man für alte Geräte keine Updates mehr bekommt.



In der Europäischen Union arbeitet man an einem Gesetz, dass Computerhersteller mindestens fünf Jahre lang Updates liefern müssen.

So kann man Computer länger nutzen.

## ? WAS IST EINE SICHERHEITSLÜCKE?

Eine Sicherheitslücke ist ein Fehler in der Software, die es Hackern ermöglicht, den Computer z.B. über das Internet anzugreifen. Bei einem solchen Angriff können Daten verändert oder gelöscht werden. Oder es wird bösartige Software installiert, die später auf deinem Computer Schaden anrichtet.



## 1.2 Wahrscheinlich kannst du schon programmieren

Wetten, dass du schon einige Programmierertechniken aus dem Alltag kennst? Programmieren ist nämlich gar nichts Besonderes.

### Algorithmus

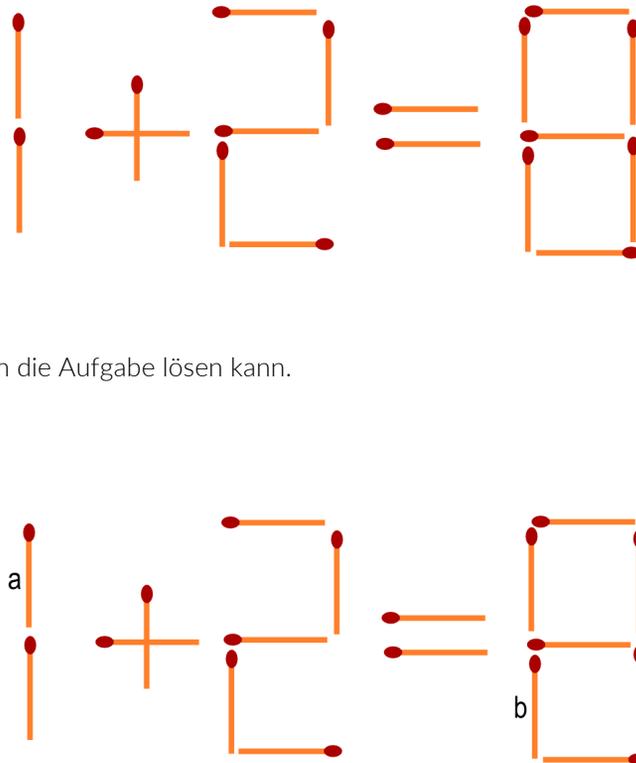
Im Prinzip ist ein Programm eine genaue Anleitung für den Computer, wie er eine bestimmte Aufgabe lösen soll. Eine solche Anleitung nennt man auch Algorithmus. Auch die Lösung zu einem Streichholzrätsel ist ein Algorithmus. Kannst du das folgende Rätsel lösen?

*Lege ein einziges Streichholz um und Sorge dafür, dass die Gleichung stimmt.*

Am Ende des Kapitels findest du ein Bild mit der Lösung. Aber auch ein Text kann erklären, was zu tun ist:

Nimm das Streichholz, das in der Zahl Acht senkrecht unten links liegt. Lege dieses Streichholz waagrecht links neben den Kopf des oberen Streichholzes der Zahl Eins.

Verstanden? Dieser Text ist eine genaue Anleitung, wie man die Aufgabe lösen kann. Er ist also ein Algorithmus.



### Namen

Jetzt kommen wir zu einer speziellen Programmieretechnik. Der Algorithmus zum Streichholzrätsel wird einfacher, wenn man an die Streichhölzer im Bild Buchstaben schreibt. Damit gibt man den Streichhölzern *Namen*, die im Algorithmus verwendet werden können.

Hier ist ein Algorithmus mit Namen für die wichtigsten Streichhölzer:

Nimm das Streichholz b und lege es waagrecht links neben den Kopf von Streichholz a.

Dieser Algorithmus ist wesentlich kürzer. Er hat nur zwei Zeilen statt vier Zeilen. Ist er nicht auch viel besser zu verstehen?



# ? WAS IST EIN ALGORITHMUS?

Ein Algorithmus ist eine Anleitung, mit der sich eine Aufgabe lösen lässt und die so genau und unmissverständlich formuliert ist, dass sie im Prinzip auch von einer Maschine ausgeführt werden kann. Beispiele sind Kochrezepte, Wegbeschreibungen oder Bauanleitungen. Auch die Lösung zu einem Streichholzrätsel ist ein Algorithmus.

## Eindeutige Anweisungen

In einem Algorithmus muss jede Anweisung eindeutig sein. Ein Computer, der einem Programm folgt, ist ein ziemlich dummer Befehlsempfänger und macht nur genau das, was man ihm sagt. Wir Menschen sind da etwas anders. Wenn wir im Alltag jemanden bitten, etwas zu tun, erwarten wir, dass die Person mitdenkt. Sonst kann es leicht zu Missverständnissen kommen.

