

Kapitel 1

Auf die Plätze ...

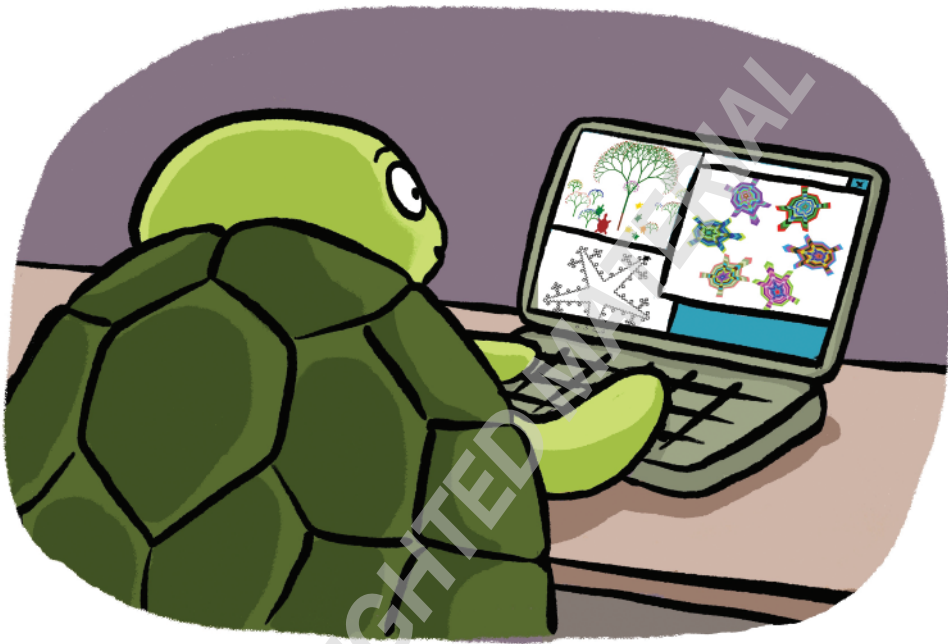


Illustration: Leonard Ermel

Computer können überhaupt nichts von alleine. Alles muss man ihnen sagen. Die Sprachen, in denen wir Computern Anweisungen erteilen, heißen *Programmiersprachen*. Sie unterscheiden sich ziemlich stark von den Sprachen, in denen Menschen miteinander reden. Der erste auffällige Unterschied ist, dass Programmiersprachen nicht gesprochen, sondern geschrieben werden. Und zwar von *Programmierern*. Die schreiben auf, was der Computer in welcher Reihenfolge tun soll. Man nennt eine Reihe solcher Anweisungen ein *Programm* oder eine *App* oder einfach *Code* oder *Software*.

Computer und Programmiersprachen

Computer unterstützen uns beim Schreiben von Texten, beim Bildermalen, beim Komponieren von Musik und beim Suchen und Einkaufen im Internet, sie können 3D-Filme zeigen, Roboter auf dem Mars steuern, . . . Bestimmt fallen dir noch weitere Dinge ein, die wir mit Computern tun.

Da ist es ganz natürlich, dass mehrere Programmiersprachen praktischer sind als eine einzige für alles. Die wichtigsten Programmiersprachen, die heute verwendet werden, heißen Java, C, C++, C#, Python, PHP und JavaScript. Es gibt aber noch viel, viel mehr! Programmierer streiten sich manchmal, welche Sprache für welche Art von Aufgaben am besten geeignet ist. Für uns ist wichtig: Sooo unterschiedlich sind die Programmiersprachen erst einmal gar nicht. Auf jeden Fall ist es leichter, eine Programmiersprache zu lernen als eine echte Fremdsprache. Und wenn du erst eine Programmiersprache kennengelernt hast, dann lernst du die nächste gleich noch einmal so schnell.

Damit ein Computer die Anweisungen in einem Programm befolgen kann, braucht er ein paar Geräte (man nennt sie die *Hardware*). Also auf jeden Fall einen Bildschirm, der Bilder und Texte anzeigt, und einen Lautsprecher für die Töne. Bildschirm und Lautsprecher nennt man **Ausgabegeräte**.



