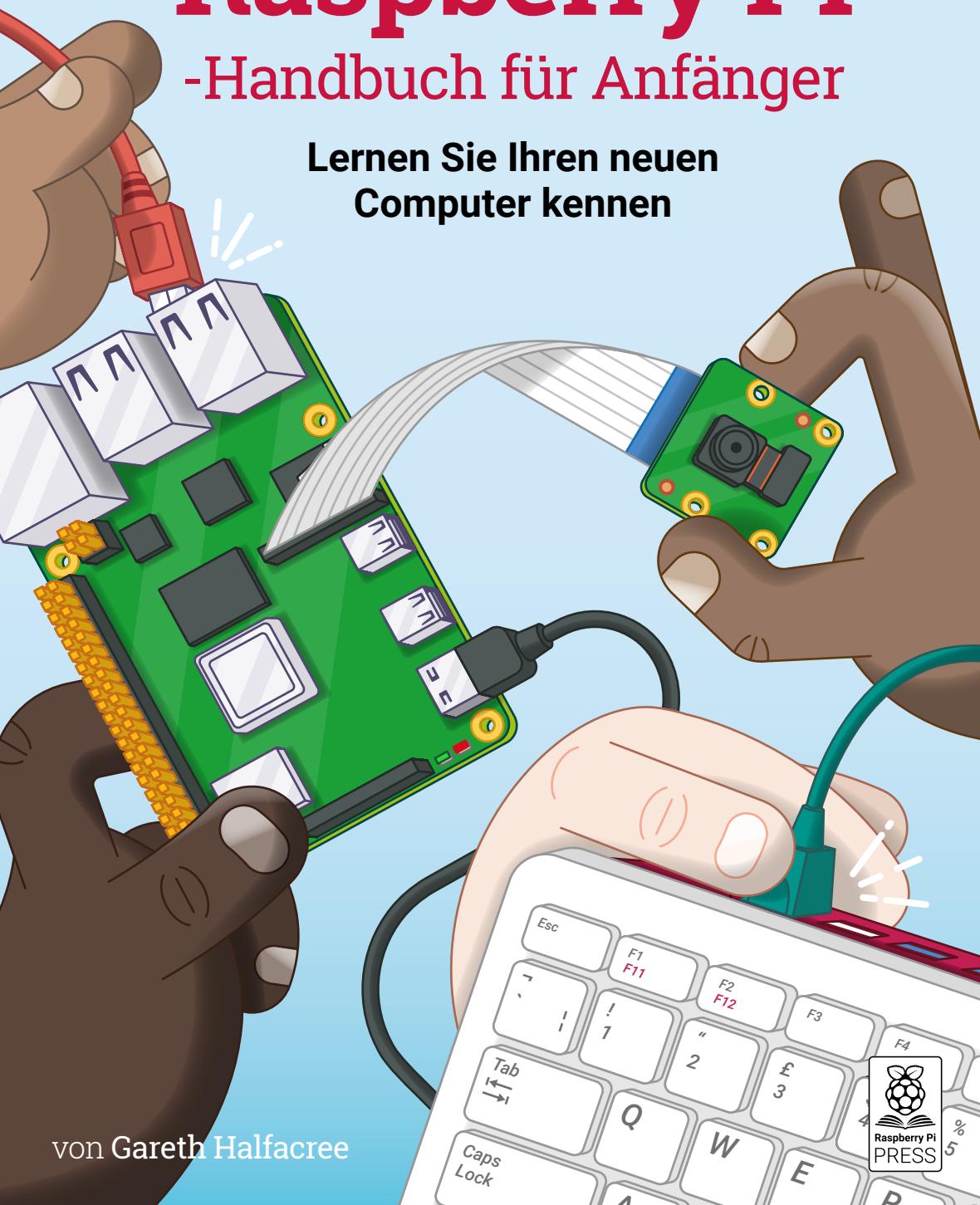


4.
Ausgabe

VOLLSTÄNDIG AKTUALISIERT FÜR RASPBERRY PI 400

DAS OFFIZIELLE Raspberry Pi -Handbuch für Anfänger

Lernen Sie Ihren neuen
Computer kennen



von Gareth Halfacree



DAS OFFIZIELLE
Raspberry Pi
Handbuch für Anfänger
Lernen Sie Ihren neuen Computer kennen



Erstmals veröffentlicht im Jahr 2020 von Raspberry Pi Trading Ltd, Maurice Wilkes Building, St. John's Innovation Park, Cowley Road, Cambridge, CB4 0DS

Verlagsleitung: Russell Barnes • Chefredakteur: Phil King

Design: Critical Media • Illustrationen: Sam Alder

Geschäftsführer: Eben Upton

ISBN: 978-1-912047-95-6

Der Herausgeber und die Mitwirkenden übernehmen keine Verantwortung in Bezug auf Auslassungen oder Fehler in Verbindung mit Waren, Produkten oder Dienstleistungen, die in diesem Buch erwähnt oder beworben werden. Sofern nicht anders angegeben, ist der Inhalt dieses Buches unter der Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported Lizenz lizenziert.

(CC BY-NC-SA 3.0)

Willkommen im offiziellen Raspberry Pi-Handbuch für Anfänger

Wir glauben, dass Sie den Raspberry Pi lieben werden. Dieser ultrakleine, erschwingliche Computer kostet weniger als die meisten Videospiele, kann aber zum Programmierenlernen, zum Bau von Robotern und für jede Menge ungewöhnlicher und faszinierender Projekte verwendet werden.

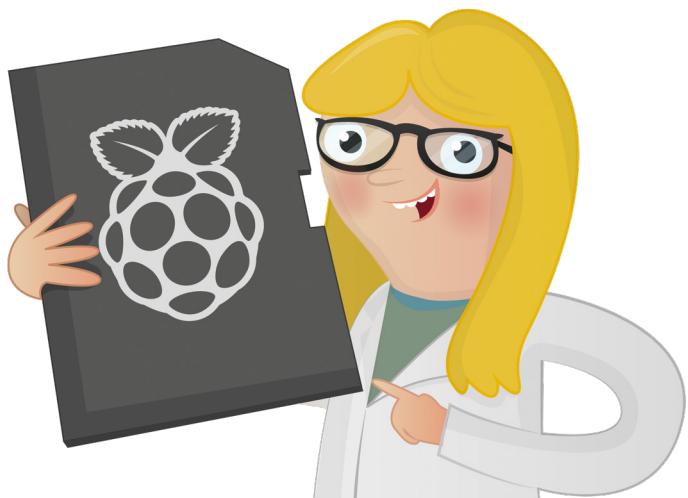
Der Raspberry Pi ist in der Lage, all die Dinge zu tun, die Sie von einem Computer erwarten – alles, vom Surfen im Internet und Spielen bis hin zum Anschauen von Filmen und Hören von Musik. Aber der Raspberry Pi ist viel mehr als nur ein moderner Computer.

Mit einem Raspberry Pi können Sie in das Herz des Computers vordringen. Sie können Ihr eigenes Betriebssystem einrichten und Kabel und Schaltkreise direkt an die Stifte auf seiner Platine anschließen. Er wurde eigens dafür entwickelt, um jungen Leuten zu zeigen, wie man in Sprachen wie Scratch und Python programmiert. Alle wichtigen Programmiersprachen sind im offiziellen Betriebssystem enthalten.

Die Welt braucht Programmierer mehr denn je, und der Raspberry Pi entfacht die Liebe und Leidenschaft für Informatik und Technologie in der jungen Generation.

Aber nicht nur Jugendliche, sondern Menschen jeden Alters nutzen den Raspberry Pi für spannende Projekte: von Retro-Spielkonsolen bis hin zu Wetterstationen mit Internetanschluss.

Wenn Sie also Spiele entwickeln, Roboter bauen oder interessante Projekte programmieren wollen, ist dieses Buch der beste Weg, um Ihnen den Einstieg zu erleichtern.



Über den Autor

Gareth Halfacree ist ein freiberuflicher Technologie-Journalist, Schriftsteller und ehemaliger Systemadministrator im Bildungssektor. Mit seiner Leidenschaft für Open-Source-Software und -Hardware war er ein früher Anhänger der Raspberry Pi-Plattform und hat mehrere Publikationen über ihre Fähigkeiten und die zahlreichen Anwendungsmöglichkeiten verfasst. Er twittert als [@ghalfacree](https://twitter.com/ghalfacree) und seine Website ist freelance.halfacree.co.uk.



Inhalt

Kapitel 1: Lernen Sie Ihren Raspberry Pi kennen	008
Eine Einführung in Ihren neuen Computer	
Kapitel 2: Erste Schritte mit Ihrem Raspberry Pi	022
Verbinden Sie alles, was Sie brauchen, um Ihren Raspberry Pi zum Laufen zu bringen	
Kapitel 3: Verwendung Ihres Raspberry Pi	036
Erfahren Sie alles über das Raspberry Pi-Betriebssystem	
Kapitel 4: Programmieren mit Scratch 3	054
Beginnen Sie mit dieser leicht erlernbaren, blockbasierten Sprache zu programmieren	
Kapitel 5: Programmieren mit Python	092
Machen Sie sich mit textbasierter Programmierung mit der Sprache Python vertraut	
Kapitel 6: Physical Computing mit Scratch und Python	120
Steuern Sie elektronische Bauteile, die mit den GPIO-Stiften Ihres Raspberry Pi verbunden sind	
Kapitel 7: Physical Computing mit dem Sense HAT	152
Verwenden Sie die Sensoren und die LED-Matrixanzeige dieser Zusatzplatine	
Kapitel 8: Raspberry Pi Kamera	196
Nehmen Sie mit dieser winzigen Kamera hochauflösende Fotos und Videos auf	
ANHÄNGE	
Anhang A: Installieren eines Betriebssystems auf einer microSD-Karte	214
Anhang B: Installieren und Deinstallieren von Software	216
Anhang C: Die Kommandozeile	222
Anhang D: Raspberry Pi-Konfigurationswerkzeug	228
Anhang E: Weiterführende Literatur	234
Anhang F: Einrichtung der High Quality Camera	240
Anhang G: Raspberry Pi-Spezifikationen	244
Anhang H: Raspberry Pi 4 Sicherheits- und Benutzerhandbuch	247

Kapitel 1

Lernen Sie Ihren Raspberry Pi kennen

Machen Sie sich bei einem geführten Rundgang des Raspberry Pi mit Ihrem neuen kreditkartengroßen Computer vertraut. Entdecken Sie seine zahlreichen Komponenten und ihre Funktion



Der Raspberry Pi ist ein bemerkenswertes Gerät: ein voll funktionsfähiger Computer in einem winzigen und kostengünstigen Formfaktor. Ganz gleich, ob Sie nach einem Gerät suchen, mit dem Sie im Internet surfen oder Spiele spielen können, ob Sie lernen möchten, wie Sie Ihre eigenen Programme schreiben können, oder ob Sie Ihre eigenen Schaltkreise und Geräte entwickeln möchten, der Raspberry Pi – und seine erstaunliche Community – wird Sie bei jedem Schritt unterstützen.

Der Raspberry Pi ist bekannt als *Einplatinen-Computer*. Das bedeutet genau das, wonach es sich anhört: Es ist ein Computer, genau wie ein Desktop PC, Laptop oder Smartphone, aber auf einer einzigen Leiterplatte. Wie die meisten Einplatinen-Computer ist der Raspberry Pi klein – ungefähr so groß wie eine Kreditkarte – aber das bedeutet nicht, dass er nicht leistungsfähig ist: Ein Raspberry Pi kann alles, was ein größerer und stromhungriger Computer kann, wenn auch nicht unbedingt so schnell.

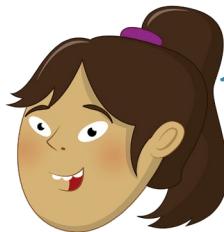
Die Raspberry Pi Familie entstand aus dem Wunsch heraus, mehr praktische Computerbildung auf der ganzen Welt zu fördern. Ihre Schöpfer, die sich zur gemeinnützigen Raspberry Pi Foundation zusammengeschlossen haben, ahnten kaum, dass sie sich als so beliebt erweisen würde: Die wenigen tausend Raspberry Pi, die 2012 gebaut wurden, um den Markt auszuloten, waren sofort vergriffen, und seitdem wurden Millionen auf der ganzen Welt ausgeliefert. Diese Platinen sind in Wohnungen, Klassenzimmer, Büros, Rechenzentren, Fabriken und sogar in selbstfahrende Boote und Raumfahrtballons anzutreffen.

Seit dem ursprünglichen Modell B sind verschiedene Modelle des Raspberry Pi erschienen, die jeweils entweder verbesserte technische Daten oder spezifische Funktionen für bestimmte Anwendungsfälle aufweisen. Die Raspberry Pi Zero-Familie, zum Beispiel, ist eine winzige Version des Raspberry Pi in Originalgröße, bei der einige wenige Merkmale – insbesondere die zahlreichen USB-Anschlüsse und der kabelgebundene Netzwerkanschluss – zugunsten eines

deutlich kleineren Layouts und eines geringeren Stromverbrauchs weggelassen wurden.

Eines haben jedoch alle Raspberry Pi-Modelle gemeinsam: Sie sind *kompatibel*. Das bedeutet, dass Software, die für ein Modell geschrieben wurde, auf jedem anderen Modell läuft. Es ist sogar möglich, die allerneueste Version des Betriebssystems des Raspberry Pi auf einem Original-Prototypen des Modells B von vor der Markteinführung laufen zu lassen. Er wird zwar in der Tat langsamer arbeiten, aber er wird funktionieren.

In diesem Buch erfahren Sie alles über Raspberry Pi 4 Modell B und Raspberry Pi 400, die neuesten und leistungsstärksten Versionen des Raspberry Pi. Was Sie lernen, lässt sich jedoch leicht auf andere Modelle der Raspberry Pi-Familie anwenden, also machen Sie sich keine Sorgen, wenn Sie eine andere Version verwenden.



RASPBERRY PI 400



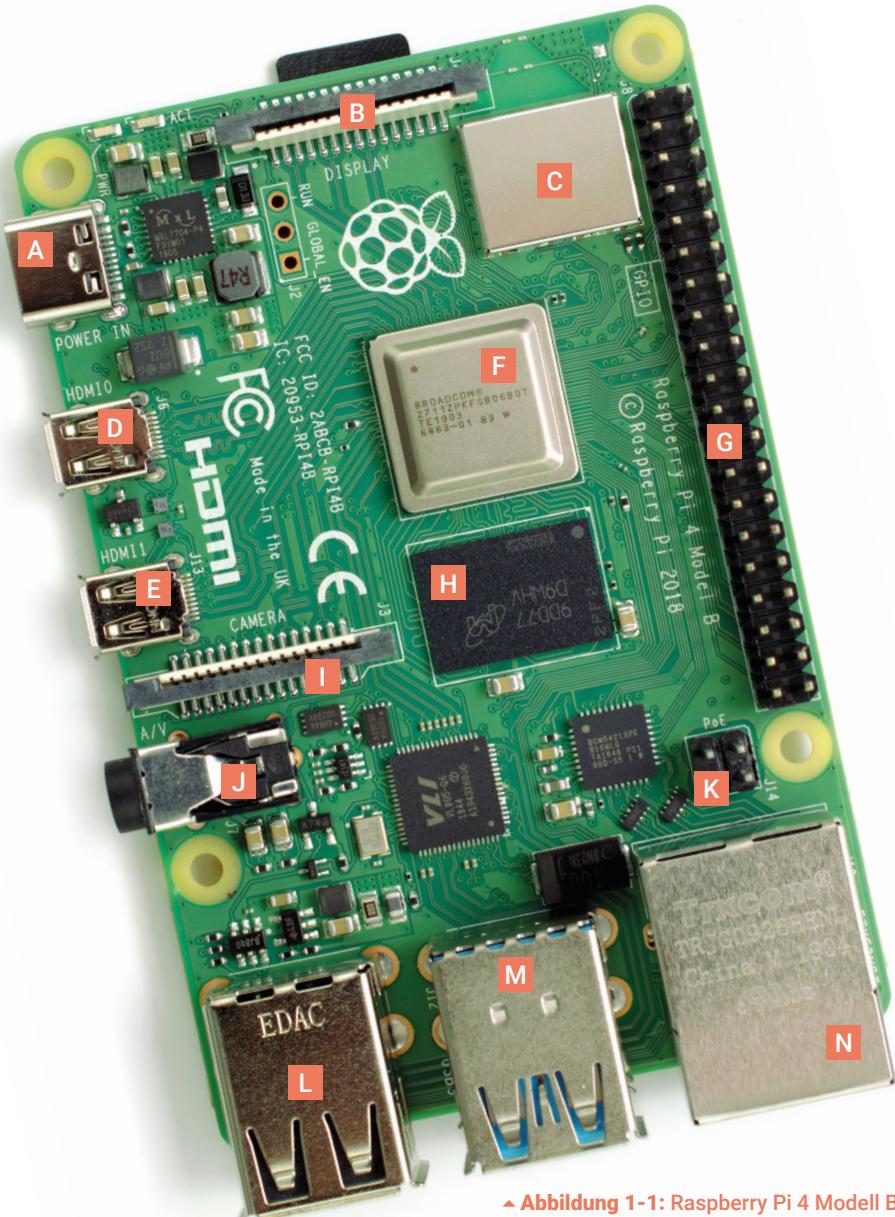
Wenn Sie einen Raspberry Pi 400 haben, ist die Platine in das Tastaturgehäuse eingebaut. Lesen Sie weiter, um mehr über alle Komponenten zu erfahren, die einen Raspberry Pi ausmachen, oder blättern Sie zur Seite 20 vor, um einen Überblick über das Gerät zu erhalten.

Eine Führung durch den Raspberry Pi

Im Gegensatz zu einem herkömmlichen Computer, der sein Innenleben in einem Gehäuse verbirgt, sind bei einem Raspberry Pi in der Standardausstattung alle Komponenten, Anschlüsse und Funktionen gut sichtbar. Sie können aber auch ein Gehäuse kaufen, das Ihnen zusätzlichen Schutz bietet. Das ist ein idealer Ausgangspunkt, um alles über die verschiedenen Teile eines Computers zu erfahren. Sie erfahren dann auch, wie die Komponenten zusammengesetzt werden müssen, wenn bestimmte Zusatzelemente – die sogenannten *Peripheriegeräte* – angeschlossen werden müssen.



A	USB Typ-C Stromversorgung	F	System-on-a-Chip	K	PoE
B	Display-Anschluss (DSI)	G	GPIO	L	USB 2.0
C	WLAN/Bluetooth	H	RAM	M	USB 3.0
D	Micro-HDMI 0	I	CSI-Kamera-Anschluss	N	Ethernet-Anschluss
E	Micro-HDMI 1	J	3,5mm AV-Buchse		



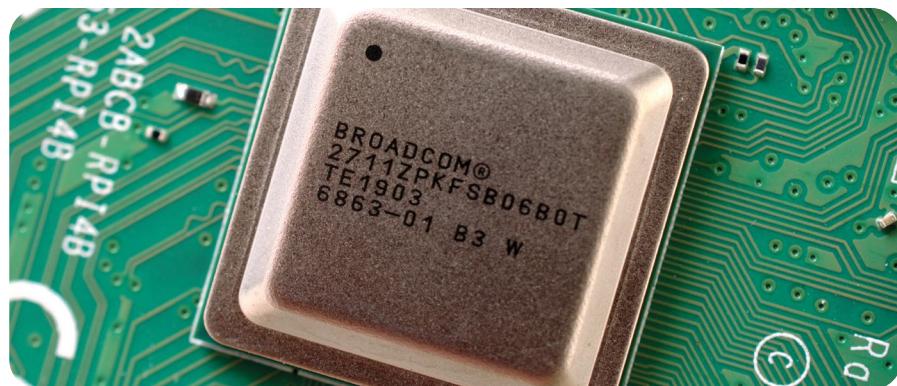
▲ Abbildung 1-1: Raspberry Pi 4 Modell B

Abbildung 1-1 zeigt einen Raspberry Pi 4 Modell B von oben gesehen. Wenn Sie bei der Lektüre dieser Anleitung einen Raspberry Pi verwenden, versuchen Sie, ihn in derselben Position zu halten wie auf dem Bild. Eine andere Ansicht könnte zu Verwirrungen führen, wenn es um Dinge wie die GPIO-Stiftleiste geht (ausführlich in **Kapitel 6, Physical Computing mit Scratch und Python** erklärt).

Auch wenn es so aussieht, als wäre viel in die winzige Platine gepackt worden, ist ein Raspberry Pi sehr einfach zu verstehen – angefangen bei seinen Komponenten, dem Innenleben, das das Gerät zum Laufen bringt.

Komponenten des Raspberry Pi

Wie jeder Computer setzt sich Raspberry Pi aus verschiedenen Komponenten zusammen, die jeweils eine bestimmte Funktion übernehmen. Die erste und wohl wichtigste davon befindet sich direkt oberhalb des Mittelpunktes auf der Oberseite der Platine (**Abbildung 1-2**), mit einer Metallkappe abgedeckt: der *System-on-a-Chip* (SoC).



▲ Abbildung 1-2: Der System-on-a-Chip (SoC) des Raspberry Pi

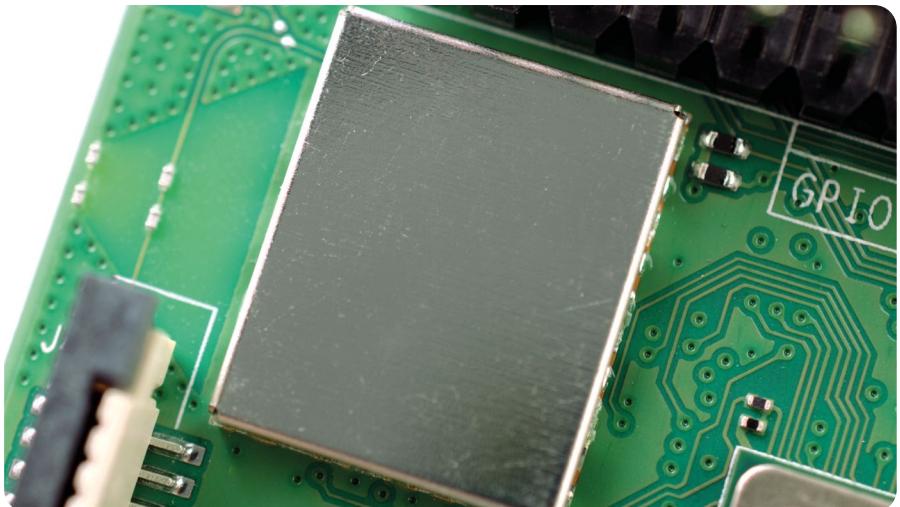
Der Name „System-on-a-Chip“ ist ein guter Hinweis darauf, was man vorfindet, wenn man die Metallabdeckung abnimmt: einen Siliziumchip, bekannt als *integrierter Schaltkreis*, der den Großteil des Systems des Raspberry Pi enthält. Dazu gehören die *Central Processing Unit* (CPU oder „Zentraleinheit“), die gemeinhin als das „Gehirn“ eines Computers angesehen wird, und der *Grafikprozessor* (GPU), der für die Grafikdarstellung sorgt.

Ein Gehirn taugt jedoch nichts ohne Gedächtnis, und neben dem SoC finden Sie genau das: einen weiteren Chip, der wie ein kleines, schwarzes Plastikquadrat aussieht (**Abbildung 1-3** umseitig). Das ist der *Arbeitsspeicher* (RAM) des Raspberry Pi. Wenn Sie mit dem Raspberry Pi arbeiten, ist es der RAM, in dem das, was Sie tun, vorgehalten wird; erst wenn Sie Ihre Arbeit speichern, wird sie auf die microSD-Karte geschrieben. Zusammen bilden diese Komponenten die flüchtigen und nichtflüchtigen Speicher des Raspberry Pi: Der flüchtige RAM verliert seinen Inhalt, wenn der Raspberry Pi ausgeschaltet wird, während die nichtflüchtige microSD-Karte ihren Inhalt behält.



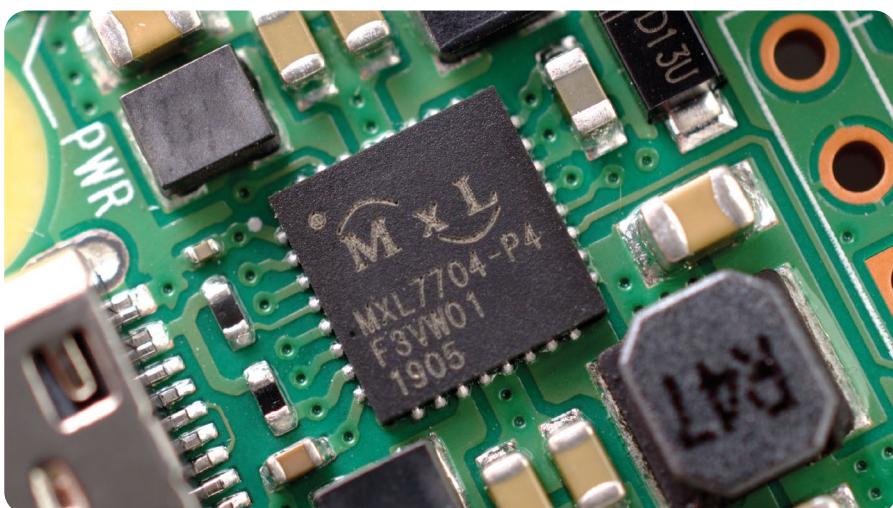
▲ Abbildung 1-3: Das ist der Arbeitsspeicher (RAM) des Raspberry Pi

Oben rechts auf der Platine finden Sie eine weitere Metallkappe (**Abbildung 1-4**), die die **Funkeinheit** abdeckt, die Komponente, mit der der Raspberry Pi drahtlos mit anderen Geräten kommunizieren kann. Die Funkeinheit selbst besteht aus zwei Hauptkomponenten: einem **WLAN-Teil** zur Verbindung mit Computernetzwerken; und einem **Bluetooth-Teil** zur Verbindung mit Peripheriegeräten wie Mäusen und zum Senden oder Empfangen von Daten an oder von nahegelegenen intelligenten Geräten wie Sensoren oder Smartphones.



▲ Abbildung 1-4: Das Funkmodul des Raspberry Pi

Ein weiterer schwarzer, kunststoffbeschichteter Chip ist am unteren Rand der Platine zu sehen, direkt hinter der mittleren Gruppe der USB-Anschlüsse. Dies ist der *USB-Controller*, der für den Betrieb der vier USB-Anschlüsse zuständig ist. Daneben befindet sich ein noch kleinerer Chip, der *Netzwerk-Controller*, der den Ethernet-Netzwerkanchluss des Raspberry Pi steuert. Ein letzter schwarzer Chip, kleiner als die anderen, befindet sich etwas oberhalb des USB-Typ C Stromversorgungsanschlusses oben links auf der Platine (**Abbildung 1-5**). Er ist bekannt als *power management integrated circuit (PMIC)* und wandelt den Strom, der vom Micro-USB-Anschluss kommt, in den Strom um, den der Raspberry Pi zum Betrieb benötigt.



▲ Abbildung 1-5: Der Power management integrated circuit (PMIC) des Raspberry Pi

Machen Sie sich keine Sorgen, wenn Sie gerade überwältigt sind: Sie brauchen nicht zu wissen, was jede Komponente ist oder wo sie auf der Platine zu finden ist, um den Raspberry Pi zu verwenden.

Anschlüsse des Raspberry Pi

Der Raspberry Pi hat eine Reihe von Anschlüssen, beginnend mit vier *Universal Serial Bus (USB)-Anschlüssen* (**Abbildung 1-6**) in der Mitte und an der rechten Seite der Unterkante.

Über diese Anschlüsse können Sie jedes USB-kompatible Peripheriegerät, von Tastaturen und Mäusen bis hin zu Digitalkameras und USB-Sticks, an den Raspberry Pi anschließen. Technisch gesehen gibt es zwei Arten von USB-Anschlüssen: Diejenigen mit schwarzen Teilen im Inneren sind USB 2.0-Anschlüsse, basierend auf Version 2 des Universal Serial Bus-Standards. Die mit blauen Teilen sind schnellere USB 3.0-Anschlüsse, basierend auf der neueren Version 3.



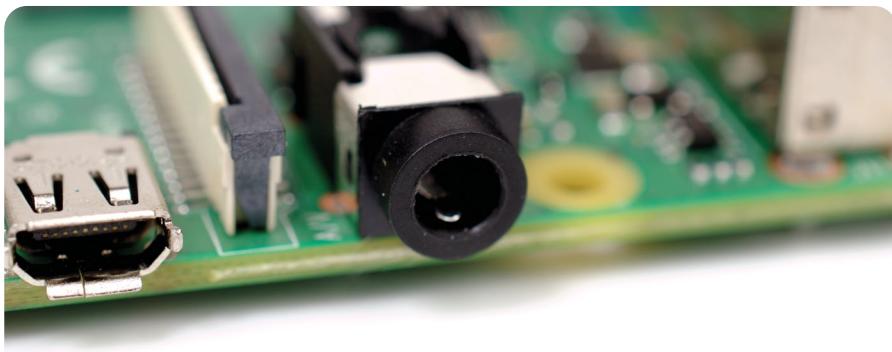
▲ Abbildung 1-6: USB-Anschlüsse des Raspberry Pi

Rechts neben den USB-Anschlüssen befindet sich ein **Ethernet-Anschluss**, auch bekannt als **Netzwerkanschluss** (**Abbildung 1-7**). Sie können diesen Anschluss verwenden, um den Raspberry Pi über ein Kabel mit einem sogenannten RJ45-Stecker an seinem Ende an ein drahtgebundenes Computernetzwerk anzuschließen. Wenn Sie sich den Ethernet-Anschluss genau anschauen, sehen Sie unten zwei Leuchtdioden (LEDs). Dies sind Status-LEDs, die anzeigen, dass die Verbindung funktioniert.



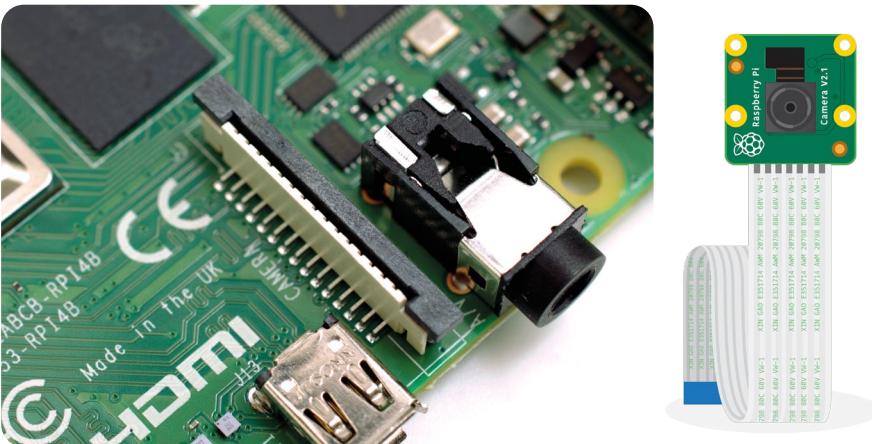
▲ Abbildung 1-7: Ethernet-Anschluss des Raspberry Pi

Knapp oberhalb des Ethernet-Anschlusses, am linken Rand des Raspberry Pi, befindet sich eine **3,5 mm-Audio-Video-Buchse (AV)** (**Abbildung 1-8**). Sie ist auch als **Kopfhörerbuchse** bekannt. Sie kann genau für diesen Zweck verwendet werden – allerdings erhalten Sie einen besseren Klang, wenn Sie sie an eine Stereoanlage statt an Kopfhörer anschließen. Sie verfügt jedoch noch über eine versteckte, zusätzliche Funktion: Neben dem Ton führt die 3,5-mm-AV-Buchse ein Videosignal, das an Fernseher, Beamer und andere Bildschirme angeschlossen werden kann, die ein **Composite Video Signal (FBAS)** unter Verwendung eines speziellen Kabels, bekannt als **tip-ring-ring-sleeve (TRRS)**-Adapter, unterstützen.



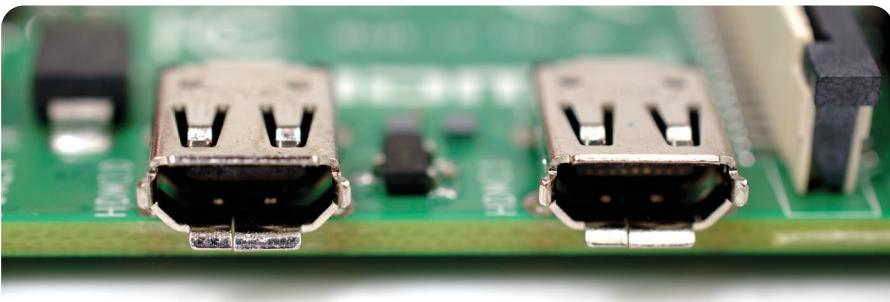
▲ Abbildung 1-8: 3,5-mm-AV-Buchse des Raspberry Pi

Direkt über der 3,5-mm-AV-Buchse befindet sich ein Stecker mit einer Kunststoffklappe, die nach oben gezogen werden kann. Dies ist der *Kamera-Anschluss*, auch bekannt als *Camera Serial Interface(CSI)* (**Abbildung 1-9**). Dies ermöglicht Ihnen die Verwendung des speziell entwickelten Raspberry Pi Camera Modules (Kamera-Moduls) – über das Sie in **Kapitel 8, Raspberry Pi Kamera** – mehr erfahren.



► Abbildung 1-9: Kamera-Anschluss des Raspberry Pi

Darüber, auch am linken Rand der Platine, befinden sich die *micro High Definition Multimedia Interface (micro-HDMI)* Anschlüsse. Diese sind eine kleinere Version der Anschlüsse, die sich auch auf einer Spielekonsole, einer Set-Top-Box oder einem Fernseher befinden (**Abbildung 1-10**). Wie der Namensbestandteil „Multimedia“ schon sagt, überträgt die Schnittstelle sowohl Audio- als auch Videosignale, während der Teil „High-Definition“ auf die ausgezeichnete Qualität hinweisen. Die Schnittstelle verwenden Sie, um den Raspberry Pi an ein oder zwei Anzeigegeräte anzuschließen: einen Computermonitor, einen Fernseher oder einen Beamer.



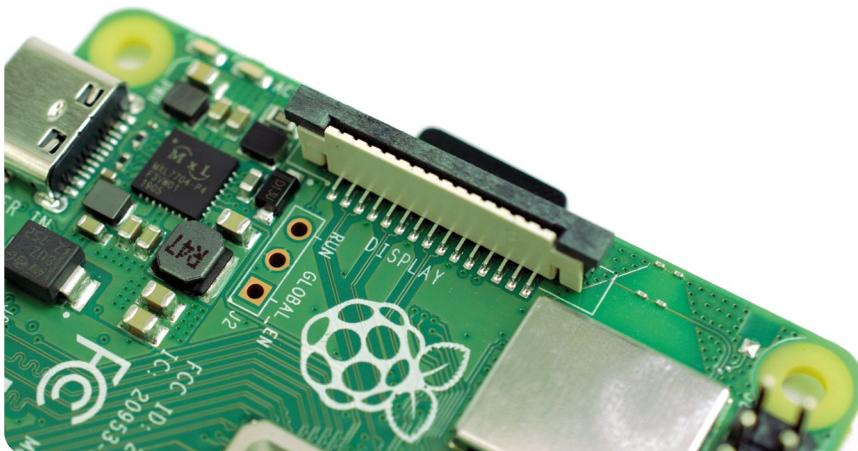
► Abbildung 1-10: Micro-HDMI-Anschlüsse des Raspberry Pi

Oberhalb der HDMI-Anschlüsse befindet sich ein *USB-C Anschluss* (**Abbildung 1-11**), mit dem Sie den Raspberry Pi an eine Stromquelle anschließen können. Der USB-C Anschluss wird häufig bei Smartphones, Tablets und anderen tragbaren Geräten verwendet. Für die Stromversorgung des Raspberry Pi können Sie zwar ein handelsübliches mobiles Ladegerät verwenden, für beste Ergebnisse empfehlen wir jedoch das offizielle Raspberry Pi USB-C Netzteil.

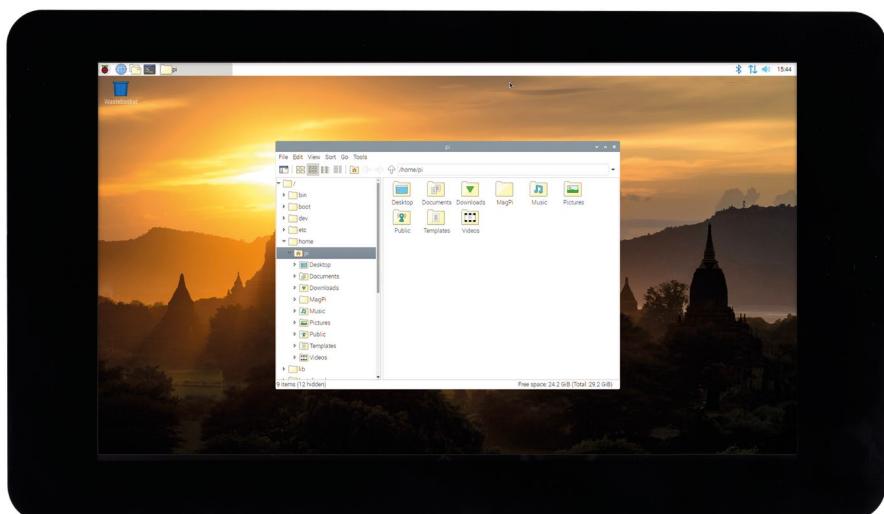


▲ Abbildung 1-11: USB-C Stromanschluss des Raspberry Pi

An der Oberkante der Platine befindet sich eine weitere Steckverbindung (**Abbildung 1-12**), die auf den ersten Blick identisch mit dem Kameraanschluss zu sein scheint. Tatsächlich ist sie jedoch das Gegenteil: ein *Display-Anschluss* oder *Display Serial Interface (DSI)*, entworfen für die Verwendung mit einem Raspberry Pi Touch Display (**Abbildung 1-13** (siehe umseitig)).

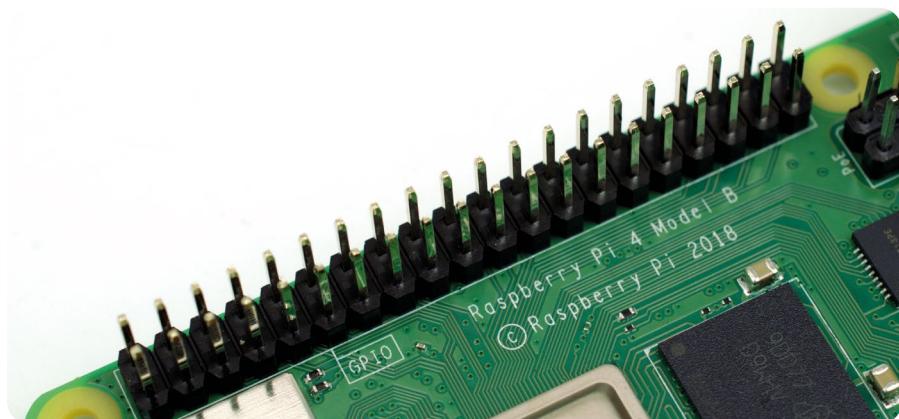


▲ Abbildung 1-12: Display-Anschluss (DSI) des Raspberry Pi



▲ Abbildung 1-13: Das Raspberry Pi-Touch-Display

Am rechten Rand der Platine finden Sie 40 Metallstifte, die in zwei Reihen zu je 20 Stiften aufgeteilt sind (**Abbildung 1-14**). Dies ist die GPIO (General-Purpose Input/Output)-Stiftleiste, ein Feature des Raspberry Pi, das verwendet wird, um mit zusätzlicher Hardware von LEDs und Tastern bis hin zu Temperatursensoren, Joysticks und Pulsfrequenzmessgeräten zu kommunizieren. Mehr über die GPIO-Stiftleiste erfahren Sie in **Kapitel 6, Physical Computing mit Scratch und Python**. Direkt unter und links von dieser Stiftleiste befindet sich eine weitere, kleinere Stiftleiste mit vier Stiften: Sie dient zum Anschluss des Power-over-Ethernet (PoE)-HAT, eines optionalen Zusatzes, mit dem der Raspberry Pi seinen Strom über die Netzwerkverbindung statt über den USB-C Anschluss beziehen kann.



▲ Abbildung 1-14: GPIO-Stiftleiste des Raspberry Pi

Es gibt noch einen letzten Anschluss am Raspberry Pi, aber den können Sie von oben nicht sehen. Drehen Sie die Platine um und Sie finden einen *microSD-Karten-Anschluss* auf der dem Display-Anschluss gegenüberliegenden Unterseite der Platine (**Abbildung 1-15**). Dies ist der Speicher des Raspberry Pi: Die hier eingesetzte microSD-Karte enthält alle Dateien, die Sie speichern, alle Software, die Sie installieren, und das Betriebssystem, das den Raspberry Pi zum Laufen bringt.



▲ **Abbildung 1-15:** microSD-Karten-Anschluss des Raspberry Pi



▲ Abbildung 1-16: Der Raspberry Pi 400 besitzt eine integrierte Tastatur

Der Raspberry Pi 400

Der Raspberry Pi 400 hat die gleichen Komponenten wie ein Raspberry Pi 4 allerdings in einem Tastaturgehäuse verbaut. Das Tastaturgehäuse schützt nicht nur das Gerät, sondern reduziert auch den Platzbedarf und den Kabelaufwand.

Der Raspberry Pi 400 besteht aus denselben Kernkomponenten wie der Raspberry Pi 4, einschließlich des System-on-a-Chip und des Speichers. Zwar sieht man sie nicht, aber sie sind vorhanden. Was Sie sehen können, sind die externen Teile, beginnend mit der Tastatur (**Abbildung 1-16**). In der rechten Ecke befinden sich drei Leuchtdioden (LEDs): Die erste leuchtet auf, wenn Sie die Taste Num Lock drücken, die bewirkt, dass gewisse Tasten wie der Nummernblock auf einer Tastatur in Standardgröße funktionieren. Die zweite leuchtet auf, wenn Sie die Feststelltaste drücken, wodurch die Buchstabentasten Großbuchstaben statt Kleinbuchstaben erzeugen. Die letzte leuchtet auf, wenn der Raspberry Pi 400 eingeschaltet ist.

Die Anschlüsse befinden sich auf der Rückseite des Raspberry Pi 400 (**Abbildung 1-17**). Der Anschluss ganz links von der Rückseite aus gesehen ist die General-Purpose Input/Output (GPIO)-Stifitleiste. Dies ist dieselbe Stifitleiste, die auf Seite 17 beschrieben ist, jedoch umgedreht: der erste Stift, Stift 1, befindet sich oben rechts, während der letzte Stift, 40, unten links liegt. Mehr über die GPIO-Stifitleiste erfahren Sie in **Kapitel 6, Physical Computing mit Scratch und Python**.



▲ Abbildung 1-17: Die Anschlüsse befinden sich an der Rückseite des Raspberry Pi 400

Neben der GPIO-Stifteiste befindet sich der microSD-Kartensteckplatz. Hier wird die microSD-Karte eingesteckt, die als Speicher des Raspberry Pi 400 für das Betriebssystem, die Anwendungen und andere Daten dient. Die microSD-Karte ist im Raspberry Pi 400 bereits vorinstalliert. Sie können sie entfernen, indem Sie leicht auf die Karte drücken, bis sie einrastet und herauspringt, und sie dann herausziehen. Wenn Sie die Karte wieder einstecken, achten Sie darauf, dass die glänzenden Metallkontakte nach unten zeigen. Die Karte sollte mit einem leichten Klick einrasten.

Die nächsten beiden Anschlüsse sind die Micro-HDMI-Anschlüsse für den Anschluss an einen Monitor, Fernseher oder ein anderes Display. Wie der Raspberry Pi 4 unterstützt auch der Raspberry Pi 400 bis zu zwei Displays. Daneben befindet sich der USB-C Stromanschluss für den Anschluss an das Raspberry Pi-Netzteil oder ein kompatibles USB-Netzteil.

Die beiden blauen Anschlüsse sind USB 3.0-Anschlüsse, die eine Hochgeschwindigkeitsverbindung zu Geräten wie Solid-State-Laufwerken (SSDs), USB-Speichersticks, Druckern und mehr ermöglichen. Der weiße Anschluss rechts davon ist ein USB 2.0-Anschluss mit niedrigerer Geschwindigkeit, an den Sie die mitgelieferte Raspberry Pi-Maus anschließen können.

Der letzte Anschluss ist ein Gigabit-Ethernet-Netzwerkanschluss, der es Ihnen ermöglicht, den Raspberry Pi 400 über ein Netzwerkkabel mit Ihrem Netzwerk zu verbinden, als Alternative zur Verwendung des integrierten drahtlosen WLAN-Netzwerks. Weitere Informationen über den Anschluss des Raspberry Pi 400 an ein Netzwerk finden Sie in **Kapitel 2, Erste Schritte mit Ihrem Raspberry Pi**.

Kapitel 2

Erste Schritte mit Ihrem Raspberry Pi

Entdecken Sie das wesentliche Zubehör, das Sie für Ihren Raspberry Pi benötigen, und erfahren Sie, wie Sie alles miteinander verbinden können, um Ihren Raspberry Pi für den Betrieb vorzubereiten



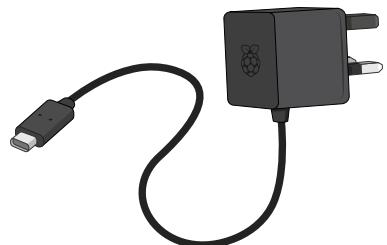
Der Raspberry Pi ist auf schnelle und einfache Einrichtung und Benutzung ausgelegt. Wie jeder Computer stützt er sich allerdings auch auf verschiedene externe Komponenten, die Peripheriegeräte genannt werden. Ein Blick auf die Leiterplatte des Raspberry Pi zeigt, dass er sich deutlich von den im Gehäuse verkapselten, verschlossenen Computern unterscheidet, die Sie vermutlich gewohnt sind. Das könnte bei Ihnen zur Sorge führen, dass es jetzt kompliziert wird, aber das ist nicht der Fall. Sie können einen Raspberry Pi in weit weniger als zehn Minuten einrichten, wenn Sie einfach den Schritten in diesem Leitfaden folgen.

Wenn Sie dieses Buch in einem Raspberry Pi Desktop Kit oder zusammen mit einem Raspberry Pi 400 erhalten haben, steht Ihnen bereits fast alles zur Verfügung, was Sie für den Anfang benötigen: Sie brauchen nur noch einen Computermonitor oder Fernseher mit einem HDMI-Anschluss – dieselbe Art von Anschluss, die auch von Set-Top-Boxen, Blu-ray-Playern und Spielkonsolen verwendet wird. So können Sie sehen, was Ihr Raspberry Pi macht.

Wenn Sie Ihren Raspberry Pi ohne Zubehör gekauft haben, dann brauchen Sie zusätzlich Folgendes:

■ **USB-Netzteil** – Ein 5 V Netzteil mit 3 A (Ampere)

Stromstärke und mit einem USB-C Stecker. Das offizielle Raspberry Pi Netzteil ist die empfohlene Wahl, da es den schnell fluktuierenden Leistungsanforderungen des Raspberry Pi gewachsen ist.



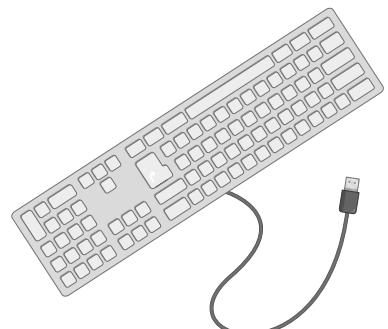
■ **microSD-Karte mit NOOBS** – Die microSD-Karte fungiert

als permanenter Speicher des Raspberry Pi. Alle Dateien, die Sie erstellen, und die Software, die Sie installieren, werden zusammen mit dem Betriebssystem selbst auf der Karte gespeichert. Eine 8 GB Karte reicht für den Einstieg, eine 16 GB Karte bietet jedoch mehr Platz für Ihre Daten, Projekte und zusätzliche Software. Eine Karte mit vorinstalliertem NOOBS (New Out-Of-Box Software) spart Ihnen Zeit; ansonsten lesen Sie **Anhang A** mit Anweisungen zur Installation eines Betriebssystems (OS) auf eine leere Karte.



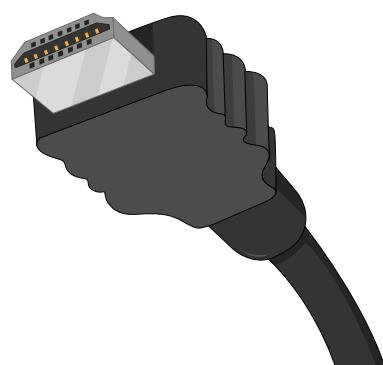
■ **USB-Tastatur und -Maus** – Mit der Tastatur und der Maus

können Sie Ihren Raspberry Pi steuern. Nahezu alle kabelgebundenen oder kabellosen Tastaturen und Mäuse mit USB-Anschluss funktionieren mit dem Raspberry Pi, obwohl einige „Gaming“-Tastaturen mit farbiger Tastenbeleuchtung möglicherweise zu viel Strom ziehen, um zuverlässig genutzt werden zu können.



■ **Micro-HDMI-Kabel** – Dieses Kabel überträgt Ton

und Bild vom Raspberry Pi zu Ihrem Fernseher oder Monitor. Ein Ende des Kabels verfügt über einen Micro-HDMI-Anschluss (Typ D) für den Raspberry Pi, das andere Ende über einen Standard-HDMI-Anschluss (Typ A) für Ihren Bildschirm. Sie können auch einen Micro-HDMI-zu-HDMI-Adapter und ein Standard-HDMI-Kabel verwenden. Wenn Sie einen Monitor ohne HDMI-Anschluss verwenden, können Sie Micro-HDMI-zu-DVI-D-, DisplayPort- oder VGA-Adapter kaufen. Zum Anschluss an ein älteres Fernsehgerät, das Composite-Video verwendet oder über eine SCART-Buchse verfügt, verwenden Sie ein 3,5 mm Tip-Ring-Ring-Sleeve (TRRS)-Audio-/Videokabel.



Der Raspberry Pi kann sicher ohne Gehäuse verwendet werden, vorausgesetzt, Sie stellen ihn nicht auf eine Metalloberfläche, die Elektrizität leiten und einen Kurzschluss verursachen könnte. Ein optionales Gehäuse kann jedoch zusätzlichen Schutz bieten; das Desktop-Kit enthält das offizielle Raspberry Pi-Gehäuse, viele Händler bieten Gehäuse von Drittanbietern an.

Wenn Sie Raspberry Pi nicht in einem drahtlosen (WLAN), sondern in einem kabelgebundenen Netzwerk verwenden möchten, benötigen Sie außerdem ein Netzwerkkabel. Dieses sollte an einem Ende mit dem Switch oder Router Ihres Netzwerks verbunden sein. Wenn Sie vorhaben, das eingebaute WLAN des Raspberry Pi zu verwenden, benötigen Sie kein Kabel. Sie müssen jedoch den Namen und den Schlüssel oder das Passwort für Ihr drahtloses Netzwerk kennen.



EINRICHTUNG DES RASPBERRY PI 400

Die folgenden Anweisungen gelten für die Einrichtung vom Raspberry Pi 4 oder einem anderen Bare-Board Mitglied der Raspberry Pi-Familie. Eine Anleitung zur Einrichtung vom Raspberry Pi 400 finden Sie auf Seite 32.



Einrichten der Hardware

Packen Sie als Erstes Ihren Raspberry Pi aus seiner Schachtel aus. Der Raspberry Pi ist robuste Hardware, aber das bedeutet nicht, dass er unzerstörbar ist: Versuchen Sie, sich daran zu gewöhnen, die Platine an den Kanten statt an den flachen Seiten zu halten, und seien Sie besonders vorsichtig mit den herausragenden Metallstiften. Wenn diese Stifte verbogen sind, erschwert dies im besten Fall die Verwendung von Zusatzplatinen und anderer zusätzlicher Hardware und kann im schlimmsten Fall einen Kurzschluss verursachen, der Ihren Raspberry Pi beschädigt.

Einzelheiten darüber, wo genau sich die verschiedenen Anschlüsse befinden und wofür sie verwendet werden, finden Sie in **Kapitel 1, Lernen Sie Ihren Raspberry Pi kennen**.

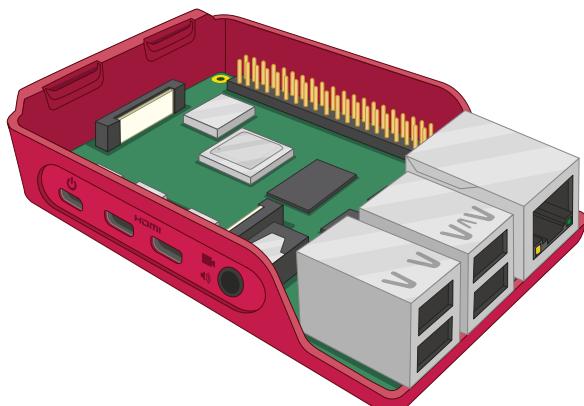
Zusammenbau des Gehäuses

Falls Sie den Raspberry Pi in einem Gehäuse benutzen möchten, sollten Sie mit diesem Schritt anfangen. Wenn Sie das offizielle Raspberry Pi-Gehäuse verwenden, zerlegen Sie es zunächst in seine zwei Einzelteile: den roten Boden und den weißen Deckel.

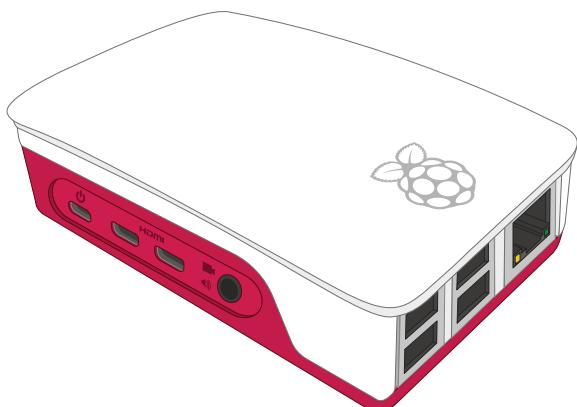
- 1 Nehmen Sie den Boden und halten Sie ihn so, dass das erhöhte Ende zu Ihrer Linken und das tieferliegende Ende zu Ihrer Rechten zeigt.**



- 2 Greifen Sie Ihren Raspberry Pi (ohne eingesteckte microSD-Karte) an den USB- und Ethernet-Anschlüssen, halten Sie ihn leicht schräg und stecken Sie die Anschlüsse (USB-C, 2 × micro-HDMI und 3,5 mm) in ihre Öffnungen an der Seite des Bodens. Senken Sie dann die andere Seite der Platine vorsichtig nach unten, sodass sie flach aufliegt.**

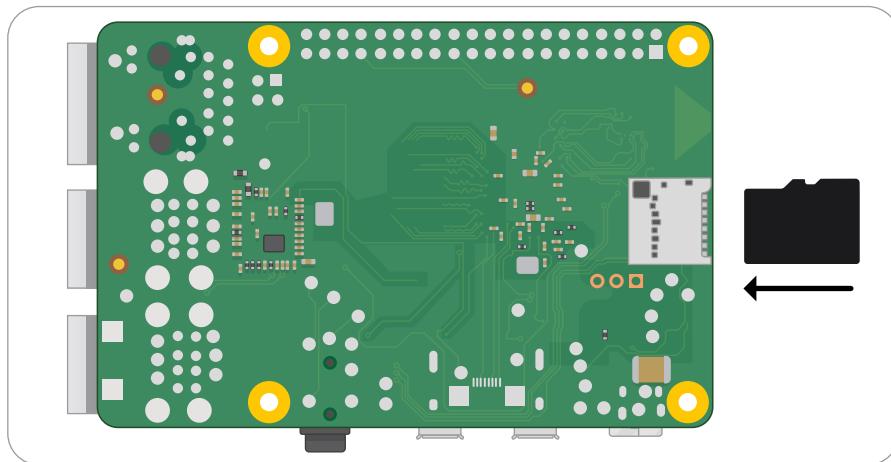


- 3 Nehmen Sie den weißen Deckel und setzen Sie die beiden Clips auf der linken Seite in die passenden Löcher auf der linken Seite des Bodens oberhalb des microSD-Kartensteckplatzes ein. Nachdem die Clips eingehängt worden sind, drücken Sie die rechte Seite (oberhalb der USB-Anschlüsse) nach unten, bis Sie ein Klicken hören.**

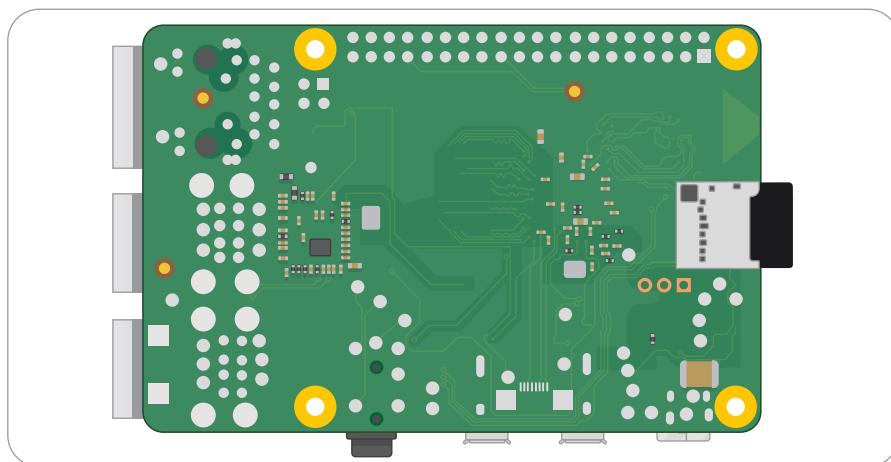


Einstecken der microSD-Karte

Zur Installation der microSD-Karte, die der Speicher des Raspberry Pi ist, drehen Sie den Raspberry Pi (im Gehäuse, wenn Sie eins verwenden), und schieben Sie die Karte, mit dem Label vom Raspberry Pi weg zeigend, in den microSD-Steckplatz. Sie kann nur in eine Richtung eingeschoben werden und sollte ohne allzu großen Druck hineingleiten.