In diesem Comic entsteht ein Missverständnis, weil der Grillmeister nicht genau erklärt hatte, was er unter »aufpassen« versteht.



## Sequenz

Piraten haben auf der Insel einen Schatz versteckt. Die Schatztruhe befindet sich unter einem der vielen Gebüsch. Aber unter welchem?

Der folgende Algorithmus beschreibt, wie du den Schatz findest:

Lege mit deinem Boot am Steg ganz im Süden der Insel an und betritt die Insel. Gehe geradeaus über den Weg Richtung Norden. Biege nach der vierten Palme nach rechts ab. Der Schatz ist unter dem zweiten Busch auf der rechten Seite des Wegs.

Weißt du, welcher Busch gemeint ist? Wenn du unsicher bist, kannst du am Ende des Kapitels nachsehen. Du bist vielleicht überrascht, wenn ich dir sage, dass in dieser Wegbeschreibung eine bestimmte Programmieretechnik verwendet worden ist.

Die Palmen an dem Weg Richtung Norden wurden durchnummeriert. Zuerst kommt Palme Nummer 1, dann Palme Nummer 2, dann Palme Nummer 3 und dann Palme Nummer 4, hinter der der Schatzsucher abbiegen muss. Dinge, die Nummern haben und in einer bestimmten Reihenfolge sind, nennt man *Sequenz*. In der Programmierung verwendet man häufig Sequenzen. Python enthält mehrere Arten von Sequenzen, zum Beispiel Listen und Zeichenketten.



## Bedingung

Es geschah mittags in der Kölner Innenstadt. Zwei maskierte Männer überfielen die Sparkasse und flüchteten mit der Beute in einem Auto. Zum Glück gab es einige Zeugenaussagen zum Fluchtfahrzeug:

- »Das Auto der Bankräuber ist blau.«
- »Ich habe das Kennzeichen gesehen. Der Wagen kommt nicht aus Köln.«
- »Das Autokennzeichen enthält eine 7.«

Überwachungskameras in der Umgebung der Sparkasse haben folgende Autos fotografiert. Welches könnte das Fluchtfahrzeug der Bankräuber sein?



Du findest die Lösung des Rätsels am Ende des Kapitels. Überlegen wir kurz, wie man die Bilder der Überwachungskameras prüfen kann.

Jede einzelne Zeugenaussage beschreibt eine *Bedingung*, die das gesuchte Auto erfüllen muss. Eine Bedingung ist eine Aussage, die entweder wahr oder falsch ist.

Untersuchen wir das erste Bild:

- Die erste Zeugenaussage enthält die Bedingung »Das Auto ist blau.«. Diese Bedingung ist für das erste Bild erfüllt.
- Aber die dritte Zeugenaussage »Das Autokennzeichen enthält eine 7.« trifft auf das erste Bild nicht zu. Diese Bedingung ist nicht erfüllt, weil im Kennzeichen M AF 122 keine 7 vorkommt.
- Also kann das Auto im ersten Bild nicht das gesuchte Fluchtfahrzeug sein, denn es sind nicht alle drei Bedingungen erfüllt, die sich aus den Zeugenaussagen ergeben.

Das Formulieren und Prüfen von Bedingungen ist wichtig, wenn du *if*-Anweisungen programmierst. Bei einer *if*-Anweisung wird eine Aktion nur dann ausgeführt, wenn eine Bedingung erfüllt ist. Mehr dazu in Kapitel 3.

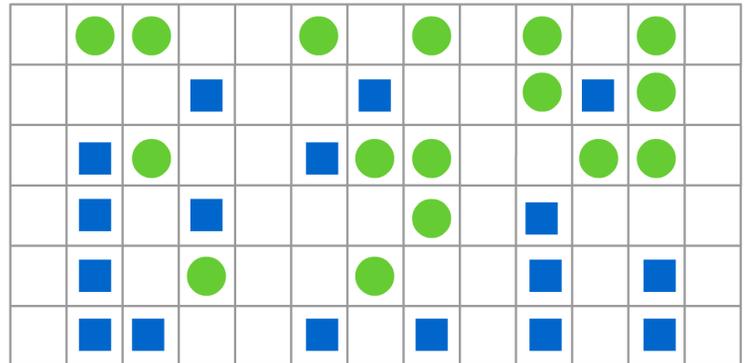


## Wiederholung

Auf dem Bild aus Punkten und Quadraten sind Buchstaben versteckt.

Mit dem folgenden Algorithmus kannst du die Buchstaben sichtbar machen:

Verschiebe jeden Punkt um 1 Feld nach unten.



Kannst du dir das Ergebnis vorstellen? Der Algorithmus ist leicht verständlich, oder? Die Besonderheit: Er enthält eine *Wiederholung*. Die Aktion »verschiebe um ein Feld nach unten« wird mehrfach ausgeführt. Genauer gesagt: Sie wird auf alle Punkte angewendet.

In Python programmierst du Wiederholungen mit *for* und *while*. Alles Wichtige zu Wiederholungen findest du in den Kapiteln 3 und 5.