Illustration: elzarkho – stock.adobe.com

Manchmal braucht der Computer auch noch Informationen von uns, die nicht von vornherein im Programm stehen: Welchen Text willst du ausdrucken, welche Zahlen sollen addiert werden, wie groß soll der Kreis sein, den du zeichnen möchtest? Solche Informationen teilen wir dem Computer dann über die Tastatur oder die Maus oder den Touchscreen mit. Das sind dann – logisch – **Eingabegeräte**.



Damit Programmierer auf der ganzen Welt die Anweisungen in Programmiersprachen verstehen und benutzen können, sind die wichtigsten Anweisungen immer auf Englisch. Möchtest du also den Text »HALLO WELT!« auf dem Bildschirm anzeigen lassen, dann benutzt du in vielen Programmiersprachen das Wort `print` (auf Deutsch »drucke«) und schreibst zum Beispiel:

`print("HALLO WELT!")` in Python,

`printf("HALLO WELT!\n");` in C oder

`System.out.println("HALLO WELT!");` in Java.

Schön, jetzt verstehen also wir Programmierer die Anweisungen und können ein Programm schreiben. Aber der Computer soll unsere Anweisungen ja auch verstehen, und der kann doch kein Englisch, oder?

Um zu verstehen, wie der Computer Anweisungen verarbeitet, schauen wir uns mal an, wie er aufgebaut ist. Von Ein- und Ausgabegeräten hast du schon gehört, ein wichtiges Stück Hardware fehlt aber noch: die *Zentraleinheit* (auf Englisch *CPU*, für *Central Processing Unit*). Sie ist das Herz des Computers und ist mit den Ein- und Ausgabegeräten über Stromleitungen verbunden.

Die CPU ist ein elektronischer Chip und enthält enorm viele kleine Schalter (sogenannte *Transistoren*), die entweder geschlossen sein können (dann fließt Strom) oder offen (dann kann kein Strom fließen). Unser Programm steuert nun, welche von den Schaltern geöffnet und welche geschlossen werden. Dafür reicht der CPU eine lange Liste, die für jeden Schalter sagt, ob er `offen` oder `geschlossen` sein soll. Ein Programm, wie die CPU es versteht, also in *Maschinensprache*, kannst du dir so vorstellen (die Nummern sind die einzelnen Schalter):

1. offen
2. geschlossen
3. offen
4. offen
5. geschlossen
6. offen
7. offen
8. offen

Weil das ein bisschen lang wird, hat man sich darauf geeinigt, stattdessen die beiden Zeichen 0 (Null) für *offen* und 1 (Eins) für *geschlossen* zu verwenden, und die Nullen und Einsen in Schalterreihenfolge als Kette anzuordnen. Das Programm oben kann dann viel kürzer geschrieben werden, nämlich als Kette 0100 1000. Der Effekt der Kette 0100 1000 ist, dass der Buchstabe »H« auf unserem Bildschirm erscheint. Für den Rest unseres Textes »HALLO WELT!« braucht die CPU noch viele weitere Nullen und Einsen.

Von Bits und Bytes

Eine Kette aus Nullen und Einsen, also eine Zahl, die nur aus zwei unterschiedlichen Ziffern aufgebaut ist, nennt man *binäre Zahl* (*bini* ist das lateinische Wort für *je zwei*). Die einzelnen Stellen der Zahl heißen *Bit*. Eine binäre Zahl aus acht Stellen, also aus acht Bits, wird *Byte* genannt.



Jetzt bleibt nur noch die Frage, wie unser Programm mit seinen englischen Anweisungen in Maschinensprache übersetzt wird, also in Ketten von Nullen und Einsen.

Bei der Übersetzung muss sehr sorgfältig vorgegangen werden, denn eine einzige falsch übersetzte Eins oder Null kann das ganze schöne Programm kaputt machen. Wenn bei einer Aufgabe sorgfältig vorgegangen werden muss und etwas genau nach Vorschrift getan werden soll, bietet es sich an, einen Computer zu verwenden und für die Aufgabe ein Programm zu schreiben.