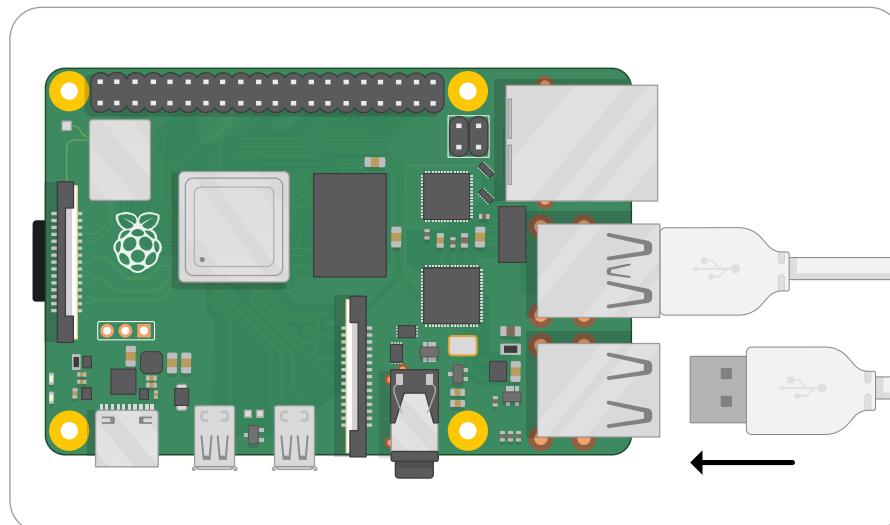
Die microSD-Karte gleitet in den Anschluss und stoppt dann ohne ein Klicken.



Wenn Sie sie später wieder entfernen möchten, ziehen Sie sie einfach vorsichtig heraus. (Bitte nur während der Raspberry Pi nicht an das Netzteil angeschlossen ist um Schäden an Ihren Daten zu vermeiden!) Wenn Sie ein älteres Raspberry Pi Modell verwenden, müssen Sie die Karte zuerst leicht drücken, um sie zu entsperren; bei einem Raspberry Pi 3 oder 4 ist dies nicht notwendig.

Anschließen einer Tastatur und Maus

Schließen Sie das USB-Kabel der Tastatur an einen der vier USB-Anschlüsse (2.0 oder 3.0) des Raspberry Pi an. Wenn Sie die offizielle Raspberry Pi Tastatur verwenden, befindet sich auf der Rückseite ein USB-Anschluss für die Maus; wenn nicht, schließen Sie einfach das USB-Kabel Ihrer Maus an einen anderen USB-Anschluss des Raspberry Pi an.



Die USB-Anschlüsse für Tastatur und Maus sollten ohne zu viel Druck hineingleiten; wenn Sie den Stecker mit Gewalt einstecken müssen, stimmt etwas nicht. Prüfen Sie, ob Sie den USB-Stecker richtig herum einstecken!



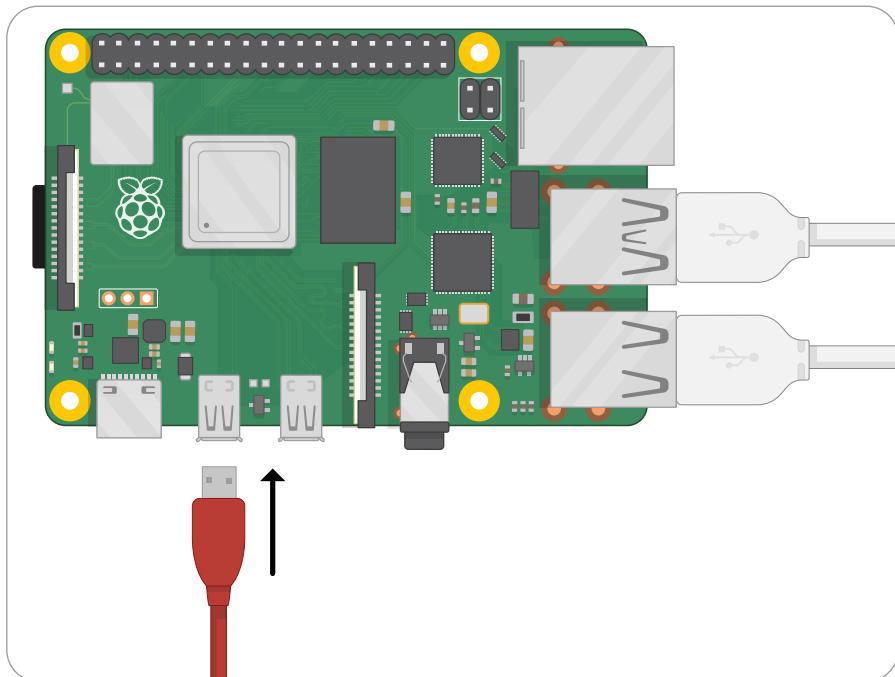
TASTATUR UND MAUS

Tastatur und Maus sind die wichtigsten Geräte, über die Sie Ihren Raspberry Pi steuern können. In der Informatik sind sie als *Eingabegeräte* bekannt, im Gegensatz zum Display, das ein *Ausgabegerät* ist.



Anschließen eines Displays

Nehmen Sie das Micro-HDMI-Kabel und verbinden Sie das kleinere Ende mit dem "HDMI0" Micro-HDMI-Anschluss, direkt neben dem USB-C Anschluss an Ihrem Raspberry Pi, und das andere Ende mit Ihrem Display. Wenn Ihr Bildschirm mehr als einen HDMI-Anschluss hat, suchen Sie nach einer Anschlussnummer neben diesem Anschluss. Sie müssen den Fernseher auf diesen Eingang schalten, um die Bildschirmausgabe Ihres Raspberry Pi zu sehen. Wenn Sie keine Anschlussnummer finden, kein Problem: Sie können einfach nacheinander durch jeden Eingang schalten, bis Sie den Raspberry Pi finden.



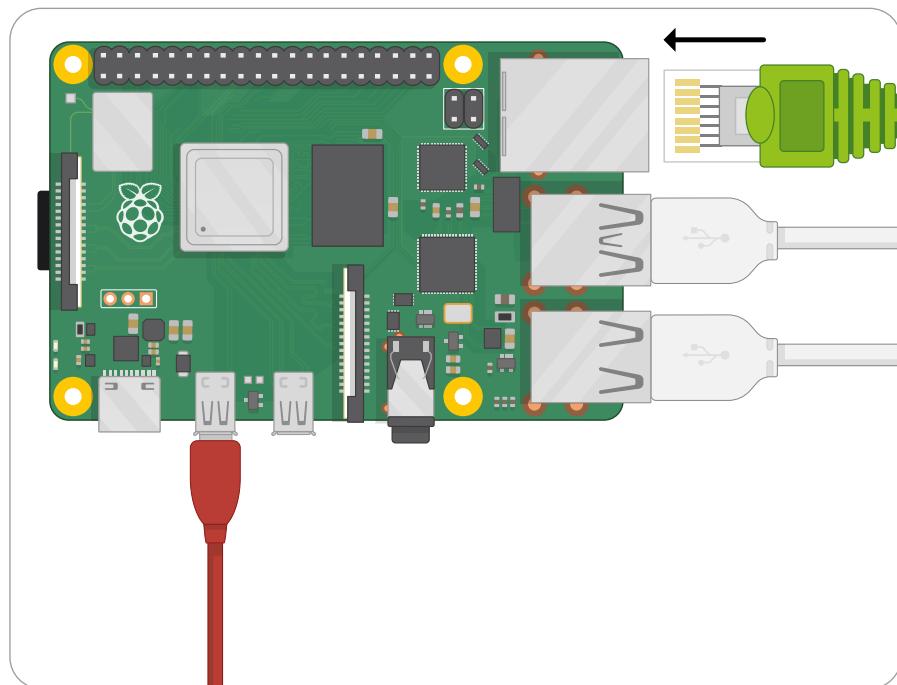
TV-ANSCHLUSS

Auch wenn Ihr Fernseher oder Monitor über keinen HDMI-Anschluss verfügt, können Sie den Raspberry Pi trotzdem daran anschließen. Adapterkabel, die bei jedem Elektronikfachhändler erhältlich sind, ermöglichen es Ihnen, den Micro-HDMI-Anschluss Ihres Raspberry Pi für die Verwendung mit älteren Computermonitoren in DVI-D, DisplayPort oder VGA umzuwandeln. Diese werden einfach an den Micro-HDMI-Anschluss des Raspberry Pi angeschlossen und dann mit einem geeigneten Kabel mit dem Monitor verbunden. Wenn Ihr Fernseher nur über einen Composite-Video- oder SCART-Eingang verfügt, können Sie 3,5 mm Tip-Ring-Ring-Sleeve (TRRS)-Adapterkabel und Composite-zu-SCART-Adapter kaufen, die an die 3,5 mm AV-Buchse angeschlossen werden.



Anschließen eines Netzwerkkabels (optional)

Um Ihren Raspberry Pi an ein kabelgebundenes Netzwerk anzuschließen, nehmen Sie ein Netzwerkkabel – ein so genanntes Ethernet-Kabel – und schieben Sie es mit der Kunststoffklammer nach unten in den Ethernet-Anschluss des Raspberry Pi, bis Sie ein Klicken hören. Wenn Sie das Kabel entfernen müssen, drücken Sie einfach die Kunststoffklammer nach innen in Richtung des Steckers und ziehen Sie das Kabel vorsichtig wieder heraus.

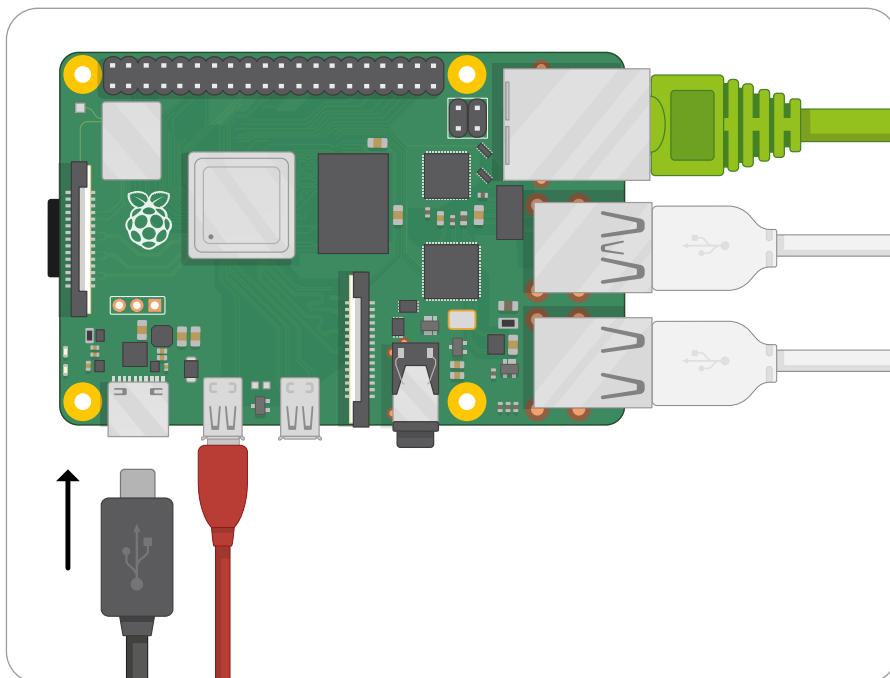


Das andere Ende Ihres Netzwerkkabels sollte auf die gleiche Weise an einen beliebigen freien Anschluss Ihres Netzwerk-Hubs, Switch oder Routers angeschlossen werden.

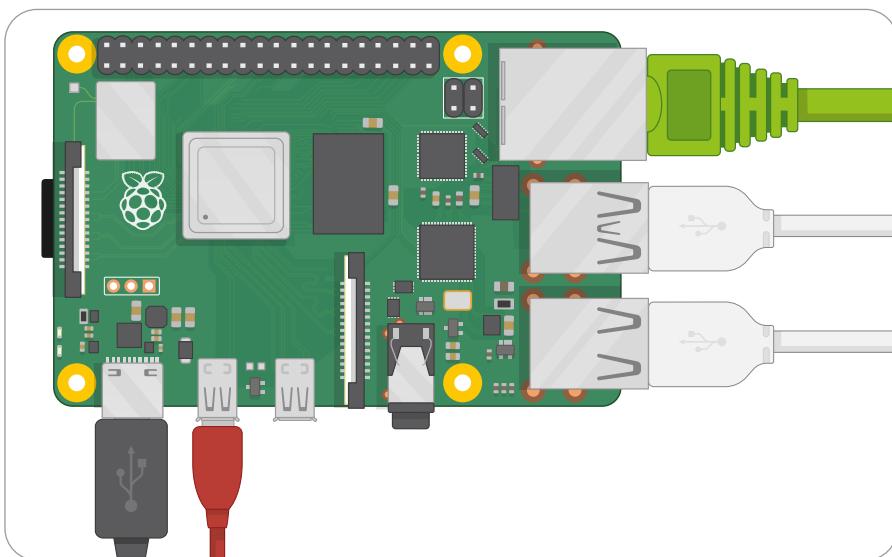
Anschließen eines Netzteils

Der Anschluss von Raspberry Pi an ein Netzteil ist der allerletzte Schritt im Hardware-Einrichtungsprozess. Sie sollten es erst dann einstecken, wenn Sie bereit sind, die Software einzurichten: Der Raspberry Pi hat keinen Netzschalter und schaltet sich sofort ein, sobald ein an den Strom angeschlossenes Netzteil in ihn eingesteckt wird.

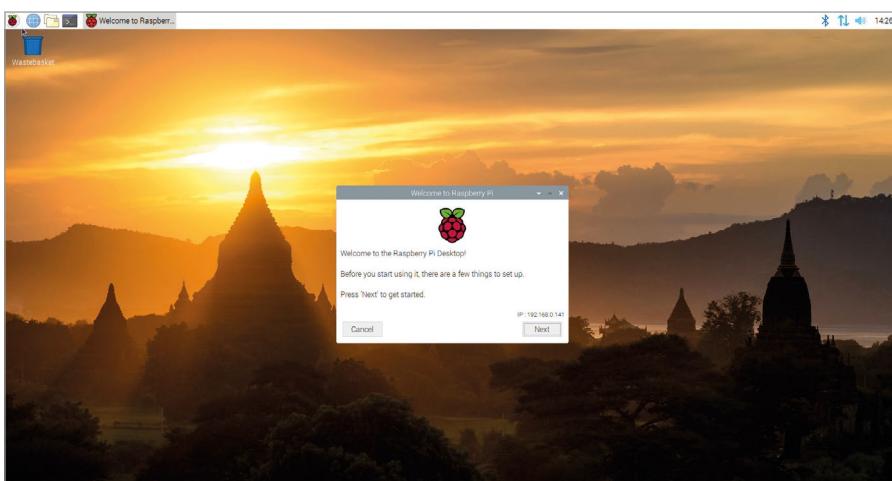
Schließen Sie zunächst das USB-C-Ende des Netzteilkabels an den USB-C-Stromanschluss des Raspberry Pi an. Der Stecker kann dabei beliebig herum eingesteckt werden und sollte sanft hineingleiten. Wenn Ihr Netzteil über ein abnehmbares Kabel verfügt, stellen Sie sicher, dass das andere Ende in das Gehäuse des Netzteils eingesteckt ist.



Schließen Sie dann das Netzteil an eine Steckdose an. Ihr Raspberry Pi wird sofort starten. Herzlichen Glückwunsch: Sie haben Ihren Raspberry Pi angeschlossen!



Vier Raspberry Pi-Logos werden kurz oben links auf einem schwarzen Bildschirm angezeigt. Möglicherweise sehen Sie auch einen blauen Bildschirm, während sich die Software selbst in der Größe anpasst, um Ihre microSD-Karte voll auszunutzen. Wenn Sie einen schwarzen Bildschirm sehen, warten Sie einige Minuten: Beim ersten Bootvorgang muss der Raspberry Pi im Hintergrund etwas Arbeit erledigen. Nach einer Weile sehen Sie die Arbeitsoberfläche und den Begrüßungsassistenten von Raspberry Pi OS, wie in **Abbildung 2-1** gezeigt. Ihr Betriebssystem ist jetzt bereit zur Konfiguration, über die Sie sich in **Kapitel 3, Verwendung Ihres Raspberry Pi** informieren können.



▲ Abbildung 2-1: Der Desktop- und Begrüßungsassistent für Raspberry Pi OS

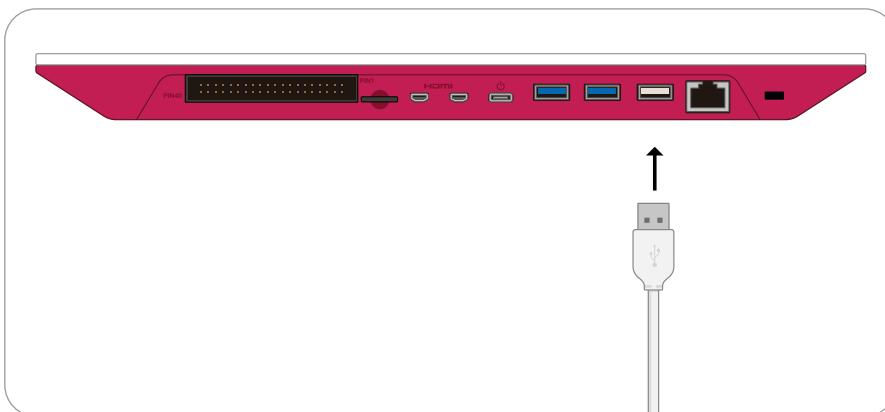
Einrichtung vom Raspberry Pi 400

Im Gegensatz zu Raspberry Pi 4 wird Raspberry Pi 400 mit einer eingebauten Tastatur und einer bereits installierten microSD-Karte geliefert. Sie müssen noch ein paar Kabel anschließen, um anfangen zu können, aber das sollte nur ein paar Minuten dauern.



Anschließen einer Maus

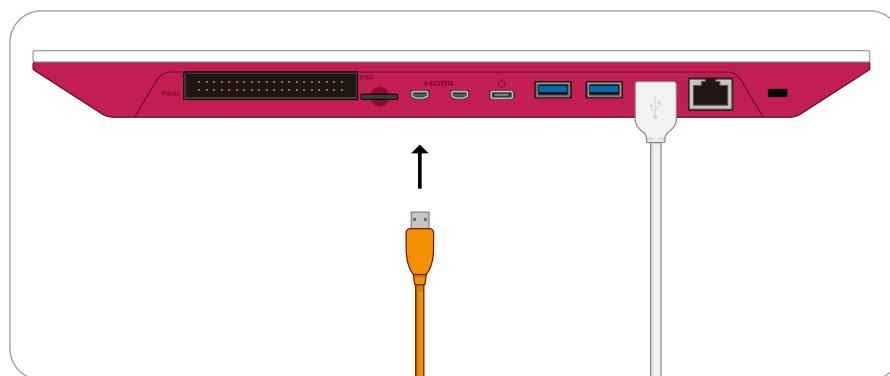
Die Tastatur des Raspberry Pi 400 ist bereits angeschlossen, sodass Sie nur noch die Maus hinzufügen müssen. Nehmen Sie das USB-Kabel am Ende der Maus und stecken Sie es in einen der drei USB-Anschlüsse (2.0 oder 3.0) auf der Rückseite des Raspberry Pi 400. Wenn Sie die beiden Hochgeschwindigkeits-USB 3.0-Anschlüsse für anderes Zubehör nutzen möchten, verwenden Sie den weißen Anschluss.



Der USB-Anschluss sollte ohne zu viel Druck hineingleiten. Wenn Sie den Stecker mit Gewalt einstecken müssen, stimmt etwas nicht. Prüfen Sie, ob Sie den USB-Stecker richtig herum einstecken!

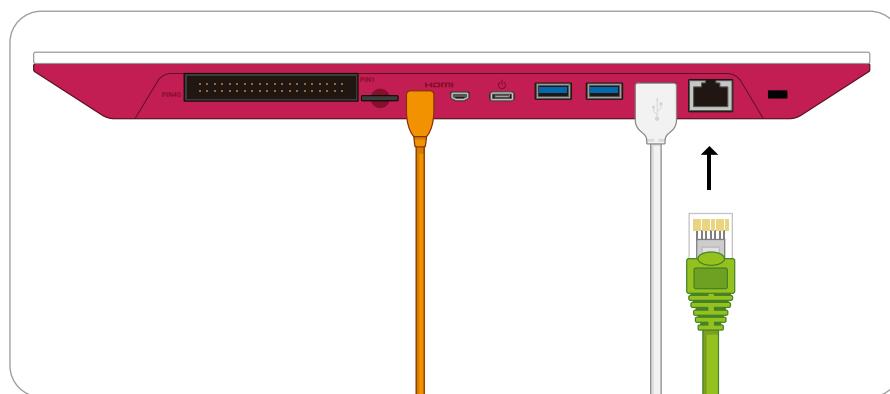
Anschließen eines Displays

Nehmen Sie das micro-HDMI-Kabel und verbinden Sie das kleinere Ende mit dem micro-HDMI-Anschluss, direkt neben dem microSD-Steckplatz Ihres Raspberry Pi 400, und das andere Ende mit Ihrem Display. Wenn Ihr Bildschirm mehr als einen HDMI-Anschluss hat, suchen Sie nach einer Anschlussnummer neben dem Anschluss. Sie müssen den Fernseher auf diesen Eingang schalten, um die Bildschirmausgabe Ihres Raspberry Pi zu sehen. Wenn Sie die Anschluss-Nummer nicht finden, kein Problem: Sie können einfach nacheinander durch jeden Eingang schalten, bis Sie den Raspberry Pi finden.



Anschließen eines Netzwerkkabels (optional)

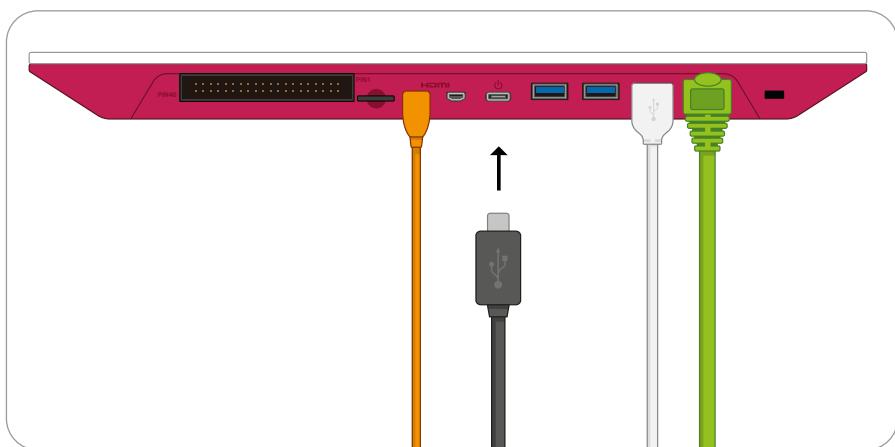
Um Ihren Raspberry Pi 400 an ein kabelgebundenes Netzwerk anzuschließen, nehmen Sie ein Netzwerkkabel – ein sogenanntes Ethernet-Kabel – und schieben Sie es mit der Kunststoffklammer nach oben in den Ethernet-Anschluss des Raspberry Pi 400, bis Sie ein Klicken hören. Wenn Sie das Kabel entfernen müssen, drücken Sie einfach die Kunststoffklammer nach innen in Richtung des Steckers und ziehen Sie das Kabel vorsichtig wieder heraus.



Das andere Ende Ihres Netzwerkkabels sollte auf die gleiche Weise an einen beliebigen freien Anschluss Ihres Netzwerk-Hubs, Switch oder Routers angeschlossen werden.

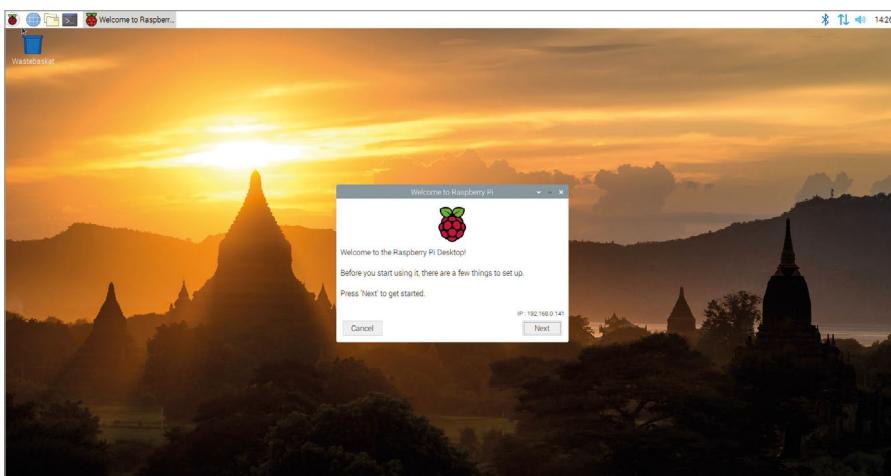
Anschießen eines Netzteils

Der Anschluss des Raspberry Pi 400 an ein Netzteil ist der allerletzte Schritt im Hardware-Einrichtungsprozess. Sie sollten es erst dann anschließen, wenn Sie bereit sind, die Software einzurichten: Der Raspberry Pi 400 verfügt über keinen Netzschalter und schaltet sich sofort ein, sobald ein an den Strom angeschlossenes Netzteil in ihn eingesteckt wird. Schließen Sie zunächst das USB-C-Ende des Netzteilkabels an den USB-C-Stromanschluss des Raspberry Pi an. Der Stecker kann dabei beliebig herum eingesteckt werden und sollte sanft hineingleiten. Wenn Ihr Netzteil über ein abnehmbares Kabel verfügt, stellen Sie sicher, dass das andere Ende in das Gehäuse des Netzteils eingesteckt ist.



Schließen Sie dann das Netzteil an eine Steckdose an. Ihr Raspberry Pi wird sofort starten. Herzlichen Glückwunsch: Sie haben Ihren Raspberry Pi 400 angeschlossen!

Vier Raspberry Pi-Logos werden kurz oben links auf einem schwarzen Bildschirm angezeigt. Möglicherweise sehen Sie auch einen blauen Bildschirm, während sich die Software selbst in der Größe anpasst, um Ihre microSD-Karte voll auszunutzen. Wenn Sie einen schwarzen Bildschirm sehen, warten Sie einige Minuten: Beim ersten Bootvorgang muss der Raspberry Pi im Hintergrund etwas Arbeit erledigen. Nach einer Weile sehen Sie die Arbeitsoberfläche und den Begrüßungsassistenten von Raspberry Pi OS, wie in **Abbildung 2-2** gezeigt. Ihr Betriebssystem ist jetzt bereit zur Konfiguration, über die Sie sich in **Kapitel 3, Verwendung Ihres Raspberry Pi** informieren können.

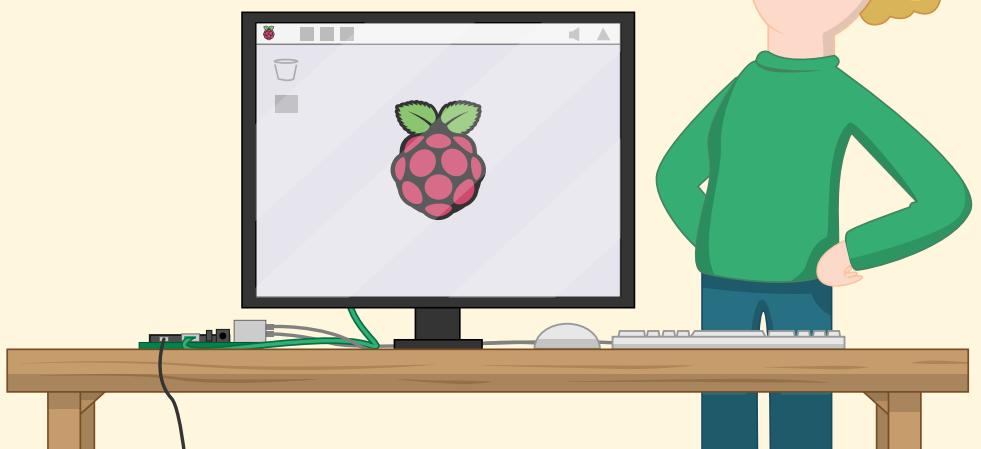


▲ **Abbildung 2-2:** Der Desktop- und Begrüßungsassistent für Raspberry Pi OS

Kapitel 3

Verwendung Ihres Raspberry Pi

Erfahren Sie mehr über das Betriebssystem

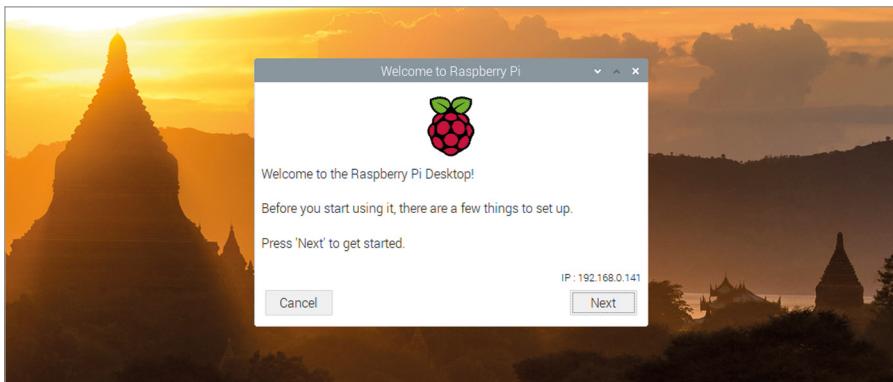


Der Raspberry Pi kann eine große Vielfalt von Softwareprogrammen ausführen, darunter eine Reihe verschiedener Betriebssysteme – die Kern-Software, die einen Computer zum Laufen bringt. Das Raspberry Pi OS ist das beliebteste unter ihnen und auch das offizielle Betriebssystem der Raspberry Pi Foundation. Es basiert auf Debian Linux, und ist maßgeschneidert für Raspberry Pi Computer, mit einigen vorinstallierten Extras. Das System ist sofort einsatzbereit.

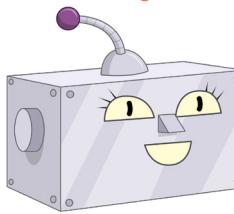
Wenn Sie bisher nur Microsoft Windows oder Apple macOS verwendet haben, machen Sie sich keine Sorgen: Raspberry Pi OS basiert ebenso auf Fenstern, Symbolen, Menüs und dem Mauszeiger (so genanntes WIMP-User Interface), sodass Sie sich ganz schnell damit anfreunden werden. Das folgende Kapitel soll Ihnen den Einstieg erleichtern und Ihnen einen Teil der mitgelieferten Software vorstellen.

Der Begrüßungsassistent

Wenn Sie das Raspberry Pi OS zum ersten Mal ausführen, sehen Sie den Begrüßungsassistenten (**Abbildung 3-1**). Dieses hilfreiche Software-Werkzeug führt Sie durch die Änderung einiger Einstellungen im Raspberry Pi OS, bekannt als *Konfiguration*, um Ihren Raspberry Pi an Verwendungszweck und -ort anzupassen.



▲ Abbildung 3-1: Der Begrüßungsassistent



SCHLIESSEN DES ASSISTENTEN

Sie können den Begrüßungsassistenten schließen, indem Sie auf den Button „Cancel“ klicken, aber bestimmte Funktionen des Raspberry Pi – wie das drahtlose Netzwerk – funktionieren erst, wenn Sie zumindest die ersten Fragen beantwortet haben.

!

Klicken Sie auf den Button „Next“ und wählen Sie dann Ihr Land, Ihre Sprache und Ihre Zeitzone, indem Sie nacheinander auf die einzelnen Dropdown-Felder klicken und Ihre Antwort aus der Liste auswählen (**Abbildung 3-2**). Wenn Sie eine Tastatur mit US-Layout verwenden, klicken Sie auf die Checkbox, um sicherzustellen, dass das Raspberry Pi OS das richtige Tastaturlayout verwendet. Wenn Sie möchten, dass der Desktop und die Programme unabhängig von der Muttersprache Ihres Landes in Englisch angezeigt werden, klicken Sie auf die Checkbox „Use English language“, um sie anzukreuzen. Wenn Sie fertig sind, klicken Sie auf „Next“.



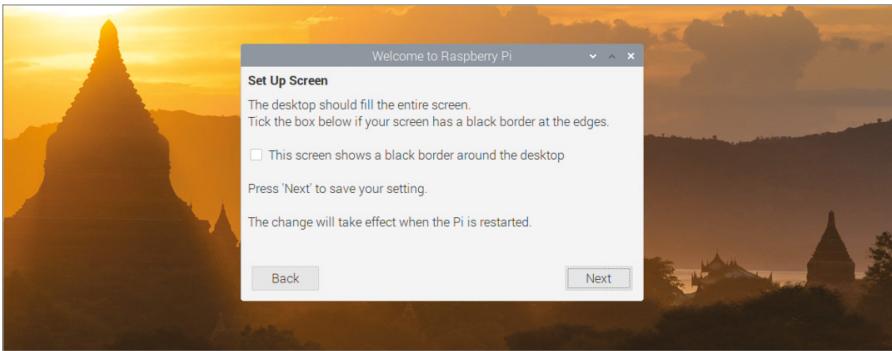
▲ Abbildung 3-2: Auswahl einer Sprache und andere Einstellungen

Auf dem nächsten Bildschirm werden Sie aufgefordert, das Passwort für den Benutzer „pi“ zu ändern (von der Standardeinstellung „raspberry“). Aus Sicherheitsgründen empfehlen wir Ihnen, hier ein neues Passwort zu vergeben. Geben Sie das neue Passwort in beiden Feldern ein (**Abbildung 3-3**). Wenn alles passt, klicken Sie auf „Next“.



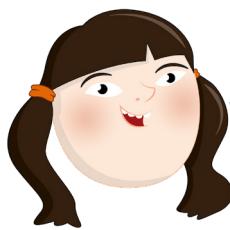
▲ Abbildung 3-3: Festlegen eines neuen Passworts

Auf dem nächsten Bildschirmbild wird gefragt, ob ein schwarzer Rand um den Bildschirrrand herum vorhanden ist (**Abbildung 3-4**). Wenn der Raspberry Pi-Desktop die gesamte Fläche Ihres Fernsehers oder Monitors ausfüllt, lassen Sie das Kästchen leer; wenn er einen schwarzen Rand hat und kleiner als Ihr Fernseher oder Monitor ist, aktivieren Sie das Kästchen. Wenn Sie soweit sind, weiterzumachen, klicken Sie auf „Next“.



▲ Abbildung 3-4: Überprüfen, ob ein schwarzer Rand vorhanden ist

Auf dem folgenden Bildschirmbild können Sie Ihr WLAN-Netzwerk aus einer Liste auswählen (**Abbildung 3-5**). Scrollen Sie mit der Maus oder der Tastatur durch die Liste der Netzwerke, suchen Sie den Namen Ihres Netzwerks, klicken Sie ihn an und klicken Sie dann auf „Next“. Unter der Annahme, dass Ihr drahtloses Netzwerk gesichert ist (was sehr zu empfehlen ist), werden Sie nach seinem Passwort (auch bekannt als sein Pre-Shared Key) gefragt. Dieses



DRAHTLOSE NETZWERKE



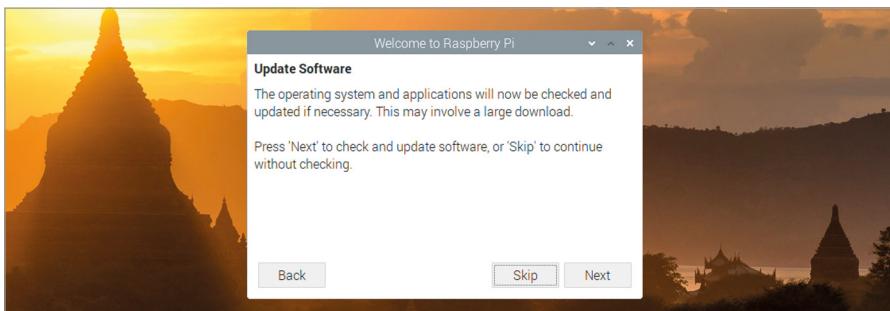
Ein eingebautes WLAN-Modul ist nur bei den Produktfamilien Raspberry Pi 3, Pi 4, Pi 00 und Pi Zero W vorhanden. Wenn Sie ein anderes Raspberry Pi Modell mit einem drahtlosen Netzwerk verbinden möchten, benötigen Sie einen USB-WLAN-Adapter.

steht normalerweise in den Unterlagen Ihres Routers oder auf der Unterseite des Routers selbst. Klicken Sie auf „Next“, um eine Verbindung mit dem Netzwerk herzustellen. Wenn Sie keine Verbindung zu einem drahtlosen Netzwerk herstellen möchten, klicken Sie auf „Skip“.



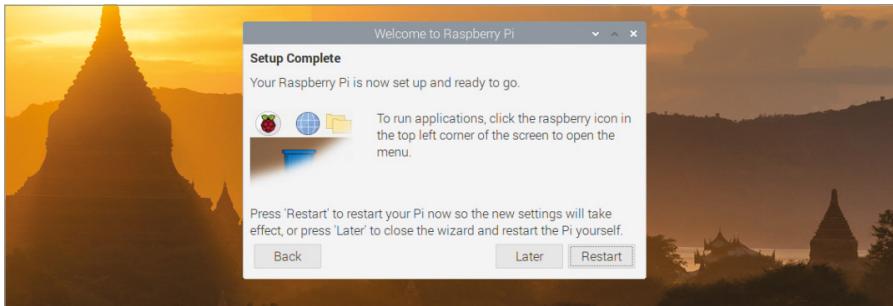
▲ Abbildung 3-5: Auswahl eines drahtlosen Netzwerks

Auf dem nächsten Bildschirmbild können Sie nach Updates für Raspberry Pi OS und den anderen installierten SoftwarePaketen auf Ihrem Raspberry Pi suchen und diese installieren lassen(**Abbildung 3-6**). Das Raspberry Pi OS wird regelmäßig aktualisiert, um Fehler zu beheben, neue Funktionen hinzuzufügen und die Leistung zu verbessern. Um diese Updates zu installieren, klicken Sie auf „Next“. Andernfalls klicken Sie auf „Skip“. Das Herunterladen der Updates kann einige Minuten dauern, haben Sie also etwas Geduld. Wenn die Updates installiert sind, erscheint ein Fenster mit der Meldung „System is up to date“. Klicken Sie auf den Button „OK“.



▲ Abbildung 3-6: Auf Updates prüfen

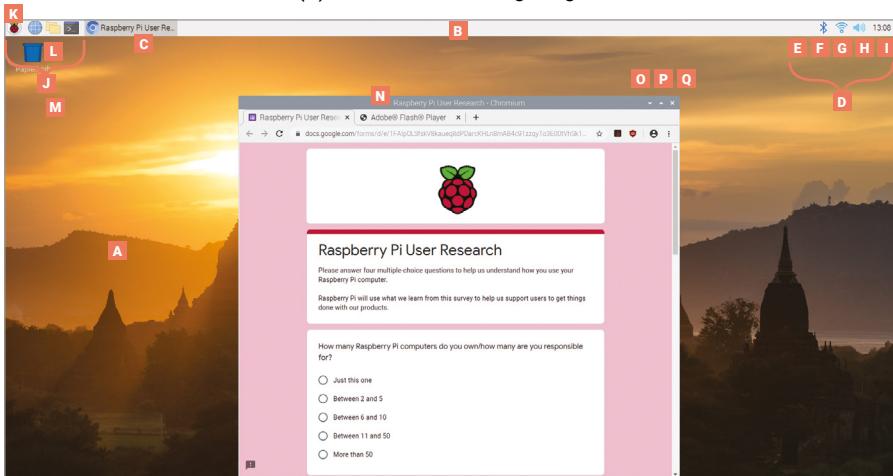
Das letzte Bildschirmbild des Begrüßungsassistenten (**Abbildung 3-7**) hat eine einfache Aufgabe: Bestimmte Einstellungen werden erst dann wirksam, wenn Sie Ihren Raspberry Pi neu starten (ein Prozess, der auch als "Rebooten" bezeichnet wird). Wenn Sie dazu aufgefordert werden, klicken Sie auf den Button „Restart“ und Ihr Raspberry Pi wird neu gestartet. Diesmal wird der Begrüßungsassistent nicht angezeigt. Seine Arbeit ist getan und Ihr Raspberry Pi ist einsatzbereit.



▲ Abbildung 3-7: Neustart des Raspberry Pis

Navigieren auf dem Desktop

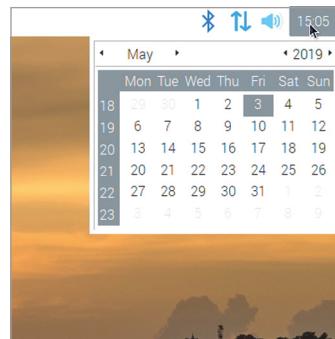
Die Version des Raspberry Pi OS, die auf den meisten Raspberry-Pi-Boards installiert ist, ist mit vollständigem Namen als „Raspberry Pi OS with Desktop“ bekannt, was sich insbesondere auf die primäre grafische Benutzeroberfläche bezieht (**Abbildung 3-8**). Der größte Teil dieses Desktops wird von einem Bild ausgefüllt, dem Hintergrundbild (**A** in **Abbildung 3-8**), auf dem die von Ihnen ausgeführten Programme angezeigt werden. Am oberen Rand des Desktops befindet sich eine Taskleiste (**B**), womit Sie jedes der Programme starten können: Diese werden in der Form von Tasks (**C**) in der Taskleiste angezeigt.



▲ Abbildung 3-8: Der Raspberry Pi OS-Desktop

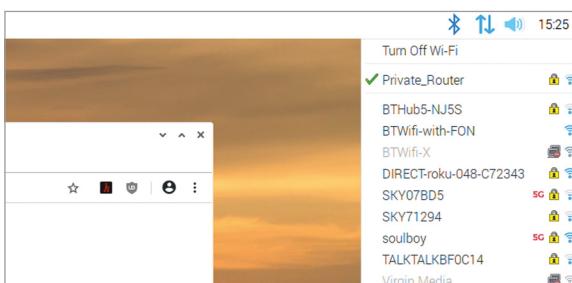
A	Bildschirmhintergrund	G	Netzwerk-Symbol	M	Symbol für entfernbare Laufwerk
B	Taskleiste	H	Lautstärke-Symbol	N	Fenster-Titelleiste
C	Task	I	Uhr	O	Verkleinern
D	Infobereich	J	Launcher	P	Vergrößern
E	Auswurf-Symbol	K	Menü- (oder Raspberry-)Symbol	Q	Schließen
F	Bluetooth-Symbol	L	Papierkorb-Symbol		

Die rechte Seite der Menüleiste enthält den **Infobereich (D)**. Wenn Sie Wechseldatenträger wie z. B. USB-Speichersticks, an den Raspberry Pi angeschlossen haben, sehen Sie ein Auswurf-Symbol (E). Wenn Sie darauf klicken, können Sie diese Datenträger sicher auswerfen und entfernen. Ganz rechts befindet sich die Uhr (I); klicken Sie darauf, um den digitalen Kalender aufzurufen (**Abbildung 3-9**).



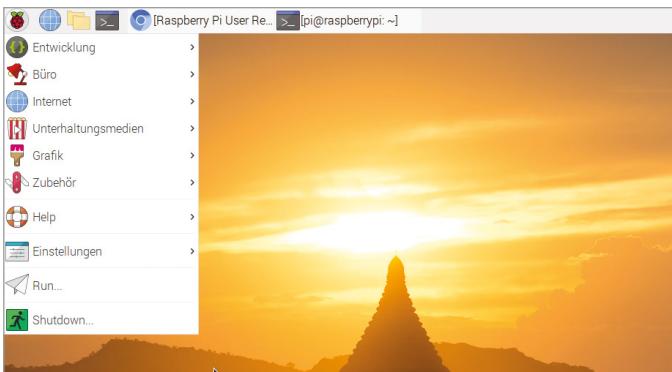
► **Abbildung 3-9:** Der digitale Kalender

Daneben befindet sich ein Lautsprechersymbol (H). Klicken Sie mit der linken Maustaste darauf, um die Audio-Lautstärke des Raspberry Pi einzustellen, oder klicken Sie mit der rechten Maustaste darauf, um auszuwählen, welche Audio-Ausgabe der Raspberry Pi verwenden soll. Daneben befindet sich ein Netzwerksymbol (G). Wenn Sie mit einem drahtlosen Netzwerk verbunden sind, wird die Signalstärke als eine Reihe von Balken angezeigt, während Sie bei einem drahtgebundenen Netzwerk nur zwei Pfeile sehen. Wenn Sie auf das Netzwerksymbol klicken, wird eine Liste der nahegelegenen drahtlosen Netzwerke angezeigt (**Abbildung 3-10**). Wenn Sie auf das Bluetooth-Symbol (F) daneben klicken, können Sie eine Verbindung zu einem Bluetooth-Gerät in der Nähe herstellen.



◀ **Abbildung 3-10:**
Auflistung nahegelegener
drahtloser Netzwerke

Auf der linken Seite der Menüleiste befindet sich der **Launcher (J)**, in dem Sie die Programme finden, die zusammen mit Raspberry Pi OS installiert wurden. Einige davon sind als Verknüpfungssymbole sichtbar. Andere sind im Menü versteckt, das Sie durch Klicken auf das Raspberry-Symbol (das Himbeeren-Symbol, K) ganz links anzeigen können (**Abbildung 3-11** (umseitig)).



▲ Abbildung 3-11: Das Raspberry Pi OS-Menü

Die Programme im Menü sind in Kategorien eingeteilt, deren Namen Ihnen sagen, was Sie erwartet: Die Kategorie „Entwicklung“ beispielsweise enthält Software, die Ihnen helfen soll, Ihre eigenen Programme zu schreiben – wie ab **Kapitel 4, Programmierung mit Scratch** erklärt. Unter „Spiele“ hingegen finden Sie Programme, mit denen Sie sich die Zeit vertreiben können. Nicht alle Programme werden in diesem Leitfaden detailliert beschrieben; experimentieren Sie einfach mit ihnen, um mehr zu erfahren.

Der Chromium-Webbrowser

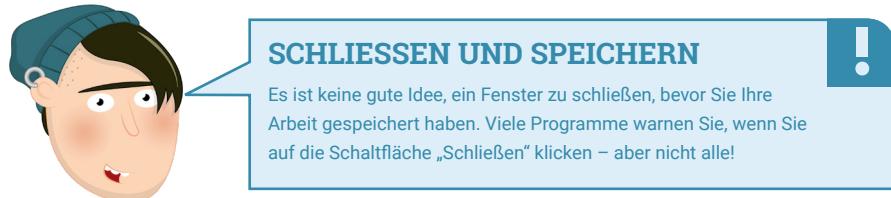
Um den Umgang mit Ihrem Raspberry Pi zu üben, beginnen Sie mit dem Starten des Chromium-Webbrowsers. Klicken Sie auf das Raspberry-Symbol oben links, um das Menü aufzurufen, bewegen Sie den Mauszeiger, um die Internet-Kategorie auszuwählen, und klicken Sie auf den Chromium-Webbrowser, um ihn zu starten (**Abbildung 3-12**).



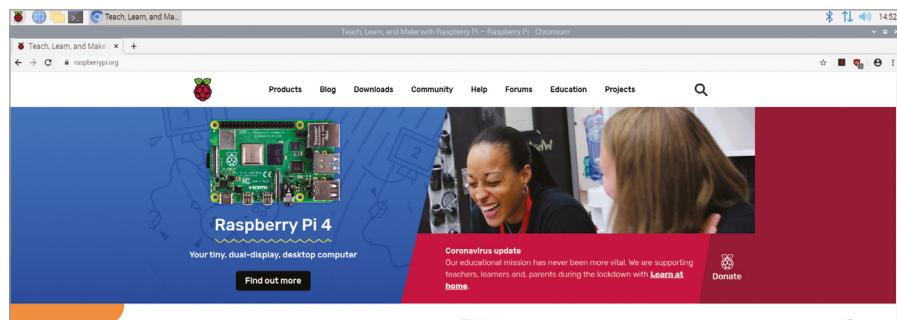
▲ Abbildung 3-12: Der Chromium-Webbrowser

Wenn Sie den Chrome-Browser von Google schon auf einem anderen Computer verwendet haben, wird Ihnen Chromium sofort vertraut sein. Mit dem Webbrowser Chromium können Sie Websites besuchen, Videos und Spiele spielen und sogar mit Menschen auf der ganzen Welt in Foren und Chat-Seiten kommunizieren.

Beginnen Sie mit der Verwendung von Chromium, indem Sie das Fenster maximieren, damit es mehr Platz auf dem Bildschirm einnimmt: In der Titelleiste des Chromium-Fensters befinden sich oben rechts drei Symbole (**N**). Klicken Sie hier auf das mittlere, nach oben gerichtete Pfeilsymbol (**P**). Dies ist der Button **Maximise**, der veranlasst, dass das Fenster den Bildschirm ganz ausfüllt. Links daneben befindet sich der **Verkleinern** (**O**) Button. Er blendet ein Fenster so lange aus bis Sie es in der Taskleiste am oberen Bildschirmrand anklicken. Das Kreuz rechts neben „**Vergrößern**“ ist **Schließen** (**Q**), welches zum Schließen des Fensters dient.



Klicken Sie auf die Adressleiste am oberen Rand des Chromium-Fensters – die große weiße Leiste mit einer Lupe auf der linken Seite – und geben Sie www.raspberrypi.org ein. Dann drücken Sie die **EINGABETASTE** auf Ihrer Tastatur. Die Website des Raspberry Pi wird geladen (**Abbildung 3-13**). Sie können auch Suchanfragen in die Adressleiste eingeben: Versuchen Sie, nach „Raspberry Pi“, „Raspberry Pi OS“ oder „Educational Computing“ zu suchen.



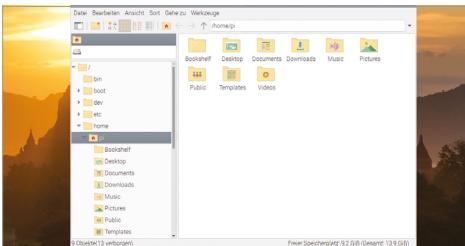
▲ Abbildung 3-13: Laden der Raspberry Pi Webseite in Chromium

Wenn Sie Chromium zum ersten Mal laden, werden möglicherweise mehrere **Tabs** (**Registerkarten**) oben im Fenster angezeigt. Um zu einem anderen Tab zu wechseln, klicken Sie darauf. Um einen Tab zu schließen, ohne Chromium selbst zu schließen, klicken Sie auf das Kreuz am rechten Rand des betreffenden Tabs. Um einen neuen Tab zu öffnen – eine praktische Methode, um mehrere Websites zu öffnen, ohne mit mehreren Chromium-Fenstern jonglieren zu müssen –, klicken Sie entweder auf den Tab-Button (mit dem Plus-Symbol) rechts neben dem letzten Tab in der Liste oder halten Sie die Taste **STRG** auf der Tastatur gedrückt und drücken Sie die Taste **T**, bevor Sie **STRG** wieder loslassen.

Wenn Sie mit Chromium fertig sind, klicken Sie auf den Button zum Schließen oben rechts im Fenster.

Der Dateimanager

Dateien, die Sie speichern möchten – z. B. Programme, Videos, Bilder – werden alle in Ihrem *Benutzerordner* gespeichert. Um den Benutzerordner anzuzeigen, klicken Sie erneut auf das Raspberry-Symbol, um das Menü aufzurufen, bewegen Sie den Mauszeiger, um „Zubehör“ auszuwählen, und klicken Sie dann auf „Dateimanager“, um ihn zu starten (**Abbildung 3-14**).



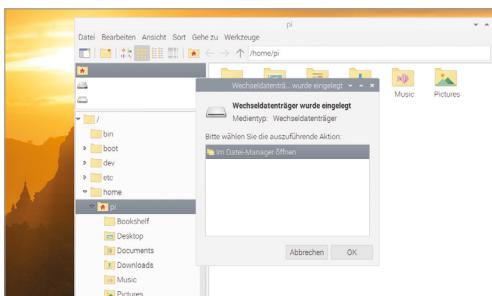
◀ Abbildung 3-14: Der Dateimanager

Mit dem Dateimanager können Sie die Dateien und Ordner, auch bekannt als *Verzeichnisse*, auf der microSD-Karte des Raspberry Pi sowie auf allen Wechseldatenträgern wie USB-Flash-Laufwerken, die Sie an die USB-Ports des Raspberry Pi anschließen, verwalten. Wenn Sie ihn zum ersten Mal öffnen, wird automatisch Ihr Benutzerordner geöffnet. Hier finden Sie eine Reihe weiterer Ordner, bekannt als *Unterverzeichnisse*, die – wie das Menü – in Kategorien gegliedert sind. Die wichtigsten Unterverzeichnisse sind:

- **Bookshelf (Bücherregal):** Dieses Verzeichnis enthält digitale Kopien von Büchern, Zeitschriften und anderen Publikationen von Raspberry Pi Press, einschließlich einer Kopie dieses Buchs, dem *Handbuch für Anfänger* (englische Version). Sie können sie mit Hilfe der Bookshelf-Anwendung lesen und damit auch weitere herunterladen; Sie finden diese Anwendung im Untermenü „Help“ des Menüs.
- **Desktop:** Diesen Ordner sehen Sie, wenn Sie das Raspberry Pi OS starten. Wenn Sie eine Datei hier speichern, erscheint sie auf der Arbeitsoberfläche, sodass sie leicht zu finden und zu öffnen ist.
- **Documents:** Dieser Ordner ist für die meisten der Dateien, die Sie erstellen werden vorgesehen, von Kurzgeschichten bis hin zu Rezepten.
- **Downloads:** Wenn Sie eine Datei mit dem Chromium-Webbrowser aus dem Internet herunterladen, wird sie automatisch unter „Downloads“ gespeichert.
- **Music (Musik):** Musik, die Sie erstellen oder auf Ihren Raspberry Pi laden, kann hier gespeichert werden.
- **Pictures (Bilder):** Dieser Ordner ist speziell für Bilder da, in der Fachsprache bekannt als *image files (Bilddateien)*.

- **Public (Öffentlich):** Während die meisten Ihrer Dateien privat sind, ist alles, was Sie in diesen Ordner „Public“ legen für andere Benutzer des Raspberry Pi zugänglich, selbst wenn diese ihren eigenen Benutzernamen und ihr eigenes Passwort haben.
- **Templates (Vorlagen):** Dieser Ordner enthält alle „templates“ (Vorlagen), die Sie erstellen oder die von Ihren Anwendungen installiert werden. Das sind leere Dateien, die mit einer Grundstruktur (Grund-Layout) ausgestattet sind.
- **Videos:** Ein Ordner für Videos und der erste Ort, an dem die meisten Programme Videos zum Abspielen suchen werden.

Das Dateimanager-Fenster selbst ist in zwei Bereiche aufgeteilt: Der linke Bereich zeigt die Verzeichnisse auf Ihrem Raspberry Pi und der rechte Bereich zeigt die Dateien und Unterverzeichnisse des im linken Bereich ausgewählten Verzeichnisses. Wenn Sie ein Wechseldatenträger an den USB-Port des Raspberry Pis anschließen, öffnet sich ein Fenster, in dem Sie gefragt werden, ob Sie es im Dateimanager öffnen möchten (**Abbildung 3-15**); klicken Sie auf „OK“ um seine Dateien und Verzeichnisse zu sehen.



◀ **Abbildung 3-15:** Anschließen eines Wechseldatenträgers

Dateien können ganz einfach zwischen der microSD-Karte des Raspberry Pis und einem Wechseldatenträger kopiert werden: Öffnen Sie sowohl Ihren Benutzerordner als auch den Wechseldatenträger in separaten Dateimanager-Fenstern, bewegen Sie Ihren Mauszeiger auf die Datei, die Sie kopieren möchten, klicken und halten Sie die linke Maustaste gedrückt, schieben Sie Ihren Mauszeiger in das andere Fenster und lassen Sie die Maustaste los (**Abbildung 3-16** (umseitig)). Dies ist bekannt als *Drag and Drop* ("Ziehen und Ablegen").



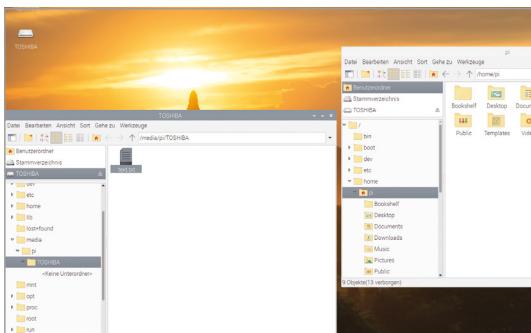
TASTATURKÜRZEL

Wenn Sie ein Tastatkürzel wie **STRG+C** sehen, bedeutet das, dass Sie die erste Taste auf der Tastatur gedrückt halten (**STRG**), dann die zweite Taste drücken (**C**), und dann beide Tasten loslassen.



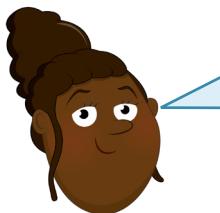
Stattdessen klicken Sie einmal auf die Datei, um sie auszuwählen, dann auf das Menü „Bearbeiten“, dort auf „Kopieren“, und anschließend auf das andere Fenster, auf dessen Menü „Bearbeiten“ und dann auf „Einfügen“.

Die Option „Ausschneiden“, die ebenfalls im Menü „Bearbeiten“ verfügbar ist, funktioniert ähnlich, außer dass sie die Datei nach dem Erstellen der Kopie von ihrem ursprünglichen Speicherort löscht. Beide Optionen können auch über Tastaturkürzel verwendet werden: **STRG+C** (Kopieren) oder **STRG+X** (Ausschneiden) und Einfügen über **STRG+V**.



◀ Abbildung 3-16: Ziehen und Ablegen einer Datei

Wenn Sie Ihre ersten Versuche mit dem Verwalten von Dateien abgeschlossen haben, schließen Sie den Dateimanager, indem Sie auf den Button „Schließen“ oben rechts im Fenster klicken. Wenn Sie mehr als ein Fenster geöffnet haben, schließen Sie sie alle. Wenn Sie einen Wechseldatenträger an Ihren Raspberry Pi angeschlossen haben, werfen Sie ihn aus, indem Sie auf das Auswurf-Symbol oben rechts auf dem Bildschirm klicken, ihn in der Liste suchen und darauf klicken, bevor Sie den Stick entnehmen.



GERÄTE AUSWERFEN

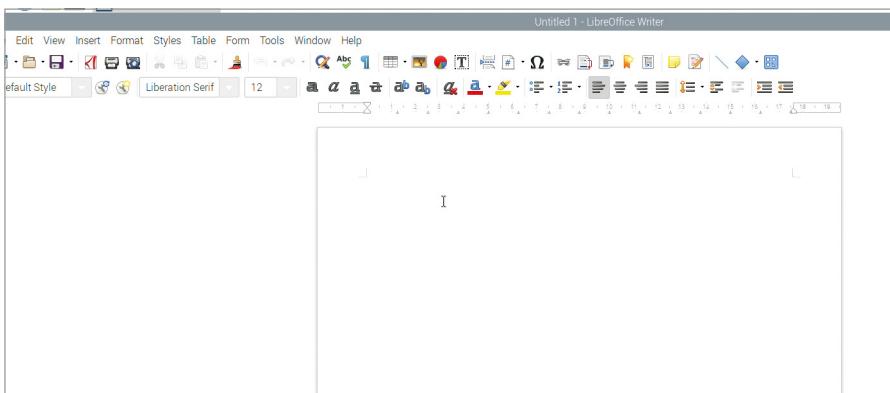
Verwenden Sie immer das Auswurf-Symbol, bevor Sie ein externes Speichergerät ausstecken. Wenn Sie das nicht tun, können die Dateien darauf beschädigt und unbrauchbar werden.



Die LibreOffice-Produktivitäts-Suite

Um einen weiteren Vorgesmack auf die Möglichkeiten eines Raspberry Pi zu erhalten, klicken Sie auf das Raspberry-Menüsymbol, bewegen Sie den Mauszeiger auf „Büro“ und klicken Sie auf LibreOffice Writer. Dadurch wird der Textverarbeitungsteil von LibreOffice geladen (**Abbildung 3-17**), einer beliebten Produktivitäts-Suite. Wenn Sie bereits Microsoft Office oder Google Docs verwenden haben, sind Sie schon mit einer Produktivitätssuite bekannt.

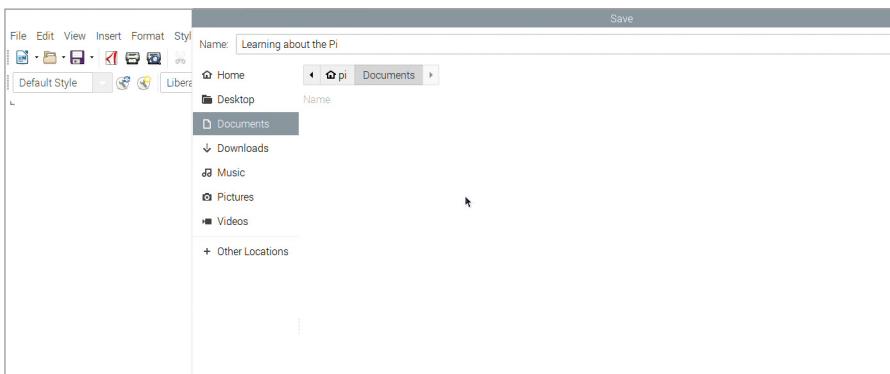
Anmerkung: LibreOffice ist möglicherweise nicht standardmäßig auf allen Raspberry Pi OS-Images installiert; falls nicht, verwenden Sie das Werkzeug Recommended Software (siehe Seite 43), um es zu installieren.



▲ Abbildung 3-17: Das LibreOffice Writer-Programm

Mit einem Textverarbeitungsprogramm können Sie Dokumente nicht nur schreiben, sondern auch auf vielseitige Weise formatieren: Sie können die Schriftart, die Farbe und Größe der Schrift ändern, Effekte hinzufügen und sogar Bilder, Diagramme, Tabellen und andere Inhalte einfügen. Mit einem Textverarbeitungsprogramm können Sie Ihre Arbeit auch auf Fehler überprüfen, wobei Rechtschreib- und Grammatikfehler während der Eingabe rot bzw. grün hervorgehoben werden.

Beginnen Sie damit, einen Absatz darüber zu schreiben, was Sie bisher über den Raspberry Pi und seine Software gelernt haben. Experimentieren Sie mit den verschiedenen Symbolen am oberen Rand des Fensters, um zu sehen, was sie bewirken: Versuchen Sie, Ihre Schrift größer zu machen und ihre Farbe zu ändern. Wenn Sie sich nicht sicher sind, wie Sie das machen können, bewegen Sie einfach den Mauszeiger nacheinander über jedes Symbol, um einen „Tooltip“ zu sehen, der Ihnen sagt, was dieses Symbol bewirkt. Wenn Sie zufrieden sind, klicken Sie auf das Menü „File“ und die Option „Save“, um Ihre Arbeit zu speichern. (**Abbildung 3-18**). Geben Sie Ihrem Dokument einen Namen und klicken Sie auf den Button „Save“.



▲ Abbildung 3-18: Speichern eines Dokuments



SPEICHERN SIE IHRE ARBEIT



Gewöhnen Sie sich an, Ihre Arbeit immer wieder zu speichern, auch wenn Sie sie noch nicht abgeschlossen haben. Es wird Ihnen viel Ärger ersparen, falls es zum Beispiel einen Stromausfall gibt und Ihr Arbeitsfortschritt dadurch verloren geht!

LibreOffice Writer ist nur ein Teil der gesamten LibreOffice-Produktivitätssuite. Die anderen Teile, die Sie in der gleichen Büro-Menükategorie wie LibreOffice Writer finden, sind folgende:

- **LibreOffice-Base:** Eine Datenbank: ein Werkzeug zum Speichern, schnellen Nachschlagen und Analysieren von Informationen.
- **LibreOffice Calc:** Eine Tabellenkalkulation: ein Werkzeug zur Handhabung von Zahlen und zur Erstellung von Diagrammen und Grafiken.
- **LibreOffice Draw:** Ein Illustrationsprogramm: ein Werkzeug zum Erstellen von Bildern und Diagrammen.
- **LibreOffice Impress:** Ein Präsentationsprogramm, zur Erstellung von Folien und dem Präsentieren von Slideshows.
- **LibreOffice Math:** Ein Formeleditor: ein Werkzeug zum Erstellen korrekt formatierter mathematischer Formeln, die dann in anderen Dokumenten verwendet werden können.

LibreOffice ist auch für andere Computer und Betriebssysteme verfügbar. Falls Ihnen die Produktivitätssuite gefällt und Sie sie gerne auf Ihrem Raspberry Pi einsetzen, können Sie sie kostenfrei von libreoffice.org herunterladen und auch auf einem beliebigen Microsoft Windows-, Apple macOS- oder Linux-Computer installieren.

Wenn Sie mehr über die Verwendung von LibreOffice erfahren möchten, klicken Sie auf das Help-Menü. Andernfalls schließen Sie LibreOffice Writer, indem Sie auf den Button zum Schließen oben rechts im Fenster klicken.



EINGEBAUTE HILFE



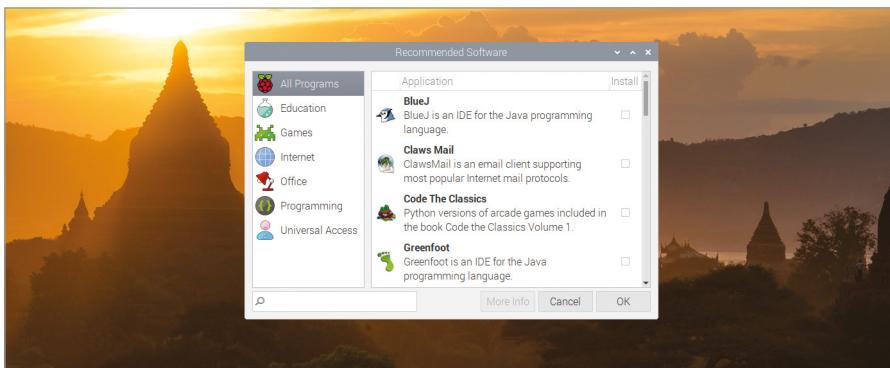
Die meisten Programme enthalten ein Hilfemenü, das Sie unterstützt: von Informationen über das Programm bis hin zu Anleitungen zur Benutzung des Programms. Wenn Sie sich jemals von einem Programm überfordert fühlen, oder nicht zurechtfinden, schauen Sie im Hilfe-Menü nach, um sich neu zu orientieren.

Das Werkzeug Recommended Software

Obwohl Raspberry Pi OS bereits mit einer großen Auswahl an vorinstallierter Software geliefert wird, ist es mit noch weit mehr Software kompatibel. Eine Auswahl der besten Software finden Sie im Werkzeug „Recommended Software“.

Beachten Sie, dass das Werkzeug „Recommended Software“ eine Verbindung zum Internet benötigt. Wenn Ihr Raspberry Pi damit verbunden ist, klicken Sie auf das Raspberry-Menüsymbol, bewegen Sie den Mauszeiger auf „Einstellungen“ und klicken auf „Recommended Software“. Das Werkzeug startet und beginnt dann mit dem Herunterladen von Informationen über verfügbare Software.

Nach einigen Sekunden erscheint eine Liste mit kompatiblen Softwarepaketen (**Abbildung 3-19**). Diese sind, wie die Software im Raspberry-Menü, in verschiedene Kategorien eingeteilt. Klicken Sie auf eine Kategorie in der linken Leiste, um Software aus dieser Kategorie anzuzeigen, oder klicken Sie auf „All Programs“, um alles zu sehen.

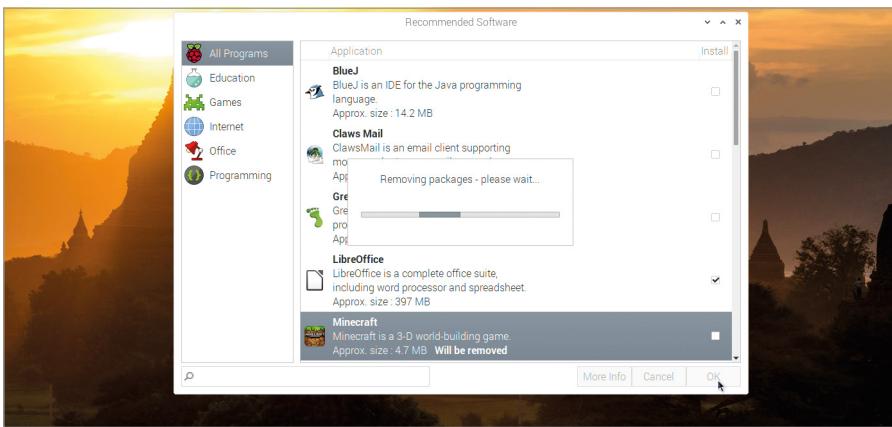


▲ Abbildung 3-19: Das Werkzeug Recommended Software

Wenn ein Softwareeintrag mit einem Häkchen versehen ist, ist sie bereits auf Ihrem Raspberry Pi installiert. Wenn nicht, können Sie die Checkbox daneben anklicken, um ein Häkchen hinzuzufügen und damit die Software zur Installation zu markieren. Sie können beliebig viele Softwareeinträge markieren, um sie anschließend alle auf einmal zu installieren, aber wenn Sie eine kleinere als die empfohlene microSD-Karte verwenden, haben Sie möglicherweise nicht genug Platz für alle.

Sie können Software auch auf die gleiche Weise deinstallieren: Suchen Sie eine Software, die bereits ein Häkchen in ihrer Checkbox hat, und klicken Sie dann auf das Häkchen, um sie zu entfernen. Wenn Sie einen Fehler gemacht oder sich anders entschlossen haben, klicken Sie einfach erneut, um das Häkchen wieder zu setzen.

Wenn Sie mit Ihrer Softwareauswahl zufrieden sind, klicken Sie auf den Button „Apply“, um den Installations- oder Deinstallationsprozess zu starten (**Abbildung 3-20** (umseitig)). Nach dem Herunterladen und Installieren der neuen Software, die Sie ausgewählt haben, erscheint ein Dialogfeld; klicken Sie auf „OK“, um das Werkzeug „Recommended Software“ zu schließen.

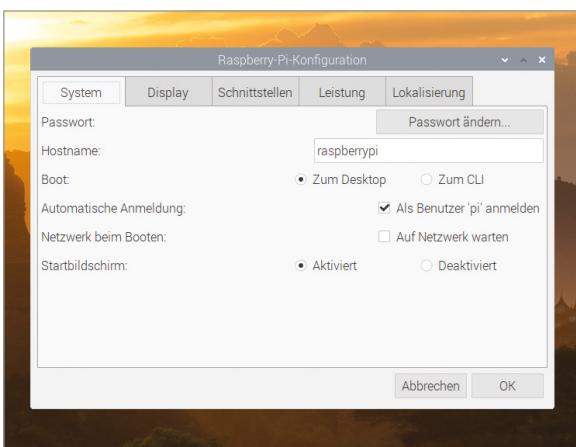


▲ Abbildung 3-20: Software deinstallieren

Ein zusätzliches Werkzeug zum Installieren oder Deinstallieren von Software, das Software-Werkzeug „Add/Remove Software“, finden Sie in der gleichen Kategorie „Einstellungen“ des Raspberry Pi OS-Menüs. Dies bietet eine größere Auswahl an Software, die jedoch nicht von der Raspberry Pi Foundation überprüft wurde.

Raspberry Pi-Konfigurationswerkzeug

Das letzte Programm, das Sie in diesem Kapitel kennenlernen werden, ist als das Raspberry Pi-Konfigurationswerkzeug bekannt, und es ähnelt dem Begrüßungsassistenten, den Sie zu Beginn verwendet haben: Es ermöglicht Ihnen, verschiedene Einstellungen in Raspberry Pi OS zu ändern. Klicken Sie auf das Raspberry-Symbol, bewegen Sie den Mauszeiger, um die Kategorie „Einstellungen“ auszuwählen, und klicken Sie dann auf „Raspberry Pi-Konfiguration“, um es zu starten(**Abbildung 3-21**).



◀ Abbildung 3-21:
Das Raspberry Pi-Konfigurationswerkzeug

Das Werkzeug ist in fünf Registerkarten unterteilt. Die erste davon ist „System“: Damit können Sie das Passwort Ihres Kontos ändern, einen Hostnamen festlegen – den Namen, den der Raspberry Pi in Ihrem lokalen drahtlosen oder drahtgebundenen Netzwerk verwendet – und eine Reihe anderer Einstellungen ändern. Die meisten davon sollten jedoch für die meisten Nutzer nicht geändert werden müssen. Klicken Sie auf die Registerkarte „Display“, um die nächste Kategorie aufzurufen. Hier können Sie bei Bedarf die Einstellungen für die Bildschirmanzeige ändern, um sie an Ihren Fernseher oder Monitor anzupassen.



MEHR DETAILS



Dieser kurze Überblick dient lediglich dazu, damit Sie anfangen können sich mit dem Werkzeug vertraut zu machen. Ausführlichere Informationen über jede Einstellung finden Sie in [Anhang E, Das Raspberry Pi-Konfigurationswerkzeug](#).

Die Registerkarte „Schnittstellen“ bietet eine Reihe von Einstellungen, die zunächst alle deaktiviert sind. Diese Einstellungen sollten nur geändert werden, wenn Sie neue Hardware, wie z. B. das Raspberry Pi Camera Module (Kameramodul), hinzufügen, und auch dann nur, wenn der Hersteller der Hardware Sie dazu anweist. Die Ausnahmen von dieser Regel sind: SSH, das eine „Secure Shell“ ermöglicht und mit dem Sie sich von einem anderen Computer in Ihrem Netzwerk aus über einen SSH-Client auf Ihrem Raspberry Pi anmelden können; VNC, das einen „Virtual Network Computer“ ermöglicht und mit dem Sie den Desktop des Raspberry Pi OS von einem anderen Computer in Ihrem Netzwerk aus über einen VNC-Client sehen und steuern können; und Remote GPIO, mit dem Sie die GPIO-Pins Ihres Raspberry Pi von einem anderen Computer in Ihrem Netzwerk aus verwenden können – mehr darüber erfahren Sie in [Kapitel 6, Physical Computing mit Scratch und Python](#).

Klicken Sie auf die Registerkarte „Leistung“, um die vierte Kategorie anzuzeigen. Hier können Sie die von der Grafikeinheit (GPU) vom Raspberry Pi verwendete Speichermenge einstellen und bei einigen Modellen die Leistung des Raspberry Pis durch einen Prozess erhöhen, der als *Overclocking* (Übertakten) bekannt ist. Wie bereits gesagt ist es jedoch am besten, diese Einstellungen nicht zu ändern, es sei denn, Sie wissen, dass Sie sie ändern müssen.



WARNUNG!



Verschiedene Länder haben unterschiedliche Regeln, welche Frequenzen ein WLAN verwenden kann. Wenn Sie das WLAN-Land im Raspberry Pi-Konfigurationswerkzeug auf ein anderes Land einstellen als das, in dem Sie sich tatsächlich befinden, wird es wahrscheinlich schwieriger sein, eine Verbindung zu Ihren Funknetzwerken herzustellen. Es kann sogar aufgrund der gesetzlichen Vorschriften für Funkfrequenzzuteilungen illegal sein – also tun Sie es nicht!

Dann klicken Sie auf die Registerkarte „Lokalisierung“, um die letzte Kategorie anzuzeigen. Hier können Sie Ihre Sprachumgebung ändern, die Dinge wie die in Raspberry Pi OS verwendete Sprache und die Anzeige von Zahlen steuert, die Zeitzone ändern, das Tastaturlayout ändern und Ihr Land für WLAN-Zwecke einstellen („WiFi-Land festlegen“). Für den Moment genügt es jedoch, auf „Abbrechen“ zu klicken, um das Werkzeug ohne Änderungen zu schließen.

Herunterfahren

Jetzt, da Sie die Arbeitsoberfläche des Raspberry Pi OS erkundet haben, ist es an der Zeit, zu erfahren, wie Sie Ihren Raspberry Pi sicher herunterfahren. Wie jeder Computer bewahrt Ihr Raspberry Pi die Dateien, an denen Sie gerade arbeiten, im *flüchtigen Speicher*(RAM, oder „*volatile memory* “auf Englisch) auf – einem Speicher, der beim Ausschalten des Systems geleert wird. Bei Dokumenten, die Sie erstellen, reicht es aus, jedes Dokument der Reihe nach zu speichern. Dadurch wird die Datei aus dem flüchtigen Speicher im *nichtflüchtigen Speicher*, auf der microSD-Karte, gespeichert, um sicherzustellen, dass nichts verloren geht.

Die Dokumente, an denen Sie arbeiten, sind jedoch nicht die einzigen geöffneten Dateien. Das Raspberry Pi OS selbst hält eine Reihe von Dateien offen, während es läuft. Das Ziehen des Stromkabels von Ihrem Raspberry Pi, während diese Dateien noch offen sind, kann dazu führen, dass das Betriebssystem beschädigt wird und neu installiert werden muss.

Um dies zu verhindern, müssen Sie das Raspberry Pi OS anwisen, alle seine Dateien zu speichern und sich für das Ausschalten vorzubereiten – ein Prozess, der als *Herunterfahren* des Betriebssystems bezeichnet wird.

Klicken Sie auf das Raspberry-Symbol oben links auf dem Desktop und klicken Sie dann auf „Shutdown“. Es erscheint ein Fenster mit drei Optionen (**Abbildung 3-22**): „Herunterfahren“, „Neu starten“ und „Abmelden“. „Herunterfahren“ ist die Option, die Sie am häufigsten verwenden werden: Wenn Sie darauf klicken, wird Raspberry Pi OS angewiesen, alle geöffneten Programme und Dateien zu schließen und den Raspberry Pi dann herunterzufahren. Sobald die Anzeige schwarz geworden ist, warten Sie ein paar Sekunden, bis das blinkende grüne LED-Licht auf dem Raspberry Pi erlischt; dann können Sie die Stromversorgung gefahrlos abziehen.

Um den Raspberry Pi nach dem Herunterfahren wieder einzuschalten, stecken Sie das Stromkabel aus und wieder ein, oder schalten Sie das Netzteil aus und wieder ein.



▲ Abbildung 3-22: Herunterfahren des Raspberry Pi

Der Neustart durchläuft einen ähnlichen Prozess wie das Herunterfahren und schließt alles. Aber anstatt den Raspberry Pi auszuschalten, startet es ihn neu – fast genauso, als hätten Sie „Herunterfahren“ gewählt und dann das Stromkabel abgezogen und wieder angeschlossen. Sie müssen „Neu starten“ verwenden, wenn Sie bestimmte Änderungen vornehmen, die einen Neustart des Betriebssystems erfordern – wie z. B. das Installieren bestimmter Aktualisierungen der Kern-Software – oder wenn Programme abgestürzt sind, bekannt als *Crash*, und das Raspberry Pi OS nicht mehr ansprechbar ist.

Die letzte Option, „Abmelden“ ist nur dann wirklich sinnvoll, wenn Sie mehr als ein Benutzerkonto auf Ihrem Raspberry Pi haben: Es schließt alle Programme, die Sie derzeit geöffnet haben, und führt Sie zu einem Anmeldebildschirm, auf dem Sie zur Eingabe eines Benutzernamens und Passworts aufgefordert werden. Wenn Sie versehentlich auf „Abmelden“ klicken und sich wieder anmelden möchten, geben Sie einfach „pi“ als Benutzernamen und das Passwort ein, das Sie im Begrüßungsassistenten am Anfang dieses Kapitels gewählt haben.



WARNUNG!

Ziehen Sie niemals das Stromkabel von einem Raspberry Pi ab, ohne das Gerät vorher abzuschalten. Das würde wahrscheinlich das Betriebssystem beschädigen und Sie könnten alle Dateien, die Sie erstellt oder heruntergeladen haben verlieren.



Kapitel 4

Programmieren mit Scratch 3

Ein einfacher Einstieg ins Programmieren mit Scratch, der blockbasierten Programmiersprache



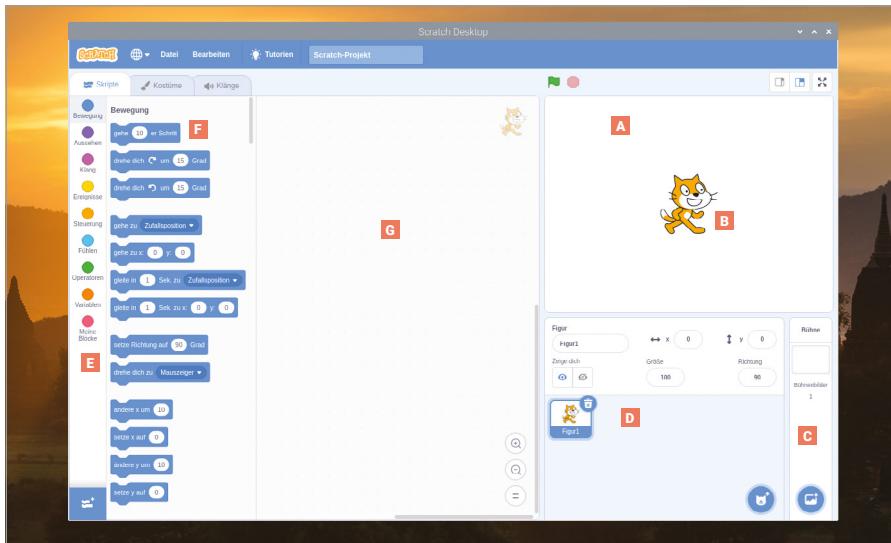
Bei der Verwendung des Raspberry Pi geht es nicht nur darum, Software zu verwenden, die andere Leute erstellt haben; es geht darum, Ihre eigene Software zu erschaffen, basierend auf fast allem, was Ihre Vorstellungskraft heraufbeschwören kann.

Unabhängig davon, ob Sie bereits Erfahrung mit der Erstellung eigener Programme haben – ein Prozess, der als Programmieren oder Coden bekannt ist – oder nicht, Sie werden entdecken, dass der Raspberry Pi eine großartige Plattform zum Erstellen und Experimentieren ist.

Der Schlüssel zur leichten Erlernbarkeit der Programmierung auf dem Raspberry Pi ist Scratch, eine visuelle Programmiersprache, die vom MIT entwickelt wurde. Während Sie in traditionellen Programmiersprachen textbasierte Anweisungen für den Computer schreiben müssen, um sie auszuführen – ähnlich wie Sie ein Rezept zum Kuchenbacken schreiben könnten – können Sie bei Scratch Ihr Programm Schritt für Schritt mit Blöcken aufbauen – vorprogrammierten Code-Stückchen, die sich in farbcodierten Puzzleteilen verbergen.

Scratch ist eine großartige erste Programmiersprache für angehende Programmierer, ob jung oder alt, aber lassen Sie sich nicht von seinem freundlichen Aussehen täuschen: Es ist eine leistungsstarke und voll funktionsfähige Programmierumgebung für die Erstellung von einfachen Spielen und Animationen bis hin zu komplexen, interaktiven Robotikprojekten.

Einführung in die Scratch 3-Nutzerschnittstelle



A Bühnenbereich – Wie Schauspieler in einem Theaterstück bewegen sich Ihre Sprites unter der Kontrolle Ihres Programms auf der Bühne.

B Sprite – Die Charaktere oder Objekte, die Sie in einem Scratch-Programm steuern, werden als Sprites bezeichnet und befinden sich auf der Bühne.

C Bühnensteuerung – Ihre Bühne kann über die Bühnensteuerung verändert werden, einschließlich des Hinzufügens eigener Bilder als Hintergründe.

D Sprites-Liste – Alle Sprites, die Sie erstellt oder in Scratch geladen haben, werden in diesem Abschnitt des Fensters angezeigt.

E Blockpalette – Alle für Ihr Programm verfügbaren Blöcke erscheinen in der Blockpalette, die farbkodierte Kategorien enthält.

SCRATCH-VERSIONEN

Zum Zeitpunkt, als dieses Buch geschrieben wurde, kamen drei Versionen von Scratch mit Raspberry Pi OS: Scratch 1, 2 und 3 sind alle im Untermenü „Entwicklung“ des Raspberry Pi OS-Menüs enthalten. Dieses Kapitel ist für Scratch 3 geschrieben. Beachten Sie, dass Scratch 3 nur auf dem Raspberry Pi 4 läuft. Scratch 2, als mögliche Alternative, läuft nicht auf Raspberry Pi Zero, Modell A, A+, B oder B+.

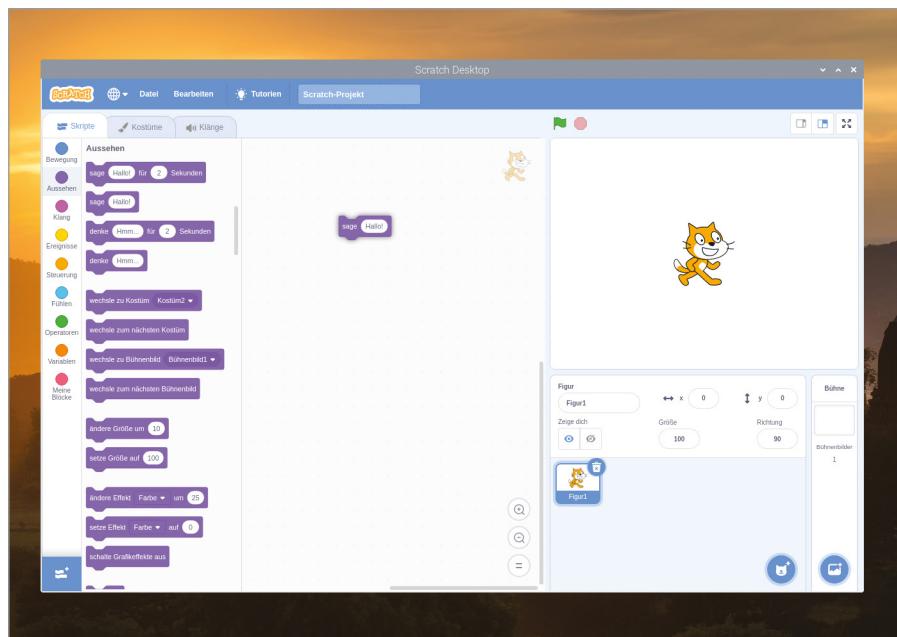
F Blöcke – Bereits programmierte Code-Blöcke ermöglichen es Ihnen, Ihr Programm Schritt für Schritt aufzubauen.

G Code-Bereich – Der Code-Bereich ist der Bereich, in dem Ihr Programm durch Ziehen und Ablegen von Blöcken aus der Blockpalette zur Bildung von Skripten erstellt wird.

Ihr erstes Scratch-Programm: Hallo, Welt!

Scratch 3 wird wie jedes andere Programm auf Raspberry Pi gestartet: Klicken Sie auf das Raspberry-Symbol, um das Raspberry Pi OS-Menü zu öffnen, zeigen Sie mit der Maus auf den Abschnitt „Entwicklung“ und klicken Sie auf „Scratch 3“. Nach einigen Sekunden wird die Benutzeroberfläche von Scratch 3 geladen.

Während man bei den meisten Programmiersprachen dem Computer durch Textbefehle mitteilen muss, was er tun soll, ist Scratch anders. Beginnen Sie mit einem Klick auf die Kategorie „Aussehen“ in der Blöcke-Palette, die sich links im Scratch-Fenster befindet. Dadurch werden die lilafarbenen Blöcke in dieser Kategorie angezeigt. Suchen Sie den Block **sage Hallo!**, klicken Sie darauf, halten Sie die linke Maustaste gedrückt und ziehen Sie ihn in den Code-Bereich in der Mitte des Scratch-Fensters. Lassen Sie dann die Maustaste wieder los (**Abbildung 4-1**).



▲ Abbildung 4-1: Ziehen Sie den Block per Drag & Drop in den Code-Bereich

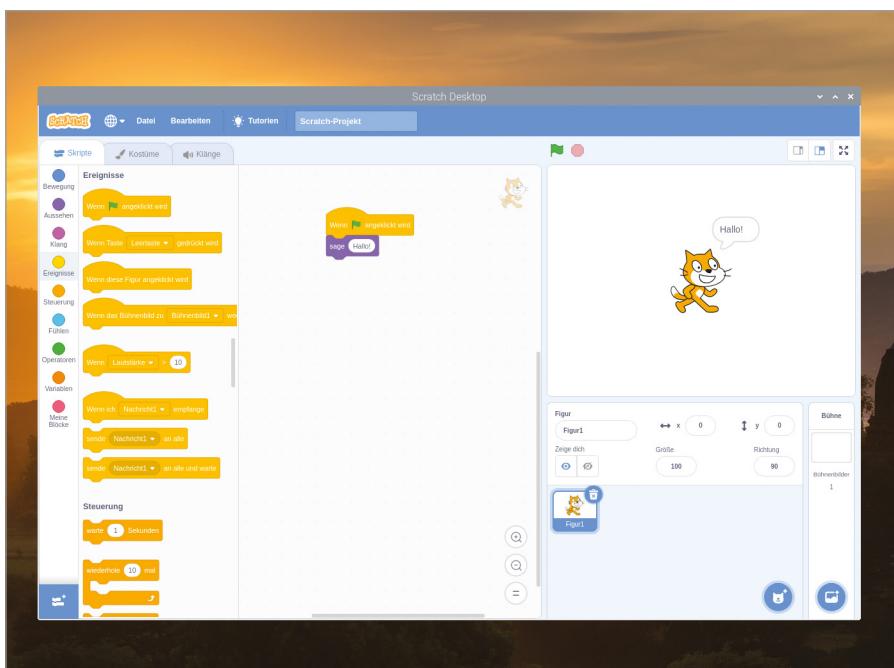
Schauen Sie die Form des Blocks an, den Sie gerade abgelegt haben: Er hat oben eine Einbuchtung und unten eine dazu passende Ausbuchtung. Wie bei einem Puzzleteil zeigt das Ihnen, dass man an diesen Block oben und unten einen anderen Block anfügen kann. Bei diesem Programm ist dieser obere Block ein *Trigger*.

Klicken Sie in der Block-Palette auf die goldfarbene Kategorie „Ereignisse“ und ziehen Sie den Block **Wenn [Flag angeklickt wird]**, (auch *hat-Block* genannt) in den Code-Bereich. Positionieren Sie ihn so, dass die untere Ausbuchtung dieses Blocks in die obere Einbuchtung

des Blocks **sage Hallo!** hereinpasst, und ein weißer Umriss erscheint. Lassen Sie dann die Maustaste los. Sie müssen dabei nicht besonders präzise sein: Wenn der Block nahe genug beim anderen ist, rastet er wie ein Puzzleteil ein. Wenn das nicht passiert, klicken Sie erneut auf den Block, halten Sie die Maustaste gedrückt und verschieben Sie ihn, bis er einrastet.

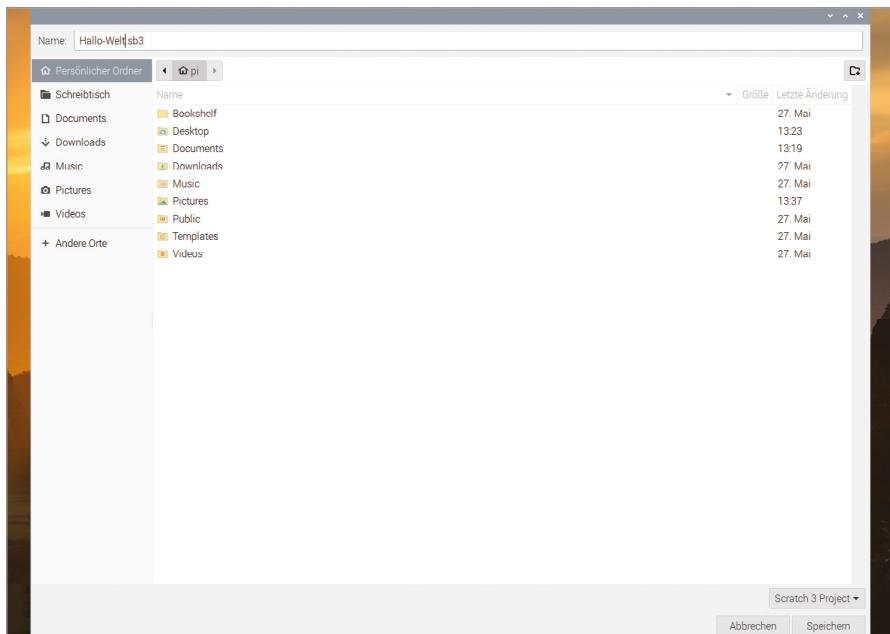


Ihr Programm ist jetzt fertig. Um es *auszuführen*, klicken Sie auf das grüne Flaggensymbol oben links im Bühnenbereich. Wenn alles funktioniert hat, begrüßt Sie die Katze auf der Bühne mit einem fröhlichen „Hallo!“ (**Abbildung 4-2**) – Ihr erstes Programm wurde erfolgreich ausgeführt!



▲ Abbildung 4-2: Klicken Sie auf die grüne Flagge über der Bühne und die Katze sagt „Hallo“

Bevor Sie weitermachen, sollten Sie Ihrem Programm einen Namen geben und es speichern. Klicken Sie auf das Menü „Datei“ und dann auf „Auf deinem Computer speichern“. Geben Sie einen Namen ein und klicken Sie auf die Schaltfläche „Speichern“ (**Abbildung 4-3**).



▲ Abbildung 4-3: Speichern Sie Ihr Programm mit einem einprägsamen Namen



WAS KANN SIE SAGEN?

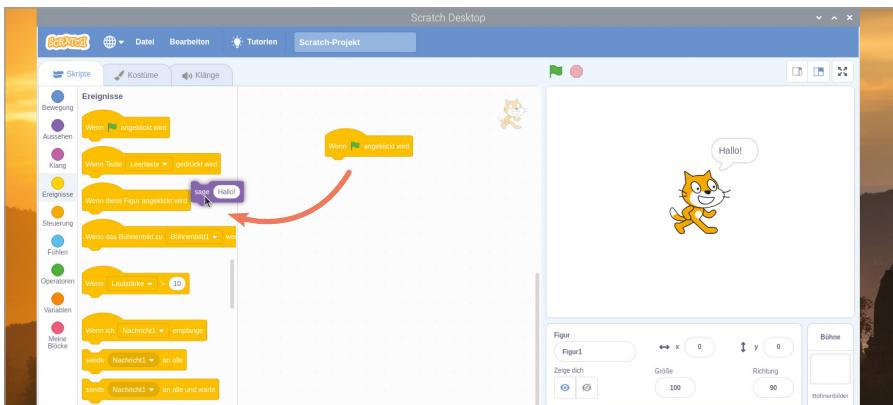


Einige Blöcke in Scratch können geändert werden. Versuchen Sie, auf das Wort „Hallo!“ zu klicken und etwas anderes einzugeben, und klicken Sie dann erneut auf die grüne Flagge. Was passiert auf der Bühne?

Nächste Schritte: Abfolgen

Ihr Programm hat zwar zwei Blöcke, aber nur eine echte Anweisung: Jedes Mal, wenn die Flagge angeklickt wird und das Programm ausgeführt wird, soll die Katze „Hallo“ sagen. Wenn Sie weitere Aktionen ausführen wollen, müssen Sie sich mit *Abfolgen* auskennen. Computerprogramme sind im Grunde nur eine Liste von Anweisungen, genau wie ein Rezept. Die Anweisungen werden in einer logischen Reihenfolge ausgeführt, die als *lineare Abfolge* bezeichnet wird.

Klicken Sie zuerst auf den Block **sage Hallo!** und ziehen Sie ihn aus dem Code-Bereich zurück in die Blöcke-Palette (**Abbildung 4-4**). Dadurch wird der Block gelöscht und aus Ihrem Programm entfernt. Jetzt ist nur noch der Trigger-Block **Wenn ♡ angeklickt wird** übrig.



▲ Abbildung 4-4: Um einen Block zu löschen, ziehen Sie ihn einfach aus dem Code-Bereich heraus

Klicken Sie auf die Kategorie „Bewegung“ in der Blockpalette, klicken Sie dann auf den Block **gehe 10er Schritt** und ziehen Sie ihn so, dass er unter dem Trigger-Block im Code-Bereich einrastet. Wie der Name schon sagt, wird Ihr Sprite – die Katze – dadurch angewiesen, sich eine Reihe von Schritten in die Richtung zu, in die es gerade ausgerichtet ist..



Fügen Sie weitere Anweisungen zu Ihrem Programm hinzu, um eine Abfolge zu erstellen. Klicken Sie auf die Klang-Palette, die rosa eingefärbt ist, klicken und ziehen Sie dann den Block **spiele Klang Miau ganz** sodass er unter dem Block **gehe 10er Schritt** einrastet. Machen Sie weiter: Klicken Sie zurück auf die Kategorie „Bewegung“ und ziehen Sie einen weiteren Block **gehe 10er Schritt** unter Ihren Klang-Block, aber dieses Mal klicken Sie auf die „-10“, um sie auszuwählen, und geben Sie „-10“ ein, um einen Block **gehe -10er Schritt** zu erstellen.



Klicken Sie auf die grüne Flagge über der Bühne, um das Programm auszuführen. Sie sehen, wie die Katze sich zunächst nach rechts bewegt und ein miauendes Geräusch macht. Stellen Sie sicher, dass Sie Lautsprecher oder Kopfhörer angeschlossen haben, um sie zu hören! Anschließend geht die Katze wieder zum Anfang zurück. Klicken Sie erneut auf die Flagge, und die Katze wiederholt ihre Handlungen.

Herzlichen Glückwunsch: Sie haben eine Abfolge von Anweisungen erstellt, die Scratch von oben nach unten nacheinander durchläuft. Obwohl Scratch nur jeweils eine Anweisung aus der Sequenz ausführt, geschieht dies sehr schnell: Versuchen Sie, den Block **spiele Klang Miau ganz** zu löschen, indem Sie den unteren Block **gehe -10er Schritt** anklicken und ziehen, um ihn zu lösen. Ziehen Sie jetzt den Block **spiele Klang Miau ganz** in die Blockpalette und ersetzen Sie ihn durch den einfacheren Block **spiele Klang Miau**. Ziehen Sie anschließend Ihren Block **gehe -10er Schritt** wieder an das untere Ende Ihres Programms.



Klicken Sie auf die grüne Flagge, um Ihr Programm erneut auszuführen, und das Katzen-Sprite scheint sich nicht zu bewegen. In Wirklichkeit bewegt sich das Sprite, aber es bewegt sich so schnell wieder zurück, dass es still zu stehen scheint. Der Grund dafür ist, dass bei Verwendung des Blocks **spiele Klang Miau** nicht darauf gewartet wird, dass der Klang zu Ende abgespielt wird, bevor der nächste Schritt ausgeführt wird; weil der Raspberry Pi so schnell „denkt“, läuft die nächste Anweisung ab, bevor Sie überhaupt sehen können, wie sich das Katzen-Sprite bewegt hat. Es gibt noch eine andere Möglichkeit, dies zu beheben, außer die Verwendung des Blocks **spiele Klang Miau ganz**: Klicken Sie auf die hellorange gefarbene Kategorie „Steuerung“ der Blockpalette. Dann klicken Sie auf den Block **warte 1 Sekunden** und ziehen ihn zwischen den Block **spiele Klang Miau** und den unteren Block **gehe -10er Schritt**.



Klicken Sie auf die grüne Flagge, um Ihr Programm ein letztes Mal auszuführen. Sie werden sehen, dass das Katzen-Sprite eine Sekunde wartet, nachdem es sich nach rechts bewegt hat, bevor es sich wieder nach links bewegt. Dies ist bekannt als **Verzögerung** und ist der wesentliche Faktor um zu steuern, wie lange die Ausführung Ihrer Befehlsabfolge dauert.



HERAUSFORDERUNG: WEITERE SCHRITTE HINZUFÜGEN



Versuchen Sie, weitere Schritte zu Ihrer Abfolge hinzuzufügen und die Werte in den vorhandenen Schritten zu ändern. Was passiert, wenn die Anzahl der Schritte in einem „gehe“-Block nicht mit der Anzahl der Schritte in einem anderen übereinstimmt? Was passiert, wenn Sie versuchen, einen Klang zu spielen, während ein anderer Klang noch gespielt wird?

Schleifen der Schleife

Die Abfolge, die Sie bisher erstellt haben, läuft nur einmal: Sie klicken auf die grüne Flagge, das Katzen-Sprite bewegt sich und miaut, und dann stoppt das Programm, bis Sie erneut auf die grüne Flagge klicken. Es muss jedoch nicht unbedingt aufhören, denn Scratch enthält eine Art Steuerungsblock, der **Schleife** genannt wird.

Klicken Sie auf die Kategorie „Steuerung“ in der Blockpalette und suchen Sie den Block **wiederhole fortlaufend**. Klicken Sie darauf und ziehen Sie ihn in den Code-Bereich, platzieren Sie ihn dann unter dem Block **Wenn Flagge angeklickt wird** und über dem Block **gehe 10er Schritt**.



Beachten Sie, wie der C-förmige Block **wiederhole fortlaufend** größer wird um die anderen Blöcke in Ihrer Abfolge zu umgeben. Klicken Sie jetzt auf die grüne Flagge und Sie können so direkt erleben, was der Block **wiederhole fortlaufend** bewirkt: Anstatt dass Ihr Programm einmal läuft und dann endet, läuft es immer und immer wieder – im wahrsten Sinne des Wortes „wiederhole fortlaufend“. In der Programmierung wird dies als *unendliche Schleife* bezeichnet – buchstäblich eine Schleife, die niemals endet.

Wenn das Geräusch des ständigen Miauens überhand nimmt, klicken Sie auf das rote Achteck neben der grünen Flagge über dem Bühnenbereich, um Ihr Programm zu stoppen. Um den Schleifentyp zu ändern, klicken und ziehen Sie den ersten Block **gehe 10er Schritt** und die Blöcke darunter aus dem Block **wiederhole fortlaufend** und platzieren Sie sie dann unter dem Block **Wenn grün angeklickt wird**. Klicken Sie auf den Block **wiederhole fortlaufend** und ziehen Sie ihn in die Blockpalette, um ihn zu löschen. Klicken und ziehen Sie dann den Block **wiederhole 10 mal** unter den Block **Wenn grün angeklickt wird**, sodass er die anderen Blöcke umgibt.



Klicken Sie auf die grüne Flagge, um Ihr neues Programm auszuführen. Auf den ersten Blick scheint es dasselbe wie Ihre ursprüngliche Version zu tun: Ihre Anweisungsabfolge immer und immer wieder zu wiederholen. Diesmal wird die Schleife jedoch nicht ewig fortgesetzt, sondern nach zehn Wiederholungen beendet. Dies ist bekannt als *endliche Schleife*: Sie bestimmen, wann sie beendet wird. Schleifen sind mächtige Hilfsmittel, und die meisten Programme – insbesondere Spiele und Messprogramme (die zum Beispiel Blöcke aus der Kategorie „Fühlen“ verwenden) – machen sowohl von unendlichen als auch endlichen Schleifen starken Gebrauch.



WAS PASSIERT WENN...?

Was passiert, wenn Sie die Zahl im „Loop“-Block ändern, um sie zu vergrößern? Was passiert, wenn sie kleiner ist? Was passiert, wenn Sie die Zahl 0 in den „Loop“-Block eingeben?

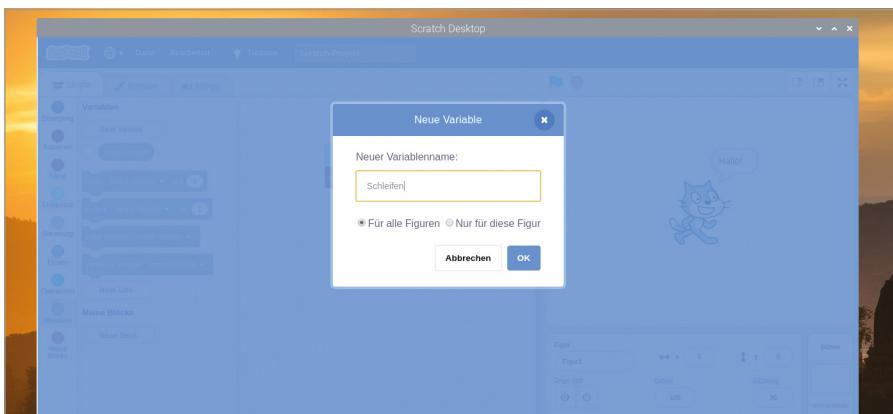


Variablen und Bedingungen

Die abschließenden Konzepte, die es zu verstehen gilt, bevor man ernsthaft mit der Programmierung von Scratch-Programmen beginnt, sind eng miteinander verbunden *Variablen* und *Bedingungen*. Eine Variable ist, wie der Name schon sagt, ein Wert, der sich im Lauf der Zeit und vom Programm gesteuert verändern kann – mit anderen Worten: veränderlich. Eine Variable hat zwei Haupteigenschaften: ihren Namen und den Wert, den sie speichert. Dieser Wert muss auch nicht unbedingt eine Zahl sein: die Variable kann Zahlen, Text, die Zustände wahr oder falsch enthalten oder auch völlig leer sein – bekannt als *Nullwert*.

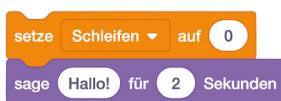
Variablen sind mächtig und wichtig. Denken Sie an die Dinge, die Sie in einem Spiel verfolgen müssen: die Gesundheit einer Spielfigur, die Geschwindigkeit sich bewegender Objekte, den Level den der Nutzer gerade spielt und die Punktzahl. All das wird in Form von Variablen abgebildet.

Klicken Sie zunächst auf das Menü „Datei“ und speichern Sie Ihr vorhandenes Programm, indem Sie auf „Auf deinem Computer speichern“ klicken. Wenn Sie das Programm bereits früher gespeichert haben, werden Sie gefragt, ob Sie es überschreiben möchten, wobei die vorher gespeicherte Kopie durch Ihre neue, aktuelle Version ersetzt wird. Klicken Sie auf „Datei“ und dann auf „Neu“, um ein neues, leeres Projekt zu beginnen (klicken Sie auf „OK“, wenn Sie gefragt werden, ob Sie die Inhalte des aktuellen Projektes ersetzen möchten). Klicken Sie in der Blockpalette auf die dunkelorangefarbene Kategorie „Variablen“ und dann auf die Schaltfläche „Neue Variable“. Geben Sie „Schleifen“ als Variablennamen ein (**Abbildung 4-5**), klicken Sie dann auf „OK“. Eine Reihe von Blöcken erscheint jetzt in der Blockpalette.



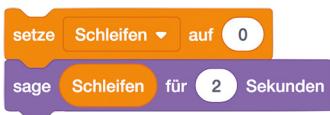
▲ Abbildung 4-5: Geben Sie Ihrer neuen Variablen einen Namen

Klicken und ziehen Sie den Block **setze Schleifen auf 0** in den Code-Bereich. Dadurch wird Ihr Programm angewiesen, die Variable mit einem Wert von 0 zu *initialisieren*. Klicken Sie als Nächstes auf die Kategorie „Aussehen“ der Blockpalette und ziehen Sie den Block **sage Hallo! für 2 Sekunden** unter Ihren Block **setze Schleifen auf 0**.

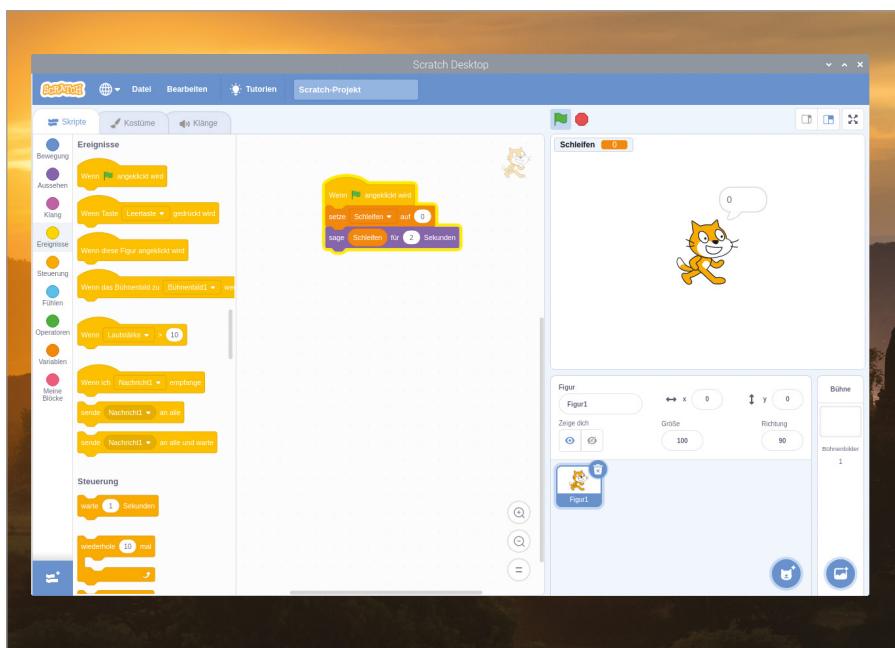


Wie Sie bereits festgestellt haben, bewirken die Blöcke **sage Hallo!**, dass das Katzen-Sprite alles sagt, was in ihnen steht. Anstatt die Nachricht selbst in den Block zu schreiben, können Sie stattdessen auch eine Variable verwenden. Klicken Sie wieder auf die Kategorie „Variablen“ in der Block-Palette. Klicken Sie dann auf den abgerundeten

Block **Schleifen**, bekannt als *Reporter-Block*, der sich ganz oben auf der Liste befindet, mit einer Checkbox daneben. Ziehen Sie diesen Reporter-Block jetzt auf das Wort „Hallo!“ in Ihrem Block **sage Hallo! für 2 Sekunden**. Dadurch wird ein neuer, kombinierter Block: **sage Schleifen für 2 Sekunden** erstellt.



Klicken Sie in der Blockpalette auf die Kategorie „Ereignisse“ und ziehen Sie den Block **Wenn grüne Flagge angeklickt wird** an die oberste Stelle Ihrer Abfolge von Blöcken. Klicken Sie auf die grüne Flagge über dem Bühnenbereich. Sie werden sehen, dass das Katzen-Sprite „0“ sagt (**Abbildung 4-6**) – der Wert, den Sie der Variable „Schleifen“ zugeteilt haben.

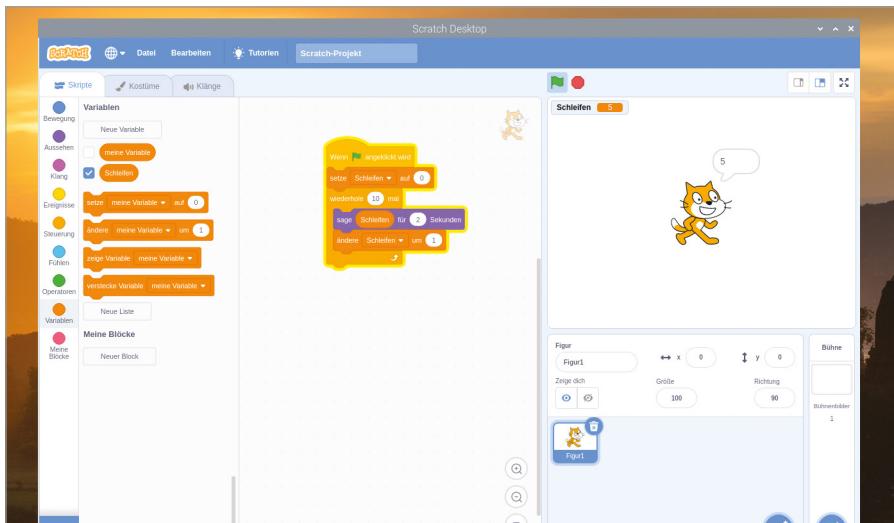


▲ Abbildung 4-6: Dieses Mal sagt die Katze den Wert der Variablen

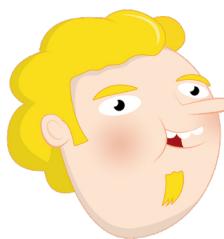
Die Variablen sind jedoch nicht fest und unveränderlich. Klicken Sie auf die Kategorie „Variablen“ in der Blockpalette. Klicken Sie dann auf den Block **ändere Schleifen um 1** und ziehen Sie ihn nach ganz unten in Ihre Abfolge von Blöcken. Klicken Sie als Nächstes auf die Kategorie „Steuerung“. Klicken und ziehen Sie dann den Block **wiederhole 10 mal** so dass er direkt unter Ihren Block **setze Schleifen auf 0** platziert ist und die restlichen Blöcke in Ihrer Abfolge umgibt.