Es gibt daher besondere Übersetzerprogramme (genannt *Compiler* oder *Interpreter*), die das für uns lesbare Programm (den *Quellcode*) in Maschinensprache übersetzen (den *Binärcode*) und das übersetzte Programm dann ausführen. Diese Übersetzerprogramme brauchen wir nicht selbst zu schreiben, es gibt sie für jede Programmiersprache. Du wirst im nächsten Abschnitt erfahren, wo du den Interpreter für Python im Internet findest und herunterladen kannst.



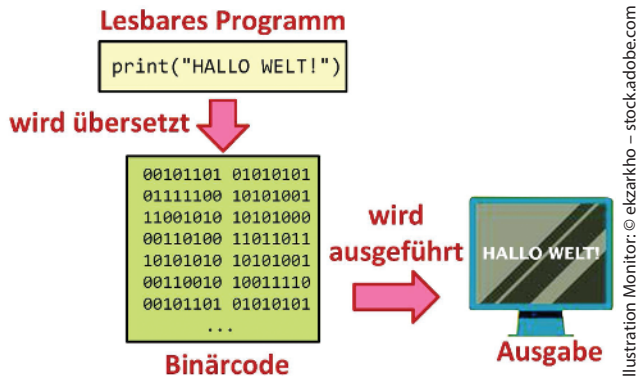
Compiler und Interpreter

Die beiden Übersetzervarianten Compiler und Interpreter unterscheiden sich in der Reihenfolge von Übersetzen und Ausführen.

Ein *Compiler* übersetzt erst das gesamte Programm in *Binärcode* und speichert es als Datei auf dem Computer. Danach, wenn der Compiler fertig übersetzt hat, kann das Programm ausgeführt werden. Ein fertig übersetztes Programm kann immer wieder aufgerufen werden und muss nicht jedes Mal neu übersetzt werden. Die Programmiersprachen C und C++ verwenden Compiler.

Im Gegensatz dazu übersetzt der *Interpreter* einzelne Anweisungen nicht bevor, sondern während das Programm läuft. Er liest also eine Anweisung aus dem Quellcode, zum Beispiel `print("HALLO WELT!")`, übersetzt sie in Binärcode und sorgt dafür, dass diese Anweisung sofort auf dem Computer ausgeführt wird. Danach liest er die nächste Anweisung, zum Beispiel `print("TSCHÜSS WELT!")`, übersetzt sie und führt sie aus. Und so weiter. Der Interpreter erzeugt keine Datei mit Binärcode. Jedes Mal, wenn das Programm laufen soll, muss er wieder loslegen und das Programm neu übersetzen und ausführen. Python ist ein Beispiel für eine Programmiersprache mit Interpreter.

Wenn du also dein Programm ausführst (auf dem Computer laufen lässt), dann startest du in Wirklichkeit erst das Übersetzerprogramm und erzeugst Binärcode aus deinem Quellcode. Mit dem Binärcode werden dann die Schalter geschaltet, und du siehst das Ergebnis auf deinem Ausgabegerät.



Python auf deinem Computer

Damit du nun an deinem Computer mit dem Programmieren in Python loslegen kannst, brauchst du zwei Dinge: einen *Python-Interpreter*, der deinen Code in Binärcode übersetzen und ausführen kann, und einen *Editor*. Das ist ein Programm zum Schreiben von Programmen. Sowohl der Interpreter als auch der Editor können von der Python-Homepage aus dem Internet heruntergeladen werden, und das sogar kostenlos. Das wollen wir jetzt tun.



Warum heißt die Programmiersprache Python?

Programmiersprachen haben manchmal seltsame Namen. Meistens entscheiden ihre Erfinder, wie sie heißen sollen. Der niederländische Erfinder von Python, Guido van Rossum, ist ein Fan der englischen Komikerguppe *Monty Python* aus den 1970er-Jahren, die mehrere Kinofilme drehte und eine Fernsehserie hatte (*Monty Python's Flying Circus*).

Van Rossum hat Python entwickelt, weil er eine möglichst einfache und übersichtliche Programmiersprache haben wollte. Das ist ihm gelungen, Python war im Jahr 2022 vor C und Java die beliebteste Programmiersprache!