Klicken Sie erneut auf die grüne Flagge. Dieses Mal werden Sie sehen, wie die Katze von 0 bis 9 zählt. Dies funktioniert, weil Ihr Programm nun die Variable ändert. Dieser Vorgang wird auch *Modifikation* genannt: Jedes Mal, wenn die Schleife ausgeführt wird, fügt das Programm dem Wert in der Variablen „Schleifen“ eins hinzu (**Abbildung 4-7**)..



▲ Abbildung 4-7: Dank der Schleife zählt die Katze jetzt aufwärts



VON NULL AN ZÄHLEN

Obwohl die Schleife, die Sie erstellt haben, zehnmal abläuft, zählt das Katzen-Sprite nur bis neun. Das liegt daran, dass wir mit dem Wert null für unsere Variable beginnen. Mit null und neun gibt es zehn Zahlen zwischen null und neun – das Programm stoppt also, bevor die Katze überhaupt dazu kommt 10 zu sagen. Um dies zu ändern, könnten Sie den Anfangswert der Variable auf 1 statt auf 0 setzen.



Sie können mit einer Variable noch mehr machen als sie nur zu ändern. Klicken Sie auf den Block **sage Schleifen für 2 Sekunden** und ziehen Sie ihn aus dem Block **wiederhole 10 mal** heraus. Platzieren Sie ihn unterhalb des Blocks **wiederhole 10 mal**. Klicken und ziehen Sie den Block **wiederhole 10 mal** in die Blockpalette, um ihn zu löschen, und ersetzen Sie ihn dann durch den Block **wiederhole bis**. Stellen Sie dabei sicher, dass der Block mit der Unterseite des Blocks **setze Schleifen auf 0** verbunden ist und die beiden anderen Blöcke in Ihrer Abfolge umgibt. Klicken Sie auf die Kategorie „Operatoren“ in der Blockpalette, grün gefärbt, dann klicken und ziehen Sie den rautenförmigen Block **$\bullet = \bullet$** und platzieren Sie ihn auf das passende rautenförmige Loch im Block **wiederhole bis**.

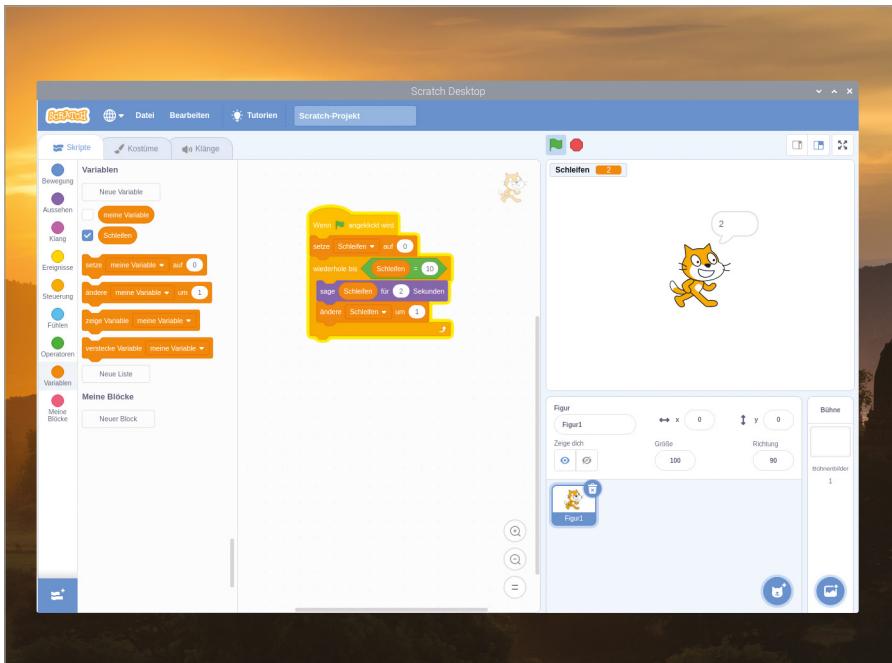


Mit dem „Operatoren“-Block können Sie zwei Werte, einschließlich Variablen, vergleichen. Klicken Sie auf die Kategorie „Variablen“, ziehen Sie den Reporter-Block **Schleifen** in das leere Feld im „Operatoren“-Block **$\bullet = \bullet$** , klicken Sie dann auf das Feld mit der 50 und geben Sie die Zahl 10 ein.



Klicken Sie auf die grüne Flagge über dem Bühnenbereich, und Sie werden feststellen, dass das Programm auf dieselbe Weise funktioniert wie zuvor: Das Katzen-Sprite zählt von 0 bis 9 (**Abbildung 4-8**) und dann stoppt das Programm. Dies liegt daran, dass der Block

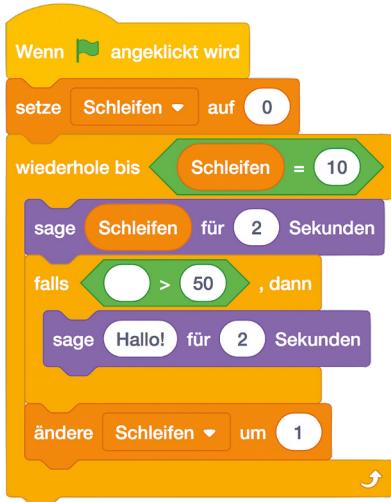
wiederhole bis genauso funktioniert wie der Block **wiederhole 10 mal**, aber anstatt die Anzahl der Schleifen selbst zu zählen, vergleicht er den Wert der Variablen „Schleifen“ mit dem Wert, den Sie auf der rechten Seite vom Block eingegeben haben. Wenn die Variable „Schleifen“ 10 erreicht, stoppt das Programm.



▲ Abbildung 4-8: Verwendung eines Blocks „wiederhole bis“ mit einem Vergleichsoperator

Dies ist bekannt als **Vergleichsoperator**: Man vergleicht buchstäblich zwei Werte. Klicken Sie auf die Kategorie „Operatoren“ der Blockpalette und suchen Sie die beiden anderen rautenförmigen Blöcke neben dem Block mit dem =-Symbol. Es handelt sich ebenfalls um Vergleichsoperatoren: „<“ vergleicht zwei Werte und wird ausgelöst, wenn der Wert des linken Teils kleiner als der des rechten Teils ist, und „>“ wird ausgelöst, wenn der Wert des linken Teils größer als der Wert des rechten Teils ist.

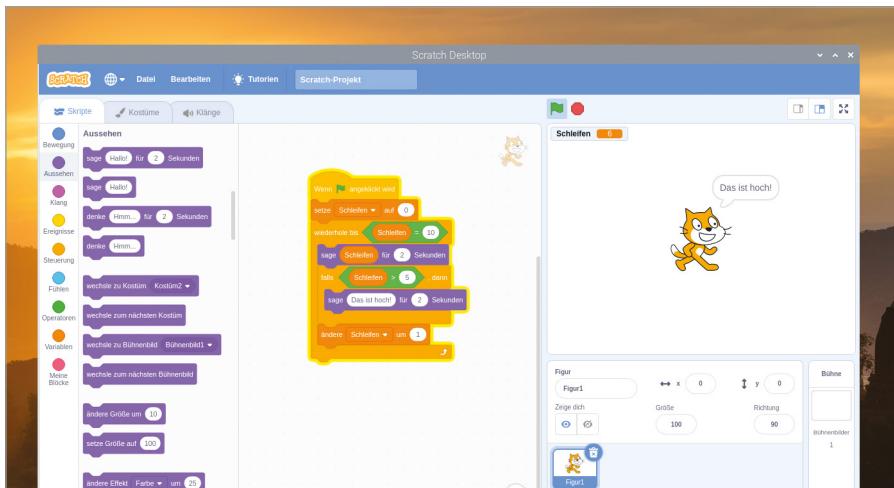
Klicken Sie auf die Kategorie „Steuerung“ der Blockpalette, suchen Sie den Block **falls dann**, klicken und ziehen Sie ihn in den Code-Bereich, um ihn direkt unter den Block **sage Schleifen für 2 Sekunden** zu platzieren. Er umgibt den Block **ändere Schleifen um 1** automatisch. Klicken Sie daher auf diesen Block und ziehen Sie ihn nach unten, um ihn so zu verschieben, dass er sich stattdessen mit dem unteren Ende Ihres Blocks **falls dann** verbindet. Klicken Sie als Nächstes auf die Kategorie „Aussehen“ der Blockpalette und ziehen Sie den Block **sage Hallo! für 2 Sekunden** in Ihren Block **falls dann**. Klicken Sie auf die Kategorie „Operatoren“ in der Blockpalette, klicken Sie dann auf den Block **<>** und ziehen Sie ihn in das rautenförmige Loch in Ihrem Block **falls dann**.



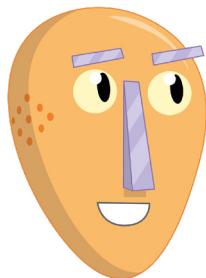
Der Block **falls dann** ist ein bedingter Block, d. h. die Blöcke innerhalb dieses Blocks werden nur ausgeführt, wenn eine bestimmte Bedingung erfüllt ist. Klicken Sie auf die Kategorie „Variablen“ der Blockpalette, ziehen Sie den Reporter-Block **Schleifen** in das leere Feld im Block **< >**, klicken Sie dann auf das Feld mit der 50 und geben Sie die Zahl 5 ein. Klicken Sie dann auf das Wort „Hallo!“ in Ihrem Block **sage Hallo! für 2 Sekunden** und tippen Sie „Das ist hoch!“ ein.



Klicken Sie auf die grüne Flagge. Zunächst funktioniert das Programm wie bisher, wobei das Katzen-Sprite von Null an aufwärts zählt. Wenn die Zahl 6 erreicht ist, die erste Zahl, die größer als 5 ist, beginnt der Block **falls dann** auszulösen und das Katzen-Sprite wird kommentieren, wie hoch die Zahlen werden (**Abbildung 4-9**). Herzlichen Glückwunsch: Sie können jetzt mit Variablen und Bedingungen arbeiten!



▲ Abbildung 4-9: Die Katze gibt einen Kommentar ab, wenn die Zahl 6 erreicht ist



HERAUSFORDERUNG: HOCH UND NIEDRIG



Wie können Sie das Programm so ändern, dass das Katzen-Sprite stattdessen kommentiert, wie niedrig die Zahlen unter 5 sind? Können Sie es so ändern, dass die Katze sowohl hohe als auch niedrige Zahlen kommentiert? Experimentieren Sie mit dem Block **falls dann sonst**, um dies zu erleichtern!

Projekt 1: Astronaut Reaktionstimer

Jetzt, da Sie verstehen, wie Scratch funktioniert, ist es an der Zeit, etwas interaktiver zu werden: einen Reaktionstimer, der zu Ehren des britischen ESA-Astronauten Tim Peake und seiner Zeit an Bord der Internationalen Raumstation (ISS, International Space Station) entwickelt wurde.

Speichern Sie Ihr bestehendes Programm, wenn Sie es behalten wollen. Dann öffnen Sie ein neues Projekt, indem Sie auf „Datei“ und „Neu“ klicken. Bevor Sie beginnen, geben Sie ihm einen Namen, indem Sie auf „Datei“ und „Auf deinem Computer speichern“ klicken: Nennen Sie es „Astronaut Reaktionstimer“.

Dieses Projekt stützt sich auf zwei Bilder, eines als Bühnenhintergrund, eines als Sprite, die nicht in den integrierten Ressourcen von Scratch enthalten sind. Um sie herunterzuladen, klicken Sie auf das Raspberry-Symbol, um das Menü von Raspberry Pi OS zu laden, bewegen Sie den Mauszeiger auf „Internet“ und klicken Sie auf Chromium-Webbrowser. Wenn der Browser geladen ist, geben Sie rpf.io/astronaut-backdrop in die Adressleiste ein, und drücken Sie die **EINGABE**-Taste. Klicken Sie mit der rechten Maustaste auf das Bild des Weltraums und klicken Sie auf „Bild speichern unter...“, dann klicken Sie auf die Schaltfläche „Speichern“ (**Abbildung 4-10**). Klicken Sie zurück in die Adressleiste, geben Sie rpf.io/astronaut-sprite ein und drücken Sie die **EINGABE**-Taste.



▲ Abbildung 4-10: Speichern Sie das Hintergrundbild

Klicken Sie erneut mit der rechten Maustaste auf das Bild von Tim Peake und klicken Sie auf „Bild speichern unter...“. Wählen Sie dann den Ordner „Downloads“ und klicken Sie auf die Schaltfläche „Speichern“. Wenn diese beiden Bilder gespeichert sind, können Sie Chromium schließen oder offen lassen und über die Taskleiste zurück zu Scratch 3 wechseln.



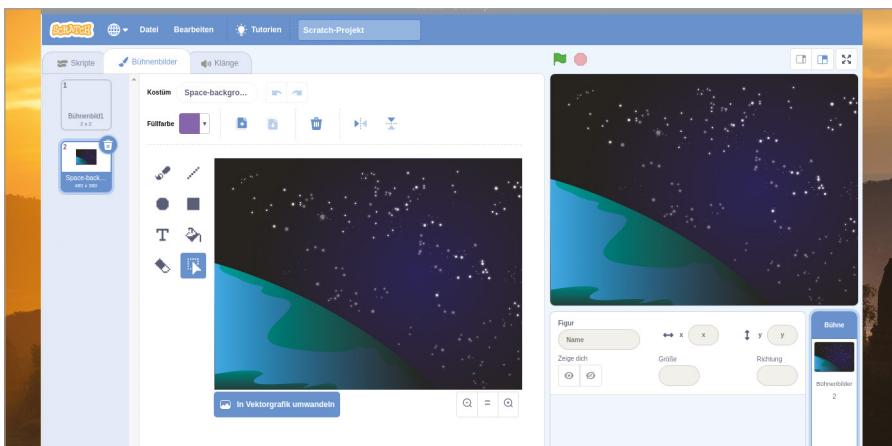
BENUTZEROBERFLÄCHE



Wenn Sie dieses Kapitel von Anfang an verfolgt haben, sollten Sie mittlerweile mit der Benutzeroberfläche von Scratch 3 vertraut sein. Die folgenden Projektanleitungen setzen voraus, dass Sie wissen, wo die Dinge sind. Falls Sie vergessen haben, wo Sie etwas finden können, schauen Sie zur Erinnerung auf das Bild der Benutzeroberfläche am Anfang dieses Kapitels zurück.

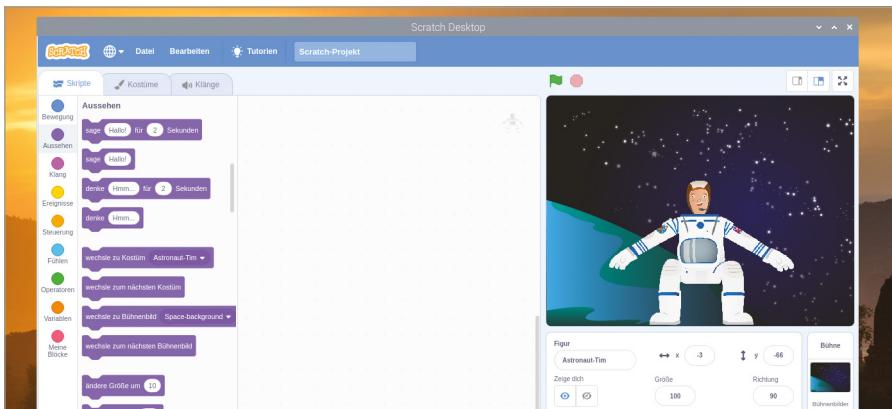
Klicken Sie mit der rechten Maustaste auf das Katzen-Sprite in der Sprites-Liste und klicken Sie auf „Löschen“. Positionieren Sie den Mauszeiger auf dem Symbol „Bühnenbild wählen“ und klicken Sie dann in der Liste die erscheint auf das Symbol „Bühnenbild hochladen“ .

. Finden Sie die Datei **Space-background.png** im Ordner „Downloads“, klicken Sie darauf, um sie auszuwählen, und klicken Sie dann auf „OK“. Der einfarbig-weiße Bühnenhintergrund ändert sich in das Bild des Weltraums, und der Code-Bereich wird durch den Hintergrundbereich ersetzt (**Abbildung 4-11**). Hier können Sie auf dem Hintergrund zeichnen, aber für den Moment klicken Sie einfach auf die Registerkarte „Skripte“ am oberen Rand des Scratch 3-Fensters.



▲ Abbildung 4-11: Der Weltraumhintergrund erscheint auf der Bühne

Laden Sie Ihr neues Sprite hoch, indem Sie den Mauszeiger auf dem Symbol „Figur wählen“ positionieren und dann auf das Symbol „Figur hochladen“ oben in der angezeigten Liste klicken. Finden Sie die Datei **Astronaut-Tim.png** im Ordner „Downloads“, klicken Sie darauf, um sie auszuwählen, und klicken Sie dann auf „OK“. Das Sprite erscheint automatisch auf der Bühne, befindet sich aber möglicherweise nicht in der Mitte: Klicken und ziehen Sie es mit der Maus und platzieren Sie es so, dass es sich in der Nähe der unteren Mitte befindet (**Abbildung 4-12**)



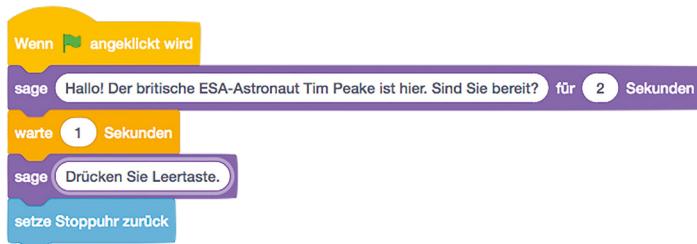
▲ Abbildung 4-12: Ziehen Sie das Astronauten-Sprite in die untere Mitte der Bühne

Mit Ihrem neuen Hintergrund und Ihrem neuen Sprite sind Sie jetzt bereit, Ihr Programm zu erstellen. Beginnen Sie, indem Sie eine neue Variable namens „Zeit“ erstellen und sicherstellen, dass „Für alle Figuren“ ausgewählt ist, bevor Sie auf „OK“ klicken. Klicken Sie auf Ihr Sprite – entweder auf der Bühne oder im Sprite-Bereich – um es auszuwählen, und fügen Sie dann einen Block **Wenn [Flagge angeklickt wird]** aus der Kategorie „Ereignisse“ zum Code-Bereich

hinzufügen. Als Nächstes fügen Sie einen Block **sage Hallo! für 2 Sekunden** aus der Kategorie „Aussehen“ hinzu und klicken Sie dann darauf, um ihn in „Hallo! Der britische ESA-Astronaut Tim Peake ist hier. Sind Sie bereit?“ zu ändern.



Als Nächstes fügen Sie einen Block **warte 1 Sekunden** aus der Kategorie „Ereignisse“ hinzu, dann einen Block **sage Hallo!**. Ändern Sie diesen Block in „Drücken Sie Leertaste.“ und fügen Sie dann einen Block **setze Stoppuhr zurück** aus der Kategorie „Fühlen“ hinzu. Diese steuert eine in Scratch eingebaute spezielle Variable, um Zeiten zu erfassen, und wird verwendet, um zu bestimmen, wie schnell Sie im Spiel reagieren können.

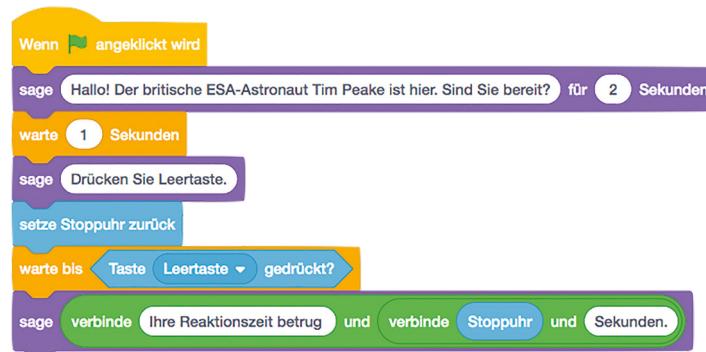


Fügen Sie einen „Steuerung“-Block **warte bis** hinzu und ziehen Sie dann einen **Taste Leertaste gedrückt?**-„Fühlen“-Block in seinen weißen Raum. Dadurch wird das Programm angehalten, bis Sie die **LEERTASTE** auf der Tastatur drücken, aber der Timer läuft weiter – er zählt genau, wie lange zwischen der Meldung „Drücken Sie Leertaste“ und dem tatsächlichen Drücken der **LEERTASTE** durch Sie vergeht.

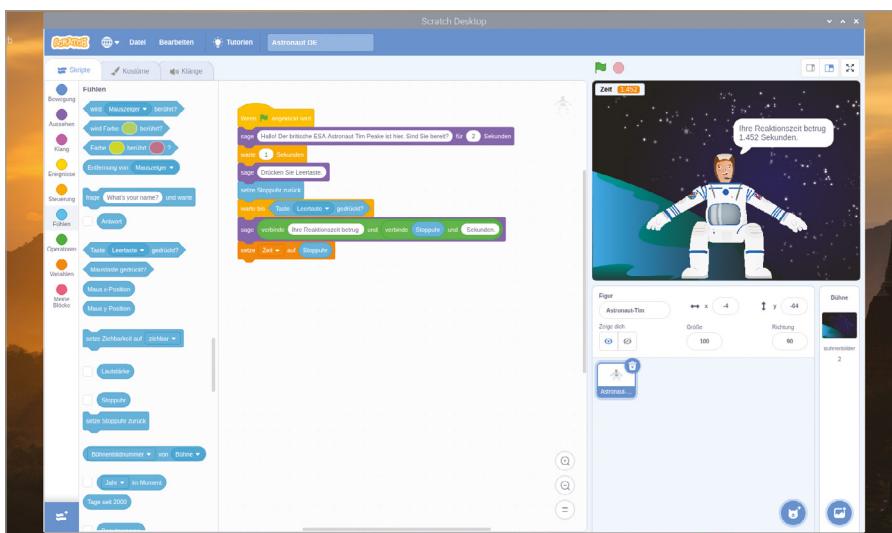


Tim muss Ihnen jetzt sagen, wie lange Sie für das Drücken der **LEERTASTE** gebraucht haben, aber auf eine Weise, die leicht zu lesen ist. Dazu benötigen Sie einen „Operatoren“-Block **verbinde**). Dabei werden zwei Werte, die natürlich auch Variablen sein können, zusammen nacheinander verbunden – bekannt als **Verkettung**.

Beginnen Sie mit einem Block **sage Hallo!** und ziehen Sie dann einen „Operatoren“-Block **verbinde** über das Wort „Hallo!“. Klicken Sie auf „apple“ und geben Sie „Ihre Reaktionszeit betrug“ ein. Achten Sie darauf, am Ende ein Leerzeichen einzufügen, und ziehen Sie dann einen weiteren Verbindungsblock auf das Wort „banana“ im zweiten Feld. Ziehen Sie einen Reporter-Block **Stoppuhr** aus der Kategorie „Fühlen“ in das nun mittlere Feld und geben Sie „Sekunden“ in das letzte Feld ein. Achten Sie darauf, am Anfang ein Leerzeichen einzufügen.



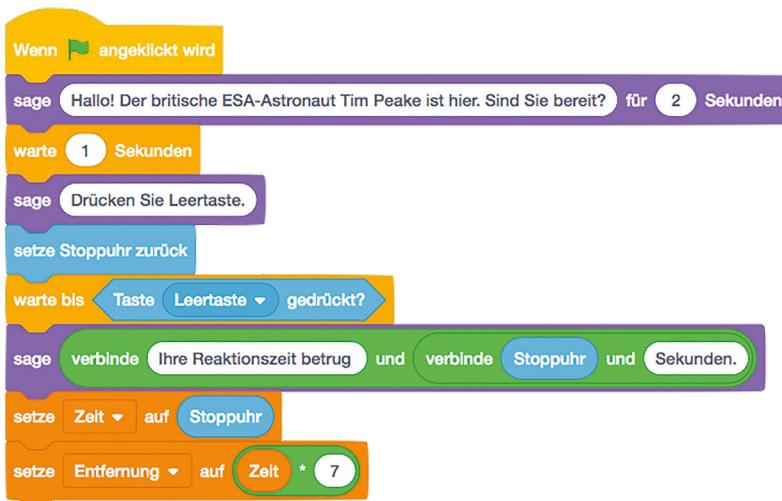
Ziehen Sie schließlich einen „Variablen“-Block **setze meine Variable auf 0** an das Ende Ihrer Abfolge. Klicken Sie auf den Dropdown-Pfeil neben „meine Variable“ und klicken Sie in der Liste auf „Zeit“. Ersetzen Sie dann die 0 durch einen Reporter-Block **Stoppuhr** aus der Kategorie „Fühlen“. Ihr Spiel ist jetzt bereit zum Testen, indem Sie auf die grüne Flagge über der Bühne klicken. Machen Sie sich bereit, und sobald Sie die Meldung „Drücken Sie Leertaste“ sehen, drücken Sie die **Leertaste** so schnell Sie können (**Abbildung 4-13**) – Können Sie unseren Highscore schlagen?



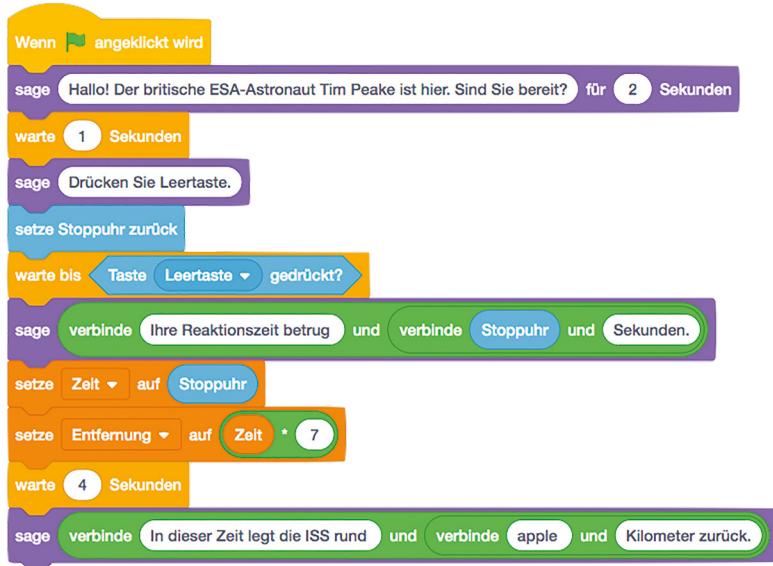
▲ Abbildung 4-13: Zeit, das Spiel zu spielen!

Sie können dieses Projekt weiter entwickeln, indem Sie grob berechnen lassen, wie weit die Internationale Raumstation in der Zeit, die Sie für die Betätigung der **LEERTASTE** gebraucht haben, geflogen ist. Wir legen dabei die offiziell angegebene Geschwindigkeit der Station von sieben Kilometern pro Sekunde zu Grunde. Erstellen Sie zunächst eine neue Variable namens „Entfernung“. Beachten Sie, wie sich die Blöcke in der Kategorie „Variablen“ automatisch ändern, um die neue Variable anzuzeigen, aber die bestehenden Variablenblöcke **Zeit** in Ihrem Programm bleiben gleich.

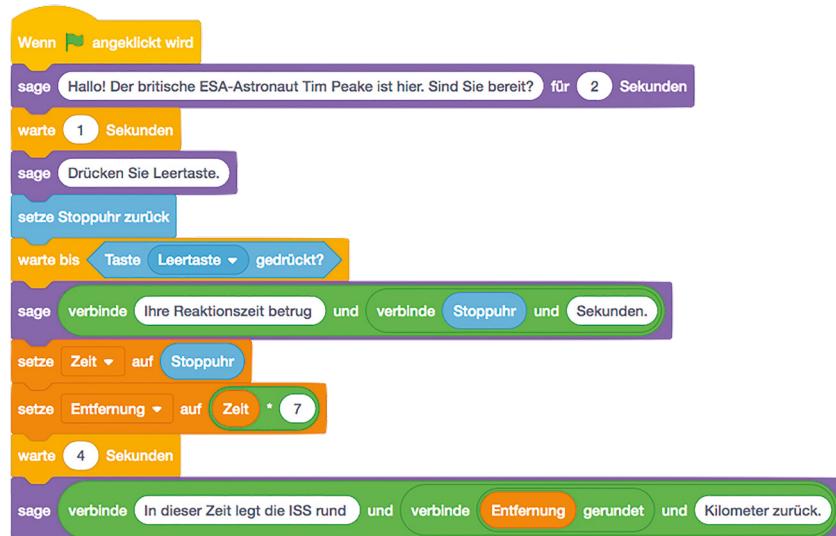
Fügen Sie einen Block **setze Entfernung auf 0** hinzu und ziehen Sie dann einen „Operatoren“-Block **0 * 0** – der Multiplikation anzeigt – auf die 0. Ziehen Sie einen Reporter-Block **Zeit** auf das erste Feld und geben Sie dann die Zahl 7 in das zweite Feld ein. Wenn Sie fertig sind, sieht Ihr kombinierter Block so aus: **setze Entfernung auf Zeit * 7**. Dies wird die Zeit, die Sie benötigt haben, um die **LEERTASTE** zu drücken mit sieben multiplizieren, um die Entfernung in Kilometern zu erhalten, die die ISS zurückgelegt hat.



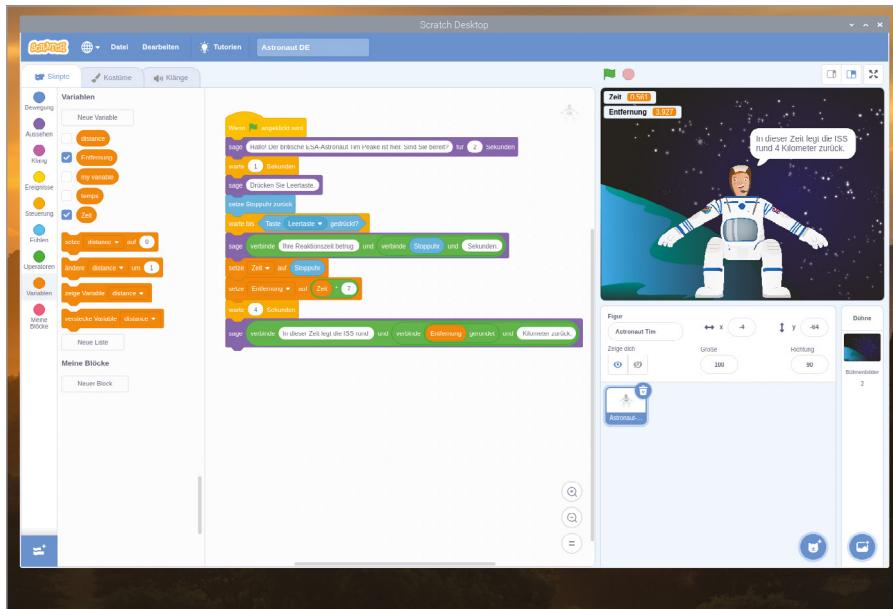
Als Nächstes fügen Sie einen Block **warte 1 Sekunden** hinzu und ändern ihn auf 4. Ziehen Sie schließlich einen weiteren Block **sage Hallo!** an das Ende Ihrer Abfolge und fügen Sie zwei Blöcke **verbinde** hinzu, so wie Sie es zuvor getan haben. Überschreiben Sie das erste Feld „apple“ mit „In dieser Zeit legt die ISS rund“. Vergessen Sie das Leerzeichen am Ende nicht. Im „banana“-Feld schreiben Sie „Kilometer zurück“, wobei Sie wiederum das Leerzeichen am Anfang setzen.



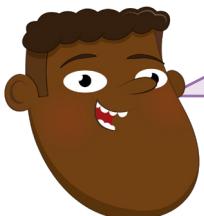
Ziehen Sie abschließend einen „Operatoren“-Block **gerundet** in den mittleren leeren Bereich und dann einen Reporter-Block **Entfernung** in den neuen leeren Bereich. Der Block **gerundet** rundet Zahlen auf die nächstliegende ganze Zahl auf oder ab, sodass Sie statt einer sehr genauen, aber schwer lesbaren Kilometerzahl eine leicht lesbare ganze Zahl erhalten.



Klicken Sie auf die grüne Flagge, um Ihr Programm auszuführen, und sehen Sie, wie weit sich die ISS in der Zeit bewegt, die Sie benötigen, um die LEER-Taste zu drücken. Denken Sie daran, Ihr Programm zu speichern, wenn Sie fertig sind, sodass Sie es in Zukunft einfach wieder laden können, ohne von vorne beginnen zu müssen!



▲ Abbildung 4-14: Tim erzählt Ihnen, wie weit die ISS gereist ist



HERAUSFORDERUNG: WER IST SCHNELL?



Welche anderen Berufe erfordern neben dem Astronauten sekundenschnelle Reflexe? Können Sie Ihre eigenen Sprites und Hintergründe zeichnen, um einen dieser Berufe zu zeigen?

Projekt 2: Synchronisiertes Schwimmen

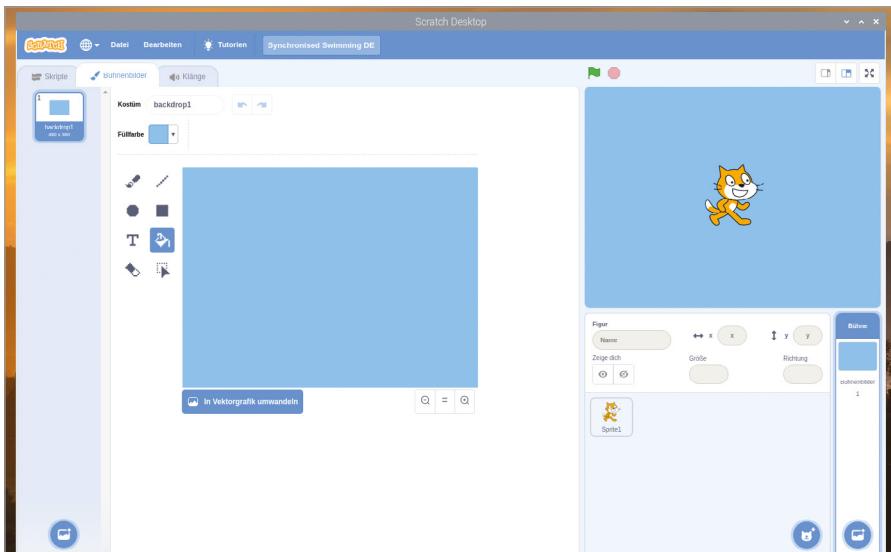
Die meisten Spiele verwenden mehr als eine einzige Taste, und dieses Projekt zeigt dies, indem es eine Zwei-Tasten-Steuerung mit den Tasten ← und → auf der Tastatur bietet.

ONLINE-PROJEKT



Dieses Projekt ist auch online verfügbar unter rpf.io/synchro-swimming

Erstellen Sie ein neues Projekt und speichern Sie es als „Synchronisiertes Schwimmen“. Klicken Sie auf die Bühne im Abschnitt „Bühnensteuerung“ und dann oben links auf die Registerkarte „Bühnenbilder“. Klicken Sie auf die Schaltfläche „In Rastergrafik unwandeln“ unterhalb des Hintergrunds. Wählen Sie eine blaue Farbe – blau wie Wasser – aus der Palette „Füllfarbe“, klicken Sie auf das Symbol „Fülleimer“  und klicken Sie dann auf den karierten Hintergrund, um ihn mit Blau zu füllen (**Abbildung 4-15**).

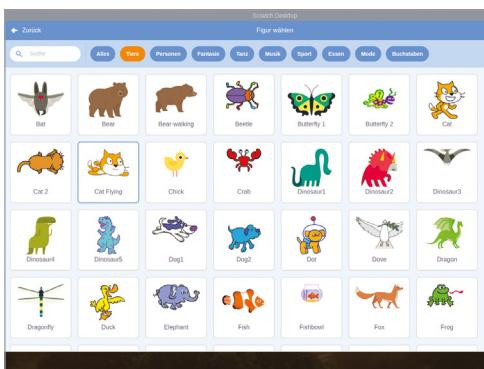


▲ Abbildung 4-15: Füllen Sie den Hintergrund mit blauer Farbe

Klicken Sie mit der rechten Maustaste auf das Katzen-Sprite in der Liste und klicken Sie auf „Löschen“. Klicken Sie auf das Symbol „Figur wählen“ , um eine Liste der integrierten

Sprites anzuzeigen. Klicken Sie auf die Kategorie „Tiere“ und dann auf „Cat Flying“ (**Abbildung 4-16**). Dieses Sprite eignet sich auch gut für Schwimmprojekte.

Klicken Sie auf das neue Sprite und ziehen Sie dann zwei „Ereignisse“-Blöcke **Wenn Taste Leertaste gedrückt wird** in den Code-Bereich. Klicken Sie auf den kleinen Abwärtspfeil neben dem Wort „Leertaste“ im ersten Block und wählen Sie den „Pfeil nach links“ aus der Liste der möglichen Optionen. Ziehen Sie einen „Bewegung“-Block



▲ Abbildung 4-16: Wählen Sie ein Sprite aus der Bibliothek

drehe dich ↗ um 15 Grad unter Ihren Block **Wenn Taste Pfeil nach links gedrückt wird**, dann machen Sie dasselbe mit Ihrem zweiten „Ereignisse“-Block, außer dass Sie den „Pfeil nach rechts“ aus der Liste wählen und einen „Bewegung“-Block **drehe dich ↙ um 15 Grad** verwenden.

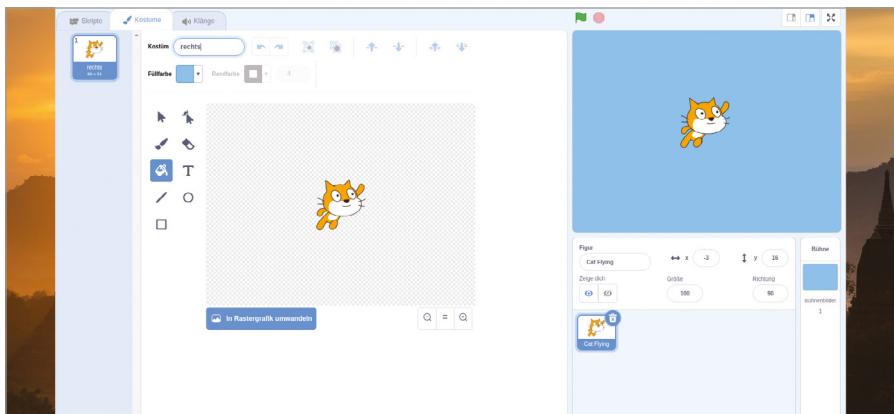


Drücken Sie die Taste ← oder →, um Ihr Programm zu testen. Sie werden sehen, wie sich das Katzen-Sprite in die Richtung dreht, die Sie auf der Tastatur wählen. Beachten Sie, dass Sie diesmal nicht auf die grüne Flagge zu klicken brauchten. Das liegt daran, dass die von Ihnen verwendeten Ereignis-Triggerblöcke immer aktiv sind, auch wenn das Programm nicht im normalen Sinne „läuft“.

Führen Sie dieselben Schritte noch zweimal aus, aber dieses Mal wählen Sie den „Pfeil nach oben“ und den „Pfeil nach unten“ für die „Ereignisse“-Triggerblöcke, dann wählen Sie **gehe 10er Schritt** und **gehe -10er Schritt** für die „Bewegung“-Blöcke. Drücken Sie jetzt die Pfeiltasten und Sie werden sehen, dass Ihre Katze sich umdrehen und auch vorwärts und rückwärts schwimmen kann!

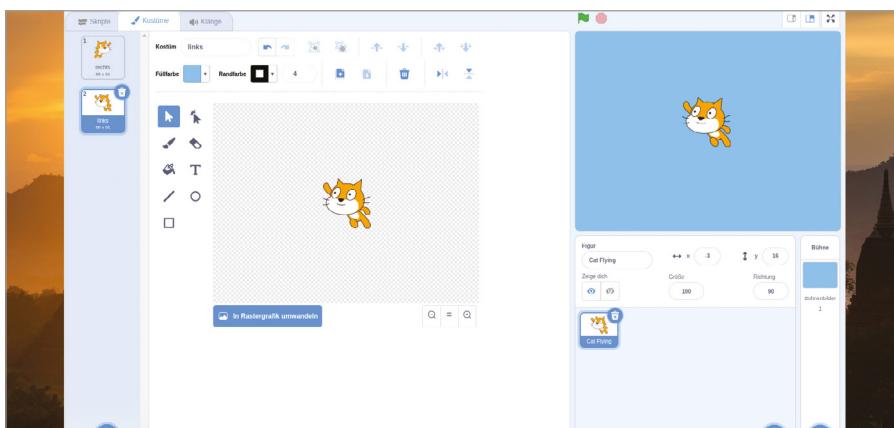


Um die Bewegung des Katzensprites realistischer zu machen, können Sie sein Aussehen verändern – in Scratch bekannt als **Kostüm**. Klicken Sie auf das Katzen-Sprite und dann auf die Registerkarte „Kostüme“ über der Blockpalette. Klicken Sie auf das Kostüm „cat flying-a“ und klicken Sie auf das „X im-Mülleimer“-Symbol  in der rechten oberen Ecke, um es zu löschen. Klicken Sie als Nächstes auf das Kostüm „cat flying-b“ und benennen Sie es im Namensfeld oben in „rechts“ um (**Abbildung 4-17**)



▲ Abbildung 4-17: Benennen Sie das Kostüm in „rechts“ um

Klicken Sie mit der rechten Maustaste auf das neu umbenannte Kostüm „rechts“ und klicken Sie auf „Duplizieren“, um eine Kopie zu erstellen. Klicken Sie auf diese Kopie, um sie auszuwählen, klicken Sie auf das Symbol  „Auswählen“, klicken Sie auf „Horizontal spiegeln“  und benennen Sie sie dann in „links“ um (**Abbildung 4-18**). Sie haben nun zwei „Kostümen“ für Ihr Sprite, die exakt spiegelbildlich sind: eines, das „rechts“ genannt wird, wobei die Katze nach rechts zeigt, und eines, das „links“ genannt wird, wobei die Katze nach links zeigt.



▲ Abbildung 4-18: Duplizieren Sie das Kostüm, drehen Sie es um und nennen Sie es „links“

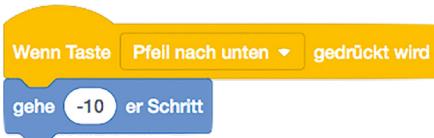
Klicken Sie auf die Registerkarte „Skripte“ über dem Kostümbereich, ziehen Sie dann zwei Aussehen-Blöcke **wechsle zu Kostüm links** unter die „Ereignisse“-Blöcke mit dem linken Pfeil und dem rechten Pfeil und ändern Sie den unter dem rechten Pfeilblock in **wechsle zu Kostüm rechts**. Versuchen Sie es noch einmal mit den Pfeiltasten. Die Katze scheint sich nun in die Richtung zu drehen, in die sie schwimmt.



Für das Synchronschwimmen im olympischen Stil brauchen wir jedoch mehr Schwimmer, und wir brauchen einen Weg, um die Position des Katzen-Sprite zurückzusetzen. Fügen Sie einen „Ereignisse“-Block **Wenn Flagge angeklickt wird** hinzu, dann fügen Sie darunter einen „Bewegung“-Block **gehe zu x: 0 y: 0** hinzu, wobei Sie die Werte ggf. ändern, und einen „Bewegung“-Block **setze Richtung auf 90 Grad**. Wenn Sie nun auf die grüne Flagge klicken, wird die Katze in die Mitte der Bühne bewegt und zeigt nach rechts.

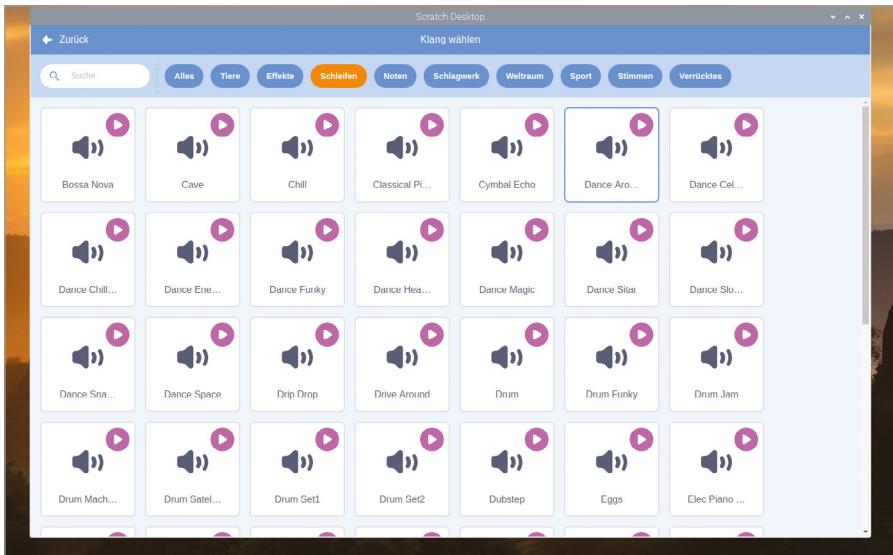


Um mehr Schwimmer zu erstellen, fügen Sie einen Block **wiederhole 6 mal** hinzu – abweichend vom Standardwert 10 – und fügen Sie einen Block **erzeuge Klon von mir selbst** darin ein. Damit die Schwimmer nicht alle in die gleiche Richtung schwimmen, fügen Sie einen Block **drehe dich ⌂ um 60 Grad** oberhalb des Blocks **erzeuge Klon**, aber immer noch innerhalb des Blocks **wiederhole 6 mal** hinzu. Klicken Sie auf die grüne Flagge und spielen Sie jetzt mit den Pfeiltasten, um Ihre Schwimmer zum Leben zu erwecken!



Um das olympische Gefühl zu vervollständigen, müssen Sie etwas Musik hinzufügen.

Klicken Sie auf die Registerkarte „Klänge“ oberhalb der Blockpalette und dann auf das Symbol „Klang auswählen“ . Klicken Sie auf die Kategorie „Schleifen“ und blättern Sie dann durch die Liste (**Abbildung 4-19**), bis Sie eine Musik finden, die Ihnen gefällt – wir haben „Dance Around“ ausgewählt. Klicken Sie auf die gewünschte Musik, und klicken Sie dann auf die Registerkarte „Skripte“, um den Codebereich wieder zu öffnen.



▲ Abbildung 4-19: Wählen Sie eine Musiksleife aus der Klangbibliothek aus

Fügen Sie einen weiteren „Ereignisse“-Block **Wenn  angeklickt wird** zu Ihrem Codebereich hinzu, und fügen Sie dann einen Block „Steuerung“ **wiederhole fortlaufend** hinzu. Fügen Sie innerhalb dieses „Steuerung“-Blocks einen Block **spiele Klang dance around ganz** hinzu. Denken Sie daran, dabei nach dem Namen des von Ihnen gewählten Musikstücks zu suchen – und klicken Sie auf die grüne Flagge, um Ihr neues Programm zu testen. Wenn Sie die Musik stoppen möchten, klicken Sie auf das rote Achteck, um das Programm zu stoppen und die Tonwiedergabe abzuschalten!

```

Wenn Taste Pfeil nach links gedrückt wird
wechsle zu Kostüm links
drehe dich ⌂ um 15 Grad

```

```

Wenn Taste Pfeil nach rechts gedrückt wird
wechsle zu Kostüm rechts
drehe dich ⌂ um 15 Grad

```

```

Wenn Taste Pfeil nach oben gedrückt wird
gehe 10 er Schritt

```

```

Wenn Taste Pfeil nach unten gedrückt wird
gehe -10 er Schritt

```

```

Wenn grün angeklickt wird
gehe zu x: 0 y: 0
setze Richtung auf 90 Grad
wiederhole 6 mal
  drehe dich ⌂ um 60 Grad
  erzeuge Klon von mir selbst

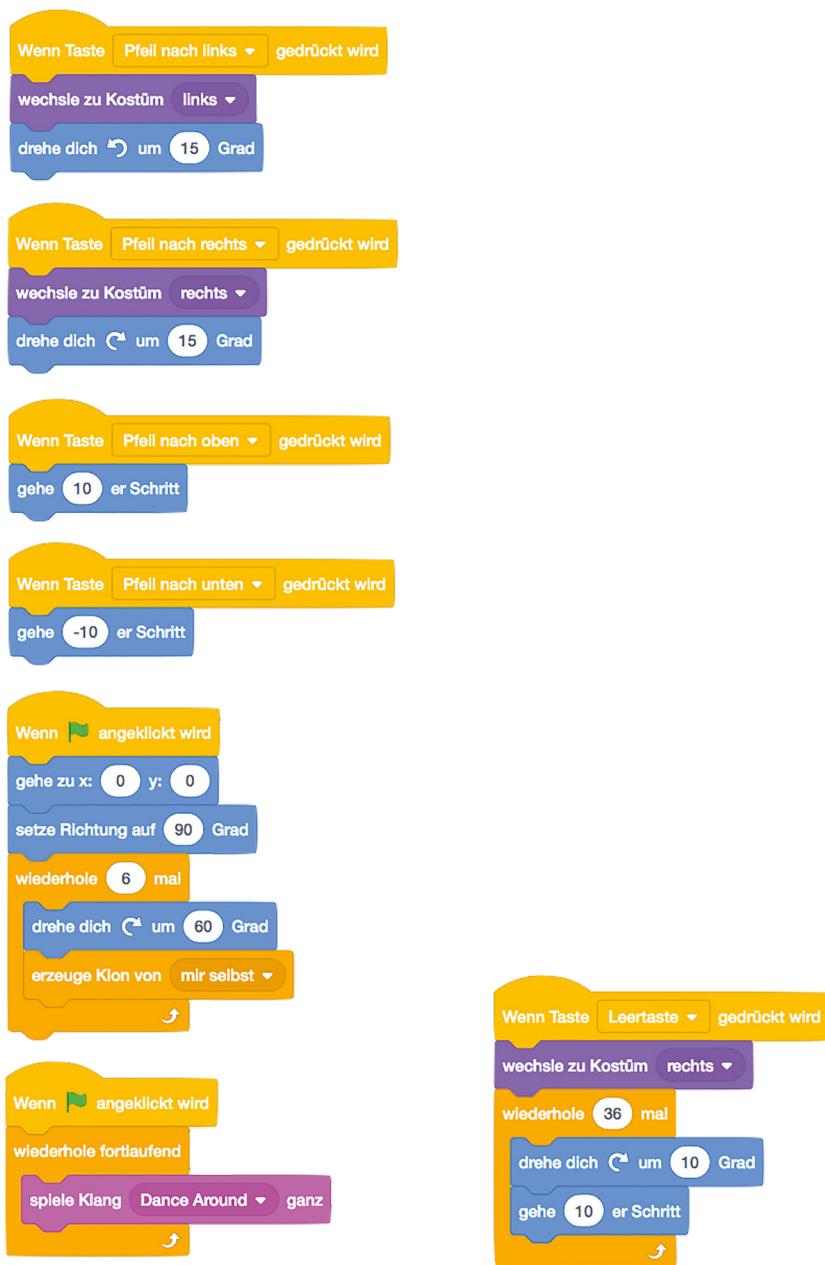
```

```

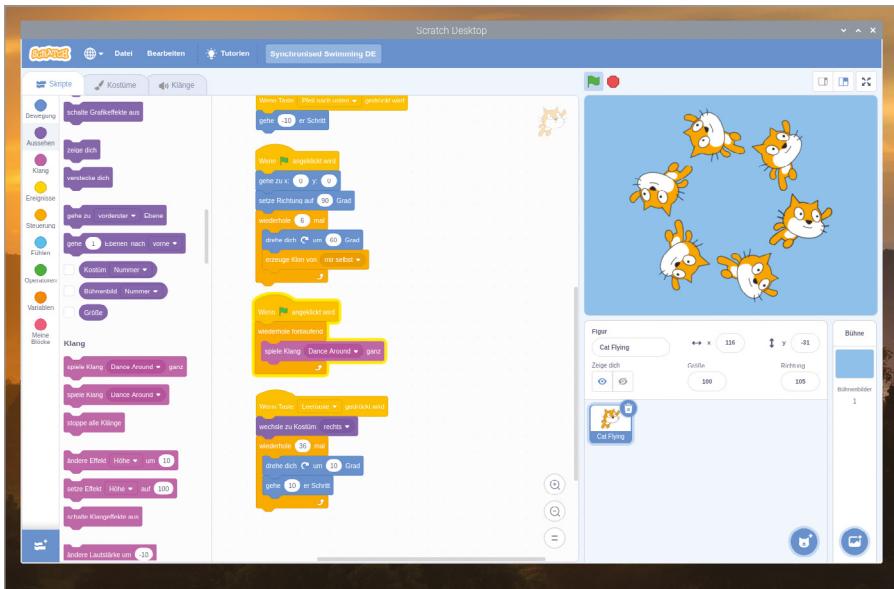
Wenn grün angeklickt wird
wiederhole fortlaufend
  spielt Klang Dance Around ganz

```

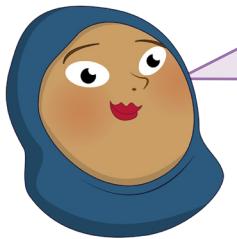
Schließlich können Sie eine vollständige Tanzroutine simulieren, indem Sie einen neuen Ereignisauslöser zu Ihrem Programm hinzufügen. Fügen Sie einen „Ereignisse“-Block **Wenn Taste Leertaste gedrückt wird** hinzu und dann einen Block **wechsle zu Kostüm rechts**. Darunter fügen Sie einen Block **wiederhole 36 mal** hinzu – denken Sie daran, den Wert gegenüber dem Standardwert zu ändern – und darin einen Block **drehe dich ⌂ um 10 Grad** und einen Block **gehe 10er Schritt**.



Klicken Sie auf die grüne Flagge, um das Programm zu starten, und drücken Sie dann die LEER-Taste, um die neue Routine auszuprobieren (**Abbildung 4-20** (umseitig)! Vergessen Sie nicht, Ihr Programm zu speichern, wenn Sie fertig sind.



▲ Abbildung 4-20: Die Routine „Synchronisiertes Schwimmen“



HERAUSFORDERUNG : BENUTZERDEFINIERTERoutine



Können Sie Ihre eigene synchronisierte Schwimmroutine mit Hilfe von Schleifen erstellen? Was müssten Sie ändern, wenn Sie mehr Schwimmer oder weniger Schwimmer haben wollten? Können Sie mehrere Schwimmroutinen hinzufügen, die über verschiedene Tasten auf der Tastatur ausgelöst werden können?

Projekt 3: Bogenschießen

Jetzt, wo Sie bei Scratch langsam zum Experten werden, ist es an der Zeit, an etwas zu arbeiten, das etwas anspruchsvoller ist: ein Bogenschießen-Spiel, bei dem der Spieler ein Ziel mit einem zufällig schwingenden Pfeil und Bogen treffen muss.

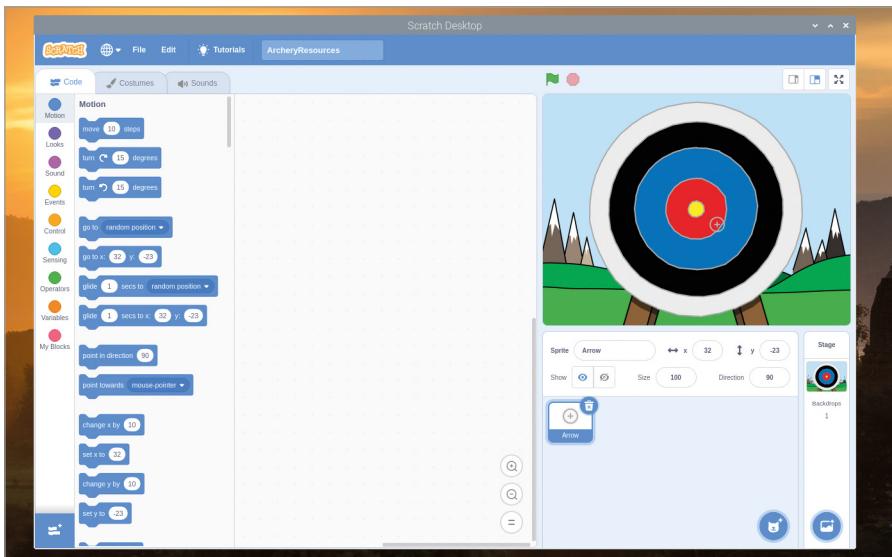
ONLINE-PROJEKT



Dieses Projekt ist auch online verfügbar unter rpf.io/archery

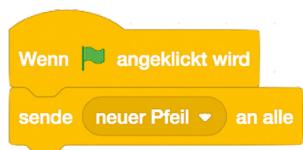
Beginnen Sie mit dem Öffnen des Chromium-Webbrowsers und tippen Sie auf rpf.io/p/en/archery-go, gefolgt von der **EINGABE**-Taste. Die Ressourcen für das Spiel werden nun als Zip-Datei heruntergeladen, sodass Sie sie entpacken müssen (klicken Sie mit der rechten Maustaste auf die heruntergeladene Datei im Ordner „Downloads“ und wählen Sie „Hier entpacken“). Wechseln Sie zurück zu Scratch 3 und klicken Sie auf das Menü „Datei“ gefolgt von „Hochladen von deinem

Computer“. Klicken Sie auf die gerade entpackte Datei **ArcheryResources.sb3** gefolgt von der Schaltfläche „Öffnen“. Sie werden gefragt, ob Sie den Inhalt Ihres aktuellen Projekts ersetzen möchten: Wenn Sie Ihre Änderungen noch nicht gespeichert haben, klicken Sie auf „Abbrechen“ und speichern Sie sie, andernfalls klicken Sie auf „OK“.



▲ Abbildung 4-21: Ressourcen-Projekt für das Bogenschießen geladen

Das Projekt, das Sie gerade geladen haben, enthält einen Hintergrund und ein Sprite (**Abbildung 4-21**), aber nichts von dem eigentlichen Code, um ein Spiel zu bauen: das Hinzufügen ist Ihre Aufgabe. Beginnen Sie mit dem Hinzufügen eines Blocks **wenn angeklickt wird**, dann eines Blocks **sende Nachricht 1 an alle**. Klicken Sie auf den Abwärtspfeil am Ende des Blocks und dann auf „Neue Nachricht“, und geben Sie „neuer Pfeil“ ein, bevor Sie auf die Schaltfläche „OK“ klicken. Ihr Block heißt jetzt **sende neuer Pfeil an alle**.



Ein Broadcast („sende ... an alle“) ist eine Nachricht von einem Teil Ihres Programms, die von jedem anderen Teil Ihres Programms empfangen werden kann. Um sie tatsächlich etwas tun zu lassen, fügen Sie einen Block **Wenn ich Nachricht 1 empfange** hinzu und ändern Sie ihn wieder auf **Wenn ich neuer Pfeil empfange**. Diesmal können Sie einfach auf den Abwärtspfeil klicken und „neuer Pfeil“ aus der Liste wählen. Sie müssen die Nachricht nicht erneut erstellen.

Fügen Sie unter Ihrem Block **Wenn ich neuer Pfeil empfange** einen Block **gehe zu x: -150 y: -150** und einen Block **setze Größe auf 400** hinzu. Denken Sie daran, dass

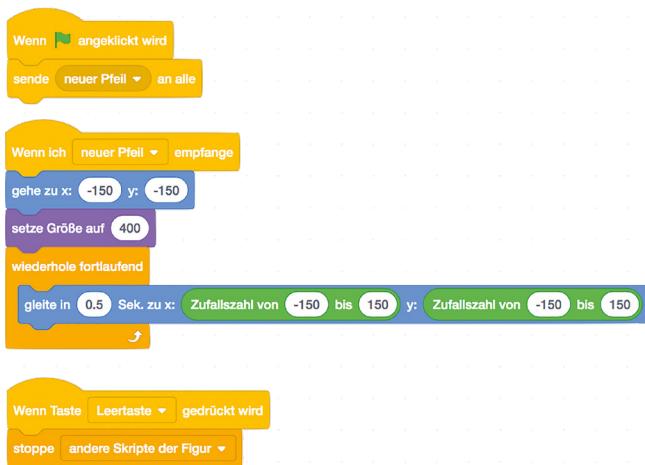
dies nicht die Standardwerte für diese Blöcke sind, sodass Sie sie ändern müssen, sobald Sie sie auf den Codebereich gezogen haben. Klicken Sie auf die grüne Flagge, um zu sehen, was Sie bisher getan haben: Das Pfeil-Sprite, mit dem der Spieler auf das Ziel zielt, springt nach links unten und vervierfacht seine Größe.



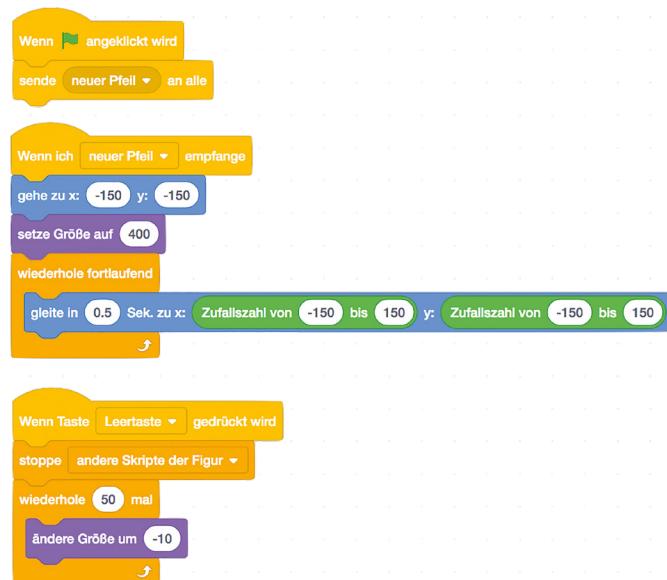
Um den Spieler herauszufordern, fügen Sie eine Bewegung hinzu, die das Schwanken beim Ziehen des Bogens und beim Zielen des Bogenschützen simuliert. Fügen Sie einen Block **wiederhole fortlaufend** hinzu, und in diesen ziehen Sie einen Block **gleite in 1 Sekunden auf x: -150 y: -150**. Überschreiben Sie die „1“ im ersten weißen Feld mit „0.5“, und setzen Sie dann einen „Operatoren“-Block **Zufallszahl von -150 bis 150** in jedes der beiden anderen weißen Felder. (Hinweis: bitte nutzen Sie den Punkt statt dem Komma, genauso wie hier angegeben - im Englischen ist der Punkt das Dezimaltrennzeichen) Das bedeutet, dass der Pfeil in einer zufälligen Richtung und über eine zufällige Distanz um die Bühne driften wird – was es viel schwieriger macht, das Ziel zu treffen!



Klicken Sie erneut auf die grüne Flagge, und Sie werden sehen, was dieser Block bewirkt: Ihr Pfeil-Sprite driftet nun auf der Bühne hin und her, und deckt verschiedene Teile des Ziels ab. Im Moment haben Sie jedoch keine Möglichkeit, den Pfeil auf das Ziel loszulassen. Ziehen Sie einen Block **Wenn Taste Leertaste gedrückt wird** in Ihren Code-Bereich, gefolgt von einem „Steuerung“-Block **stoppe alles**. Klicken Sie auf den Abwärtspfeil am Ende des Blocks und ändern Sie ihn in den Block **stoppe andere Skripte der Figur**.

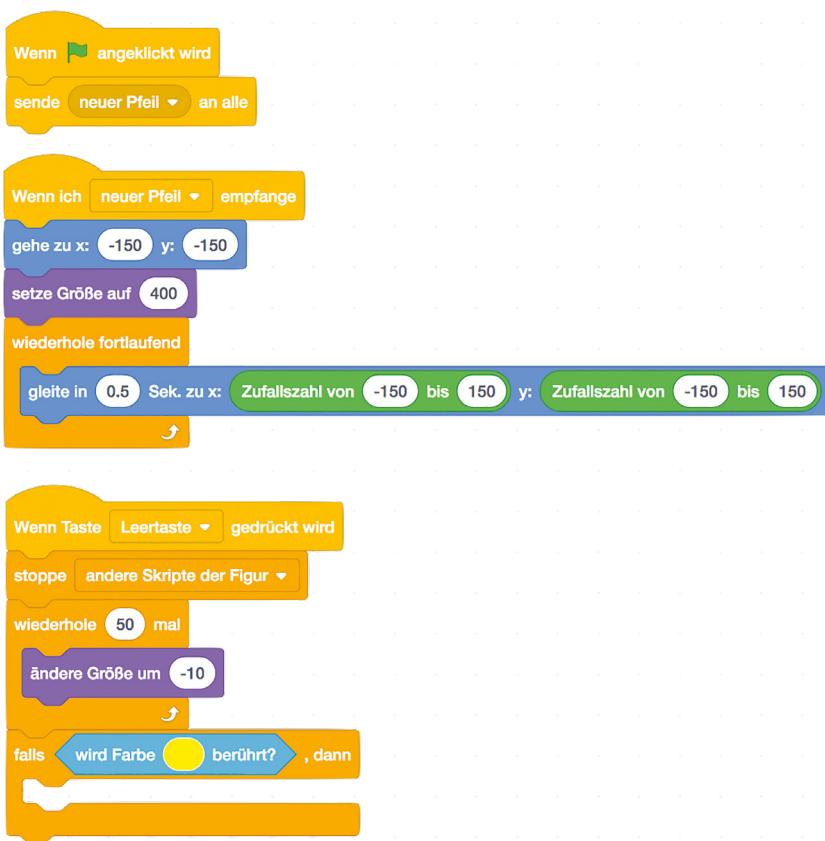


Wenn Sie Ihr Programm angehalten haben, um die neuen Blöcke hinzuzufügen, klicken Sie auf die grüne Flagge, um es erneut zu starten, und drücken Sie dann die **LEER**-Taste: Sie werden sehen, dass sich das Pfeil-Sprite nicht mehr bewegt. Das ist ein Anfang, aber Sie müssen es so aussehen lassen, als ob der Pfeil zum Ziel fliegt. Fügen Sie einen Bock **wiederhole 50 mal** hinzu, in den Sie einen Block **ändere Größe um -10** einfügen und klicken Sie dann auf die grüne Flagge, um Ihr Spiel erneut zu testen. Diesmal scheint der Pfeil von Ihnen weg und auf das Ziel zuzufliegen.

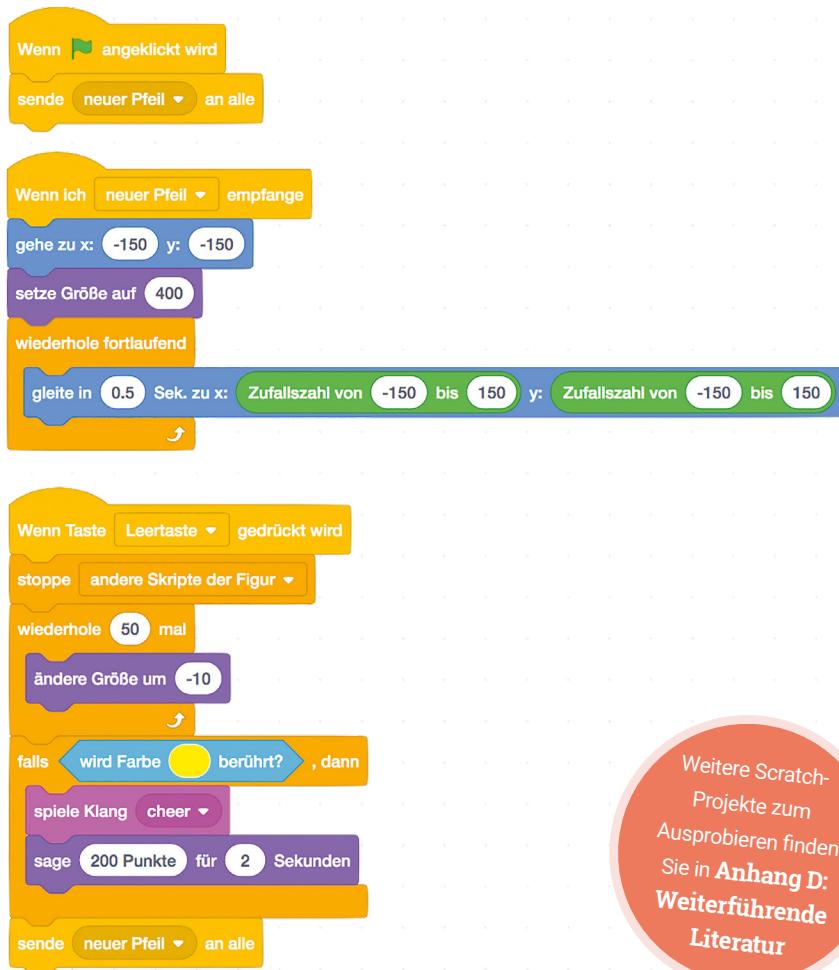


Damit das Spiel Spaß macht, müssen Sie eine Möglichkeit hinzufügen, den Punktestand festzuhalten. Immer noch im gleichen Stapel von Blöcken, fügen Sie einen Block **falls dann** hinzu,

wobei darauf zu achten ist, dass er unterhalb des Blocks **wiederhole 50 Mal** und nicht darin zu platzieren ist, mit einem **wird Farbe berührt?** „Fühlen“-Block in seiner rautenförmigen Lücke. Um die richtige Farbe zu wählen, klicken Sie auf das farbige Kästchen in der Mitte dieses „Fühlen“-Blocks, dann auf das Pipettensymbol und dann auf die gelbe Mitte Ihres Ziels auf der Bühne.



Damit der Spieler weiß, dass er einen Treffer erzielt hat, fügen Sie einen Block **spielle Klang cheer** und einen Block **sage 200 Punkte für 2 Sekunden** innerhalb des Blocks **falls dann** hinzu. Schließlich fügen Sie einen Block **sende neuer Pfeil an alle** ganz unten auf dem Blockstapel, unterhalb und außerhalb des Blocks **falls dann**, um dem Spieler bei jedem Schuss einen weiteren Pfeil zu geben. Klicken Sie auf die grüne Flagge, um Ihr Spiel zu beginnen, und versuchen Sie, die gelbe Mitte zu treffen: Wenn Sie das schaffen, werden Sie mit dem Jubel der Menge und 200 Punkten belohnt!



Weitere Scratch-Projekte zum Ausprobieren finden Sie in **Anhang D: Weiterführende Literatur**

Das Spiel funktioniert, ist aber etwas schwierig. Verwenden Sie das in diesem Kapitel Gelernte und versuchen Sie, das Spiel zu erweitern, um Punkte für das Treffen anderer Teile der Zielscheibe als dem Volltreffer hinzuzufügen: 100 Punkte für Rot, 50 Punkte für Blau, und so weiter.



HERAUSFORDERUNG: KÖNNEN SIE ES VERBESSERN?



Wie würden Sie das Spiel einfacher machen? Wie würden Sie das Spiel schwieriger machen? Können Sie Variablen verwenden, um die Punktzahl der Spieler mit jedem abgefeuerten Pfeil weiter zu erhöhen? Können Sie einen Countdown-Timer hinzufügen, um mehr Druck auf den Spieler auszuüben?

Kapitel 5

Programmieren mit Python

Jetzt, wo Sie sich mit Scratch auskennen, zeigen wir Ihnen, wie man textbasierten Code in Python schreibt

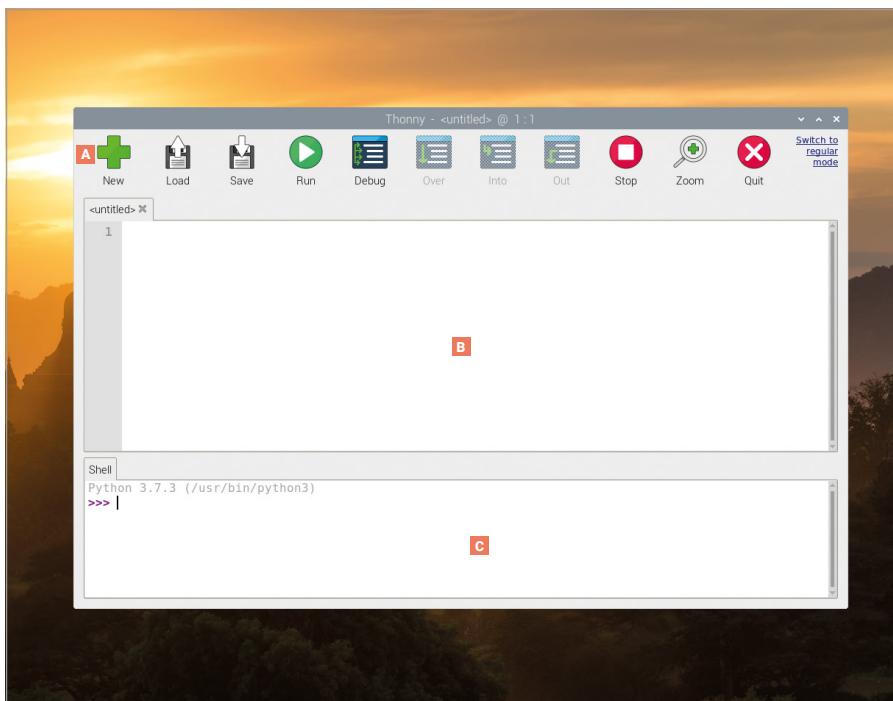


Die von Guido van Rossum nach der Comedy-Truppe Monty Python benannte Programmiersprache Python hat sich von einem Hobbyprojekt, das 1991 erstmals der Öffentlichkeit vorgestellt wurde, zu einer beliebten Programmiersprache entwickelt, die eine große Vielfalt an Projekten unterstützt. Im Gegensatz zur visuellen Umgebung von Scratch ist Python textbasiert: Sie schreiben Anweisungen in einer vereinfachten Sprache und einem spezifischen Format, die der Computer dann ausführt.

Python ist ein großartiger nächster Schritt für diejenigen, die Scratch bereits verwendet haben, da diese Sprache mehr Flexibilität und eine „traditionellere“ Programmierumgebung bietet. Das heißt aber nicht, dass mit größerem Lernaufwand zu rechnen ist: Mit ein wenig Übung kann jedermann Python-Programme für alles schreiben: von einfachen Berechnungen bis hin zu überraschend komplexen Spielen.

Dieses Kapitel baut auf Begriffen und Konzepten auf, die im **Kapitel 4, Programmierung mit Scratch 3** eingeführt wurden. Falls Sie die Übungen in diesem Kapitel noch nicht durchgearbeitet haben, werden Sie dieses Kapitel leichter nachvollziehen können, wenn Sie zurückgehen und dies zuerst tun.

Einführung in die Thonny-Python-IDE



A Symbolleiste – Die Thonny-

Benutzeroberfläche im „Einfachen Modus“ („simple mode“) verwendet eine Leiste mit sympathischen Symbolen als Menü, die es Ihnen erlaubt, eigene Python-Programme zu erstellen, zu speichern, zu laden und auszuführen, sowie sie auf verschiedene Weise zu testen.

geschrieben werden. Er ist in einen

Hauptbereich für Ihr Programm und einen kleinen Seitenrand zur Anzeige der Zeilennummern unterteilt.

B Skriptbereich – Der Skriptbereich ist der

Bereich, in dem Ihre Python-Programme

C Python-Shell – Die Python-Shell erlaubt die Eingabe individueller Anweisungen, die ausgeführt werden, sobald Sie die **EINGABETASTE** drücken. Sie liefert auch Infos über laufende Programme.

THONNY-VERSIONEN

Thonny hat zwei Versionen der Benutzeroberfläche: „Normaler Modus“ („regular mode“) und „Einfacher Modus“ („simple mode“), der sich für Anfänger besser eignet. Dieses Kapitel verwendet den „Einfachen Modus“, der standardmäßig geladen wird, wenn man Thonny über den Abschnitt „Entwicklung“ des Raspberry-Menüs öffnet.



Ihr erstes Python-Programm: Hallo, Welt!

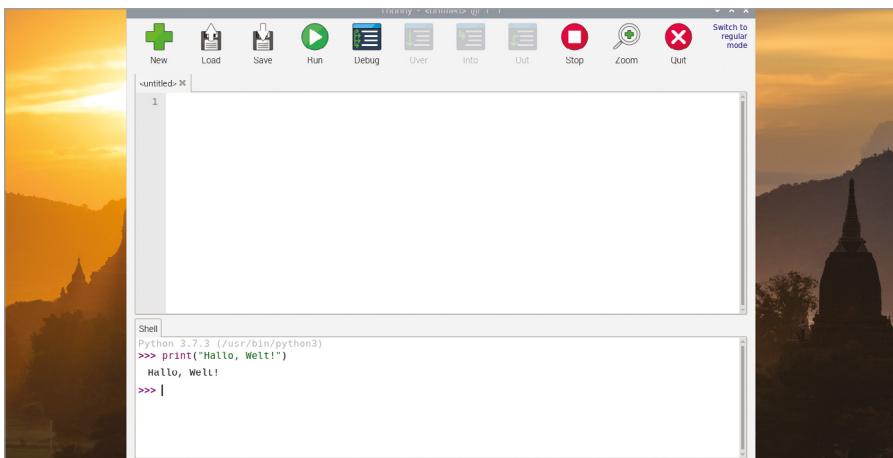
Wie die anderen vorinstallierten Programme auf dem Raspberry Pi ist Thonny über das Menü verfügbar: Klicken Sie auf das Raspberry-Symbol, bewegen Sie den Cursor zum Abschnitt „Entwicklung“ und klicken Sie auf Thonny Python IDE. Nach einigen Sekunden wird die Thonny Benutzeroberfläche (standardmäßig im „Simple Mode“ (Einfacher Modus)) geladen.

Thonny ist ein Paket, das als *integrierte Entwicklungsumgebung (IDE)* bekannt ist. Dieser kompliziert klingende Name hat eine einfache Erklärung: Das Paket vereint oder *integriert* alle Tools, die Sie zum Schreiben oder *Entwickeln* von Software in einer zentralen Benutzeroberfläche oder *Umgebung* benötigen. Es sind viele IDEs verfügbar. Einige davon unterstützen viele verschiedene Programmiersprachen, während andere, wie Thonny, sich auf die Unterstützung einer einzigen Sprache konzentrieren.

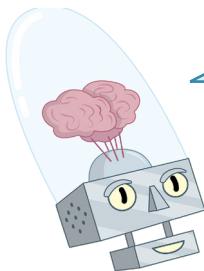
Im Gegensatz zu Scratch, das Ihnen visuelle Bausteine als Grundlage für Ihr Programm bereitstellt, ist Python eine traditionellere Programmiersprache, bei der alles als Text geschrieben wird. Starten Sie Ihr erstes Programm, indem Sie auf den Python-Shell-Bereich unten links im Thonny-Fenster klicken und dann die folgende Anweisung eingeben, bevor Sie anschließend die **EINGABETASTE** drücken:

```
print("Hallo Welt!")
```

Wenn Sie die **EINGABETASTE** drücken, sehen Sie, dass Ihr Programm sofort ausgeführt wird: Python antwortet im selben Shell-Bereich mit der Nachricht „Hallo, Welt! **Abbildung 5-1**“, ganz wie gewünscht. Das liegt daran, dass die Shell eine direkte Verbindung zum Python-*Interpretierprogramm* darstellt, dessen Aufgabe es ist, Ihre Anweisungen entgegenzunehmen und zu *interpretieren*, das heißt ihre Bedeutung umzusetzen. Es handelt sich dabei um den *interaktiven Modus*. Stellen Sie sich das wie ein persönliches Gespräch vor: Sobald Sie eine Aussage formuliert haben, antwortet Ihr Gesprächspartner und wartet auf das, was Sie entgegen werden.



▲ Abbildung 5-1: Python gibt die Nachricht „Hallo, Welt!“ im Shell-Bereich aus



SYNTAXFEHLER



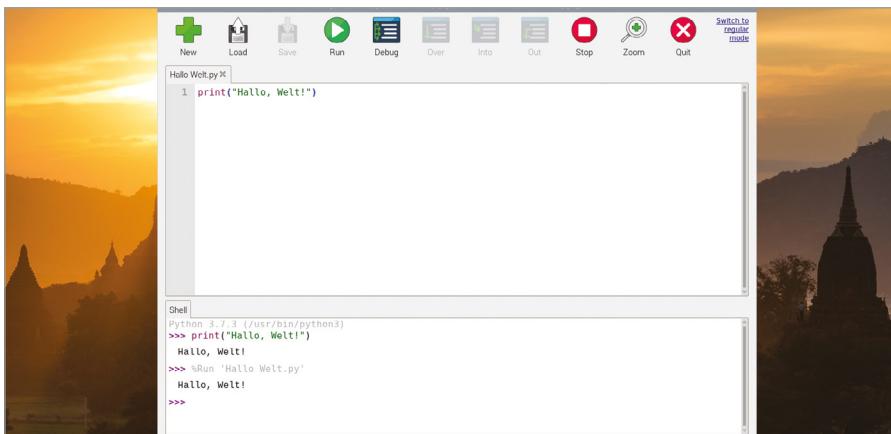
Falls Ihr Programm nicht läuft, sondern stattdessen eine `SyntaxError` („Syntaxfehler“)-Meldung an den Shell-Bereich ausgibt, dann liegt irgendwo in dem, was Sie geschrieben haben, ein Fehler vor. Python erwartet seine Anweisungen in einer ganz bestimmten Art und Weise: Lässt man mal eine Klammer oder ein Anführungszeichen weg, schreibt "print" falsch oder zum Beispiel mit einem großen P, oder fügt man irgendwo in der Anweisung zusätzliche Symbole hinzu, wird es nicht funktionieren. Geben Sie die Anweisung erneut ein und achten Sie darauf, dass sie mit der Schreibweise in diesem Buch exakt übereinstimmt, bevor Sie die **EINGABETASTE** drücken!

Python muss jedoch nicht zwangsläufig immer im interaktiven Modus verwendet werden. Klicken Sie auf den Skriptbereich im Thonny-Fenster und geben dann das Programm erneut ein:

```
print("Hallo Welt!")
```

Wenn Sie dieses Mal die **EINGABETASTE** drücken, passiert nichts – außer, dass im Skriptbereich eine neue, leere Zeile angezeigt wird. Um diese Version Ihres Programms zum Laufen zu bringen müssen Sie auf das Symbol „Run“ ((Ausführen)) in der Thonny-Symbolleiste klicken. Wenn Sie dies tun, werden Sie zunächst aufgefordert, das Programm zu speichern. Geben Sie einen beschreibenden Begriff ein, z. B. „Hallo Welt“, und klicken Sie auf den Button „Save“ (Speichern). Sobald das Programm gespeichert wurde, Sehen Sie zwei Meldungen im Python-Shell-Bereich (**Abbildung 5-2**):

```
>>> %Run 'Hallo Welt.py'
Hallo Welt!
```



▲ Abbildung 5-2: So bringen Sie Ihr kleines Programm zum Laufen

Die erste dieser Zeilen ist eine Anweisung von Thonny, die den Python-Interpreter anweist, das soeben gespeicherte Programm auszuführen. Die zweite Zeile enthält die Ausgabe des Programms – die Nachricht die Sie Python angewiesen haben auszugeben. Herzlichen Glückwunsch: Sie haben gerade Ihr erstes Python-Programm sowohl im interaktiven als auch im Skript-Modus geschrieben und ausgeführt!



HERAUSFORDERUNG: NEUE MELDUNG



Können Sie die Meldung ändern, die das Python-Programm ausgibt?

Falls Sie weitere Meldungen hinzufügen wollen, würden Sie dafür den interaktiven Modus oder den Skriptmodus verwenden? Was passiert, wenn Sie die Klammern oder Anführungszeichen aus dem Programm entfernen und dann versuchen, es erneut auszuführen?.

Nächste Schritte: Schleifen und Code-Einrückung

Genauso wie Scratch Stapel von puzzleartigen Blöcken verwendet, um zu kontrollieren, welche Teile des Programms miteinander verbunden sind, so kontrolliert Python auf seine eigene Weise die Reihenfolge, in der seine Programme ausgeführt werden: mit der *Einrückung*. Erstellen Sie ein neues Programm, indem Sie auf das Symbol „New“ (Neu) in der Thonny-Symbolleiste klicken. Sie verlieren Ihr bestehendes Programm nicht. Stattdessen wird Thonny eine neue Registerkarte oberhalb des Skriptbereichs erstellen. Geben Sie zunächst folgenden Text ein:

```
print("Schleife beginnt!")
for i in range(10):
```

Die erste Zeile gibt eine einfache Nachricht an die Shell aus, genau wie Ihr „Hallo Welt“-Programm. Die zweite startet eine *endliche Schleife*, die genauso funktioniert wie in Scratch: ein Zähler – **i** – wird der Schleife zugeordnet und mit einer Reihe von Zahlen versehen. Die Anweisung **range** gibt diesen Zahlenbereich vor, der mit 0 beginnt und aufwärts in Richtung der Zahl 10 zählt, jedoch stets vor Erreichen der Zahl 10 endet. Das Doppelpunkt-Symbol (:) teilt Python mit, dass die nächste Anweisung Teil der Schleife sein soll.

Bei Scratch sind die Anweisungen, die in die Schleife aufgenommen werden sollen, buchstäblich im Inneren des C-förmigen Blocks enthalten. Python verfolgt einen anderen Ansatz: Einrücken von Codezeilen. Die nächste Zeile beginnt mit vier Leerzeichen, die Thonny selbstständig hinzugefügt haben sollte, als Sie nach Zeile 2 die **EINGABETASTE** gedrückt haben:

```
print("Schleife Nummer", i)
```

Die Leerzeichen schieben diese neue Zeile im Verhältnis zu den anderen Zeilen nach innen. Durch diese Einrückung macht Python den Unterschied zwischen Anweisungen außerhalb der Schleife und Anweisungen innerhalb der Schleife deutlich. Der eingerückte Code wird als *verschachtelt* bezeichnet.

Nachdem Sie die **EINGABETASTE** am Ende der dritten Zeile gedrückt haben, werden Sie feststellen, dass Thonny die nächste Zeile automatisch einrückt, in der Annahme, dass sie ebenso Teil der Schleife wird. Um die Einrückung zu entfernen, drücken Sie einfach einmal die **RÜCKTASTE**, bevor Sie die vierte Zeile eingeben:

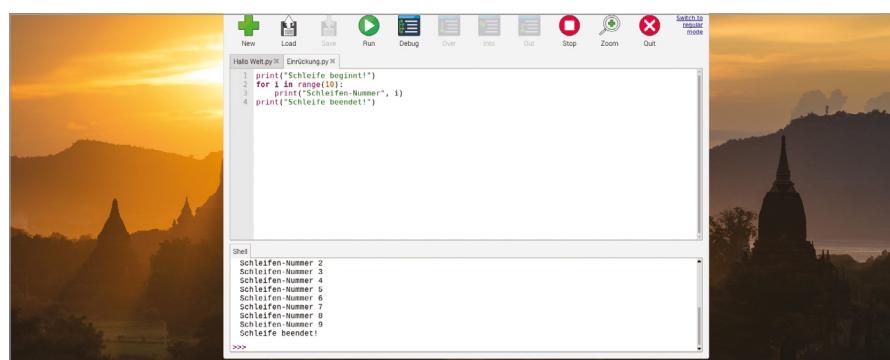
```
print("Schleife beendet!")
```

Ihr Vier-Zeilen-Programm ist jetzt abgeschlossen. Die erste Zeile sitzt außerhalb der Schleife und läuft nur einmal; die zweite Zeile bildet die Schleife; die dritte Zeile sitzt innerhalb der Schleife und wird in jedem Durchlauf der Schleife einmal ausgeführt; und die vierte Zeile sitzt wiederum außerhalb der Schleife.

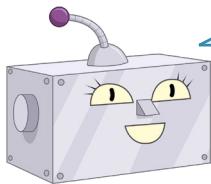
```
print("Schleife beginnt!")
for i in range (10):
    print("Schleife Nummer", i)
print("Schleife beendet!")
```

Klicken Sie auf das Symbol „Run“ zum Ausführen, speichern Sie das Programm unter dem Namen **Einrückung** und schauen Sie sich den Shell-Bereich für dessen Ausgabe an (**Abbildung 5-3**):

```
Schleife beginnt!
Schleife Nummer 0
Schleife Nummer 1
Schleife Nummer 2
Schleife Nummer 3
Schleife Nummer 4
Schleife Nummer 5
Schleife Nummer 6
Schleife Nummer 7
Schleife Nummer 8
Schleife Nummer 9
Schleife beendet!
```



▲ Abbildung 5-3: Ausführen einer Schleife



MIT DEM ZÄHLEN BEI NULL BEGINNEN



Python ist eine nullindizierte Sprache: Das bedeutet, dass sie beim Zählen mit der 0 und nicht der 1 beginnt. Das Programm gibt daher die Zahlen 0 bis 9 und nicht 1 bis 10 aus. Sie könnten dieses Verhalten auch ändern, indem Sie die Anweisung `range(10)` in `range(1, 11)` – oder jede andere beliebige Zahl – ändern.

Das Einrücken ist ein wichtiger und mächtiger Teil von Python und einer der häufigsten Gründe dafür, dass ein Programm nicht erwartungsgemäß funktioniert. Bei der Fehlersuche in einem Programm – ein Vorgang, der als *Debugging* bezeichnet wird, – sollten Sie stets noch einmal die Einrückung überprüfen, vor allem, wenn Sie mit der anfangen Schleifen innerhalb von Schleifen zu verschachteln.

Python unterstützt auch *Endlosschleifen*. Um Ihr Programm von einer endlichen Schleife in eine unendliche (Endlos-) Schleife zu ändern, bearbeiten Sie Zeile 2 wie folgt:

while True:

Wenn Sie jetzt auf das Symbol „Run“ zum Ausführen klicken, erhalten Sie eine Fehlermeldung: **name 'i' is not defined**. Das liegt daran, dass Sie die Zeile gelöscht haben, die den Wert erstellt und der Variablen **i** zugewiesen hat. Um dies zu beheben, überarbeiten Sie die Zeile 3 so, dass die Variable nicht mehr verwendet wird:

print("Schleife wird ausgeführt!")

Klicken Sie auf das Symbol „Run“. Ganz kurz sieht man dann die Meldung „Schleife beginnt!“, gefolgt von einer nicht enden wollenden Reihe von „Schleife wird ausgeführt“-Meldungen (**Abbildung 5-4**). Die Meldung „Schleife beendet!“ wird niemals ausgegeben, weil die Schleife endlos ist: Jedes Mal, wenn Python die Meldung „Schleife wird ausgeführt!“ ausgegeben hat, kehrt es zum Anfang der Schleife zurück und gibt die Meldung erneut aus.

```
print("Schleife beginnt!")
while True:
    print("Schleife läuft!")
    print("Schleife läuft!")
```

▲ Abbildung 5-4: Eine Endlosschleife läuft so lange, bis Sie das Programm stoppen

Klicken Sie auf das Stop-Symbol  in der Thonny-Symbolleiste, um dem Programm mitzuteilen, dass es die Ausführung anhalten soll. Dies wird als Unterbrechung des Programms bezeichnet. Es erscheint eine Meldung im Bereich der Python-Shell und das Programm wird beendet – und zwar ohne jemals Zeile 4 zu erreichen.



HERAUSFORDERUNG:
LASSEN SIE DIE SCHLEIFE LAUFEN

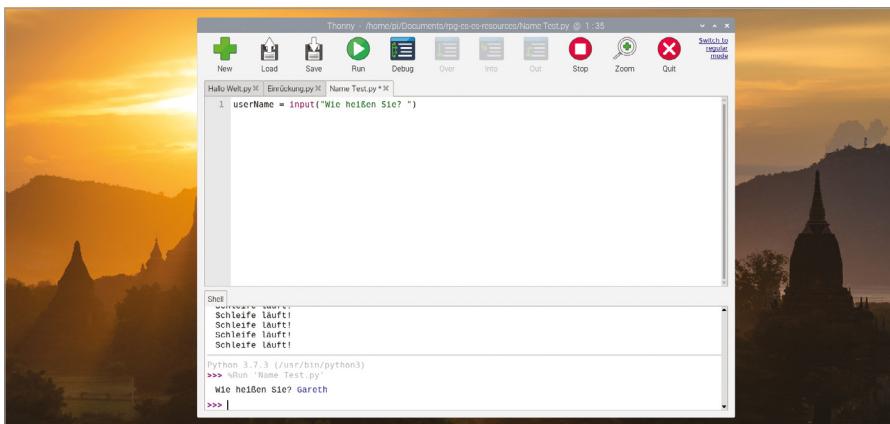
Können Sie die Schleife wieder zurück in eine endliche Schleife ändern? Können Sie eine zweite endliche Schleife in das Programm einfügen? Wie würden Sie eine Schleife innerhalb einer Schleife hinzufügen? Wie würden Sie erwarten dass Ihr Programm sich damit verhält?

Bedingungen und Variablen

Wie in allen Programmiersprachen dienen Variablen nicht nur dazu, Schleifen zu steuern. Starten Sie ein neues Programm, indem Sie im Thonny-Menü auf das Symbol  (Neu) klicken, und geben Sie dann Folgendes in den Skriptbereich ein:

```
userName = input ("Wie heißen Sie? ")
```

Klicken Sie auf das Symbol „Run“, speichern Sie das Programm unter **Name Test** und achten dann auf das, was im Shell-Bereich passiert: Sie werden nach Ihrem Namen gefragt. Geben Sie Ihren Namen in den Shell-Bereich ein und drücken Sie dann die **EINGABETASTE**. Da das die einzige Anweisung in Ihrem Programm ist, wird nichts weiter passieren (**Abbildung 5-5**). Wenn Sie mit den Daten, die Sie in die Variable geschrieben haben, tatsächlich etwas anfangen wollen, dann benötigt das Programm noch weitere Zeilen.



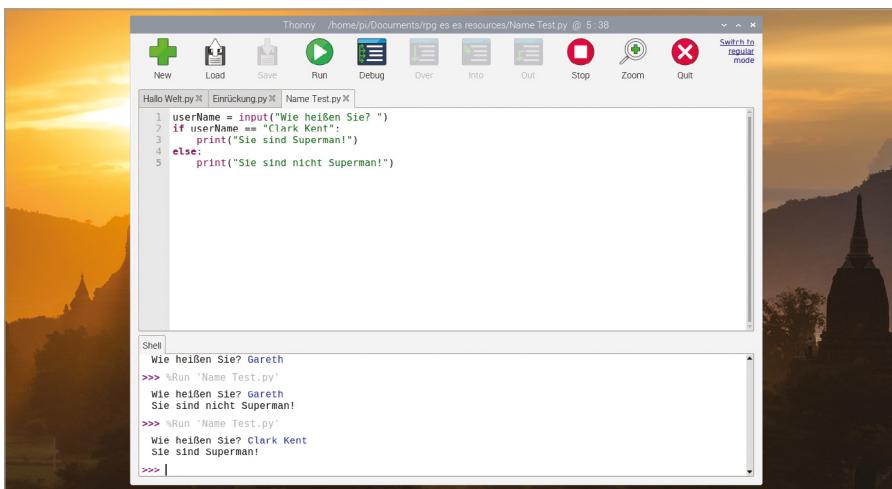
► Abbildung 5-5: Mit der `input`-Funktion können Sie einen Benutzer zur Eingabe von Text auffordern

Um das Programm dazu zu bringen, etwas Nützliches mit dem Namen anzufangen, fügen Sie eine *bedingte Anweisung* hinzu, indem Sie Folgendes eingeben:

```
if userName == "Clark Kent":  
    print("Sie sind Superman!")  
else:  
    print("Sie sind nicht Superman!")
```

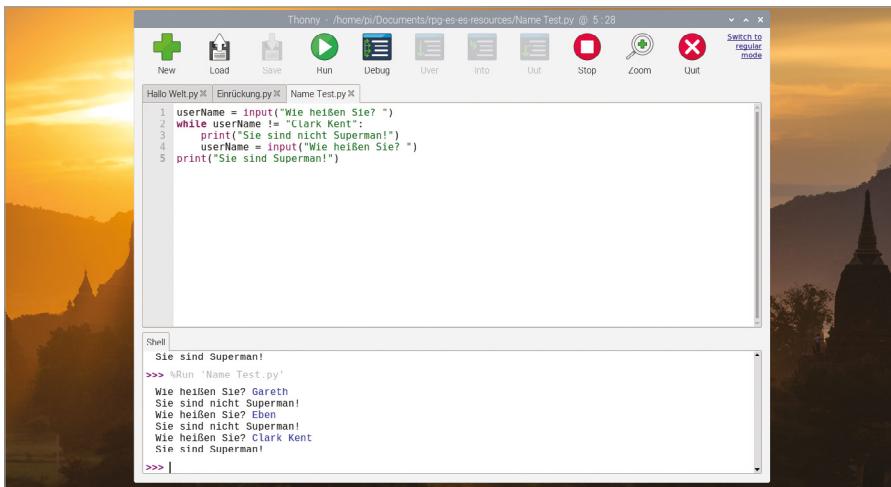
Denken Sie daran: Wenn Thonny feststellt, dass Ihr Code eingerückt werden muss, macht es das automatisch. Aber Thonny weiß nicht, wo die Einrückung enden soll, sodass Sie die Leerzeichen von Hand löschen müssen.

Klicken Sie auf das Symbol „Run“ und geben Sie Ihren Namen in den Shell-Bereich ein. Wenn Ihr Name nicht zufälligerweise Clark Kent ist, bekommen Sie die Nachricht „Sie sind nicht Superman!“. Klicken Sie erneut auf „Run“ und geben Sie dieses Mal den Namen ‚Clark Kent‘ ein. Achten Sie darauf, dass er genau wie im Programm geschrieben wird, mit einem großen C und K. Dieses Mal erkennt das Programm, dass Sie tatsächlich Superman sind (**Abbildung 5-6**).

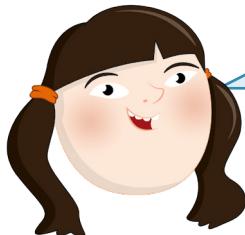


▲ Abbildung 5-6: Sollten Sie nicht gerade irgendwo die Welt retten?

Die `==`-Symbole weisen Python an, einen direkten Vergleich durchzuführen, um festzustellen, ob der Wert der Variablen `userName` dem Text – bekannt als *String* oder *Zeichenfolge* – in Ihrem Programm entspricht. Bei der Arbeit mit Zahlen, gibt es weitere Vergleiche, die man anstellen kann: `>`, um festzustellen, ob eine Zahl größer als eine andere ist, `<` um festzustellen, ob sie kleiner ist, `=` um festzustellen, ob sie gleich oder größer ist, `=<` um festzustellen, ob sie gleich oder kleiner ist. Außerdem gibt es noch `!=`. Das bedeutet „nicht gleich“ – und ist das genaue Gegenteil von `==`. Diese Symbole werden in der Fachsprache als *Vergleichsoperatoren* bezeichnet.



▲ Abbildung 5-7: Das Programm fragt so lange nach Ihrem Namen, bis Sie ‚Clark Kent‘ eingeben



VERWENDEN VON = UND ==

Bei der Verwendung von Variablen kommt es darauf an, den Unterschied zwischen `=` und `==` zu verstehen. Merken Sie sich: `=` steht für „Variable gleich diesem Wert machen“, während `==` für „prüfen, ob die Variable gleich diesem Wert ist“ steht. Wenn Sie das verwechseln, können Sie davon ausgehen, dass Ihr Programm nicht funktionieren wird!



Vergleichsoperatoren können auch in Schleifen verwendet werden. Löschen Sie die Zeilen 2 bis 5 und geben Sie dann an ihrer Stelle Folgendes ein:

```

while(userName != "Clark Kent"):
    print("Sie sind nicht Superman!")
    userName = input("Wie heißen Sie? ")
print("Sie sind Superman!")

```

Klicken Sie erneut auf das Symbol „Run“. Dieses Mal wird das Programm nicht aufhören, sondern immer weiter nach Ihrem Namen fragen, bis es bestätigen kann, dass Sie Superman sind (**Abbildung 5-7**). Damit haben Sie effektiv eine Art sehr einfacher Kennwortabfrage programmiert. Um aus der Schleife herauszukommen, geben Sie entweder ‚Clark Kent‘ ein oder klicken Sie auf das Symbol „Stop“ in der Thonny-Symbolleiste. Herzlichen Glückwunsch: Sie wissen jetzt, wie man Bedingungen und Vergleichsoperatoren verwendet!



HERAUSFORDERUNG: WEITERE FRAGEN HINZUFÜGEN



Können Sie das Programm dahingehend ändern, dass es mehrere Fragen stellt und die Antworten in mehreren Variablen ablegt?
Können Sie ein Programm erstellen, das Bedingungen und Vergleichsoperatoren verwendet, um dann auszugeben, ob eine vom Benutzer eingegebene Zahl größer oder kleiner als 5 ist, wie schon im Programm, das Sie in **Kapitel 4, Programmieren mit Scratch 3** erstellt haben?

Projekt 1: Turtle-Schneeflocken

Jetzt verstehen Sie, wie Python funktioniert. Und nun ist an der Zeit, mit Grafiken zu spielen und eine Schneeflocke mit einem Tool namens *Turtle* zu erstellen.

ONLINE-PROJEKT



Dieses Projekt ist auch online verfügbar unter rpf.io/turtle-snowflakes.

Schildkröten (engl. Turtles) sind ursprünglich Roboter in der Form ihrer tierischen Namensvetter, und so konstruiert, dass sie sich in einer geraden Linie bewegen, sich drehen und einen Stift heben und senken können – in der digitalen Version bedeutet dies einfach, dass sie mit dem Zeichnen einer Linie beginnen oder aufhören, während sie sich bewegen. Im Gegensatz zu einigen anderen Sprachen, namentlich Logo und seinen vielen Varianten, enthält Python kein integriertes Turtle-Tool – aber es wird mit einer *Bibliothek* von Add-On-Code geliefert, die ihm diese Turtle-Fähigkeiten verleihen. Bibliotheken sind Code-Bündel, die Anweisungen enthalten, mit denen sich die Fähigkeiten von Python erweitern und sich daher auch in eigene Programme mithilfe eines import-Befehls integrieren lassen.

Erstellen Sie ein neues Programm, indem Sie auf das „New“-Symbol (Neu) klicken und Folgendes eingeben:

```
import turtle
```

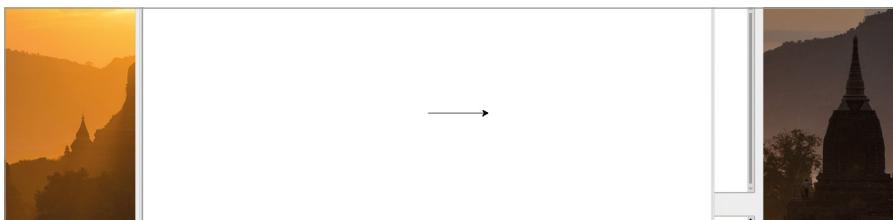
Falls Sie Anweisungen verwenden, die in einer Bibliothek enthalten sind, müssen Sie den Bibliotheksnamen gefolgt von einem Punkt und anschließend den Namen der Anweisung verwenden. Es kann ziemlich nervig sein, all das jedes Mal aufs Neue einzugeben. Sie können stattdessen einen kürzeren Variablennamen vergeben. Ein einzelner Buchstabe würde ausreichen. Aber wir fanden es netter, der Schildkröte damit gleich einen eigenen Namen zu geben, Pat. Geben Sie Folgendes ein:

```
pat = turtle.Turtle()
```

Um das Programm zu testen, müssen Sie der Schildkröte eine Aufgabe geben. Geben Sie ein:

```
pat.forward(100)
```

Klicken Sie auf das Symbol „Run“, und speichern Sie das Programm unter **Turtle-Schneeflocken**. Nach dem Speichern wird ein neues Fenster mit der Bezeichnung „Python Turtle Graphics“ angezeigt, in dem das Ergebnis des Programms zu sehen ist: Pat, Ihre Schildkröte, bewegt sich 100 Einheiten vorwärts und zeichnet dabei eine gerade Linie (**Abbildung 5-8**).



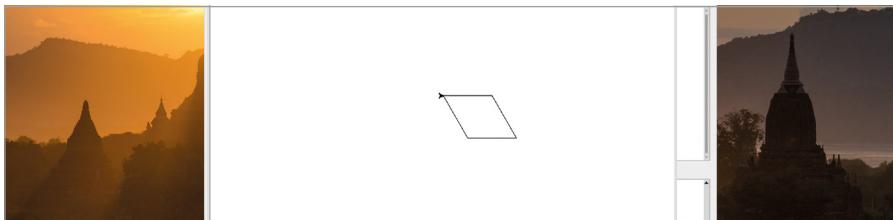
▲ Abbildung 5-8: Die Schildkröte bewegt sich vorwärts und zeichnet dabei eine gerade Linie

Gehen Sie zurück zum Thonny-Hauptfenster. Sollte es hinter dem Fenster der Turtle-Grafik verborgen sein, klicken Sie entweder auf den Button „Minimieren“ im Fenster „Python Turtle Graphics“ oder auf den Thonny-Eintrag in der Taskleiste am oberen Bildschirmrand. Klicken Sie auf den Button „Stop“, um das Fenster Turtle Graphics zu schließen.

Es wäre mühsam, jede einzelne Bewegungsanweisung manuell einzugeben. Löschen Sie daher Zeile 3 und erstellen Sie eine Schleife, die Formen zeichnet:

```
for i in range(2):
    pat.forward(100)
    pat.right(60)
    pat.forward(100)
    pat.right(120)
```

Wenn Sie das Programm ausführen, wird Pat ein einzelnes Parallelogramm zeichnen (**Abbildung 5-9**).



▲ Abbildung 5-9: Durch die Kombination aus Drehungen und Bewegungen lassen sich Formen zeichnen

Um diesen eine Schneeflocken-Form zu verleihen, klicken Sie auf das Stop-Symbol im Thonny-Hauptfenster und erstellen Sie eine Schleife um Ihre Schleife herum, indem Sie folgende Zeile als Zeile 3 hinzufügen:

```
for i in range(10):
```

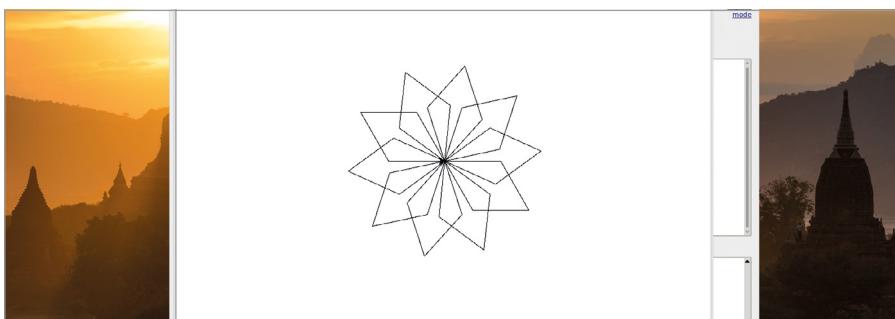
Und fügen Sie folgenden Zeilen an das Ende des Programms an:

```
    pat.right(36)
```

Ihr Programm wird in dieser Form nicht laufen, da die darin enthaltene Schleife nicht korrekt eingerückt wurde. Um dies zu korrigieren, klicken Sie auf den Anfang jeder Zeile der Schleife – Zeilen 4 bis 8 – und drücken Sie die **LEERTASTE** viermal für die richtige Einrückung. Ihr Programm sollte jetzt folgendermaßen aussehen:

```
import turtle
pat = turtle.Turtle()
for i in range(10):
    for i in range(2):
        pat.forward(100)
        pat.right(60)
        pat.forward(100)
        pat.right(120)
    pat.right(36)
```

Klicken Sie auf das Symbol „Run“ und achten Sie auf die Schildkröte: Sie zeichnet ein Parallelogramm wie schon zuvor. Sobald sie aber damit fertig ist, dreht sie sich um 36 Grad und zeichnet ein weiteres, dann noch ein weiteres und so weiter, bis zehn, einander überlappende, Parallelogramme auf dem Bildschirm erscheinen, die ein wenig an eine Schneeflocke erinnern (**Abbildung 5-10**).

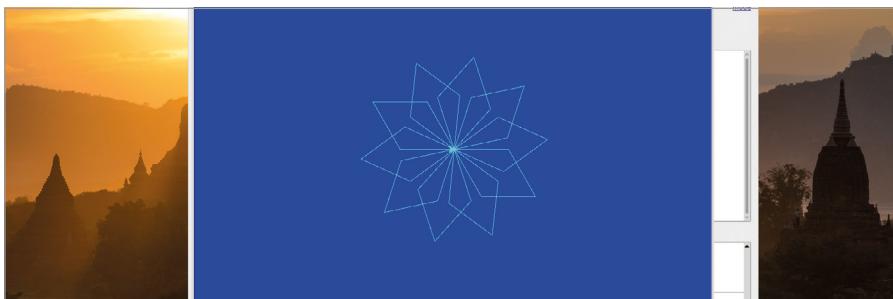


▲ Abbildung 5-10: Die Form wird wiederholt, um eine komplexere Form zu erhalten

Während eine Roboterschildkröte in einer einzigen Farbe auf ein großes Stück Papier zeichnet, kann die simulierte Python-Schildkröte eine ganze Reihe von Farben verwenden. Fügen Sie die neuen Zeilen 3 und 4 ein, wobei die vorhandenen Zeilen nach unten verschoben werden:

```
turtle.Screen().bgcolor("blue")
pat.color("cyan")
```

Führen Sie das Programm erneut aus. Sie sehen nun, was der neue Code bewirkt hat: Die Hintergrundfarbe des Turtle-Grafik-Fensters ist jetzt blau und die Schneeflocke hat die Farbe Cyan angenommen (**Abbildung 5-11**).



▲ Abbildung 5-11: Ändern der Hintergrund- und Schneeflockenfarben

Sie können die Farben auch zufällig aus einer Liste auswählen lassen, indem Sie die Bibliothek **random** verwenden. Gehen Sie wieder an den Anfang des Programms und fügen Sie Folgendes als Zeile 2 ein:

```
import random
```

Ändern Sie die Hintergrundfarbe in der jetzigen Zeile 4 von "blue" (Blau) in "grey" (Grau) und erstellen Sie anschließend eine neue Variable mit der Bezeichnung „colours“ (Farben), indem Sie eine neue Zeile 5 einfügen:

```
colours = ["cyan", "purple", "white", "blue"]
```



U.S.-RECHTSCHREIBUNG

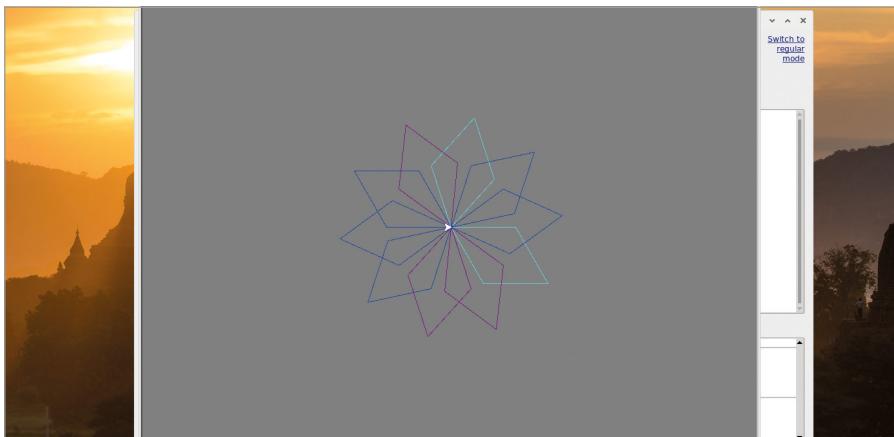
Viele Programmiersprachen verwenden die Schreibweise des amerikanischen Englisch, und Python ist da keine Ausnahme: Der Befehl zum Ändern der Farbe des Schildkrötenstifts wird color geschrieben. Falls Sie also versuchen die britische Variante des Wortes, colour, zu verwenden, wird es einfach nicht funktionieren. Variablen können jedoch beliebig geschrieben werden. Deshalb kann man eine neue Variable auch colours nennen und Python wird es verstehen.



Diese Variablen-Art wird als Liste bezeichnet und wird durch eckige Klammern kenntlich gemacht. In diesem Fall ist die Liste mit möglichen Farben für die Schneeflockensegmente gefüllt. Sie müssen Python jetzt noch anweisen, bei jedem Schleifendurchlauf eine Farbe davon zu wählen. Geben Sie ganz am Ende des Programms Folgendes ein und achten Sie darauf, dass die Eingabe mit vier Leerzeichen eingerückt ist, sodass sie Teil der äußeren Schleife ist, genau wie die Zeile darüber:

```
pat.color(random.choice(colours))
```

Klicken Sie auf das Symbol „Run“, und der Schneeflocken-Ninja-Stern wird erneut gezeichnet. Dieses Mal wählt Python jedoch beim Zeichnen jedes Elements eine zufällige Farbe aus Ihrer Liste aus. Die Schneeflocke erhält damit ein ansprechendes, mehrfarbiges Aussehen (**Abbildung 5-12**).



▲ Abbildung 5-12: Verwendung von zufälligen Farben für die Flockenelemente

Damit die Schneeflocke nicht wie ein Ninja-Stern, sondern wie eine echte Schneeflocke aussieht, fügen Sie eine neue Zeile 6 direkt unter Ihre Liste **colours** hinzu und geben Sie Folgendes ein:

```
pat.penup()  
pat.forward(90)  
pat.left(45)  
pat.pendown()
```

Die Anweisungen **penup** und **pendown** würden bei Verwendung eines Schildkrötenroboters einen Stift vom Papier weg und wieder auf dieses hin bewegen, in der virtuellen Welt jedoch weisen Sie die Schildkröte einfach an, aufzuhören Linien zu zeichnen oder damit

wieder anzufangen. Anstatt einer Schleife verwenden wir dieses Mal eine *Funktion* – ein Codesegment, das Sie jederzeit aufrufen können, vergleichbar damit Ihre persönliche Python-Anweisung zu erschaffen.

Beginnen Sie zunächst damit, den Code zum Zeichnen der parallelogrammförmigen Schneeflocken zu löschen: das umfasst alles zwischen und einschließlich der Anweisung `pat.color("cyan")` in Zeile 10 bis `pat.right(36)` in Zeile 17. Die Anweisung `pat.color(random.choice(colours))` bleibt erhalten, setzen Sie aber ein Rauten-Symbol (#) an den Anfang der Zeile. Dies ist bekannt als *Auskommentieren* einer Anweisung und bedeutet, dass Python diese dann ignoriert. Sie können Kommentare verwenden, um Erläuterungen zu den Codezeilen hinzuzufügen. Auf diese Weise bleiben diese leichter verständlich, falls Sie nach einigen Monaten wieder darauf zurückkommen oder sie an Dritte weitergeben wollen!

Erstellen Sie Ihre Funktion, die wir ‚branch‘ nennen werden, indem Sie folgende Anweisung in Zeile 10 unterhalb von `pat.pendown()` eingeben:

```
def branch():
```

Damit definieren Sie Ihre Funktion `branch`. Wenn Sie die **EINGABETASTE** drücken, fügt Thonny automatisch eine Einrückung für die Anweisungen der Funktion ein. Geben Sie Folgendes ein und achten Sie dabei genau auf die Einrückung, weil Sie hierbei den Code bis zu drei Einrückungsebenen tief verschachteln!

```
for i in range(3):
    for i in range(3):
        pat.forward(30)
        pat.backward(30)
        pat.right(45)
        pat.left(90)
        pat.backward(30)
        pat.left(45)
    pat.right(90)
    pat.forward(90)
```

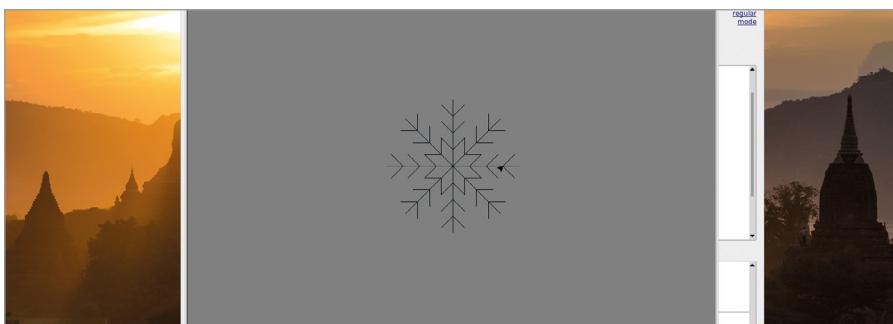
Erstellen Sie schließlich eine neue Schleife am Ende des Programms, jedoch oberhalb der auskommentierten "color"-Zeile, um die neue Funktion auszuführen bzw. aufzurufen (*call*):

```
for i in range(8):
    branch()
    pat.left(45)
```

Das fertige Programm sollte so aussehen:

```
import turtle
import random
pat = turtle.Turtle()
turtle.Screen().bgcolor("grey")
colours = ["cyan", "purple", "white", "blue"]
pat.penup()
pat.forward(90)
pat.left(45)
pat.pendown()
def branch():
    for i in range(3):
        for i in range(3):
            pat.forward(30)
            pat.backward(30)
            pat.right(45)
        pat.left(90)
        pat.backward(30)
        pat.left(45)
    pat.right(90)
    pat.forward(90)
for i in range(8):
    branch()
    pat.left(45)
#    pat.color(random.choice(colours))
```

Klicken Sie auf „Run“ und schauen Sie zu, wie Pat im „Graphics“-Fenster Ihre Anweisungen ausführt und zeichnet. Herzlichen Glückwunsch: Ihre Schneeflocke sieht jetzt wirklich wie eine Schneeflocke aus (**Abbildung 5-13**)!



▲ Abbildung 5-13: Zusätzliche Zweige (engl. branches) machen daraus eine Schneeflocke



HERAUSFORDERUNG: WAS FOLGT ALS NÄCHSTES?



Können Sie Ihre auskommentierte Anweisung verwenden, um die Elemente der Schneeflocke in verschiedenen Farben zeichnen zu lassen? Können Sie eine „schneeflocke“-Funktion erstellen und damit viele Schneeflocken auf den Bildschirm zeichnen? Können Sie in Ihrem Programm die Größe und Farbe der Schneeflocken zufällig variiieren lassen?

Projekt 2: Das unheimliche Vexier-Suchbild

Python kann genauso gut wie auf Schildkröten basierende Grafiken auch Bilder und Klänge verarbeiten, die sich recht wirkungsvoll auch für einen Streich für Ihre Freunde verwenden lassen – ein Spiel, bei dem Unterschiede gesucht werden müssen, mit einem gruseligen Clou, perfekt für Halloween!

ONLINE-PROJEKT



Dieses Projekt ist auch online verfügbar unter rpf.io/scary-spot

Dieses Projekt benötigt zwei Bilder: Ihr Suchbild sowie ein „gruseliges“ Überraschungsbild und eine Audiodatei. Klicken Sie auf das Raspberry-Symbol, um das Menü des Raspberry Pi OS zu laden. Wählen Sie die Kategorie „Internet“ und klicken Sie auf den Chromium-Webbrowser. Nachdem er geladen wurde, geben Sie **rpf.io/spot-pic** in die Adresszeile ein und drücken Sie die **EINGABETASTE**. Klicken Sie mit der rechten Maustaste auf das Bild und anschließend auf „Bild speichern unter ...“. Wählen Sie den Ordner „Persönlicher Ordner“ aus der Liste auf der linken Seite und klicken Sie dann auf „Speichern“. Gehen Sie zurück zur Adressleiste von Chromium und geben Sie **rpf.io/scary-pic** ein, gefolgt von der **EINGABETASTE**. Klicken Sie – wie schon zuvor – mit der rechten Maustaste auf das Bild und auf „Bild speichern unter ...“. Wählen Sie den Ordner „Persönlicher Ordner“ aus und klicken Sie dann auf „Speichern“.

Die Audiodatei schließlich bekommen Sie, indem Sie zurück in die Adressleiste gehen und **rpf.io/scream** eingeben, gefolgt von der **EINGABETASTE**. Diese Datei enthält einen Schrei, der dem Spieler einen gehörigen Schreck einjagen dürfte. Sie wird automatisch abgespielt, muss allerdings gespeichert werden, damit Sie sie nutzen können. Klicken Sie mit der rechten Maustaste auf den kleinen Audioplayer und auf „Speichern unter ...“. Wählen Sie den Ordner „Persönlicher Ordner“ und klicken Sie auf „Speichern“. Das Chromium-Fenster können Sie jetzt schließen.

Klicken Sie auf das „New“-Symbol in der Thonny-Symbolleiste, um ein neues Projekt zu starten. Wie bereits zuvor verwenden wir jetzt eine Bibliothek, um die Fähigkeiten von Python zu erweitern: die Bibliothek Pygame, die – wie der Name schon sagt – speziell für die Programmierung von Spielen (engl. Games) erstellt wurde. Geben Sie Folgendes ein:

```
import pygame
```

Sie benötigen außerdem einige Teile aus anderen Bibliotheken und auch aus einem Unterabschnitt der Pygame-Bibliothek. Diese importieren wir, indem wir Folgendes eingeben:

```
from pygame.locals import *
from time import sleep
from random import randrange
```

Die Anweisung **from** funktioniert anders als die Anweisung **import**. Damit lassen sich nur die benötigten Bestandteile einer Bibliothek importieren, statt die Bibliothek vollständig einzulesen. Als Nächstes muss Pygame eingerichtet werden. Dies wird als *Initialisierung* bezeichnet. Pygame muss die Breite und Höhe des Monitors oder Fernsehers des Spielers kennen – also dessen **Auflösung**. Geben Sie Folgendes ein:

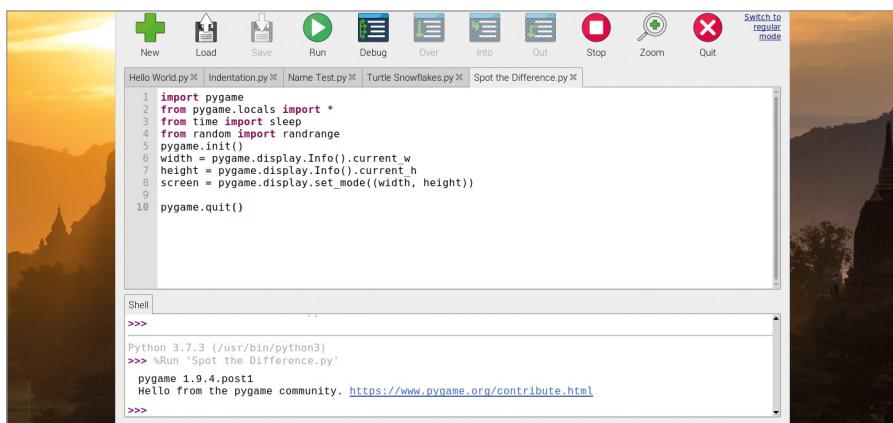
```
pygame.init()
width = pygame.display.Info().current_w
height = pygame.display.Info().current_h
```

Der letzte Schritt bei der Einrichtung von Pygame ist die Erstellung seines Fensters, das Pygame als **screen** (Bildschirmbild) bezeichnet. Geben Sie Folgendes ein:

```
screen = pygame.display.set_mode((width, height))

pygame.quit()
```

Achten Sie auf die Leerzeile in der Mitte; das ist der Platz, wo später Ihr Programm eingefügt wird. Im Moment klicken Sie jedoch nur auf das Symbol für Ausführen, speichern Sie Ihr Programm unter dem Namen **Spot the Difference** und schauen Sie dann zu: Pygame erstellt ein Fenster mit schwarzem Hintergrund, das dann fast sofort wieder verschwindet, sobald die Anweisung zum Beenden erreicht wird. Abgesehen von einer kurzen Nachricht in der Shell (**Abbildung 5-14**), hat das Programm bisher noch nicht viel bewirkt.



▲ Abbildung 5-14: Ihr Programm ist funktionsfähig, leistet aber noch nicht viel

Um das Suchbild anzuzeigen, geben Sie die folgende Zeile in den Freiraum oberhalb von `pygame.quit()` ein:

```
difference = pygame.image.load('spot_the_diff.png')
```

Damit das Bild auch wirklich den Bildschirm ausfüllt, müssen Sie es entsprechend der Auflösung Ihres Monitors bzw. Fernsehers skalieren. Geben Sie Folgendes ein:

```
difference = pygame.transform.scale(difference, (width, height))
```

Nachdem sich das Bild jetzt im Speicher befindet, müssen Sie Pygame anweisen, es tatsächlich auf dem Bildschirm anzuzeigen – ein Vorgang, der als *blitting* oder *bit block transfer* bezeichnet wird. Geben Sie Folgendes ein:

```
screen.blit(difference, (0, 0))
pygame.display.update()
```

Die erste dieser Zeilen kopiert das Bild auf den Bildschirm, beginnend in der linken oberen Ecke. Die zweite Zeile weist Pygame an, den Bildschirm neu zu zeichnen. Ohne diese zweite Zeile wird das Bild zwar an der richtigen Stelle im Speicher abgelegt, aber man würde es nie zu sehen bekommen!

Klicken auf das „Run“ -Symbol zum Ausführen und schon wird das Bild kurz auf dem Bildschirm angezeigt (**Abbildung 5-15**).



▲ Abbildung 5-15: Ihr Suchbild

Um das Bild länger sehen zu können, fügen Sie folgende Zeile direkt oberhalb von `pygame.quit()` ein:

```
sleep(3)
```

Klicken Sie erneut auf „Ausführen“. Nun bleibt das Bild länger auf dem Bildschirm sichtbar. Das Überraschungsbild fügen Sie hinzu, indem Sie direkt unter der Zeile `pygame.display.update()` eingeben:

```
zombie = pygame.image.load('scary_face.png')
zombie = pygame.transform.scale(zombie, (width, height))
```

Fügen Sie eine Verzögerung hinzu, damit das Zombie-Bild nicht sofort erscheint:

```
sleep(3)
```

Blenden Sie dann das Bild auf dem Bildschirm aus und aktualisieren Sie es so, dass es dem Spieler angezeigt wird:

```
screen.blit(zombie, (0,0))
pygame.display.update()
```

Klicken auf das Symbol „Run“ und beobachten Sie, was passiert: Pygame lädt Ihr Suchbild, allerdings wird es nach drei Sekunden durch den furchterregenden Zombie ersetzt (**Abbildung 5-16**)!



▲ **Abbildung 5-16:** Damit kann man jemanden ganz schön erschrecken

Wenn man die Verzögerung auf drei Sekunden festlegt, wird die Sache allerdings ziemlich vorhersehbar. Ändern Sie daher die Zeile **sleep(3)** oberhalb von **screen.blit(zombie, (0,0))** in:

```
sleep(randrange(5, 15))
```

Auf diese Weise wird eine Zufallszahl zwischen 5 und 15 gewählt, um die das Programm dann entsprechend verzögert wird. Fügen Sie dann folgende Zeile direkt über der Anweisung **sleep** hinzu, um die Audiodatei mit dem Schrei zu laden:

```
scream = pygame.mixer.Sound('scream.wav')
```

Gehen Sie dann unter die Anweisung **sleep** und geben Sie dort folgende Anweisung in eine neue Zeile ein, die dann die Tonwiedergabe unmittelbar vor Anzeige des furchteinflößenden Bildes startet:

```
scream.play()
```

Abschließend weisen Sie Pygame an, die Wiedergabe des Sounds zu beenden, indem Sie folgende Zeile direkt oberhalb von `pygame.quit()` eingeben:

```
scream.stop()
```

Klicken auf das „Run“-Symbol zum Ausführen und bewundern Sie Ihre Arbeit: Nachdem ein paar Sekunden lang das harmlose Suchbild angezeigt wird, in dem man die Unterschiede erkennen soll, erscheint plötzlich der furchterregende Zombie mit einem entsetzlichen Schrei. So jagen Sie Ihren Freunden sicher einen gehörigen Schrecken ein! Falls Sie feststellen, dass das Zombie-Bild bereits erscheint, noch bevor der Ton abgespielt wird, dann können Sie dies korrigieren, indem Sie eine kleine Verzögerung direkt nach der Anweisung `scream.play()` und vor der Anweisung `screen.blit` einfügen:

```
sleep(0.4)
```

Das fertige Programm sollte so aussehen:

```
import pygame
from pygame.locals import *
from time import sleep
from random import randrange
pygame.init()
width = pygame.display.Info().current_w
height = pygame.display.Info().current_h
screen = pygame.display.set_mode((width, height))
difference = pygame.image.load('spot_the_diff.png')
difference = pygame.transform.scale(difference, (width, height))
screen.blit(difference, (0, 0))
pygame.display.update()
zombie = pygame.image.load('scary_face.png')
zombie = pygame.transform.scale(zombie, (width, height))
scream = pygame.mixer.Sound('scream.wav')
sleep(randrange(5, 15))
scream.play()
screen.blit(zombie, (0,0))
pygame.display.update()
sleep(3)
scream.stop()
pygame.quit()
```

Jetzt müssen Sie nur noch Freunde einladen, die in dem Suchbild einen Unterschied finden sollen – und natürlich dürfen Sie nicht vergessen, zuvor die Lautsprecher aufzudrehen!



HERAUSFORDERUNG: ÄNDERE DAS AUSSEHEN



Können Sie die Bilder ändern, um diesen Streich an andere Ereignisse wie Weihnachten anzupassen? Können Sie Ihre eigenen Such- und Gruselbilder zeichnen (mit einem Bildbearbeitungsprogramm wie beispielsweise GIMP)? Können Sie das Spiel verfolgen lassen, wie der Benutzer auf einen Unterschied klickt, um das Spiel noch überzeugender zu machen?

Projekt 3: RPG Maze

Da Sie jetzt den Dreh bei Python heraus haben, können wir mit Pygame etwas Komplexeres entwickeln: ein voll funktionsfähiges, textbasiertes Labyrinthspiel zum Beispiel, das auf klassischen Rollenspielen basiert. Diese Spiele, die als Textadventure, Textabenteuer oder interaktive Fiktion bekannt sind, stammen aus einer Zeit, in der Computer noch nicht mit Grafiken umgehen konnten. Aber es gibt immer noch Fans, die überzeugt sind, dass keine Grafik je mit der eigenen Fantasie mithalten kann!

ONLINE-PROJEKT



Dieses Projekt ist auch online verfügbar unter rpf.io/python-rpg

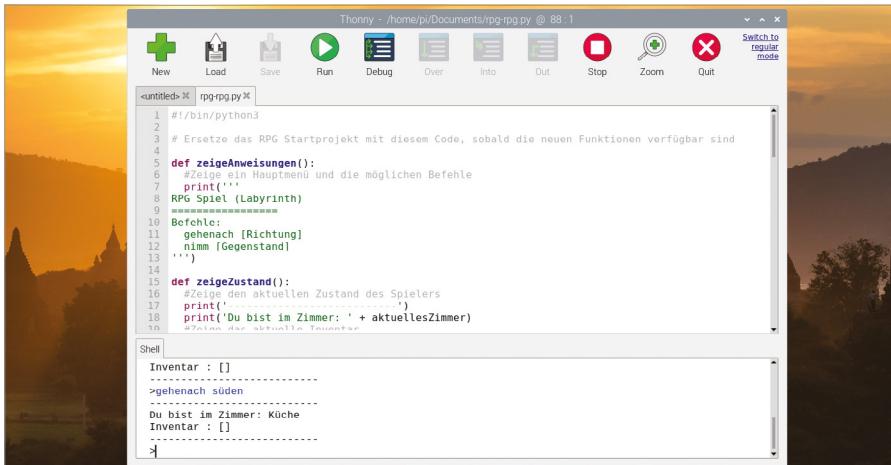
Dieses Programm ist um einiges komplexer als die anderen in diesem Kapitel. Um es Ihnen leichter zu machen, beginnen wir mit einer zum Teil bereits geschriebenen Version. Öffnen Sie den Chromium-Webbrowser und gehen Sie zu folgender Adresse rpf.io/p/de-DE/rpg-go.

Der Chromium-Webbrowser lädt den Code für das Programm in Form eines zip-Archivs, „rpg-de-de-resources.zip“ automatisch in Ihren Downloads-Ordner herunter. Sie müssen die zip-Datei noch entpacken. Klicken Sie dazu rechts auf die in Chromium, und wählen Sie die Option „In Ordner anzeigen“. Das öffnet das „Downloads“-Verzeichnis. Suchen Sie hier nach der Datei, „rpg-de-de-resources.zip“, und klicken Sie rechts auf sie. Wählen Sie im erscheinenden Menü die Option „Hier entpacken“.. Gehen Sie dann zurück zu Thonny und klicken Sie auf das „Load“-Symbol (Laden). Suchen Sie nach der Datei **rpg-rpg.py** in Ihrem Downloads-Ordner und klicken Sie auf den Button „OK“.

Klicken Sie zunächst auf das Symbol „Run“ für Ausführen, um sich mit der Funktionsweise eines Textabenteuers vertraut zu machen. Die Ausgabe des Spiels erscheint im Shell-Bereich am unteren Rand des Thonny-Fensters. Vergrößern Sie das Thonny-Fenster, indem Sie auf die Schaltfläche „Maximieren“ klicken. So wird das Fenster leichter lesbar.

Das Spiel ist in seiner derzeitigen Form sehr einfach: Es gibt zwei Räume und keine Objekte. Der Spieler beginnt in der Diele, dem ersten der beiden Räume. Um in die Küche zu kommen, geben Sie einfach ‚gehenach süden‘ gefolgt von der **EINGABETASTE**, ein (**Abbildung 5-17**). Wenn Sie sich

in der Küche befinden, können Sie „gehenach norden“ eingeben, um in der Diele zurückzukehren. Sie können auch versuchen, „gehenach westen“ und „gehenach osten“ einzugeben. Da es aber in diesen Himmelsrichtungen keine Räume gibt, wird das Spiel eine Fehlermeldung ausgeben.



▲ Abbildung 5-17: Es gibt bisher nur zwei Räume

Scrollen Sie hinunter bis zu Zeile 29 des Programms im Skriptbereich, um eine Variable namens **zimmer** zu finden. Dieser VariablenTyp ist bekannt als *Wörterbuch (dictionary)* und informiert das Spiel über die Räume, deren Ausgänge und zu welchem Raum ein bestimmter Ausgang führt.

Um das Spiel interessanter zu gestalten, fügen wir einen weiteren Raum hinzu, und zwar einen Speiseraum, östlich von der Diele. Suchen Sie nach der Variable **zimmer** im Skriptbereich und erweitern Sie sie, indem Sie in Zeile 38 ein Komma anfügen (,) nach der }. Anschließend geben Sie Folgendes ein (eine exakte Einrückung ist bei einem Wörterbuch nicht unbedingt erforderlich):

```

'Esszimmer' : {
    'westen' : 'Diele'
}
```

Sie benötigen auch noch einen neuen Ausgang in der Diele, da dieser nicht automatisch angelegt wird. Daher gehen Sie ans Ende von Zeile 33 und fügen ein Komma und dann folgende Zeile hinzu:

```
'osten' : 'Esszimmer'
```

Beenden Sie das Spiel mit dem „Stop“-Symbol falls es noch läuft. Klicken Sie auf das „Run“-Symbol, um den neuen Raum auszuprobieren: Geben Sie „gehenach osten“ ein, während Sie noch in der Diele stehen, um dann das Esszimmer zu betreten (**Abbildung 5-18** (nächste Seite)). Geben Sie anschließend „gehenach westen“ ein, während Sie im Esszimmer stehen, um in die Diele zu gehen. Herzlichen Glückwunsch: Sie haben gerade einen eigenen Raum angelegt!

```

thonny - /home/pi/Documents/rpg.py @ 30 · 1
New Load Save Run Debug Over Into Out Stop Zoom Quit Switch to regular mode
rpg.py [x]
29     inventar = []
30
31 #Ein Dictionary (Wörterbuch) verbindet ein Zimmer mit anderen Zimmern
32 zimmer = {
33     'Diele' : { 'süden' : 'Küche',
34                 'osten' : 'Esszimmer',
35             },
36
37     'Küche' : { 'norden' : 'Diele',
38                 },
39
40     'Esszimmer' : {
41         'westen' : 'Diele',
42         },
43     }
44
Shell [x]
Inventar : []
>gehenach ostern
-----
Du bist im Zimmer: Esszimmer
Inventar : []
-----
|
```

▲ Abbildung 5-18: Sie haben einen weiteren Raum hinzugefügt

Leere Räume machen allerdings nicht sonderlich viel Spaß. Um einem Raum einen Gegenstand hinzuzufügen, müssen Sie das Wörterbuch dieses Raums ändern. Halten Sie das Programm an, indem Sie auf das „Stop“-Symbol klicken. Suchen Sie das Wörterbuch **Diele** im Skriptbereich und setzen Sie ein Komma ans Ende der Zeile '**osten**' : '**Esszimmer**' hinzu, bevor Sie die **EINGABETASTE** drücken. Anschließend geben Sie folgende Zeile ein:

'Gegenstand' : 'Schlüssel'

Klicken Sie erneut auf „Run“. Dieses Mal teilt Ihnen das Spiel mit, dass Sie den neuen Gegenstand sehen können: einen Schlüssel. Geben Sie 'himm Schlüssel' ein (**Abbildung 5-19**). Sie können ihn an sich nehmen und in die Liste der Gegenstände einfügen, die Sie mit sich führen – bekannt als das *Inventar*. Sie haben Ihr Inventar stets dabei, wenn Sie durch die einzelnen Räume gehen.

```

thonny - /home/pi/Documents/rpg-finished-rpg.py @ 109 · 1
New Load Save Run Debug Over Into Out Stop Zoom Quit Switch to regular mode
rpg-finished-rpg.py [x]
27
28 #Das Inventar ist beim Start leer
29 inventar = []
30
31 #Ein Dictionary (Wörterbuch) verbindet ein Zimmer mit anderen Zimmern
32 zimmer = {
33     'Diele' : { 'süden' : 'Küche',
34                 'osten' : 'Esszimmer',
35                 'Gegenstand' : 'Schlüssel',
36             },
37
38     'Küche' : { 'norden' : 'Diele',
39                 'Gegenstand' : 'Monster',
40             },
41     }
42
Shell [x]
>nimm Schlüssel
Schlüssel wurde genommen!
-----
Du bist im Zimmer: Diele
Inventar : ['Schlüssel']
-----
```

▲ Abbildung 5-19: Der aufgenommene Schlüssel wird Ihrem Inventar hinzugefügt

Klicken Sie auf das Stop-Symbol und machen Sie das Ganze noch interessanter, indem Sie buchstäblich ein Monster ins Spiel bringen, vor dem man auf der Hut sein sollte. Suchen Sie nach dem Wörterbuch **Küche** (Küche) und fügen Sie einen „Monster“-Eintrag ein, wie Sie es auch schon mit dem „Schlüssel“-Eintrag getan haben. Vergessen Sie nicht, an die Zeile darüber ein Komma anzuhängen:

'Gegenstand' : 'Monster'

Damit das Monster den Spieler auch angreifen kann, müssen Sie dem Spiel etwas zusätzliche Logik beibringen. Dazu scrollen Sie im Skriptbereich bis ans Ende des Programms und tragen die folgenden Zeilen nach – einschließlich des Kommentars, der mit einem Rautensymbol gekennzeichnet ist und Ihnen hilft, das Programm auch dann noch zu verstehen, wenn Sie Tage später daran weiterarbeiten wollen. Achten Sie auch darauf, die Zeilen einzurücken:

```
# Der Spieler verliert, wenn er einen Raum mit einem
#Monster betritt
if 'Gegenstand' in zimmer[aktuellesZimmer] and 'Monster' in
zimmer[aktuellesZimmer]['Gegenstand']:
    print('Du wurdest von einem hungrigen Monster gefressen...
DAS SPIEL IST AUS!')
    break
```

Klicken Sie auf „Run“ und versuchen Sie, in die Küche zu gehen (**Abbildung 5-20**) – Das Monster wird nicht allzu beeindruckt sein, wenn Sie das tun.

The screenshot shows the Thonny IDE interface. The top menu bar includes options like New, Load, Save, Run, Debug, Over, Into, Out, Stop, Zoom, and Quit. A status bar at the bottom indicates the file path is /home/pi/Documents/rpg-finished-rpg.py and the port is 109:1. The main window displays the Python code for a game script. The code defines a function that checks if the player is in a room with a monster and an item. If so, it prints a message and ends the game. The terminal window below shows the game's output: the player is in a room with an inventory, sees a key, and then moves south. However, the game ends because the player is in a room with a monster and an item. The terminal output is as follows:

```
Du bist im Zimmer: Diele
Inventar: []
Du siehst einen Schlüssel
>gehenach süden
Du wurdest von einem hungrigen Monster gefressen... DAS SPIEL IST AUS!
```

▲ Abbildung 5-20: Und vergessen Sie übrigens die Ratten, schließlich ist da ein Monster in der Küche!

Um aus diesem Abenteuer ein richtiges Spiel zu machen, benötigen Sie noch weitere Gegenstände, einen weiteren Raum und die Fähigkeit zu „gewinnen“, indem Sie das Haus mit sämtlichen Gegenständen in Ihrem Inventar sicher verlassen. Zunächst fügen Sie einen weiteren Raum hinzu, so wie Sie es beim Esszimmer getan haben – nur, dass es sich dieses Mal um einen Garten handelt. Fügen Sie einen Ausgang aus dem Wörterbuch Esszimmer hinzu. Denken Sie auch daran, ans Ende der vorigen Zeile ein Komma anzufügen:

```
'süden' : 'Garten'
```

Dann tragen Sie diesen neuen Raum in das Hauptwörterbuch **zimmer** ein und vergessen dabei nicht, ein Komma ans Ende } der Zeile darüber nachzutragen:

```
'Garten' : {  
    'norden' : 'Esszimmer'  
}
```

Fügen Sie dem Wörterbuch „Esszimmer“ den Gegenstand „Zaubertrank“ hinzu. Vergessen Sie nicht, in der Zeile darüber das erforderliche Komma einzufügen:

```
'Gegenstand' : 'Zaubertrank'
```

Scrollen Sie schließlich zum Ende des Programms und fügen Sie die Logik hinzu, die erforderlich ist, um zu prüfen, ob der Spieler alle Gegenstände hat. Falls ja, teilen Sie ihm mit, dass er das Spiel gewonnen hat:

```
# Der Spieler gewinnt, wenn er den Garten mit einem Schlüssel  
und einem Trank betritt  
if aktuellesZimmer == 'Garten' and 'Schlüssel' in inventar  
and 'Zaubertrank' in inventar:  
    print('Du bist aus dem Haus geflohen... DU HAST  
GEWINNEN!')  
    break
```

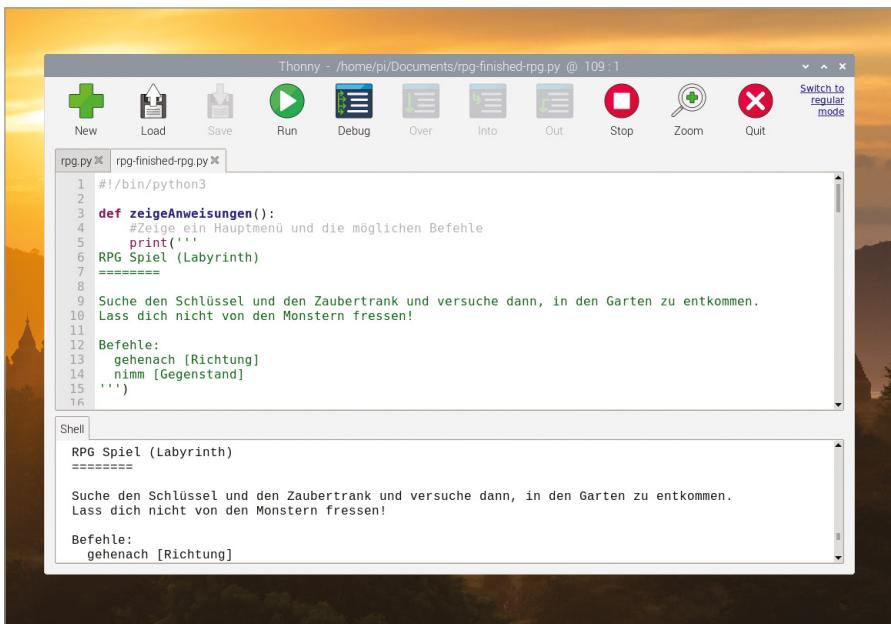
Klicken Sie auf „Run“ und versuchen Sie, das Spiel zu beenden, indem Sie Schlüssel und Zaubertrank an sich nehmen, bevor Sie den Garten betreten. Und immer daran denken, nicht die Küche zu betreten: Dort lauert das Monster auf Sie!

Als letzten Feinschliff für das Spiel fügen Sie einige Anweisungen hinzu, die dem Spieler sagen, wie er das Spiel zu Ende spielen soll. Scrollen Sie zum Anfang des Programms, wo die Funktion **zeigeAnweisungen()** definiert wird, und fügen Sie Folgendes ein:

Suche den Schlüssel und den Zaubertrank und versuche dann, in den Garten zu entkommen.

Lass dich nicht von den Monstern fressen!

Spielen das Spiel noch ein letztes Mal durch. Sie werden sehen, dass die neuen Anweisungen gleich zu Beginn erscheinen (**Abbildung 5-21**). Herzlichen Glückwunsch: Sie haben ein interaktives, textbasiertes Labyrinth-Spiel entwickelt!



▲ Abbildung 5-21: Jetzt wissen die Spieler, was sie zu tun haben



HERAUSFORDERUNG: DAS SPIEL ERWEITERN



Können Sie weitere Räume hinzufügen, damit das Spiel länger dauert? Können Sie einen Gegenstand hinzufügen, um sich vor dem Monster zu schützen? Wie würden Sie eine Waffe hinzufügen, um das Monster zu erlegen? Können Sie Räume hinzufügen, die oberhalb und unterhalb der bereits bestehenden liegen und über eine Treppe zugänglich sind?

Kapitel 6

Physical Computing mit Scratch und Python

Die Welt des Programmierens geht weiter über interaktive Anwendungen am Computerbildschirm hinaus – Sie können auch elektronische Bauteile steuern, die mit den GPIO-Stiften Ihres Raspberry Pi verbunden sind



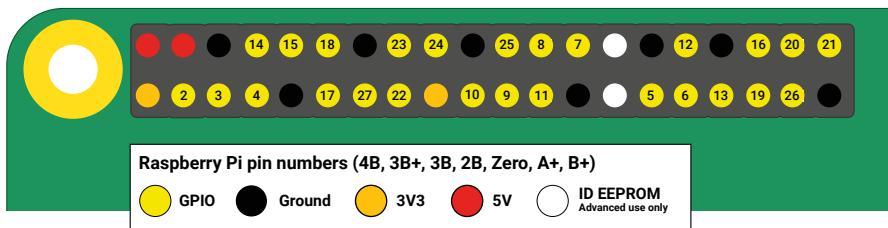
Wenn von „Programmieren“ oder „Coden“ die Rede ist, denkt man meist an Software. Programmieren kann jedoch mehr sein als nur Software zu schreiben. Es kann die reale Welt durch Hardware beeinflussen. Dies nennen wir *Physical Computing*.

Wie der Name schon sagt, geht es beim Physical Computing darum, die Dinge in der realen Welt über Programme zu steuern – also Hardware, nicht Software. Wenn Sie das Programm an Ihrer Waschmaschine einstellen, die Temperatur an Ihrem programmierbaren Thermostat ändern oder den Knopf an einer Ampel drücken, um die Straße sicher zu überqueren, verwenden Sie Physical Computing.

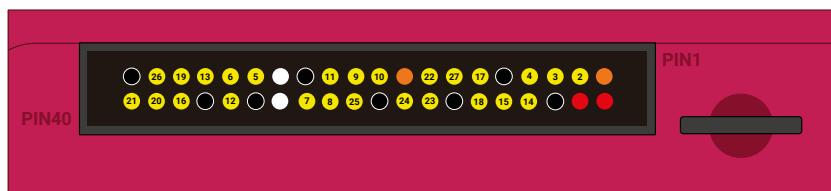
Der Raspberry Pi ist ein großartiges Gerät zum Erlernen des Physical Computing. Das liegt an einem seiner wichtigsten Merkmale: der Allzweckeingabe/-ausgabe-Stiftleiste *general-purpose input/output (GPIO)*.

Einführung in die GPIO-Stifteleiste

Die GPIO-Stifteleiste (GPIO = General-Purpose Input/Output) befindet sich an der Oberkante der Leiterplatte des Raspberry Pi oder an der Rückseite des Raspberry Pi 400. Sie sieht aus wie zwei lange Reihen von Metallstiften. Hier können Sie Hardware wie LEDs und Schalter an Ihren Raspberry Pi anschließen und sie dann mithilfe Ihrer Programme steuern. Die Stifte (Pins) können sowohl als Eingang als auch als Ausgang verwendet werden.



Die GPIO-Stifteleiste des Raspberry Pi besteht aus 40 Steckerstiften. Einige davon stehen Ihnen für die Verwendung in Ihren Projekten zum Physical Computing zur Verfügung, einige liefern Strom und wieder andere sind für die Kommunikation mit Zusatzhardware wie dem Sense HAT reserviert (siehe **Kapitel 7**).



Der Raspberry Pi 400 ist mit der gleichen GPIO-Stifteleiste mit den gleichen Stiften ausgestattet, ist aber im Vergleich zu anderen Raspberry Pi-Modellen auf den Kopf gestellt. Dieses Diagramm geht davon aus, dass Sie die GPIO-Stifteleiste von der Rückseite des Raspberry Pi 400 aus betrachten. Überprüfen Sie Ihre Verdrahtung immer doppelt, wenn Sie etwas an die GPIO-Stifteleiste des Raspberry Pi 400 anschließen. Man kann es leicht vergessen, trotz der „Pin 40“- und „Pin 1“-Beschriftungen auf dem Gehäuse!

GPIO-ERWEITERUNGEN



Es ist durchaus möglich, die GPIO-Stifteleiste des Raspberry Pi 400 so zu verwenden, wie sie ist. Aber vielleicht finden Sie es einfacher, eine Erweiterung zu verwenden. Mit einer Erweiterung werden die Stifte seitlich an den Raspberry Pi 400 herangeführt, sodass Sie Ihre Verdrahtung überprüfen und anpassen können, ohne immer wieder hinten herum gehen zu müssen.

Zu den kompatiblen Erweiterungen gehört die Black HAT Hack3r-Reihe von [pimoroni.com](https://www.pimoroni.com) und der Pi T-Cobbler Plus von adafruit.com.

Wenn Sie eine Erweiterung kaufen, überprüfen Sie immer, wie sie verdrahtet ist. Einige, wie der Pi T-Cobbler Plus, haben ein anderes Layout der GPIO-Stifte. Halten Sie sich im Zweifelsfall immer an die Anweisungen des Herstellers.

Es gibt mehrere Kategorien von Stifttypen, jede mit einer bestimmten Funktion:

3V3	3,3 Volt Stromversorgung	Eine ständig eingeschaltete 3,3-V-Stromquelle mit derselben Spannung, mit der der Raspberry Pi intern läuft
5 V	5 Volt Stromversorgung	Eine ständig eingeschaltete 5-V-Stromquelle mit derselben Spannung, die der Raspberry Pi am microUSB-Stromanschluss aufnimmt
Masse (GND)	0 Volt Masse	Eine Erdverbindung, die zur Vervollständigung eines an die Stromquelle angeschlossenen Stromkreises verwendet wird
GPIO XX	Allzweck-Eingangs-/Ausgangs-Stiftnummer XX	Die für Ihre Programme verfügbaren GPIO-Stifte, gekennzeichnet durch eine Zahl von 2 bis 27
ID EEPROM	Reservierte Stifte für besondere Zwecke	Stifte reserviert für die Verwendung mit Hardware Attached on Top (HAT) und anderem Zubehör

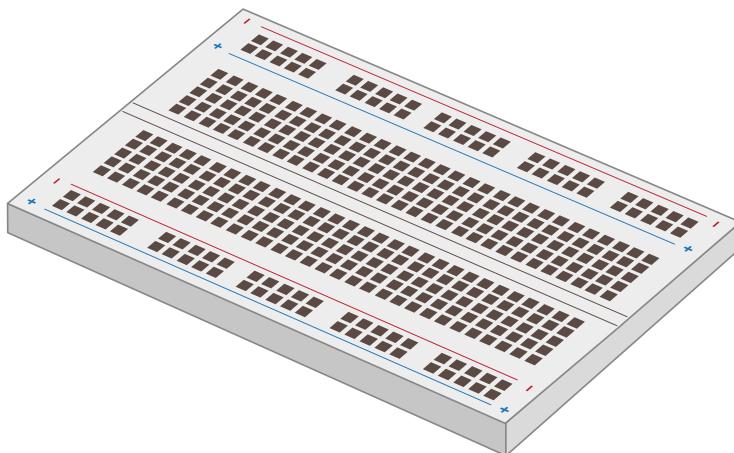
WARNUNG!



Die GPIO-Stifteleiste des Raspberry Pi ist eine unterhaltsame und sichere Art, mit Physical Computing zu experimentieren. Dabei ist aber Vorsicht geboten. Achten Sie darauf, die Stifte beim Anschließen und Demontieren von Hardware nicht zu verbiegen. Verbinden Sie niemals zwei Stifte direkt miteinander, weder versehentlich noch absichtlich, es sei denn, dies wird in den Projektanweisungen ausdrücklich so angegeben: Eine direkte Verbindung wird als Kurzschluss bezeichnet und kann je nach den Stiften Ihren Raspberry Pi dauerhaft beschädigen.

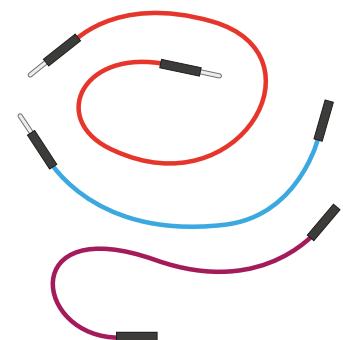
Elektronische Bauteile

Die GPIO-Stifteleiste ist nur ein Teil dessen, was Sie für den Einstieg in das Physical Computing brauchen. Die andere Hälfte besteht aus elektronischen Bauteilen, also den Geräten, die Sie über die GPIO-Stifteleiste steuern. Es gibt Tausende von Bauteilen, aber bei den meisten GPIO-Projekten kommen die Folgenden zum Einsatz:

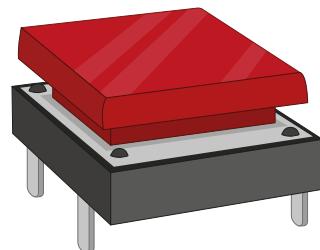


Eine *Steckplatine*, auch bekannt als *Breadboard*, kann Physical Computing-Projekte erheblich erleichtern. Anstatt einen Haufen loser Bauteile zu haben, die miteinander mit Drähten verbunden werden müssen, können Sie auf einer Steckplatine Komponenten einstecken und dadurch mit Hilfe von Metallbahnen verbinden, die unter ihrer Oberfläche verborgen sind. Viele Steckplatten enthalten auch Abschnitte für die Stromverteilung, was den Aufbau von Schaltkreisen erleichtert. Man braucht keine Steckplatine, um mit Physical Computing zu beginnen, aber sie erleichtert auf jeden Fall den Einstieg.

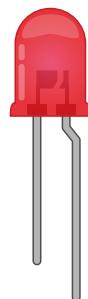
Jumperkabel, auch bekannt als *Dupont Kabel* oder *flexible Drahtbrücken*, verbinden die Bauteile mit dem Raspberry Pi und, wenn Sie keine Steckplatine verwenden, untereinander. Sie sind in drei Versionen erhältlich: männlich-zu-weiblich (M2F), die Sie benötigen, um eine Steckplatine mit den GPIO-Stiften zu verbinden; weibliche-zu-weiblichen (F2F), die verwendet werden, um einzelne Bauteile miteinander zu verbinden, wenn Sie dafür keine Steckplatine verwenden; und männliche-zu männlichen (M2M), um Verbindungen von einem Teil einer Steckplatine zu einem anderen herzustellen. Abhängig von Ihrem Projekt benötigen Sie möglicherweise alle drei Arten von Jumperkabeln. Wenn Sie eine Steckplatine verwenden, kommen Sie in der Regel mit M2F- und M2M-Jumperkabeln aus.



Ein *Drucktaster*, ist die Art von Schalter, die Sie zur Steuerung einer Spielkonsole verwenden würden. Diese Schalter sind üblicherweise mit zwei oder vier Beinchen erhältlich. Beide Typen können mit dem Raspberry Pi eingesetzt werden. Der Drucktaster ist ein Eingabegerät: Ihr Programm kann überwachen, ob er gedrückt wird, und dann eine entsprechende Aufgabe auszuführen. Ein weiterer gängiger Schaltertyp ist ein *rastender Schalter*. Während ein Drucktaster nur aktiv ist, wenn Sie ihn gedrückt halten, wird ein rastender Schalter aktiviert, wenn Sie ihn umschalten. Er bleibt dann aktiv, bis Sie ihn wieder umschalten. Die meisten Lichtschalter sind ein gutes Beispiel für diese Art von Schalter.



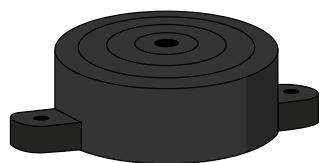
Eine *Leuchtdiode* (*Light Emitting Diode = LED*) ist ein Ausgabegerät. Sie steuern sie direkt von Ihrem Programm aus. Eine LED leuchtet, wenn sie eingeschaltet ist. Sie finden sie überall in Ihrem Haus, in allen Größen: von den kleinen LEDs, die Ihnen anzeigen, dass Ihre Waschmaschine eingeschaltet ist, bis hin zu den großen, die Ihre Räume erhellen. LEDs sind in einer Vielzahl von Formen, Farben und Größen erhältlich. Aber nicht alle sind für die Verwendung mit einem Raspberry Pi geeignet: Vermeiden Sie die LEDs, bei denen angegeben wird, sie seien für 5 V- oder 12 V-Versorgungsspannung bestimmt.



Widerstände sind Bauteile, die den Fluss des *elektrischen Stroms* steuern. Sie sind in verschiedenen Werten erhältlich, die in der Einheit *Ohm* (Ω) angegeben werden. Je höher die Ohm-Zahl, desto höher der Widerstand. Bei Physical-Computing-Projekten mit dem Raspberry Pi werden Widerstände am häufigsten dazu verwendet, LEDs davor zu schützen, zu viel Strom zu ziehen und sich selbst oder den Raspberry Pi zu beschädigen. Dazu benötigen Sie Widerstände mit einer Nennleistung von etwa $330\ \Omega$. Viele Elektronikshops verkaufen handliche Packungen mit einer Reihe verschiedener, häufig verwendeter Werte, um Ihnen mehr Gestaltungsspielraum zu bieten.



Ein weiteres Ausgabegerät ist der *piezoelektrische Summer*, der normalerweise einfach als Summer, Buzzer oder akustisches Signal bezeichnet wird. Während eine LED Licht erzeugt, verursacht der Summer ein Geräusch – ein summendes Geräusch, um genau zu sein. Im Inneren des Kunststoffgehäuses des Summers befinden sich zwei Metallplatten. Wenn sie aktiviert werden, vibrieren diese Platten



gegeneinander und erzeugen den Summton. Es gibt zwei Arten von Summern: *aktive Summer* und *passive Summer*. Achten Sie beim Kauf darauf, aktive Summer zu erwerben, da diese am einfachsten zu benutzen sind.

Andere häufig verwendete elektrische Bauteile sind: Motoren, die eine spezielle Steuerplatine benötigen, bevor sie an den Raspberry Pi angeschlossen werden können, Infrarotsensoren, die Bewegungen erkennen, Temperatur- und Feuchtigkeitssensoren, die zur Wettervorhersage verwendet werden können, und lichtabhängige Widerstände (Fotowiderstände / LDRs) – Eingabegeräte, die wie eine umgekehrte LED funktionieren, indem sie Licht erkennen.

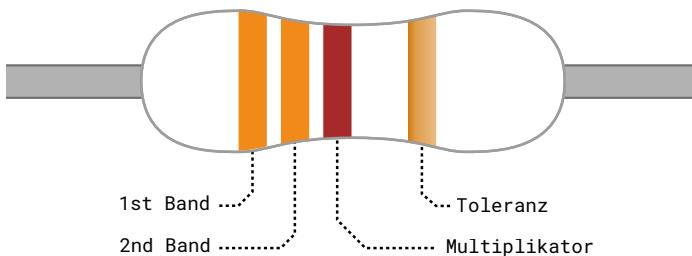
Verkäufer auf der ganzen Welt bieten Bauteile für Physical Computing mit dem Raspberry Pi an. Sie finden sie als Einzelteile oder in Bausätzen, die alles enthalten, was Sie für den Einstieg benötigen. Um Anbieter zu finden, besuchen Sie rpf.io/products, klicken Sie auf Raspberry Pi 4. Sie erhalten eine Liste von Raspberry Pi Partner-Onlineshops (approved resellers) für Ihr Land oder Ihre Region.

Für die Projekte in diesem Kapitel brauchen Sie mindestens folgende Bauteile:

- 3 × LEDs: rot, grün und gelb oder orange
- 2 × Drucktaster
- 1 × aktiver Summer
- Jumperkabel männlich-auf-weiblich (M2F) und weiblich-auf-weiblich (F2F)
- Optional eine Steckplatine und männliche-zu-männlichen Jumperkabel (M2M)

Widerstandswerte anhand von Farbcodes ermitteln

Widerstände gibt es in einer breiten Palette von Werten, von Null-Widerstandversionen, die praktisch nur Drahtstücke sind, bis hin zu Hoch-Widerstandversionen in der Dicke eines Unterschenkels. Bei den wenigsten dieser Widerstände sind die Werte jedoch in Zahlen aufgedruckt: Stattdessen verwenden sie einen speziellen Code, der in Form farbiger Streifen oder Bänder um den Widerstandskörper herum verläuft.



	1st/2nd Band	Multiplikator	Toleranz
Schwarz	0	$\times 10^0$	-
Braun	1	$\times 10^1$	$\pm 1\%$
Rot	2	$\times 10^2$	$\pm 2\%$
Orange	3	$\times 10^3$	-
Gelb	4	$\times 10^4$	-
Grün	5	$\times 10^5$	$\pm 0.5\%$
Blau	6	$\times 10^6$	$\pm 0.25\%$
Violett	7	$\times 10^7$	$\pm 0.1\%$
Grau	8	$\times 10^8$	$\pm 0.05\%$
Weiß	9	$\times 10^9$	-
Gold	-	$\times 10^{-1}$	$\pm 5\%$
Silber	-	$\times 10^{-2}$	$\pm 10\%$
Keine	-	-	$\pm 20\%$

Um den Wert eines Widerstands abzulesen, positionieren Sie ihn so, dass die Gruppe der Streifen links und der einzelne Streifen rechts liegt. Ausgehend vom ersten Streifen schauen Sie in der Spalte „1./2. Band“ der Tabelle nach, um die erste und zweite Ziffer zu erhalten. Dieses Beispiel hat zwei orangefarbene Streifen, die beide einen Wert von „3“ haben und damit für „33“ stehen. Wenn Ihr Widerstand vier statt drei gruppierte Streifen hat, notieren Sie sich auch den Wert des dritten Streifens (für Fünf-/Sechs-Streifen-Widerstände siehe [rpf.io/5-6band](#)).

Wenn Sie zum letzten gruppierten Streifen – dem dritten oder vierten – übergehen, sehen Sie in der Spalte „Multiplikator“ nach, welche Farbe er hat. Dies sagt Ihnen, mit welcher Zahl Sie Ihre aktuelle Zahl multiplizieren müssen, um den tatsächlichen Wert des Widerstands zu erhalten. Dieses Beispiel hat einen braunen Streifen und das bedeutet „ $\times 10^1$ “. Das mag verwirrend aussehen, aber ist einfach die sogenannte *wissenschaftliche Notation*: $\times 10^1$ bedeutet „Fügen Sie am Ende Ihrer Zahl eine Null an“. Wenn der Streifen blau wäre, stünde dies für $\times 10^6$ und bedeutet: „Fügen Sie sechs Nullen an das Ende Ihrer Zahl an“.

33, von den orangefarbenen Streifen, plus die hinzugefügte Null vom braunen Streifen, ergibt 330 – das ist der Wert des Widerstands, gemessen in Ohm. Der letzte Streifen, rechts, ist die Toleranz des Widerstands. Damit wird einfach angegeben, wie nahe er seinem Nennwert voraussichtlich sein wird. Billigere Widerstände haben möglicherweise einen silbernen Streifen, der anzeigt, dass sie 10 Prozent höher oder niedriger als ihr Nennwert sein können. Vielleicht haben sie auch gar keinen letzten Streifen, was bedeutet, dass sie 20 Prozent höher oder niedriger sein können. Die teuersten Widerstände haben einen grauen Streifen, der anzeigt, dass sie innerhalb von 0,05 Prozent ihres Nennwerts liegen werden. Für Hobby-Projekte ist die Genauigkeit nicht so wichtig: Jeder Toleranzwert wird in der Regel gut funktionieren.

Wenn Ihr Widerstandswert über 1000 Ohm liegt, wird er normalerweise in Kiloohm ($k\Omega$) angegeben. Wenn er über einer Million Ohm liegt, sind dies Megaohm ($M\Omega$). Ein $2200\ \Omega$ -Widerstand wird $2,2\ k\Omega$ geschrieben. Ein $2.200.000\ \Omega$ -Widerstand wird $2,2\ M\Omega$ geschrieben.



KÖNNEN SIE DIE FRAGEN BEANTWORTEN?



Welche Farbstreifen würde ein $100\ \Omega$ -Widerstand haben? Welche Farbstreifen würde ein $2,2\ M\Omega$ -Widerstand haben? Wenn Sie die billigsten Widerstände finden, nach welchem Farbtoleranzstreifen würden Sie suchen?

Ihr erstes Physical Computing-Programm: Hallo, LED!

So wie die Ausgabe von „Hallo, Welt“ auf dem Bildschirm ein fantastischer erster Schritt beim Erlernen einer Programmiersprache ist, steht das Aufleuchten einer LED für die traditionelle Einführung in das Erlernen von Physical Computing. Für dieses Projekt benötigen Sie eine LED und einen Widerstand mit $330\ \Omega$ ($330\ \Omega$) oder so nah wie möglich an $330\ \Omega$. Dazu brauchen Sie weibliche-zu-weiblichen-Jumperkabel (F2F).



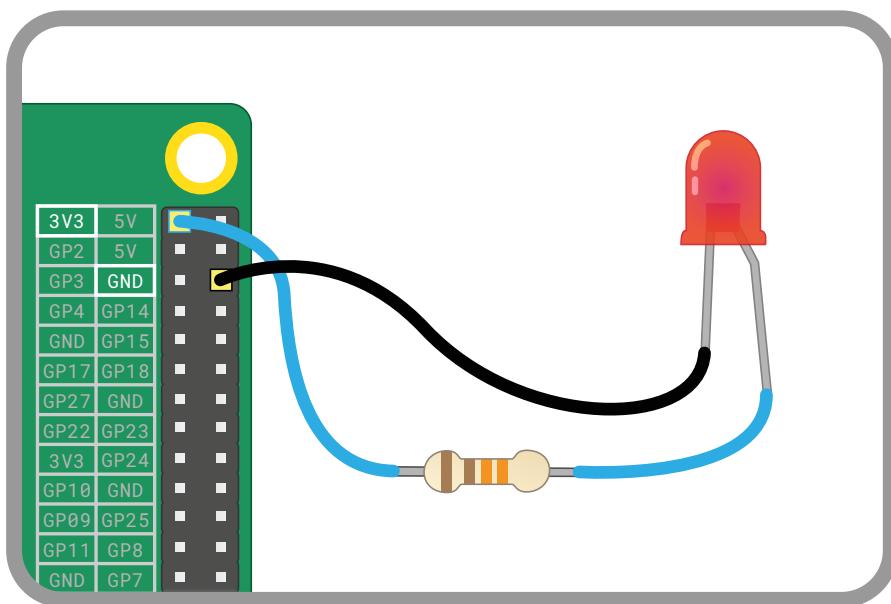
WIDERSTAND IST LEBENSWICHTIG



Der Widerstand ist ein wichtiges Bauteil in diesem Schaltkreis: Er schützt den Raspberry Pi und die LED, indem er die Menge des elektrischen Stroms, den die LED aufnehmen kann, begrenzt. Ohne ihn kann die LED zu viel Strom ziehen und sich selbst – oder den Raspberry Pi – beschädigen. Wenn der Widerstand in dieser Form verwendet wird, wird er als *strombegrenzender Widerstand* bezeichnet. Der genaue Widerstandswert, den Sie benötigen, hängt von der LED ab, die Sie verwenden, aber $330\ \Omega$ funktioniert für die meisten gängigen LEDs. Je höher der Wert, desto dunkler ist die LED. Je niedriger der Wert, desto heller ist die LED.

Schließen Sie niemals eine LED ohne einen Strombegrenzungswiderstand an einen Raspberry Pi an, es sei denn, Sie wissen, dass die LED einen eingebauten Widerstand mit entsprechendem Wert hat.

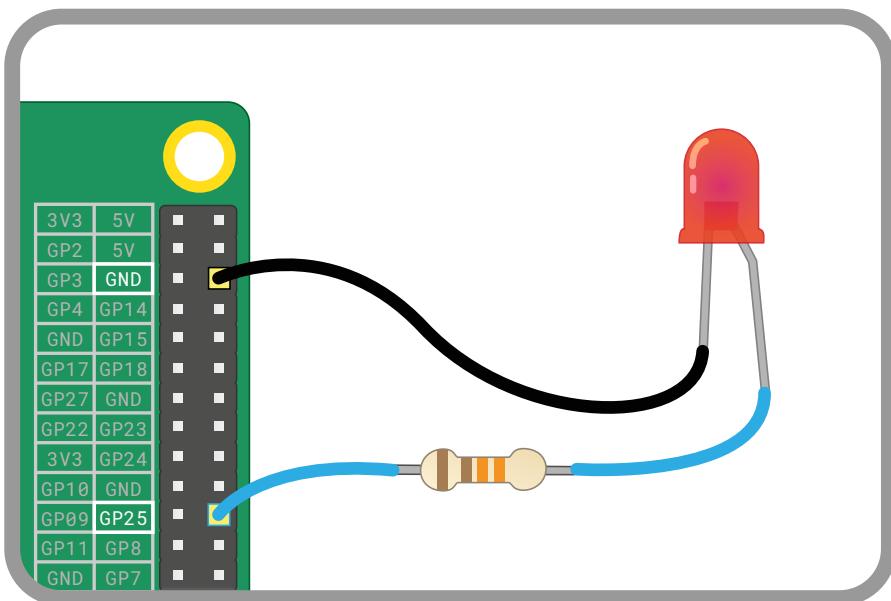
Überprüfen Sie zunächst, ob Ihre LED funktioniert. Orientieren Sie Ihren Raspberry Pi so, dass sich die GPIO-Stiftleiste in zwei vertikalen Streifen auf der rechten Seite der Platine befindet. Verbinden Sie ein Ende Ihres 330 Ω -Widerstands mit dem ersten 3,3-V-Stift (beschriftet mit 3V3 in **Abbildung 6-1**) mithilfe eines weiblichen-zu-weiblichen-Jumperkabels und verbinden Sie dann das andere Ende mit dem langen Beinchen – positiv oder Anode – Ihrer LED mit einem weiteren weiblichen-zu-weiblichen-Jumperkabel. Nehmen Sie ein weiteres weibliches-zu-weibliches-Jumperkabel und verbinden Sie das kurze Beinchen – den Minuspol oder die Kathode – Ihrer LED mit dem ersten Massestift (beschriftet mit GND in **Abbildung 6-1**).



▲ **Abbildung 6-1:** Verdrahten Sie Ihre LED mit diesen Stiften – und vergessen Sie den Widerstand nicht!

Wenn Ihr Raspberry Pi eingeschaltet ist, sollte die LED aufleuchten. Wenn dies nicht der Fall ist, überprüfen Sie Ihre Schaltung sorgfältig: Vergewissern Sie sich, dass der Widerstandswert nicht zu hoch ist, dass alle Kabel richtig angeschlossen sind und Sie definitiv die richtigen GPIO-Stifte wie auf dem Diagramm ausgewählt haben. Überprüfen Sie auch die Beinchen der LED, da LEDs nur in einer Richtung funktionieren: mit dem längeren Beinchen an die positive Seite der Schaltung und dem kürzeren an die negative.

Funktioniert Ihre LED in diesem ersten Schritt, ist es nun an der Zeit, sie zu programmieren. Trennen Sie das Jumperkabel vom 3,3-V-Stift (beschriftet mit 3V3 in **Abbildung 6-2**) und verbinden Sie es mit dem GPIO 25-Stift (gekennzeichnet mit GP25 in **Abbildung 6-2**). Die LED erlischt, aber keine Sorge – das ist normal.



▲ Abbildung 6-2: Trennen Sie das Kabel von 3V3 und schließen Sie es an den GPIO 25-Stift an

Sie sind jetzt soweit, ein Scratch- oder Python-Programm zu schreiben, um Ihre LED ein- und auszuschalten.



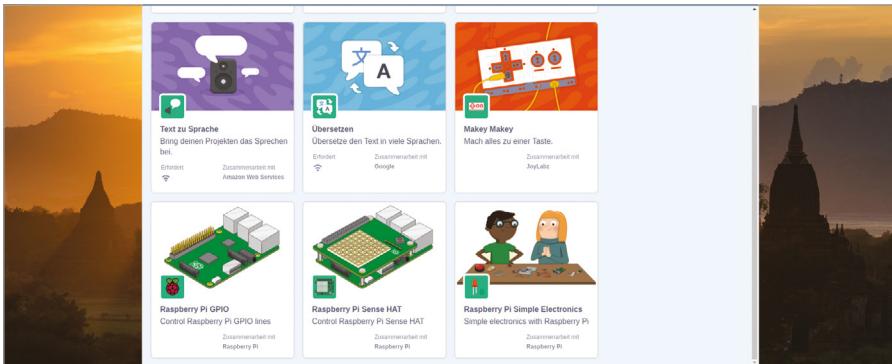
PROGRAMMIERKENNTNISSE

Die Projekte in diesem Kapitel setzen voraus, dass Sie mit der Verwendung von Scratch 3 und der integrierten Entwicklungsumgebung (IDE) Thonny Python vertraut sind. Wenn Sie dies noch nicht getan haben, arbeiten Sie zunächst die Projekte in **Kapitel 4, Programmieren mit Scratch 3** und **Kapitel 5, Programmieren mit Python** durch.



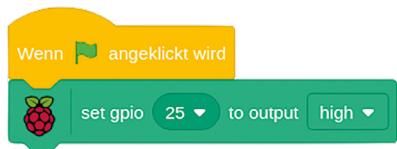
LED-Steuerung in Scratch

Starten Sie Scratch 3 und klicken Sie auf das Symbol zum Hinzufügen einer Erweiterung . Scrollen Sie nach unten, um die Erweiterung „Raspberry Pi GPIO“ zu finden (**Abbildung 6-3** umseitig). Dann klicken Sie darauf. Dadurch werden die Blöcke geladen, die Sie zur Steuerung der GPIO-Stifte des Raspberry Pi aus Scratch 3 benötigen. Sie sehen die neuen Blöcke in der Blockpalette, sie sind jetzt in der neuen Kategorie „Raspberry Pi GPIO“ verfügbar.

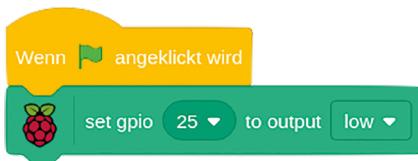


▲ Abbildung 6-3: Hinzufügen der Raspberry Pi GPIO-Erweiterung zu Scratch 3

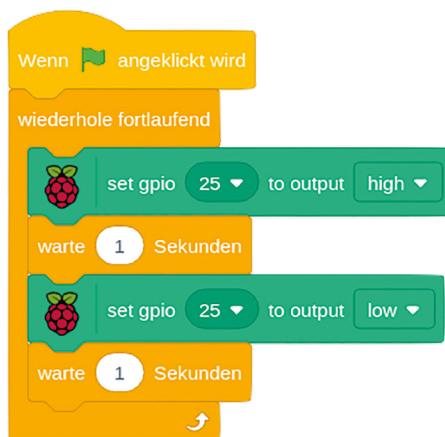
Fangen Sie damit an, einen **Wenn angeklickt wird**-Ereignisblock auf den Codebereich zu ziehen und platzieren Sie dann einen **set gpio to output high**-Block darunter. Sie müssen die Nummer des von Ihnen verwendeten Stifts auswählen: Klicken Sie auf den kleinen Pfeil, um die Dropdown-Auswahl zu öffnen, und klicken Sie auf 25, um Scratch mitzuteilen, dass Sie den GPIO 25-Stift steuern möchten.



Klicken Sie auf die grüne Flagge, um Ihr Programm auszuführen. Sie werden sehen, dass Ihre LED aufleuchtet: Sie haben Ihr erstes Physical Computing-Programm programmiert! Klicken Sie auf das rote Achteck, um Ihr Programm zu stoppen: Sie werden feststellen, dass die LED weiterhin leuchtet. Das liegt daran, dass Ihr Programm den Raspberry Pi nur angewiesen hat, die LED einzuschalten – das ist nämlich das, was der „output high“-Teil Ihres **set gpio 25 to output high**-Blocks bedeutet. Um sie wieder auszuschalten, klicken Sie auf den Abwärtspfeil am Ende des Blocks und wählen Sie „low“ aus der Liste.



Klicken Sie erneut auf die grüne Flagge, und jetzt schaltet Ihr Programm die LED aus. Um die Dinge noch interessanter zu machen, fügen Sie einen **wiederhole fortlaufend**-Steuerblock und ein paar **warte 1 Sekunde**-Blöcke hinzu, um ein Programm zu erstellen, das die LED im Sekundentakt ein- und ausschaltet.



Klicken Sie auf die grüne Flagge und beobachten Sie Ihre LED. Sie geht eine Sekunde lang an, eine Sekunde lang aus, eine Sekunde lang an und wiederholt dieses Muster so lange, bis Sie auf das rote Achteck klicken, um den Vorgang zu stoppen. Probieren Sie aus, was passiert, wenn Sie auf das Achteck klicken, während die LED gerade ein- oder ausgeschaltet ist.



HERAUSFORDERUNG: KÖNNEN SIE ÄNDERUNGEN VORNEHMEN?



Wie würden Sie das Programm ändern, damit die LED länger leuchtet? Wie wäre es, wenn sie länger erloschen ist? Welche kleinste Verzögerung können Sie verwenden, während Sie noch sehen können, dass die LED ein- und ausgeschaltet wird?

LED-Steuerung in Python

Laden Sie Thonny aus dem Abschnitt „Entwicklung“ des Raspberry-Menüs. Klicken Sie dann auf den Button „New“, um ein neues Projekt zu starten, und auf „Save“, um es unter **Hallo LED** zu speichern. Um die GPIO-Stifte von Python aus zu verwenden, benötigen Sie eine Bibliothek namens GPIO Zero. Für dieses Projekt benötigen Sie nur den Teil der Bibliothek für die Arbeit mit LEDs. Importieren Sie nur diesen Teil der Bibliothek, indem Sie Folgendes in den Python-Shell-Bereich eingeben:

```
from gpiozero import LED
```

Als Nächstes müssen Sie GPIO Zero mitteilen, an welchem GPIO-Stift die LED angeschlossen ist.

Geben Sie Folgendes ein:

```
led = LED(25)
```

Zusammen verleihen diese beiden Zeilen Python die Fähigkeit, LEDs zu steuern, die an die GPIO-Stifte Ihres Raspberry Pi angeschlossen sind, und auszuwählen, welchen Stift – oder welche Stifte, wenn Sie mehr als eine LED in Ihrer Schaltung haben – gesteuert werden soll. Um die LED nun tatsächlich zu steuern, geben Sie Folgendes ein:

```
led.on()
```

Und um die LED wieder auszuschalten:

```
led.off()
```

Herzlichen Glückwunsch! Sie haben jetzt die Kontrolle über die GPIO-Stifte Ihres Raspberry Pi in Python erlangt! Versuchen Sie, diese beiden Anweisungen erneut einzugeben. Wenn die LED bereits aus ist, wird durch `led.off()` nichts ausgelöst. Dasselbe gilt, wenn die LED bereits leuchtet und Sie `led.on()` eingeben.

Um ein echtes Programm zu erstellen, geben Sie Folgendes in den Skriptbereich ein:

```
from gpiozero import LED
from time import sleep

led = LED(25)

while True:
    led.on()
    sleep(1)
    led.off()
    sleep(1)
```

Dieses Programm importiert die LED-Funktion aus der Bibliothek `gpiozero` (GPIO Zero) und die `sleep`-Funktion aus der Bibliothek `time`. Es konstruiert dann eine Endlosschleife, um die LED eine Sekunde lang einzuschalten, eine Sekunde lang auszuschalten und dies zu wiederholen. Klicken Sie auf den Button „Run“, um das Programm in Aktion zu sehen: Ihre LED beginnt zu blinken. Wie auch beim Scratch-Programm beobachten Sie das Verhalten, wenn Sie auf die Stop-Taste klicken, während die LED an ist, im Vergleich dazu, wenn sie aus ist.



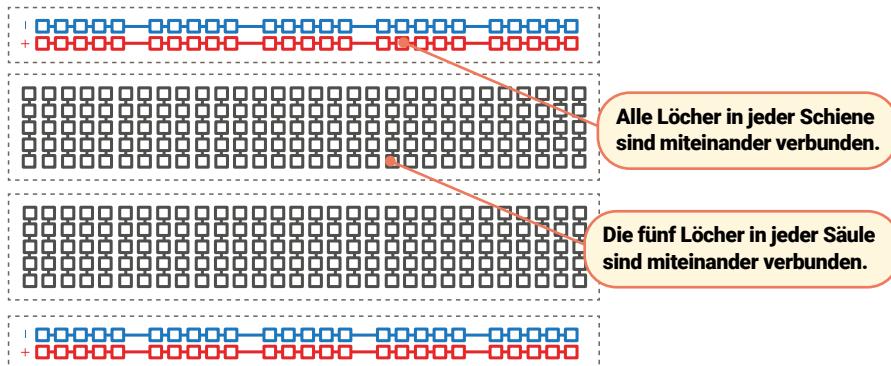
HERAUSFORDERUNG: LÄNGERES LEUCHTEN



Wie würden Sie das Programm ändern, damit die LED länger leuchtet? Und wie, damit die LED länger ausgeschaltet bleibt? Welches ist die kleinste Verzögerung, die es Ihnen noch erlaubt, zu sehen, die LED ein- und ausgeschaltet wird?

Verwendung einer Steckplatine

Die nächsten Projekte in diesem Kapitel können Sie viel einfacher durcharbeiten, wenn Sie eine Steckplatine zur Aufnahme der Bauteile und zur Herstellung der elektrischen Verbindungen verwenden.



Eine Steckplatine ist mit Löchern überzogen, die einen Abstand von 2,54 mm untereinander haben. Das entspricht dem Abstand der Beinchen gängiger Bauteile. Unter diesen Löchern befinden sich Metallstreifen, die wie die Jumperkabel wirken, die Sie bisher verwendet haben. Diese verlaufen in Reihen über das gesamte Breadboard hinweg, wobei die meisten Breadboards in der Mitte eine Lücke haben, um sie in zwei Hälften zu teilen. Viele Steckplatten haben auch Buchstaben auf der Oberseite und Zahlen an den Seiten. Mit ihrer Hilfe können Sie ein bestimmtes Loch finden: A1 ist die linke obere Ecke, B1 ist das Loch unmittelbar rechts daneben, während B2 ein Loch weiter unten von B1 aus liegt. A1 ist durch die verborgenen Metallstreifen mit B1 verbunden, aber es wird niemals ein Loch aus der ersten Reihe (1) mit einem Loch aus der zweiten Reihe (2) verbunden, es sei denn, Sie fügen selbst ein Jumperkabel hinzu.

Größere Steckplatten haben auch Reihen von Löchern an den Seiten, die typischerweise mit roten und schwarzen oder roten und blauen Streifen markiert sind. Dies sind die **Stromschienen**, die die Verdrahtung erleichtern sollen: Sie können ein einzelnes Kabel vom Massestift des Raspberry Pi mit einer der Stromschienen verbinden. Normalerweise ist diese mit einem blauen oder schwarzen Streifen und einem Minus-Symbol gekennzeichnet. Damit können Sie eine **Masseverbindung** (*wird auch als Erde bezeichnet*) für viele Bauteile auf der Steckplatine bereitstellen. Natürlich können Sie dasselbe auch tun, wenn Ihre Schaltung 3,3-V- oder 5-V-Strom benötigt. Benutzen Sie dazu die andere Stromschiene, um einen Kurzschluss zu vermeiden.

Das Hinzufügen von elektronischen Bauteilen zu einer Steckplatine ist einfach: Man muss nur ihre Anschlüsse (die herausstehenden Metallbeinchen) an den Löchern ausrichten und das Bauteil vorsichtig einstecken, bis es sicher fixiert ist. Für weitere Verbindungen, die die Steckplatine nicht von sich aus vorsieht, können Sie männliche-zu-männlichen-Jumperkabel (M2M) verwenden. Für Verbindungen von der Steckplatine zum Raspberry Pi verwenden Sie männliche-zu-weiblichen-Jumperkabel(M2F).

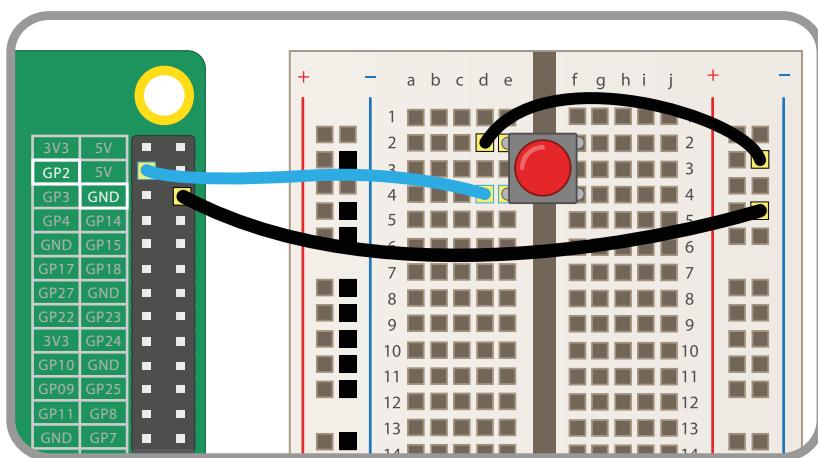
Versuchen Sie niemals, mehr als einen Bauteilanschluss oder mehr als ein Jumperkabel in ein einziges Loch auf der Steckplatine zu stopfen. Wie bereits erwähnt, sind die Löcher in

Spalten verbunden, abgesehen von der Teilung in der Mitte, sodass ein Bauteilanschluss in A1 elektrisch mit allem verbunden ist, was Sie in B1, C1, D1 und E1 einstecken.

Nächste Schritte: Lesen eines Tasters

Ausgabe, wie beispielsweise über LEDs ist eine Sache, aber der „Input/Output“-Teil in „GPIO“ bedeutet, dass Sie Stifte auch als Eingänge verwenden können. Für dieses Projekt benötigen Sie eine Steckplatine, männliche-zu-männlichen-Jumperkabel (M2M) und männliche-zu-weiblichen-Jumperkabel (M2F) sowie einen Drucktaster. Wenn Sie keine Steckplatine haben, können Sie weibliche-zu-weiblichen-Jumperkabel (F2F) verwenden, aber die Taste ist für Sie dann viel schwerer zu drücken, ohne den Stromkreis versehentlich zu unterbrechen.

Fangen Sie, fügen Sie den Drucktaster zu Ihrer Steckplatine hinzu. Wenn Ihr Drucktaster nur zwei Beinchen hat, achten Sie darauf, dass sie in unterschiedlich nummerierten Reihen auf der Steckplatine stecken. Hat sie vier Beinchen, drehen Sie ihn so, dass die Seiten, aus denen die Beine herauskommen, entlang der Reihen der Steckplatine stecken und die flachen Seiten ohne Beine oben und unten sind. Verbinden Sie die Erdungsschiene (Masseschiene) Ihrer Steckplatine mit einem Massestift des Raspberry Pi (gekennzeichnet mit GND auf Abbildung 6-4) mit einem männlichen-zu-weiblichen-Jumperkabel und verbinden Sie dann ein Bein des Drucktasters mit einem männlichen-zu-männlichen-Jumperkabel mit der Erdungsschiene, indem Sie das Jumperkabel in der selben Spalte bleibend direkt daneben einstecken. Verbinden Sie schließlich das andere Bein – das Bein auf der gleichen Seite wie das soeben angeschlossene Bein, wenn Sie einen vierbeinigen Schalter verwenden – mit dem GPIO 2-Stift (gekennzeichnet mit GP2 auf Abbildung 6-4) des Raspberry Pi mit einem männlichen-zu-weiblichen-Jumperkabel.

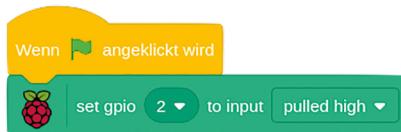


► Abbildung 6-4: Verdrahtung eines Drucktasters mit den GPIO-Stiften

Auslesen eines Tasters in Scratch

Starten Sie ein neues Scratch-Programm und ziehen Sie einen **Wenn [Flagge angeklickt wird]-Block** auf den Bereich mit dem Programmcode. Schließen Sie einen **set gpio to input pulled high**-Block

-Block an und wählen Sie die Zahl 2 aus der Dropdown-Liste aus, die dem GPIO-Stift entspricht, den Sie für den Drucktaster verwendet haben.



Wenn Sie jetzt auf die grüne Flagge klicken, wird nichts passieren. Das liegt daran, dass Sie Scratch angewiesen haben, den Stift als Input (Eingang) zu verwenden, aber nicht, was mit diesem Input geschehen soll. Ziehen Sie einen **wiederhole fortlaufend**-Block an das Ende Ihrer Sequenz und dann einen **falls dann sonst**-Block hinein. Suchen Sie den **gpio is high?**-Block, ziehen Sie ihn in den rautenförmigen Bereich in dem **falls dann**-Teil des Blocks und wählen Sie aus der Dropdown-Liste die Zahl 2, um einzustellen, welcher GPIO-Stift geprüft werden soll. Ziehen Sie einen **sage Hallo! für 2 Sekunden**-Block in den **sonst**-Teil des Blocks und bearbeiten Sie ihn so, dass „Knopf gedrückt!“ angezeigt wird. Lassen Sie den „falls dann“-Teil des Blocks vorerst leer.



Das war jetzt sehr viel auf einmal, aber fangen Sie mit dem Testen an: Klicken Sie auf die grüne Flagge und drücken Sie dann den Taster auf Ihrer Steckplatine. Ihr Sprite sollte Ihnen anzeigen, dass der Knopf gedrückt wurde: Sie haben erfolgreich eine Eingabe vom GPIO-Stift gelesen!

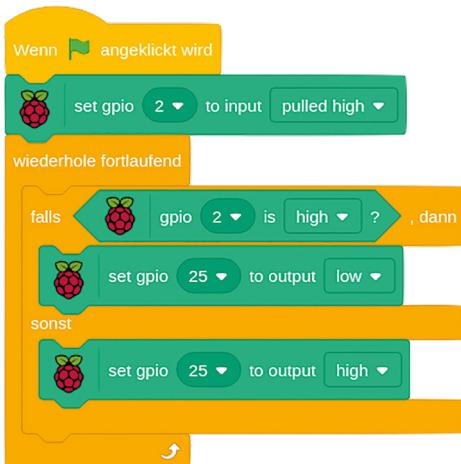
Velleicht haben Sie bemerkt, dass der **falls gpio 2 is high? dann**-Teil des Blocks leer ist. Der Code, der läuft, wenn der Taster tatsächlich gedrückt wird, befindet sich hingegen im **sonst**-Teil des Blocks. Das wirkt verwirrend, steigt die Spannung etwa nicht, wenn man den Taster drückt? Tatsächlich ist das Gegenteil der Fall: Die GPIO-Stifte des Raspberry Pi sind normalerweise unter Spannung („high“ = 3,3 V), also eingeschaltet, wenn sie als Eingang eingestellt sind. Beim Drücken auf den Taster wird ihre Spannung auf Masse heruntergezogen („low“ = 0 V).

Schauen Sie sich Ihre Schaltung noch einmal an: Der Taster ist an einem Bein mit dem GPIO 2-Stift, der den positiven Teil des Stromkreises bereitstellt, verbunden und auf dem anderen

Bein mit dem Massestift. Wenn das alles verwirrend klingt, merken Sie sich das einfach so: Ein Taster an einem Raspberry Pi GPIO-Stift ist gedrückt, wenn der Stift auf Masse gezogen wird, und der Taster ist nicht gedrückt, wenn der Stift unter Spannung steht!

Um Ihr Programm zu erweitern, fügen Sie die LED und den Widerstand wieder in die Schaltung ein: Denken Sie daran, den Widerstand an den GPIO 25-Stift und das lange Bein der LED anzuschließen und das kürzere Bein der LED an die Erdungsschiene auf Ihrer Steckplatine.

Ziehen Sie den **sage Taste gedrückt! für 2 Sekunden**-Block aus dem Codebereich auf die Blockpalette, um ihn zu löschen, und ersetzen Sie ihn dann durch einen **set gpio 25 to output high**-Block. Denken Sie daran, dass Sie die GPIO-Zahl mithilfe des Dropdown-Pfeils ändern müssen. Fügen Sie einen **set gpio 25 to output low**-Block zum derzeit leeren **falls gpio 2 is high? dann**-Teil des Blocks hinzu – und denken Sie daran, die Einstellungen des Blocks zu ändern.



Klicken Sie auf die grüne Flagge und drücken Sie den Taster. Die LED leuchtet, solange Sie den Taster gedrückt halten. Lassen Sie ihn los und sie erlischt wieder. Herzlichen Glückwunsch: Sie steuern einen GPIO-Stift basierend auf der Eingabe von einem anderen!



HERAUSFORDERUNG: LED SOLL LÄNGER LEUCHTEN



Wie würden Sie das Programm ändern, damit die LED ein paar Sekunden lang weiter leuchtet, nachdem Sie die Taste losgelassen haben? Was müssten Sie ändern, damit die LED an ist, wenn Sie den Knopf nicht drücken, und aus, wenn Sie ihn drücken?

Lesen eines Tasters in Python lesen

Klicken Sie in Thonny auf den Button „New“, um ein neues Projekt zu starten, und auf „Save“, um es unter **Button Input** zu speichern. Die Verwendung eines GPIO-Stifts als Eingang für einen Taster ist der Verwendung eines Stifts als Ausgang für eine LED sehr ähnlich, aber Sie müssen einen anderen Teil der GPIO Zero-Bibliothek importieren. Geben Sie Folgendes in den Skriptbereich ein:

```
from gpiozero import Button
button = Button(2)
```

Damit der Code beim Drücken der Taste ausgeführt wird, bietet GPIO Zero die **wait_for_press**-Funktion. Geben Sie Folgendes ein:

```
button.wait_for_press()
print("Sie haben mich gedrückt!")
```

Klicken Sie auf den Button „Run“ und drücken Sie dann den Drucktasterr. Ihre Nachricht wird in der Python-Shell am unteren Rand des Thonny-Fensters ausgegeben: Sie haben erfolgreich eine Eingabe vom GPIO-Stift gelesen! Wenn Sie Ihr Programm erneut ausprobieren möchten, müssen Sie erneut auf den Button „Run“ klicken. Da es keine Schleife im Programm gibt, wird es beendet, sobald es die Nachricht an die Shell ausgegeben hat.

Um Ihr Programm zu erweitern, fügen Sie die LED und den Widerstand wieder in die Schaltung ein, falls Sie dies noch nicht getan haben. Denken Sie daran, den Widerstand an den GPIO 25-Stift und das lange Bein der LED sowie das kurze Bein der LED an die Erdungsschiene Ihrer Steckplatine anzuschließen.

Um sowohl eine LED zu steuern als auch einen Taster zu lesen, müssen Sie beide Funktionen, **Button** und **LED**, aus der GPIO Zero-Bibliothek importieren. Sie benötigen außerdem die Funktion **sleep** aus der Bibliothek **time**. Gehen Sie zurück zum Anfang Ihres Programms und geben Sie Folgendes als die neuen ersten beiden Zeilen ein:

```
from gpiozero import LED
from time import sleep
```

Unterhalb der Zeile **button = Button(2)** geben Sie ein:

```
led = LED(25)
```

Löschen Sie die Zeile **print ("Sie haben mich gedrückt!")** und ersetzen Sie sie durch:

```
led.on()  
sleep(3)  
led.off()
```

So sollte Ihr fertiges Programm aussehen:

```
from gpiozero import LED  
from time import sleep  
from gpiozero import Button  
  
button = Button(2)  
led = LED(25)  
button.wait_for_press()  
led.on()  
sleep(3)  
led.off()
```

Klicken Sie auf den Button „Run“ und drücken Sie dann den Drucktaster: Die LED leuchtet drei Sekunden lang auf, erlischt dann wieder und das Programm beendet sich. Herzlichen Glückwunsch: Sie können jetzt in Python eine LED am GPIO-Ausgang mit einem Taster am GPIO-Eingang steuern!



HERAUSFORDERUNG: EINE SCHLEIFE HINZUFÜGEN



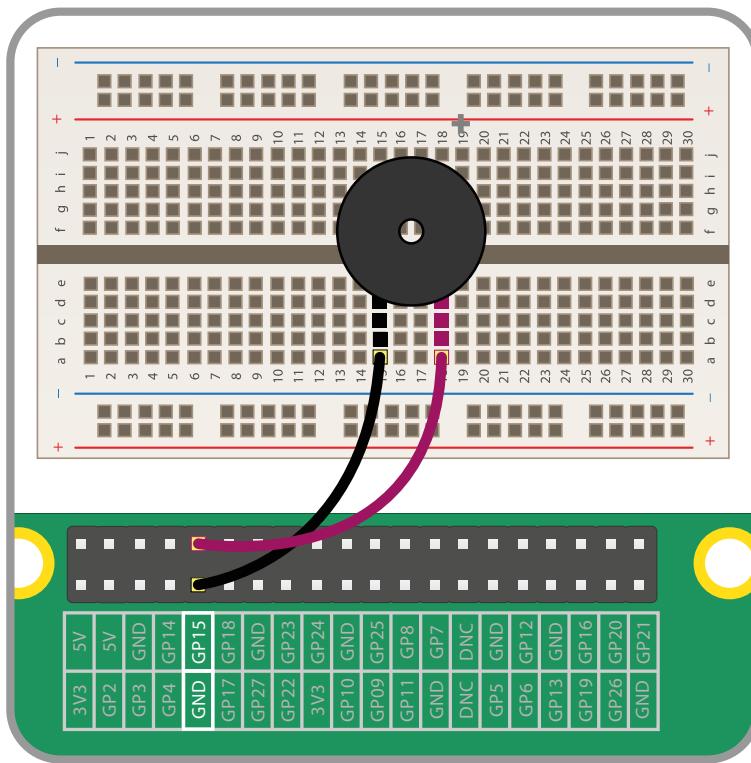
Wie würden Sie eine Schleife hinzufügen, um das Programm zu wiederholen, anstatt es nach einem Tastendruck zu beenden? Was müssten Sie ändern, damit die LED an ist, wenn Sie den Knopf nicht drücken, und aus, wenn Sie ihn drücken?

Lärm machen: Steuerung eines Summers

LEDs sind ein großartiges Ausgabegerät, aber nicht sehr nützlich, wenn man sie nicht im Auge hat. Die Lösung: Summer, die ein Geräusch überall im Raum hörbar machen. Für dieses Projekt benötigen Sie eine Steckplatine, ein männliche-zu-weiblichen-Jumperkabel (M2F) und einen aktiven Summer. Wenn Sie keine Steckplatine haben, können Sie den Summer stattdessen mit weiblichen-zu-weiblichen-Jumperkabeln (F2F) anschließen.

Ein aktiver Summer kann hinsichtlich Schaltung und Programmierung genau wie eine LED behandelt werden. Wiederholen Sie die Schaltung, die Sie für die LED gemacht haben, aber ersetzen Sie die LED durch den aktiven Summer und lassen Sie den Widerstand weg, da der Summer mehr Strom benötigt, um zu funktionieren. Verbinden Sie ein Bein des Summers mit dem GPIO-15-Stift (in der Abbildung mit GP15 bezeichnet) **Abbildung 6-5**) und den anderen mit dem Massestift (in der Abbildung mit GND bezeichnet) unter Verwendung Ihrer Steckplatine und eines männlichen-zu-weiblichen-Jumperkabels.

Wenn Ihr Summer drei Beine hat, vergewissern Sie sich, dass das mit einem Minus-Symbol (-) markierte Bein mit dem Massestift und das mit S oder SIGNAL markierte Bein mit GPIO 15 verbunden ist. Verbinden Sie dann das verbleibende Bein – normalerweise das mittlere Bein – mit dem 3,3-V-Stift (mit der Bezeichnung 3V3.)



▲ Abbildung 6-5: Anschließen eines Summers an die GPIO-Stifte

Steuerung eines Summers in Scratch

Erstellen Sie dasselbe Programm wie für das Blinken der LED – oder laden Sie es, wenn Sie es vor dem Erstellen des Taster-Projekts gespeichert haben. Wählen Sie in der Dropdown-Liste in den `set gpio to output high`-Blöcken die Nummer 15 aus, damit Scratch den richtigen GPIO-Stift steuert.



Klicken Sie auf die grüne Flagge und Ihr Summer beginnt zu summen: eine Sekunde an und eine Sekunde aus. Wenn Sie den Summer nur einmal pro Sekunde klicken hören, verwenden Sie einen passiven anstelle eines aktiven Summers. Wenn ein aktiver Summer das sich schnell ändernde Signal erzeugt, das als *Schwingung* bezeichnet wird, um die Metallplatten selbst zum Schwingen zu bringen, benötigt ein passiver Summer ein Schwingungssignal. Wenn Sie ihn einfach mit Scratch einschalten, bewegen sich die Platten nur einmal und stoppen – das „Klick“-Geräusch ertönt, bis Ihr Programm den Stift das nächste Mal ein- oder ausschaltet.

Klicken Sie auf das rote Achteck, um Ihren Summer zu stoppen, aber stellen Sie sicher, dass Sie dies tun, wenn kein Ton zu hören ist, sonst summt der Summer weiter, bis Sie Ihr Programm erneut ausführen!



HERAUSFORDERUNG: SUMMER ÄNDERN

Wie könnten Sie das Programm ändern, dass der Summer kürzer ertönt? Können Sie eine Schaltung bauen, sodass der Summer mit einer Taste gesteuert wird?

Steuerung eines Summers in Python

Die Steuerung eines aktiven Summers durch die GPIO Zero-Bibliothek ist fast identisch mit der Steuerung einer LED, da er Ein- und Aus-Zustände hat. Sie benötigen jedoch eine andere Funktion: **buzzer**. Starten Sie ein neues Projekt in Thonny und speichern Sie es als **Buzzer**. Geben Sie dann Folgendes ein:

```
from gpiozero import Buzzer
from time import sleep
```

Wie bei den LEDs muss GPIO Zero wissen, an welchem Stift Ihr Summer angeschlossen ist, um ihn steuern zu können. Geben Sie Folgendes ein:

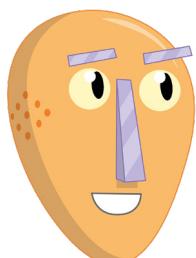
```
buzzer = Buzzer(15)
```

Von hier aus ist Ihr Programm fast identisch mit dem, das Sie zur Steuerung der LED geschrieben haben. Der einzige Unterschied (abgesehen von einer anderen GPIO-Stiftnummer) besteht darin, dass Sie **buzzer** anstelle von **led** verwenden. Geben Sie Folgendes ein:

```
while True:
    buzzer.on()
    sleep(1)
    buzzer.off()
    sleep(1)
```

Klicken Sie auf den Button „Run“, und Ihr Summer beginnt zu summieren: eine Sekunde an und eine Sekunde aus. Wenn Sie statt eines aktiven Summers einen passiven Summer verwenden, hören Sie statt eines Dauersummtos nur ein kurzes Klicken pro Sekunde: Das liegt daran, dass ein passiver Summer keinen Oszillator hat, um das schnell wechselnde Signal zu erzeugen, das die Platten im Inneren des Summers in Schwingung versetzt.

Klicken Sie auf den Button „Stop“, um das Programm zu beenden. Stellen Sie aber sicher, dass der Summer zu diesem Zeitpunkt keinen Ton von sich gibt, sonst summert er weiter, bis Sie Ihr Programm erneut starten!



HERAUSFORDERUNG: EIN BESSERER SUMMER



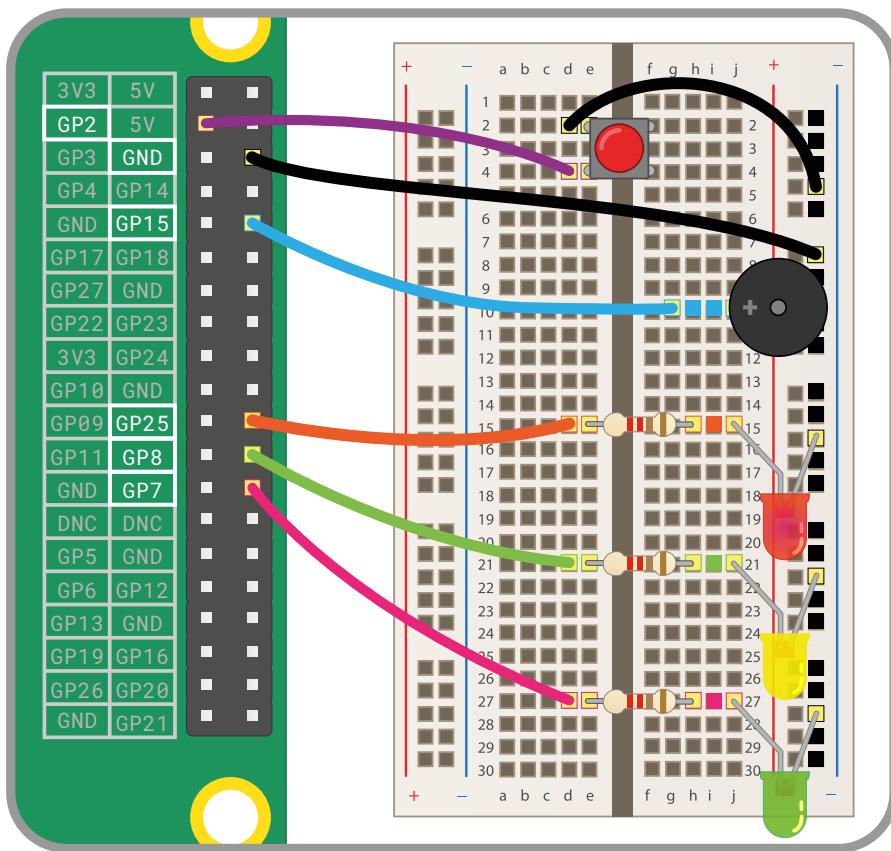
Wie könnten Sie das Programm ändern, dass der Summer kürzer ertönt? Können Sie eine Schaltung bauen, sodass der Summer mit einer Taste gesteuert wird?

Scratch-Projekt: Verkehrsampeln

Jetzt wissen Sie, wie man Taster, Summer und LEDs als Ein- und Ausgänge verwendet.

Sie sind nun soweit, ein Beispielprogramm für die reale Welt zu bauen: Ampeln, komplett mit einem Knopf, den Sie drücken können, um die Straße zu überqueren. Für dieses Projekt benötigen Sie eine Steckplatine, eine rote, eine gelbe und eine grüne LED, drei $330\ \Omega$ -Widerstände, einen Summer, einen Drucktaster und eine Auswahl an männlichen-zu-männlichen-(M2M)- und männlichen-zu-weiblichen-(M2F)-Jumperkabeln.

Beginnen Sie mit dem Aufbau der Schaltung (**Abbildung 6-6**), wobei der Summer an den GPIO 15-Stift angeschlossen wird (gekennzeichnet mit GP15 in **Abbildung 6-6**), die rote LED auf den GPIO 25-Stift (beschriftet mit GP25), die gelbe LED auf GPIO 8 (GP8), die grüne LED auf GPIO 7 (GP7) und der Schalter auf GPIO 2 (GP2). Denken Sie daran, die $330\ \Omega$ -Widerstände zwischen den GPIO-Stiften und den langen Beinen der LEDs anzuschließen, und verbinden Sie die zweiten Beine aller Ihrer Bauteile mit der Erdungsschiene Ihrer Steckplatine. Verbinden Sie schließlich die Erdungsschiene mit einem Massestift (mit GND bezeichnet) an den Raspberry Pi, um die Schaltung zu vervollständigen.



▲ Abbildung 6-6: Schaltplan für das Ampelprojekt

Starten Sie ein neues Scratch 3-Projekt und ziehen Sie dann einen **Wenn grüne Flagge angeklickt wird**-Block auf den Codebereich. Als Nächstes müssen Sie Scratch mitteilen, dass der GPIO 2-Stift, der mit dem Drucktaster in Ihrer Schaltung verbunden ist, ein Eingang und kein Ausgang ist: Ziehen Sie einen **set gpio to input pulled high**-Block aus der Kategorie Raspberry Pi GPIO der Blockpalette unter Ihren **Wenn grüne Flagge angeklickt wird**-Block. Klicken Sie auf den Abwärtspfeil neben der 0 und wählen Sie aus der Dropdown-Liste „2“.



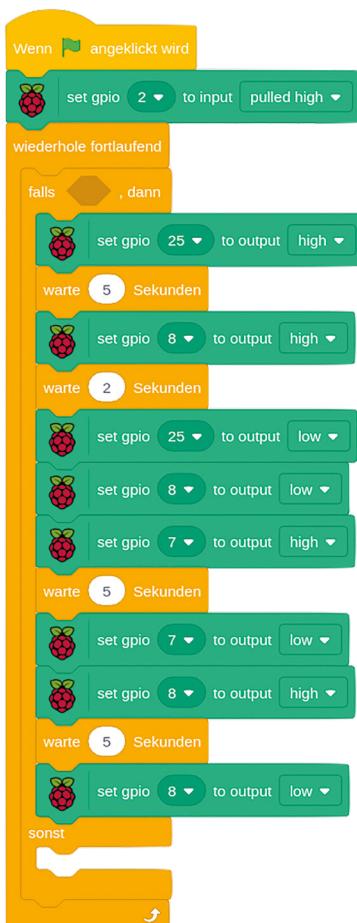
Als Nächstes müssen Sie Ihre Ampelsequenz erstellen. Ziehen Sie einen **wiederhole fortlaufend**-Block in Ihr Programm und füllen Sie ihn dann mit Blöcken, um die Ampel-LEDs nach einem Muster an- und auszuschalten. Denken Sie daran, an welche GPIO-Stifte welche Bauteile angeschlossen sind: Wenn Sie Stift 25 verwenden, verwenden Sie die rote LED, bei Stift 8 die gelbe LED und bei Stift 7 die grüne LED.



Klicken Sie auf die grüne Flagge und beobachten Sie Ihre LEDs. Zuerst leuchtet die rote Flagge, dann die rote und die gelbe, dann die grüne, dann die gelbe und schließlich

wiederholt sich die Sequenz mit dem roten Licht noch einmal. Dies ist das Mutter der Ampeln in Großbritannien. Auf Wunsch können Sie die Sequenz dem Muster in Ihrem eigenen Land anpassen.

Um einen Fußgängerstreifen zu simulieren, muss Ihr Programm überwachen, ob der Knopf gedrückt wird. Klicken Sie auf das rote Achteck, um Ihr Programm zu stoppen, falls es gerade läuft. Ziehen Sie einen **falls dann sonst**-Block auf Ihren Skriptbereich und verbinden Sie ihn so, dass er sich direkt unter Ihrem **wiederhole fortlaufend**-Block befindet, mit Ihrer Ampelsequenz im „falls dann“-Abschnitt. Lassen Sie die rautenförmige Lücke vorderhand leer.

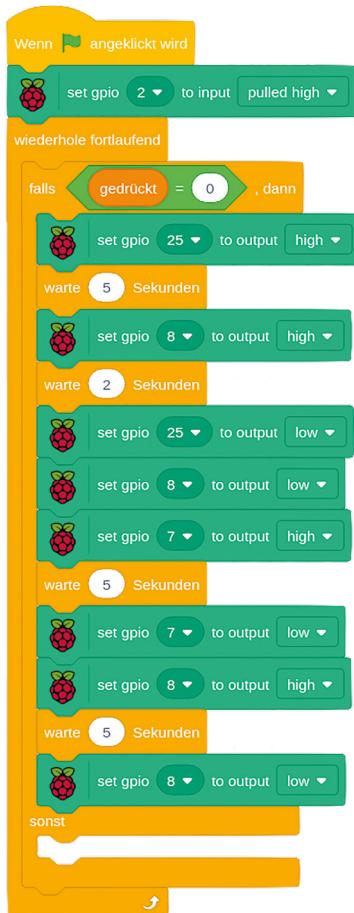


Bei einem echten Fußgängerstreifen schaltet die Ampel beim Drücken des Knopfs nicht sofort auf Rot, sondern wartet auf das nächste rote Licht in der Sequenz. Um dies in Ihr eigenes Programm einzubauen, ziehen Sie einen **when gpio is low**-Block auf den Code-Bereich und wählen 2 aus der Dropdown-Liste. Ziehen Sie dann einen **setze gedrückt auf 1**-Block darunter.



Dieser Blockstapel wartet darauf, dass der Knopf gedrückt wird und setzt dann die Variable „gedrückt“ auf 1. Wenn Sie eine Variable auf diese Weise einstellen, können Sie die Tatsache, dass der Knopf gedrückt wurde, speichern, auch wenn keine sofortige Reaktion erfolgt.

Gehen Sie zurück zu Ihrem ursprünglichen Blockstapel und suchen Sie den **falls dann**-Block. Ziehen Sie einen **=**-Operator-Block in das karoförmige Rohteil des **falls dann**-Blocks und dann einen **gedrückt**-Reporter-Block in den ersten freien Raum. Überschreiben Sie die „50“ auf der rechten Seite des Blocks durch eine „0“.



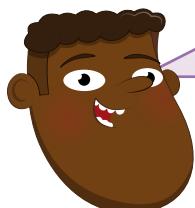
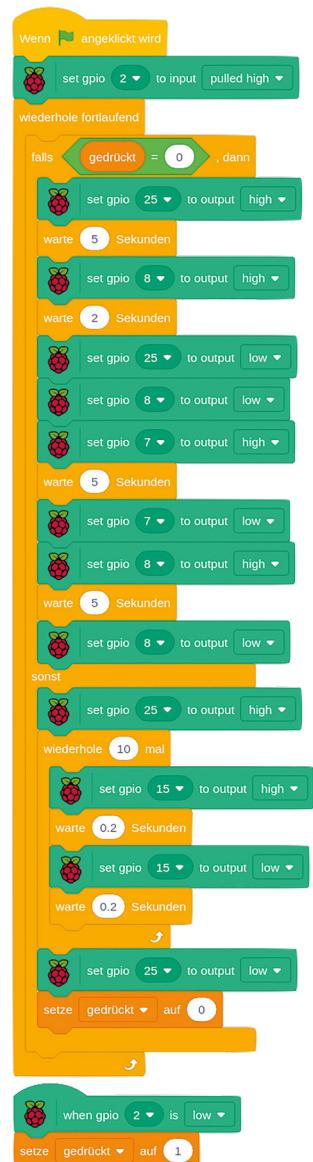
Klicken Sie auf die grüne Flagge und beobachten Sie, wie die Ampeln die Abfolge durchlaufen. Drücken Sie den Drucktaster: Zunächst sieht es so aus, als ob nichts passiert, aber sobald die Sequenz zu Ende ist – nur die gelbe LED leuchtet – geht die Ampel aus und bleibt dank der Variablen „gedrückt“ aus.

Jetzt müssen Sie nur noch dafür sorgen, dass der Knopf für den Fußgängerstreifen tatsächlich etwas anderes tut als das Licht auszuschalten. Suchen Sie im Hauptblockstapel den **sonst**-Block und ziehen Sie einen **set gpio 25 to output high**-Block hinein. Denken Sie daran, die Standard-GPIO-Stiftnummer so zu ändern, dass sie mit dem Stift übereinstimmt, an dem Ihre rote LED angeschlossen ist.

Erstellen Sie darunter, noch im **sonst**-Block, ein Muster für den Summer: Ziehen Sie einen **wiederhole 10 mal**-Block und füllen Sie ihn dann mit den Blöcken **set gpio 15 to output high**, **warte 0.2 Sekunden**, **set gpio 15 to output low** und **warte 0.2 Sekunden**. Dabei ändern Sie die GPIO-Stiftwerte so, dass sie mit dem Stift für das Summer-Bauteil übereinstimmen. Schließlich fügen Sie unter dem unteren Ende Ihres **wiederhole 10 mal**-Blocks, aber immer noch im **sonst**-Block, einen **set gpio 25 to output low**-Block und einen **setze gedrückt auf 0**-Block hinzu. Der letzte Block setzt die Variable zurück, die das Drücken des Knopfs speichert, damit sich die Summersequenz nicht einfach endlos wiederholt.

Klicken Sie auf die grüne Flagge und drücken Sie dann den Schalter auf Ihrer Steckplatine. Nachdem die Sequenz abgeschlossen ist, sehen Sie, dass die rote Ampel aufleuchtet und der Summer ertönt, damit die Fußgänger wissen, dass das Überqueren sicher ist. Nach einigen Sekunden hört der Summer auf, die Ampelsequenz beginnt erneut und dauert bis zum nächsten Knopfdruck an.

Herzlichen Glückwunsch: Sie haben Ihre eigene, voll funktionsfähige Ampelanlage für einen Fußgängerstreifen programmiert!



HERAUSFORDERUNG: KÖNNEN SIE ES VERBESSERN?

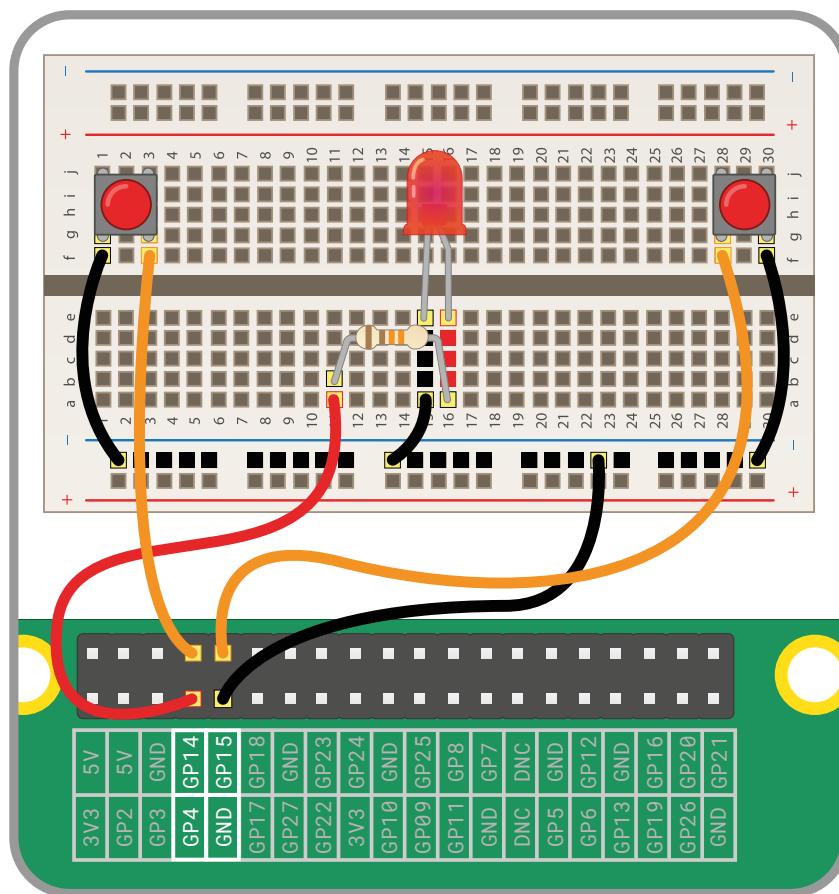
Können Sie das Programm ändern, damit dem Fußgänger mehr Zeit zum Überqueren bleibt? Können Sie Informationen über die Ampelmuster anderer Länder finden und Ihre Ampeln entsprechend umprogrammieren? Wie könnten Sie die LEDs weniger hell machen?



Python-Projekt: Reaktionsspiel

Jetzt, da Sie wissen, wie man Taster und LEDs als Ein- und Ausgänge verwendet, sind Sie soweit, ein Beispielprogramm für die reale Welt zu bauen: ein Reaktionsspiel für zwei Spieler, bei dem es darum geht, wer am schnellsten reagiert! Für dieses Projekt benötigen Sie eine Steckplatine, eine LED und einen 330-Ω-Widerstand, zwei Drucktaster, einige männliche-zu-weibliche-(M2F)-Jumperkabel und einige männliche-zu-männliche-(M2M)-Jumperkabel.

Beginnen Sie mit dem Aufbau der Schaltung (**Abbildung 6-7**): Verbinden Sie den ersten Schalter auf der linken Seite Ihrer Steckplatine mit dem GPIO 14-Stift (beschriftet mit GP14 in **Abbildung 6-7**), den zweiten Schalter auf der rechten Seite Ihrer Steckplatine mit dem GPIO 15-Stift (gekennzeichnet mit GP15), das längere Bein der LEDs mit dem 330-Ω-Widerstand, der dann an den GPIO 4-Stift (gekennzeichnet mit GP4) des Raspberry Pi angeschlossen wird, und die zweiten Beine aller Ihrer Bauteile mit der Erdungsschiene Ihrer Steckplatine. Verbinden Sie schließlich die Erdungsschiene mit dem Massestift des Raspberry Pi (mit GND bezeichnet).



▲ Abbildung 6-7: Schaltplan für das Reaktionsspiel

Starten Sie ein neues Projekt in Thonny und speichern Sie es als **Reaktionsspiel**. Sie werden die Funktionen **LED** und **button** aus der Bibliothek GPIO Zero und die Funktion **sleep** aus der Bibliothek „time“ verwenden. Anstatt die beiden GPIO Zero-Funktionen in zwei getrennte Zeilen zu importieren, können Sie Zeit sparen und sie zusammen importieren, indem Sie sie mit einem Komma-Symbol (,) trennen. Geben Sie Folgendes in den Skriptbereich ein:

```
from gpiozero import LED, Button  
from time import sleep
```

Wie zuvor müssen Sie GPIO Zero mitteilen, an welche Stifte die beiden Taster und die LED angeschlossen sind. Geben Sie Folgendes ein:

```
led = LED(4)  
right_button = Button(15)  
left_button = Button(14)
```

Fügen Sie nun Anweisungen zum Ein- und Ausschalten der LED hinzu, damit Sie überprüfen können, ob sie korrekt funktioniert:

```
led.on()  
sleep(5)  
led.off()
```

Klicken Sie auf den Button „Run“: Die LED leuchtet fünf Sekunden lang auf, dann erlischt sie und das Programm beendet sich. Für ein Reaktionsspiel ist es jedoch nicht sinnvoll, die LED nach jeweils genau 5 Sekunden ausgehen zu lassen. Fügen Sie unter der Zeile **from time import sleep** Folgendes hinzu:

```
from random import uniform
```

Mit der Zufallsbibliothek (`random`) können Sie, wie der Name sagt, Zahlen nach Zufallsprinzip erzeugen (hier mit einer gleichmäßigen Verteilung – siehe rpf.io/uniform). Suchen Sie die Zeile `sleep(5)` und ändern Sie wie folgt:

```
sleep(uniform(5, 10))
```

Klicken Sie erneut auf den Button „Run“: Diesmal leuchtet die LED für eine „zufällige“ Anzahl Sekunden zwischen 5 und 10. Zählen Sie, um zu sehen, wie lange es dauert, bis die LED erlischt, und klicken Sie dann noch einige Male auf den Button „Run“: Sie werden sehen, dass die Zeit bei jedem Durchlauf anders ist.

Um die Tasten in Auslöser für jeden Spieler zu verwandeln, müssen Sie noch eine Funktion hinzufügen. Geben Sie ganz unten in Ihrem Programm Folgendes ein:

```
def pressed(button):
    print(str(button.pin.number) + " hat das Spiel gewonnen")
```

Denken Sie daran, dass Python Einrückung verwendet, um zu wissen, welche Zeilen Teil der Funktion sind. Thonny wird die zweite Zeile automatisch für Sie einrücken. Fügen Sie schließlich die folgenden beiden Zeilen zum Erkennen der Spieler hinzu, die eine Taste drücken. Denken Sie daran, dass sie nicht eingerückt sein dürfen, sonst behandelt Python sie als Teil Ihrer Funktion.

```
right_button.when_pressed = pressed
left_button.when_pressed = pressed
```

Führen Sie Ihr Programm aus und drücken Sie eine der beiden Tasten, sobald die LED erlischt. Eine Meldung zur ersten gedrückten Taste wird in der Python-Shell unten im Thonny-Fenster angezeigt. Diese Meldung erscheint jedes Mal, wenn eine der beiden Tasten gedrückt wird – und dabei wird für die Taste die Stiftnummer verwendet und nicht ein aussagekräftiger Name.

Um Abhilfe zu schaffen, fragen Sie die Spieler nach ihren Namen. Unterhalb der Linie **from random import uniform** geben Sie Folgendes ein:

```
left_name = input("Der Name des linken Spielers ist ")
right_name = input("Der Name des rechten Spielers ist ")
```

Gehen Sie zurück zu Ihrer Funktion und ersetzen Sie die Zeile **print(str(button.pin.number) + " hat das Spiel gewonnen")** durch:

```
if button.pin.number == 14:
    print(left_name + " hat das Spiel gewonnen")
else:
    print(right_name + " hat das Spiel gewonnen")
```

Klicken Sie auf den Button „Run“ und geben Sie dann die Namen der beiden Spieler in den Python-Shell-Bereich ein. Versuchen Sie, die Taste so schnell wie möglich nach dem Erlöschen der LED zu drücken. Sie werden sehen, dass jetzt anstelle der Stiftnummer der Spielername ausgegeben wird.

Zur Behebung des Problems, dass alle Tastendrücke als „hat das Spiel gewonnen“ gemeldet werden, müssen Sie eine neue Funktion aus der Bibliothek „sys“ – kurz für *System* hinzufügen: **exit**. Unter der letzten Zeile **import**, geben Sie Folgendes ein:

```
from os import _exit
```

Am Ende Ihrer Funktion geben Sie unter der Zeile `print(right_name + " hat das Spiel gewonnen")` Folgendes ein:

```
_exit(0)
```

Die Einrückung ist hier wichtig: `_exit(0)` sollte um vier Leerzeichen eingerückt werden, ausgerichtet mit `exit`: zwei Zeilen darüber und `if` wiederum zwei Zeilen darüber. Diese Anweisung weist Python an, das Programm zu stoppen, nachdem die die erste Taste gedrückt wurde, d. h. der Spieler, dessen Taste als zweite gedrückt wird, erhält keine Belohnung!

Ihr fertiges Programm sollte so aussehen:

```
from gpiozero import LED, Button
from time import sleep
from random import uniform
from os import _exit

left_name = input("Der Name des linken Spielers ist ")
right_name = input ("Der Name des rechten Spielers ist ")
led = LED(4)
right_button = Button(15)
left_button = Button(14)

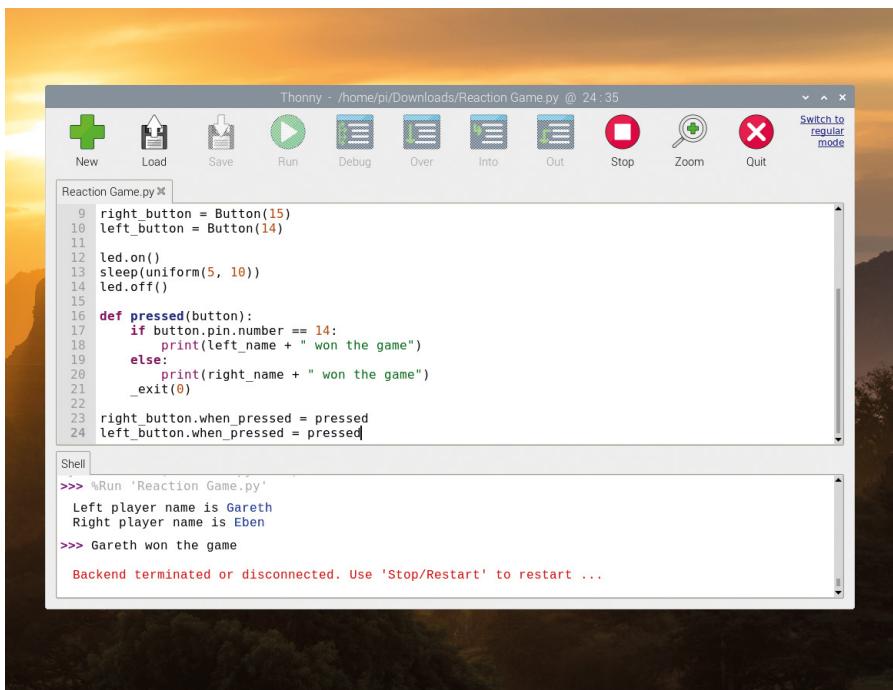
led.on()
sleep(uniform(5, 10))
led.off()

def pressed(button):
    if button.pin.number == 14:
        print(left_name + " hat das Spiel gewonnen")
    else:
        print(right_name + " hat das Spiel gewonnen")
    _exit(0)

right_button.when_pressed = pressed
left_button.when_pressed = pressed
```

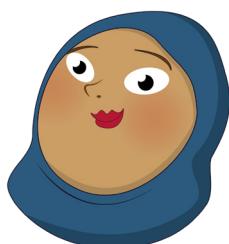
Klicken Sie auf den Button „Run“, geben Sie die Namen der Spieler ein, warten Sie, bis die LED erlischt, und Sie sehen den Namen des siegreichen Spielers. Sie werden auch eine Nachricht von Python selbst sehen: `Backend terminated or disconnected`. Use 'Stop/Restart' to restart ... Dies bedeutet einfach, dass Python Ihren

`_exit(0)`-Befehl erhalten und das Programm pausiert hat, aber Sie müssen auf das Stoppsymbol klicken, um das Programm vollständig zu beenden und für eine weitere Runde vorzubereiten (**Abbildung 6-8**).



▲ Abbildung 6-8: Sobald der Gewinner feststeht, müssen Sie das Programm beenden

Herzlichen Glückwunsch: Sie haben mit Physical Computing Ihr eigenes Spiel programmiert!



HERAUSFORDERUNG: DAS SPIEL VERBESSERN

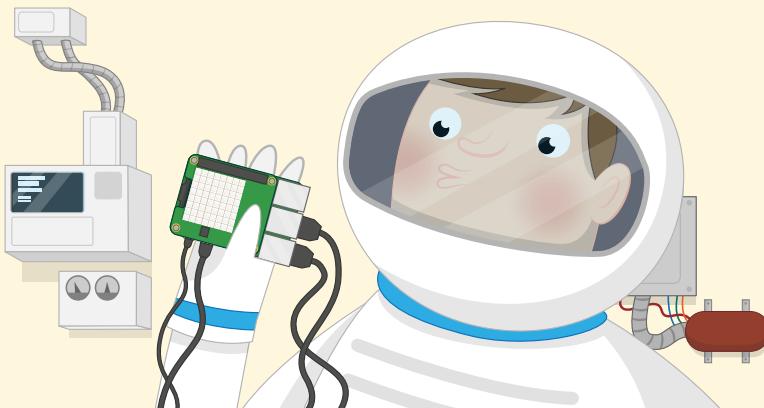


Können Sie eine Schleife hinzufügen, damit das Spiel kontinuierlich läuft? Denken Sie daran, die Anweisung `_exit(0)` zuerst zu entfernen! Können Sie einen Punktezähler hinzufügen, damit Sie sehen können, wer über mehrere Runden gewinnt? Wie wäre es mit einem Timer, damit Sie sehen können, wie lange Sie gebraucht haben, um auf das Erlöschen des Lichts zu reagieren?

Kapitel 7

Physical Computing mit dem Sense HAT

Der Sense HAT, der sogar auf der Internationalen Raumstation zum Einsatz kommt, ist eine multifunktionale Zusatzplatine für Raspberry Pis mit eingebauten Sensoren und einer LED-Matrixanzeige



Der Raspberry Pi unterstützt eine spezielle Art von Zusatzplatine namens *Hardware Attached on Top* (HAT - oben angebrachtes geschlossene Hardware). HATs können alles von Mikrofonen und Lichtern bis hin zu elektronischen Relais und Bildschirmen zu einem Raspberry Pi hinzufügen, aber vor allem eine HAT ist etwas ganz Besonderes: der Sense HAT.

Er wurde speziell für die Weltraummission Astro Pi entwickelt. Astro Pi ist ein gemeinsames Projekt der Raspberry Pi Foundation, der britischen Raumfahrtbehörde und der Europäischen Raumfahrtbehörde, bei dem Raspberry Pi-Boards und Sense HATs an Bord einer Orbital Science Cygnus-Frachtrakete zur Internationalen Raumstation gebracht wurden. Seitdem sie ihre neue Umlaufbahn hoch über der Erde sicher erreicht haben, werden die Sense HATs – von den Astronauten Ed und Izzy getauft – zur Ausführung von Code und zur Durchführung wissenschaftlicher Experimente verwendet, die von Schulkindern aus ganz Europa beigesteuert werden.

Ed und Izzy zu weit entfernt, als dass Sie sie selbst benutzen könnten, aber die gleiche Sense HAT-Hardware finden Sie auch hier auf der Erde bei allen Raspberry Pi-Händlern. Wenn Sie zurzeit keinen Sense HAT kaufen wollen, können Sie ihn auch einfach mit Hilfe von Software simulieren!

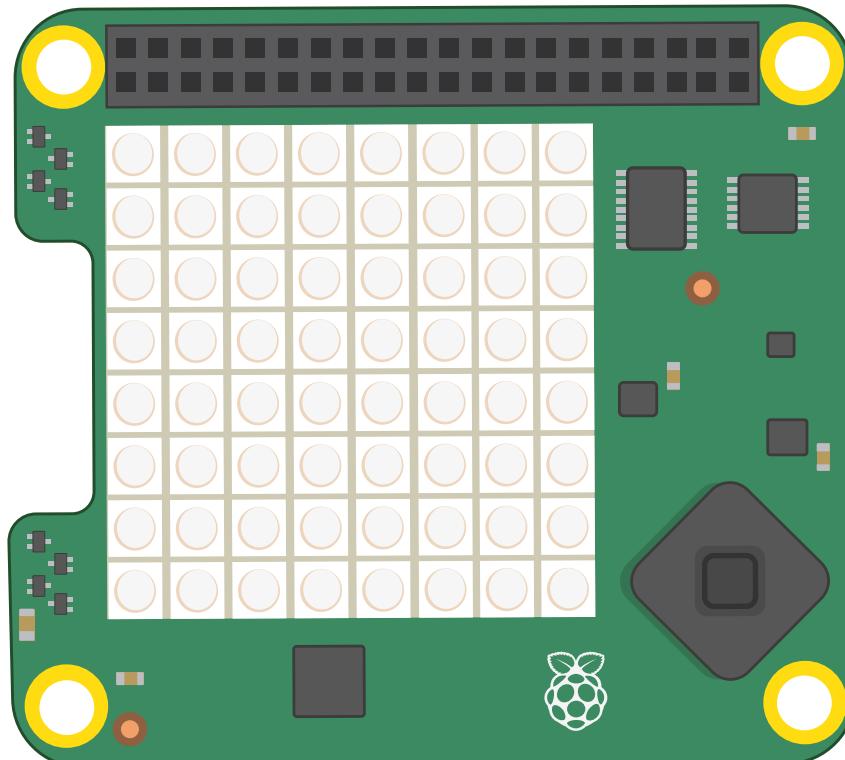
REAL ODER SIMULIERT



Dieses Kapitel erarbeiten Sie sich am besten zusammen mit einem echten Sense HAT, der an eine GPIO-Stiftleiste des Raspberry Pi angeschlossen wird. Sollten Sie keinen haben, kann der Abschnitt mit dem Titel „Installation des Sense HAT“ übersprungen werden und Sie probieren die Projekte einfach im Sense HAT-Emulator aus. Sie funktionieren genauso gut!

Einführung in den Sense HAT

Der Sense HAT ist ein mächtiges, multifunktionales Add-on für den Raspberry Pi. Neben einer 8×8-Matrix aus 64 programmierbaren roten, grünen und blauen (RGB) LEDs, die so gesteuert werden können, dass sie jede beliebige Farbe aus einem Bereich von Millionen von Farben erzeugen, umfasst der Sense HAT einen Fünf-Wege-Joystick-Controller und sechs On-Board-Sensoren.



Gyroskopsensor: Wird verwendet, um Winkeländerungen im Lauf der Zeit zu erfassen, dies ist technisch bekannt als *Winkelgeschwindigkeit*. Der Sensor verfolgt dazu die Richtung des Schwerkraftfeldes der Erde – die Kraft, die Dinge zum Zentrum des Planeten hinunterzieht. Vereinfacht ausgedrückt, kann der Gyroskopsensor erkennen, dass Sie den Sense HAT relativ zur Erdoberfläche drehen und wie schnell er sich dreht.

Beschleunigungsmesser: Ähnlich wie der Gyroskopsensor, aber anstatt einen Winkel relativ zur Schwerkraft der Erde zu überwachen, misst er die Beschleunigungskraft in mehreren Richtungen. Kombinierte Messwerte (Daten) der beiden Sensoren können Ihnen helfen zu verfolgen, in welche Richtung ein Sense HAT zeigt und wie er bewegt wird.

Magnetometer: Misst die Stärke eines Magnetfeldes und ist ein weiterer Sensor, der helfen kann, die Bewegungen des Sense HAT zu verfolgen: Durch die Messung des natürlichen Magnetfeldes der Erde kann das Magnetometer die Richtung des magnetischen Nordens herausfinden. Derselbe Sensor kann auch verwendet werden, um metallische Objekte und sogar elektrische Felder zu erkennen. Alle drei dieser Sensoren sind in einem einzigen Chip eingebaut, der auf der Platine des Sense HAT mit „ACCEL/GYRO/MAG“ beschriftet ist.

Feuchtigkeitssensor: Misst die Menge des Wasserdampfs in der Luft, bekannt als *relative Luftfeuchtigkeit*. Die relative Luftfeuchtigkeit kann zwischen 0 %, wenn gar kein Wasser vorhanden ist, und 100 %, wenn die Luft vollständig gesättigt ist, liegen. Feuchtigkeitsdaten können verwendet werden, um festzustellen, ob es binnen kurzem regnen wird!

Sensor für den barometrischen Druck: auch bekannt als *Barometer*. Er misst den Luftdruck. Obwohl die meisten Menschen den barometrischen Druck aus der Wettervorhersage kennen, hat das Barometer noch einen weiteren, geheimen anderen NutzenEinsatzzweck: Es kann verfolgen, ob Sie einen Hügel oder Berg hinauf- oder hinabsteigen, da die Luft dünner wird und der Druck sinkt, je weiter Sie sich vom Meeresspiegel entfernen.

Temperatursensor: Misst, wie heiß oder kalt die Umgebung ist, obwohl er auch davon beeinflusst wird, wie heiß oder kalt der Sense HAT ist: Wenn Sie eine Abdeckung verwenden,

SENSE HAT AUF RASPBERRY PI 400

Die Sense HAT ist vollständig kompatibel mit Raspberry Pi 400 und kann direkt in die GPIO-Stifteleiste auf der Rückseite eingesetzt werden. Dies bedeutet jedoch, dass die LEDs von Ihnen weg zeigen und die Platine verkehrt herum ausgerichtet ist.

Um dies zu beheben, benötigen Sie ein GPIO-Verlängerungskabel oder eine GPIO-Platine.

Zu den kompatiblen Erweiterungen gehört die Black HAT Hack3r-Reihe von Pimoroni. Sie

können die Sense HAT mit der Black HAT Hack3r-Platine selbst verwenden oder einfach das mitgelieferte 40-polige Flachbandkabel als Verlängerung verwenden. Überprüfen Sie jedoch immer die Anweisungen des Herstellers, um sicherzustellen, dass Sie das Kabel und die Sense HAT richtig herum anschließen!



kann es sein, dass Ihre Messwerte höher als erwartet ausfallen. Der Sense HAT hat keinen separaten Temperatursensor. Stattdessen verwendet er Temperatursensoren, die in die Feuchte- und Luftdrucksensoren mit eingebaut sind. Ein Programm kann einen oder beide dieser Sensoren verwenden – wie Sie wollen.

Installation des Sense HAT

Wenn Sie den Sense HAT als reale Hardware haben, packen Sie ihn zunächst aus und vergewissern Sie sich, dass alle Teile vorhanden sind: Neben dem Sense HAT selbst sind vier Metall- oder Plastiksäulen, bekannt als *Abstandshalter*, und acht Schrauben enthalten. Möglicherweise sind auch einige Metallstifte, verbaut in einer schwarzen Kunststoffleiste, vorhanden, ähnlich zu den GPIO-Stiften beim Raspberry Pi. In diesem Fall schieben Sie diese Leiste mit den Metallstiften voran durch die Unterseite des Sense HAT, bis Sie ein Klicken hören.

Die Abstandshalter sollen verhindern, dass sich der Sense HAT bei der Verwendung des Joysticks verbiegt oder verformt. Dieer Sense HAT funktioniert zwar auch ohne Abstandshalter, aber Sie schützen durch ihren Einsatz damitlhren Sense HAT-, den Raspberry Pi- und die GPIO-Stiftleiste vor Beschädigung.

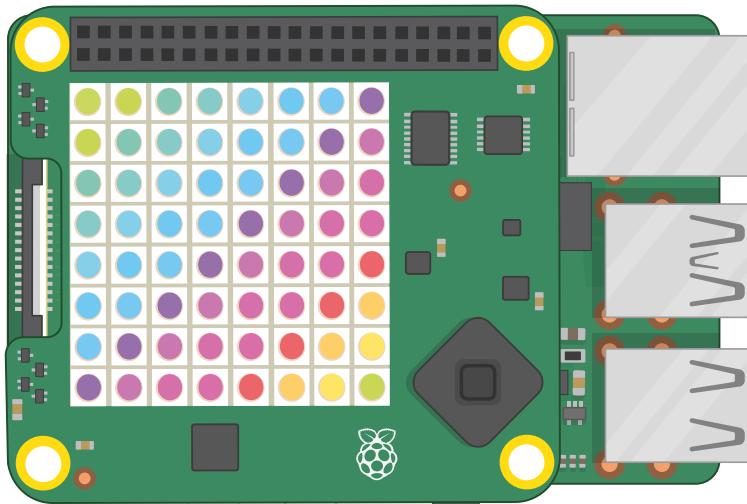
WARNUNG!



Hardware Attached on Top (HAT)-Module sollten immer nur in die GPIO-Stiftleiste eingesteckt und aus dieser entfernt werden, während Ihr Raspberry Pi ausgeschaltet und von der Stromversorgung getrennt ist. Achten Sie bei der Installation immer darauf, den HAT flach zu halten, und überprüfen Sie vor dem Herunterdrücken, ob er mit den GPIO-Stiften ausgerichtet ist.

Installieren Sie die Abstandshalter, indem Sie vier der Schrauben von der Unterseite des Raspberry Pi aus durch die vier Befestigungslöcher an jeder Ecke nach oben drücken und dann die Abstandshalter auf die Schrauben drehen. Stecken Sie den Sense HAT nun auf die GPIO-Stiftleiste des Raspberry Pi auf, achten Sie darauf, dass er richtig mit den Stiften darunter ausgerichtet ist, und schieben Sie ihn mit Gefühl bis auf Anschlag nach unten. Achten Sie beim Einsticken darauf, den Sense HAT soweit möglich parallel zur Raspberry Pi Platine zu halten, und ihn nicht gegen die Platine zu verkanten. Schrauben Sie abschließend die letzten vier Schrauben durch die Montagelöcher des Sense HAT und in die Abstandshalter, die Sie zuvor installiert haben. Wenn sieer richtig installiert ist, sollte der Sense HAT flach und eben sein und sich nicht verbiegen oder wackeln, wenn Sie auf den Joystick drücken.

Schließen Sie den Strom nun wieder an Ihren Raspberry Pi an, und Sie werden sehen, wie die LEDs auf dem Sense HAT in einem Regenbogenmuster aufleuchten (**Abbildung 7-1**), und wieder ausgehenden Raspberry Pi . Ihre Sense HAT ist jetzt installiert!



▲ Abbildung 7-1: Ein Regenbogenmuster erscheint, wenn der Strom gerade eingeschaltet worden ist

Wenn Sie den Sense HAT wieder entfernen möchten, lösen Sie einfach die oberen Schrauben und heben Sie den HAT ab – achten Sie darauf, die Stifte der GPIO-Stiftleiste nicht zu verbiegen, da der HAT recht fest sitzt (Sie müssen sie ihn eventuell mit einem kleinen Schraubendreher abhebeln) – und entfernen Sie dann die Abstandshalter vom Raspberry Pi.

Hallo, Sense HAT!

Wie bei allen Programmierprojekten ist beginnt man auch bei dem Sense HAT damit, eine Begrüßungsnachricht über die LED-Anzeige laufen zu lassen. Wenn Sie den Sense HAT-Emulator verwenden, starten Sie ihn jetzt, indem Sie auf das Raspberry Pi OS Menüsymbol des klicken, die Kategorie Entwicklung wählen und auf den Sense HAT Emulator klicken.



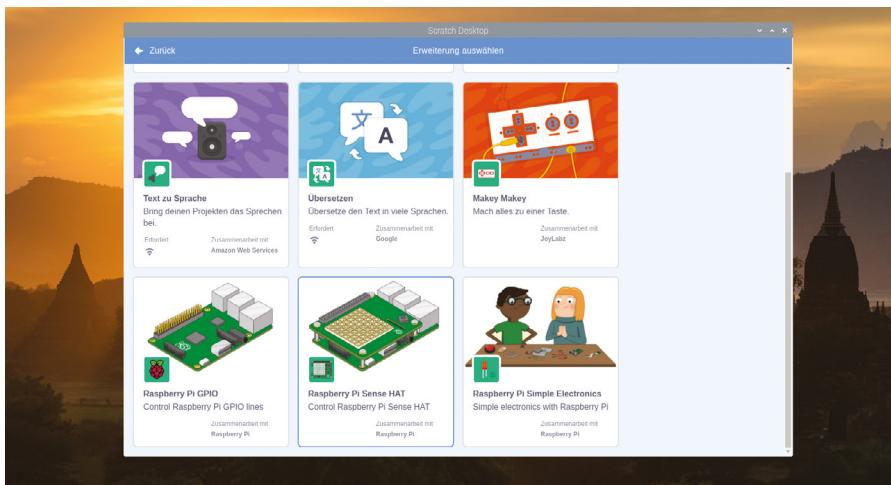
PROGRAMMIERERFAHRUNG

Dieses Kapitel setzt Erfahrung mit Scratch 3 oder Python und der integrierten Entwicklungsumgebung (IDE) Thonny voraus, je nachdem, ob Sie die Code-Beispiele für Scratch oder Python durcharbeiten – oder beides! Falls Sie dies noch nicht getan haben, lesen Sie bitte **Kapitel 4, Programmieren mit Scratch** oder **Kapitel 5, Programmieren mit Python** und arbeiten Sie zuerst die dort enthaltenen Projekte durch.



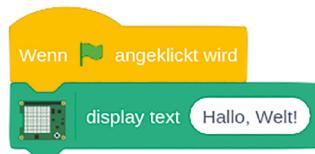
Grüße von Scratch

Starten Sie Scratch 3 aus dem Raspberry Pi OS-Menü. Klicken Sie auf den Button „Erweiterung hinzufügen“ unten links im Scratch-Fenster. Klicken Sie auf die Erweiterung Raspberry Pi Sense HAT (**Abbildung 7-2**). Dadurch werden die Blöcke geladen, die Sie zur Steuerung der verschiedenen Funktionen des Sense HAT, einschließlich ihrer LED-Anzeige, benötigen. Sie finden sie bei Bedarf in der Kategorie „Raspberry Pi Sense HAT“.

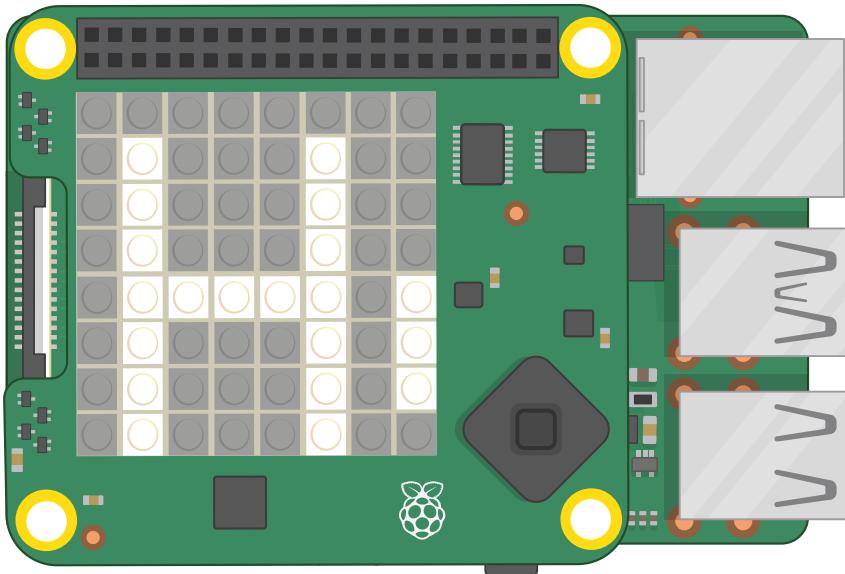


▲ Abbildung 7-2: Hinzufügen der Raspberry Pi Sense HAT-Erweiterung in Scratch 3

Beginnen Sie mit dem Ziehen eines „Ereignisse“-Blocks **Wenn [Flagge] angeklickt wird** auf den Skriptbereich und ziehen Sie dann einen Block **display text Hello!** direkt darunter. Bearbeiten Sie den Text so, dass der Block so aussieht: **display text Hallo Welt!**.

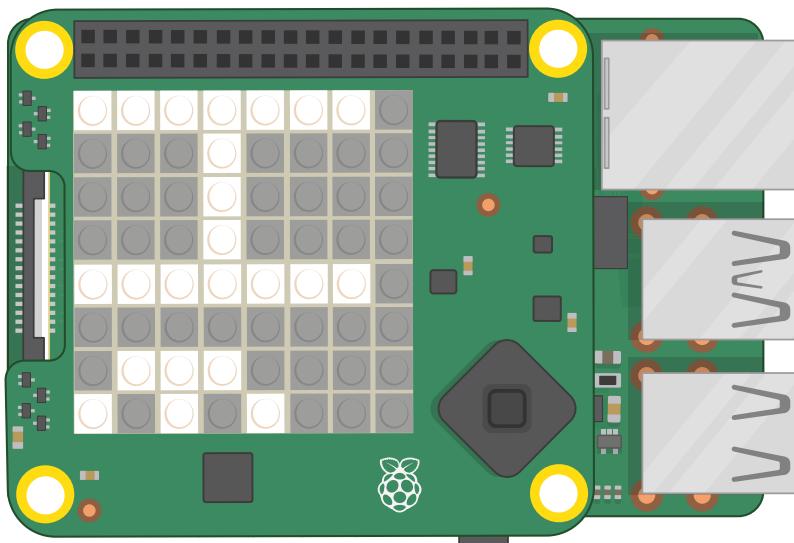


Klicken Sie auf die grüne Flagge im Bühnenbereich und beobachten Sie Ihren Sense HAT oder den Sense HAT-Emulator: Die Nachricht läuft langsam über die LED-Matrix des Sense HAT. Die LED-Pixel leuchten auf, um nacheinander die einzelnen Buchstaben zu bilden (**Abbildung 7-3** umseitig). Herzlichen Glückwunsch: Ihr Programm ist ein Erfolg!



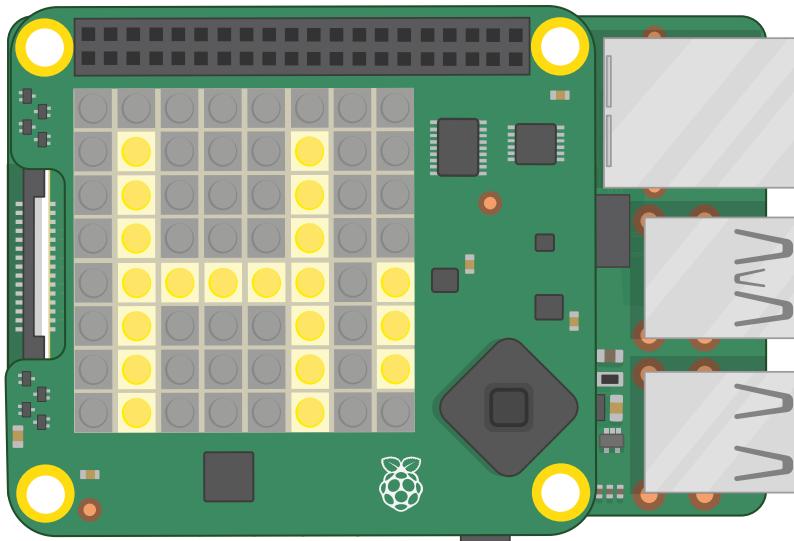
▲ Abbildung 7-3: Ihre Nachricht läuft über die LED-Matrix

Jetzt da Sie eine einfache Nachricht durchlaufen lassen können, wird es Zeit, sich damit zu befassen, wie diese Nachricht angezeigt wird.. Sie können nicht nur die anzuzeigende Nachricht ändern, sondern auch die Rotation –wie herum die Nachricht auf dem Sense HAT angezeigt wird. Ziehen Sie einen Block `set rotation to 0 degrees` aus der Blockpalette und fügen Sie ihn unter `Wenn grüne Flagge angeklickt wird` und über `display text Hallo Welt!` ein, klicken Sie dann auf den Abwärtspfeil neben 0 und ändern Sie den Wert auf 90. Klicken Sie auf die grüne Flagge. Sie sehen dieselbe Nachricht wie zuvor, aber anstatt von links nach rechts durchzulaufen, läuft sie von unten nach oben durch (**Abbildung 7-4**) – Sie müssen den Kopf oder den Sense HAT drehen, um den Text zu lesen!



▲ Abbildung 7-4: Diesmal läuft die Nachricht vertikal durch

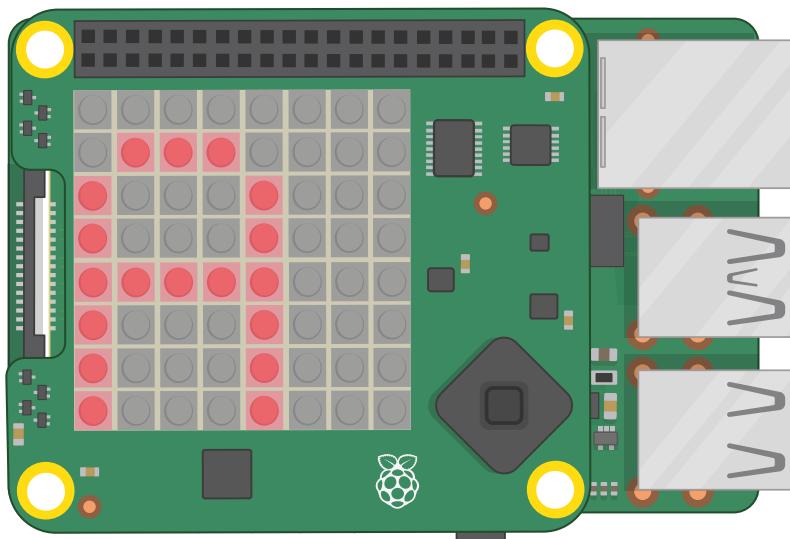
Ändern Sie nun die Rotation wieder auf 0 und ziehen Sie dann einen Block `set colour` zwischen `set rotation to 0 degrees` und `display text Hallo Welt!`. Klicken Sie auf die Farbe am Ende des Blocks, um die Farbauswahl von Scratch aufzurufen und ein schön leuchtendes Gelb zu finden. Klicken Sie dann auf die grüne Flagge, um zu sehen, wie sich die Ausgabe Ihres Programms verändert hat ([Abbildung 7-5](#)).



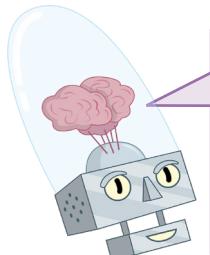
▲ Abbildung 7-5: Ändern der Farbe des Textes

Zum Schluss ziehen Sie einen Block **set background** zwischen **set colour to gelb** und **display text Hallo Welt!**, dann klicken Sie auf die Farbe, um die Farbauswahl wieder aufzurufen. Diesmal hat die Wahl einer Farbe keinen Einfluss auf die LEDs, aus denen die Botschaft besteht, sondern auf die übrigen – den sogenannten Hintergrund. Finden Sie eine schöne blaue Farbe und klicken Sie dann erneut auf die grüne Flagge: Diesmal wird Ihre Botschaft in leuchtendem Gelb auf blauem Hintergrund erscheinen. Versuchen Sie, diese Farben zu ändern, um Ihre Lieblingskombination zu finden – nicht alle Farben passen gut zusammen!

Sie können nicht nur ganze Nachrichten durchlaufen lassen, sondern auch einzelne Buchstaben anzeigen. Ziehen Sie Ihren Block **display text** aus dem Skriptbereich, um ihn zu löschen, und ziehen Sie dann stattdessen einen anderen Block **display character A** an seine Stelle. Klicken Sie auf die grüne Flagge und Sie sehen den Unterschied: Dieser Block zeigt immer nur einen Buchstaben auf einmal an und der Buchstabe bleibt auf dem Sense HAT, ohne durchzulaufen oder zu verschwinden, solange bis Sie eine andere Anweisung übermitteln. Für diesen Block gelten die gleichen Farbkontrollblöcke wie für den Block **display text**: Versuchen Sie, die Farbe des Buchstabens in Rot zu ändern (**Abbildung 7-6**).



▲ Abbildung 7-6: Anzeigen eines einzelnen Buchstabens



HERAUSFORDERUNG: DIE NACHRICHT WIEDERHOLEN



Können Sie Ihr Wissen über Schleifen nutzen, um eine scrollende Nachricht zu wiederholen? Können Sie ein Programm erstellen, das ein Wort Buchstabe für Buchstabe mit verschiedenen Farben buchstabiert?

Grüße von Python

Laden Sie Thonny, indem Sie auf das Raspberry Pi OS-Menüsymbol klicken, Entwicklung wählen und auf Thonny klicken. Wenn Sie den Sense HAT-Emulator verwenden und dieser vom Thonny-Fenster verdeckt wird, klicken Sie auf die Titelleiste eines der beiden Fenster – oben, in blau –, halten Sie die Maustaste gedrückt und ziehen Sie es, um es auf dem Desktop so zu verschieben, bis Sie beide Fenster sehen können.



PYTHON-ZEILENÄNDERUNG



Python-Code, der für einen realen Sense HAT geschrieben wurde, läuft auf dem Sense HAT-Emulator und umgekehrt, mit nur einer Änderung. Wenn Sie den Sense HAT-Emulator mit Python verwenden, müssen Sie die Zeile von `sense_hat import SenseHat` in allen Programmen in diesem Kapitel in `from sense_emu import SenseHat` ändern. Wenn Sie sie dann wieder mit einem echten Sense HAT verwenden wollen, ändern Sie einfach die Zeile zurück!

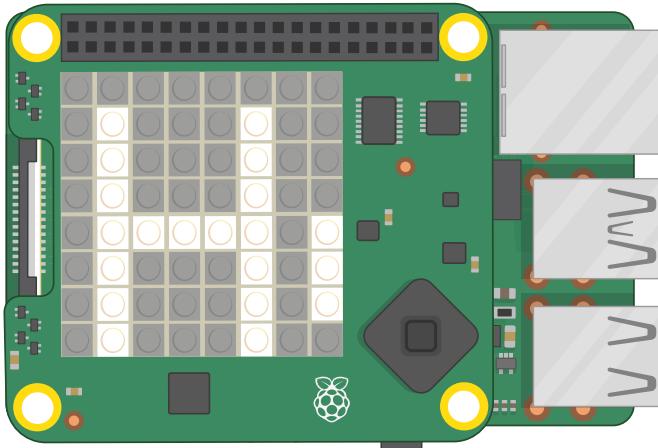
Um den Sense HAT oder den Sense HAT-Emulator in einem Python-Programm zu verwenden, müssen Sie die Sense HAT-Bibliothek importieren. Geben Sie Folgendes in den Skriptbereich ein, und benutzen Sie dabei, `sense_emu` (anstelle von `sense_hat`), wenn Sie den Sense HAT-Emulator verwenden:

```
from sense_hat import SenseHat
sense = SenseHat()
```

Die Sense HAT-Bibliothek verfügt über eine einfache Funktion, um eine Nachricht anzunehmen, sie so zu formatieren, dass sie auf dem LED-Display angezeigt werden kann, und sie reibungslos durchlaufen zu lassen. Geben Sie Folgendes ein:

```
sense.show_message("Hallo Welt!")
```

Speichern Sie Ihr Programm unter **Hallo Sense HAT** und klicken Sie dann auf „Run“. Sie werden sehen, wie Ihre Nachricht langsam über die LED-Matrix des Sense HAT durchläuft, wobei die LED-Pixel aufleuchten und nacheinander die einzelnen Buchstaben bilden (**Abbildung 7-7** umseitig). Herzlichen Glückwunsch: Ihr Programm ist ein Erfolg!



▲ Abbildung 7-7: Durchlaufen einer Nachricht über die LED-Matrix

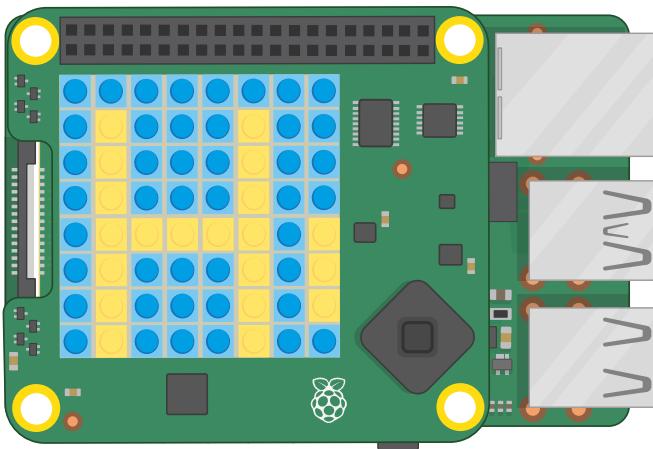
Die `show_message()`-Funktion hat aber noch mehr Tricks in petto. Gehen Sie zurück zu Ihrem Programm und bearbeiten Sie die letzte Zeile so:

```
sense.show_message("Hallo Welt!", text_colour=(255, 255, 0),  
back_colour=(0, 0, 255), scroll_speed=0.05))
```

Diese zusätzlichen, durch Komma getrennten Anweisungen sind bekannt als *Parameter* und kontrollieren verschiedene Aspekte der `show_message()`-Funktion. Der einfachste ist `scroll_speed=()`, wodurch sich die ScrollDurchlauf-Geschwindigkeit der Nachricht über den Bildschirm ändert. Ein hier angegebener Wert von 0,05 scrollt etwa doppelt so schnell wie normalerweise. Je größer die Zahl, desto geringer die Geschwindigkeit.

Die Parameter `text_colour=()` und `back_colour=()` im Gegensatz zu den meisten Python-Anweisungen in britischem Englisch geschrieben – legen die Farbe der Schrift bzw. des Hintergrunds fest. Sie akzeptieren jedoch keine Farbnamen; man muss die gewünschte Farbe in Form von drei Zahlen angeben. Die erste Zahl steht für den Rot-Anteil der Farbe, von 0 für gar kein Rot bis 255 für ein Maximum an Rot; die zweite Zahl steht für den Grün-Anteil der Farbe; und die dritte Zahl für den Blau-Anteil. Zusammen sind sie bekannt als *RGB* – Rot, Grün und Blau.

Klicken Sie auf das Symbol zum Ausführen und betrachten Sie den Sense HAT: Diesmal wird die Nachricht wesentlich schneller durchlaufen und in einem leuchtenden Gelb auf blauem Hintergrund erscheinen (**Abbildung 7-8**). Versuchen Sie, die Parameter zu ändern, um eine Geschwindigkeits- und Farbkombination nach Ihrem Gutdünken zu finden.



▲ Abbildung 7-8: Ändern der Farbe der Botschaft und des Hintergrunds

Wenn Sie anstelle von RGB-Werten einfach zu verstehende Namen verwenden möchten, um Ihre Farben einzustellen, müssen Sie Variablen erstellen. Über Ihrer Zeile `sense.show_message()` fügen Sie Folgendes hinzu:

```
gelb = (255, 255, 0)
blau = (0, 0, 255)
```

Gehen Sie zurück zu Ihrer Zeile `sense.show_message()` und ändern Sie sie wie folgt:

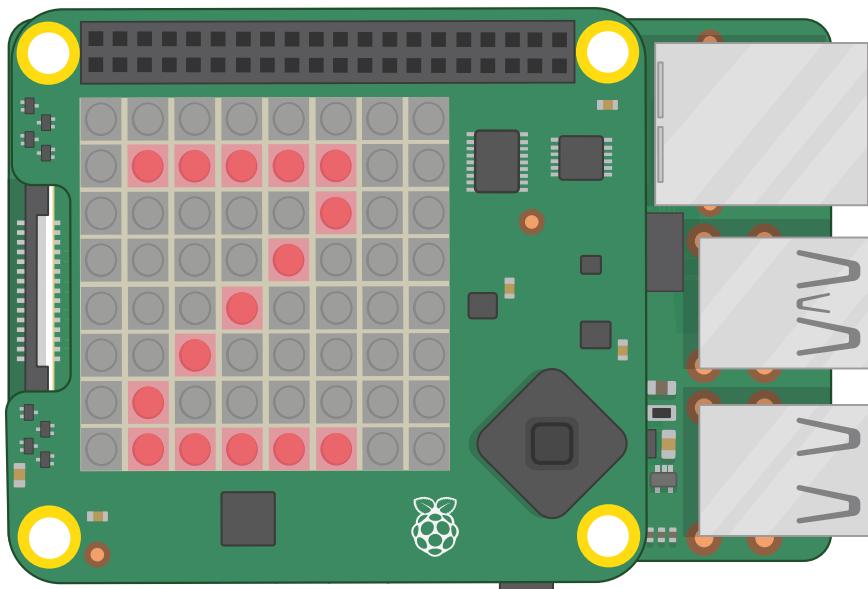
```
sense.show_message("Hallo, Welt!", text_colour=(gelb), back_
colour=(blau), scroll_speed=(0.05))
```

Klicken Sie erneut auf das Symbol „Run“ und Sie werden sehen, dass sich nichts geändert hat: Ihre Nachricht ist immer noch gelb auf blauem Hintergrund. Diesmal haben Sie jedoch die Variablennamen verwendet, um Ihren Code besser lesbar zu machen: Statt einer Zahlenfolge wird der Farbname verwendet. Sie können beliebig viele Farben definieren: Versuchen Sie, eine Variable namens „rot“ mit den Werten 255, 0 und 0, eine Variable namens „weiss“ mit den Werten 255, 255, 255 und eine Variable namens „schwarz“ mit den Werten 0, 0 und 0 hinzuzufügen.

Sie können nicht nur ganze Nachrichten scrollen, sondern auch einzelne Buchstaben anzeigen. Löschen Sie Ihre Zeile `sense.show_message()` vollständig und geben Sie an ihrer Stelle Folgendes ein:

```
sense.show_letter("Z")
```

Klicken Sie auf „Run“. Der Buchstabe Z erscheint auf der Anzeige des Sense HAT. Diesmal bleibt er dort: einzelne Buchstaben, anders als Nachrichten, scrollen nicht von sich aus. Sie können `sense.show_letter()` mit den gleichen Farbparametern wie `sense.show_message()` steuern. Versuchen Sie, die Farbe des Buchstabens in Rot zu ändern (**Abbildung 7-9**).



▲ Abbildung 7-9: Anzeigen eines einzelnen Buchstabens



HERAUSFORDERUNG: DIE NACHRICHT WIEDERHOLEN

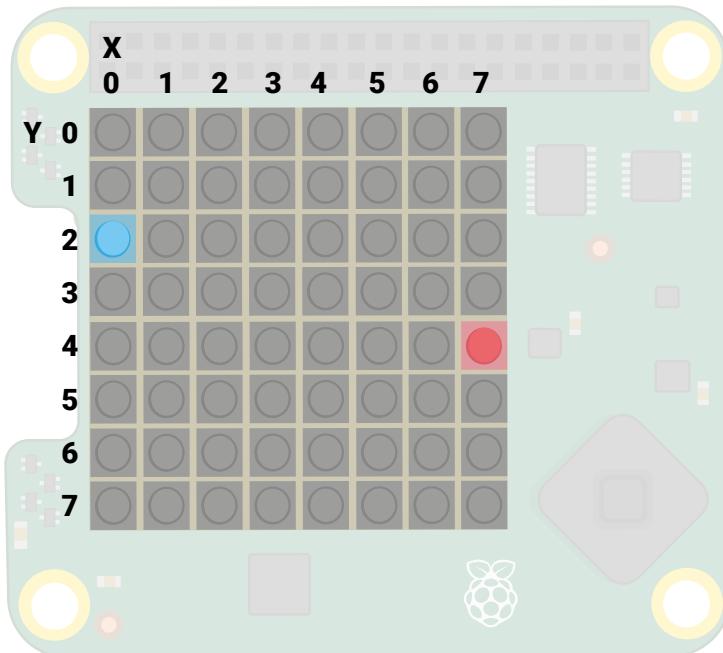


Können Sie Ihr Wissen über Schleifen nutzen, um eine scrollende Nachricht zu wiederholen? Können Sie ein Programm erstellen, das ein Wort Buchstabe für Buchstabe mit verschiedenen Farben buchstabiert? Wie schnell können Sie eine Nachricht scrollen lassen?

Nächste Schritte: Zeichnen mit Licht

Die LED-Anzeige des Sense HAT ist nicht nur für Nachrichten gedacht: Sie können auch Bilder anzeigen. Jede LED kann als ein einzelnes Pixel eines Bildes Ihrer Wahl behandelt werden. Pixel ist die Kurzform von „Picture Element“ (*Bildelement*). Sie können damit Ihre Programme mit Bildern und sogar Animationen aufpeppen.

Um mit Licht zu zeichnen, müssen Sie jedoch in der Lage sein, einzelne LEDs anzusteuernd. Dazu müssen Sie verstehen, wie die LED-Matrix des Sense HAT aufgebaut ist, um ein Programm zu schreiben, das die richtigen LEDs ein- oder ausschaltet



▲ Abbildung 7-10: LED-Matrix-Koordinatensystem

Es gibt acht LEDs in jeder Zeile des Displays und acht in jeder Spalte (**Abbildung 7-10**).

Beim Zählen der LEDs sollten Sie jedoch – wie bei den meisten Programmiersprachen – bei 0 beginnen und bei 7 enden. Die erste LED befindet sich in der oberen linken Ecke, die letzte in der unteren rechten. Anhand der Zahlen aus den Zeilen und Spalten können Sie die *Koordinaten* einer beliebigen LED auf der Matrix finden. Die blaue LED in der abgebildeten Matrix befindet sich an den Koordinaten 0; 2; die rote LED an den Koordinaten 7; 4. Die X-Achse (horizontal) kommt zuerst, gefolgt von der Y-Achse (vertikal).

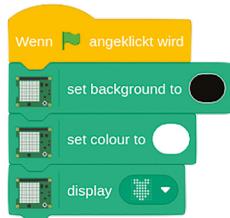
Beim Planen von Bildern, die mit dem Sense HAT gezeichnet werden sollen, kann es hilfreich sein, sie zunächst von Hand auf gerastertem Papier zu zeichnen, oder Sie tun dies mithilfe einer Tabelle wie beispielsweise LibreOffice Calc.

Bilder in Scratch

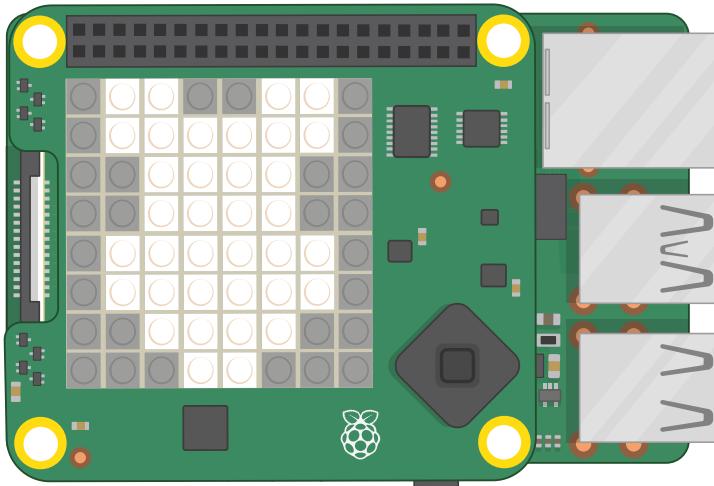
Starten Sie in Scratch ein neues Projekt – speichern Sie vorher Ihr bestehendes Projekt, wenn Sie es behalten wollen. Wenn Sie die Projekte in diesem Kapitel durchgearbeitet haben, bleibt die Raspberry Pi Sense HAT-Erweiterung in Scratch 3 geladen; wenn Sie Scratch 3 seit Ihrem letzten Projekt geschlossen und wieder geöffnet haben, laden Sie die Erweiterung über den Button „Erweiterung hinzufügen“. Beginnen Sie mit dem Ziehen eines „Ereignisse“-Blocks

Wenn [Flagge angeklickt wird] auf den Codebereich und ziehen Sie dann Block **set background** und **set colour** direkt darunter. Bearbeiten Sie beide, um die Hintergrundfarbe (background) auf Schwarz und die (Anzeige-)Farbe (colour) auf Weiß zu setzen. Schwarz bekommen Sie,

indem Sie die Schieberegler Helligkeit und Sättigung auf 0 stellen. Und Weiß bekommen Sie, indem Sie Helligkeit auf 100 und Sättigung auf 0 stellen. Sie müssen dies zu Beginn jedes Sense HAT-Programms tun, sonst verwendet Scratch einfach die von Ihnen zuletzt gewählten Farben – auch wenn Sie diese in einem anderen Programm gewählt haben. Ziehen Sie zum Schluß noch einen Block **display Raspberry** an den unteren Rand Ihres Programms.



Klicken Sie auf die grüne Flagge: Sie werden sehen, wie die LEDs des Sense HAT in Form einer Himbeere (engl. raspberry) aufleuchten (**Abbildung 7-11**).

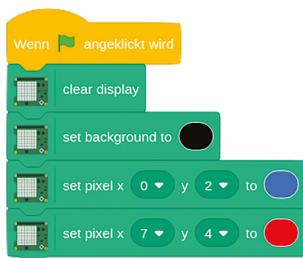


▲ Abbildung 7-11: Schauen Sie nicht direkt auf die LEDs, wenn sie hellweiß sind

Sie sind auch nicht auf die vorgegebene Raspberry-Form beschränkt. Klicken Sie auf den Abwärtspfeil neben dem Raspberry, um den Zeichenmodus zu aktivieren: Sie können auf eine beliebige LED auf dem Muster klicken, um sie ein- oder auszuschalten, während die beiden Buttons unten alle LEDs aus- oder einschalten. Versuchen Sie jetzt, Ihr eigenes Muster zu zeichnen, und klicken Sie dann auf die grüne Flagge, um es auf dem Sense HAT zu sehen. Versuchen Sie auch, die Farbe und die Hintergrundfarbe mit Hilfe der Blöcke darüber zu ändern.

Wenn Sie fertig sind, ziehen Sie die drei Blöcke in die Block-Palette, um sie zu löschen, und platzieren Sie einen Block **clear display** unter **Wenn grüne Flagge angeklickt wird**. Klicken Sie auf die grüne Flagge, und alle LEDs schalten sich aus.

Um ein Bild zu erstellen, müssen Sie in der Lage sein, einzelne Pixel zu steuern und ihnen unterschiedliche Farben zu geben. Sie können dies tun, indem Sie angepasste Blöcke **display Raspberry** mit Blöcken **set colour** verketten, oder Sie behandeln jedes Pixel einzeln. Um Ihre eigene Version des zu Beginn dieses Abschnitts gezeigten LED-Matrix-Beispiels mit zwei gezielt ausgewählten LEDs, die rot und blau leuchten, zu erstellen, lassen Sie den Block **clear display** am oberen Rand Ihres Programms stehen und ziehen Sie einen Block **set background** darunter. Ändern Sie den Block **set background** auf Schwarz und ziehen Sie dann zwei Blöcke **set pixel x 0 y 0** darunter. Bearbeiten Sie diese Blöcke schließlich wie folgt:

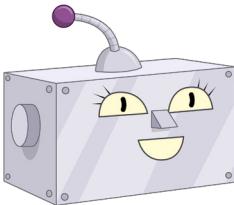


Klicken Sie auf die grüne Flagge, und Sie sehen, dass Ihre LEDs entsprechend dem Matrixbild aufleuchten (**Abbildung 7-10**) auf Seite 165. Herzlichen Glückwunsch: Sie können einzelne LEDs steuern!

Bearbeiten Sie Ihre vorhandenen Pixelblöcke (**set pixel**) wie folgt, und ziehen Sie weitere unter sie, bis Sie das folgende Programm haben:



Bevor Sie auf die grüne Flagge klicken, schauen Sie, ob Sie anhand der von Ihnen verwendeten LED-Matrix-Koordinaten erraten können, welches Bild erscheinen wird, und führen Sie dann Ihr Programm aus, um zu sehen, ob Sie Recht haben!



HERAUSFORDERUNG: NEUE ENTWÜRFE



Können Sie mehr Bilder entwerfen? Versuchen Sie, Millimeter- oder Gitterpapier zu besorgen und Ihr Bild damit zunächst von Hand zu planen. Können Sie ein Bild zeichnen und dann die Farben ändern?

Bilder in Python

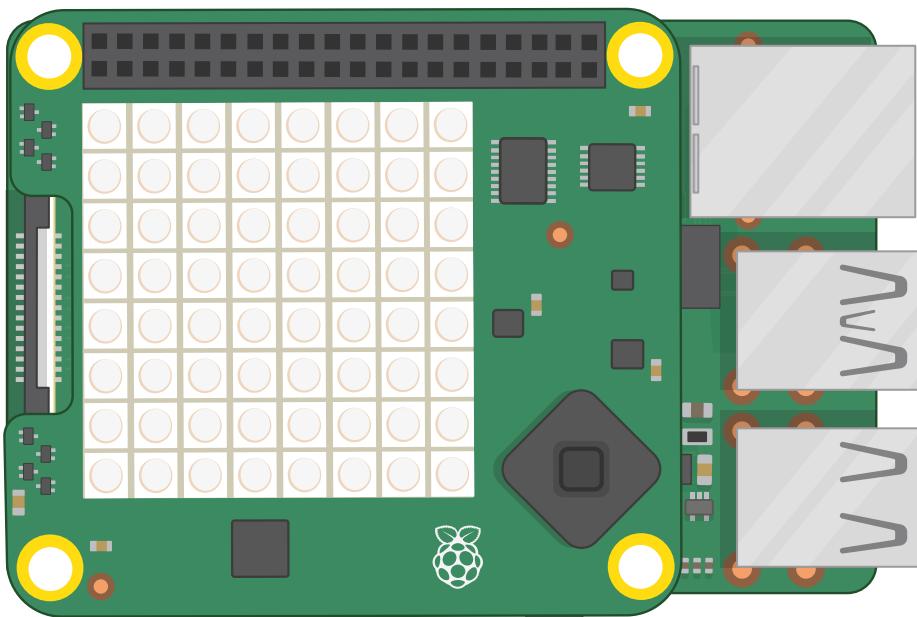
Starten Sie ein neues Programm in Thonny und speichern Sie es als Sense HAT Drawing. Dann geben Sie den folgenden Code ein, und benutzen Sie dabei `sense_emu` (anstelle von `sense_hat`), wenn Sie den Emulator benutzen:

```
from sense_hat import SenseHat  
sense = SenseHat()
```

Denken Sie daran, dass Sie diese beiden Zeilen immer in Ihrem Programm benötigen, um den Sense HAT verwenden zu können. Als Nächstes tippen Sie:

```
sense.clear(255, 255, 255)
```

Schauen Sie nicht direkt auf die LEDs des Sense HAT und klicken Sie auf das Symbol „Run“: Sie sollten jetzt alle leuchtend grell weiß leuchten (**Abbildung 7-12**) – deshalb sollten Sie bei Ausführung des Programms vermeiden, direkt daraufauf sie zu schauen.

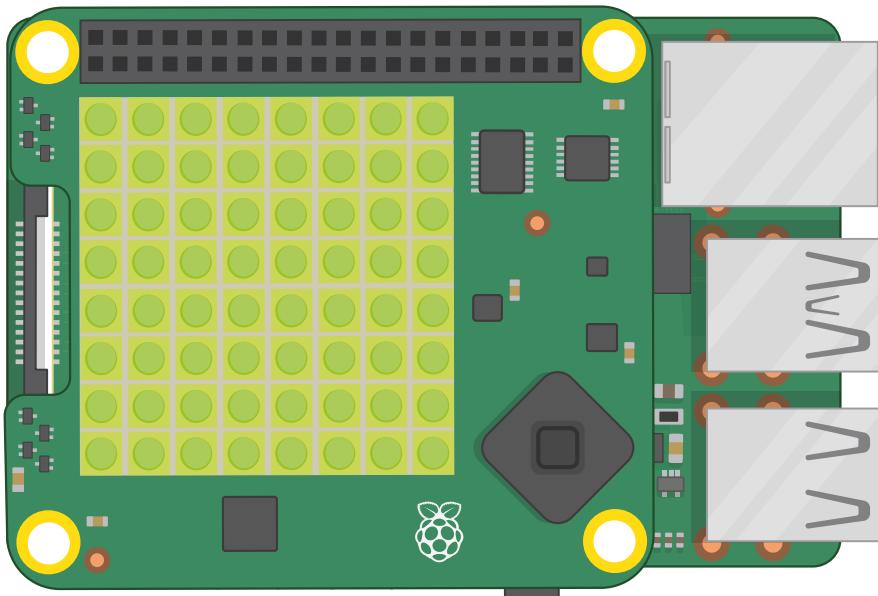


▲ Abbildung 7-12: Schauen Sie nicht direkt auf die Matrix, wenn die LEDs leuchtend grell weiß sind

Die Anweisung `sense.clear()` ist so konzipiert, dass die LEDs aller vorherigen Programmierungen gelöscht werden, akzeptiert aber RGB-Farbparameter. Das heißt, Sie können so die Anzeige auf jede beliebige Farbe umstellen. Versuchen Sie, die Zeile zu bearbeiten:

```
sense.clear(0, 255, 0)
```

Klicken Sie auf „Run“. Der Sense HAT wird hellgrün (**Abbildung 7-13** (umseitig)). Experimentieren Sie mit verschiedenen Farben, oder fügen Sie die Farbnamen-Variablen hinzu, die Sie für Ihr Hallo-Welt-Programm erstellt haben, um die Lesbarkeit zu verbessern.

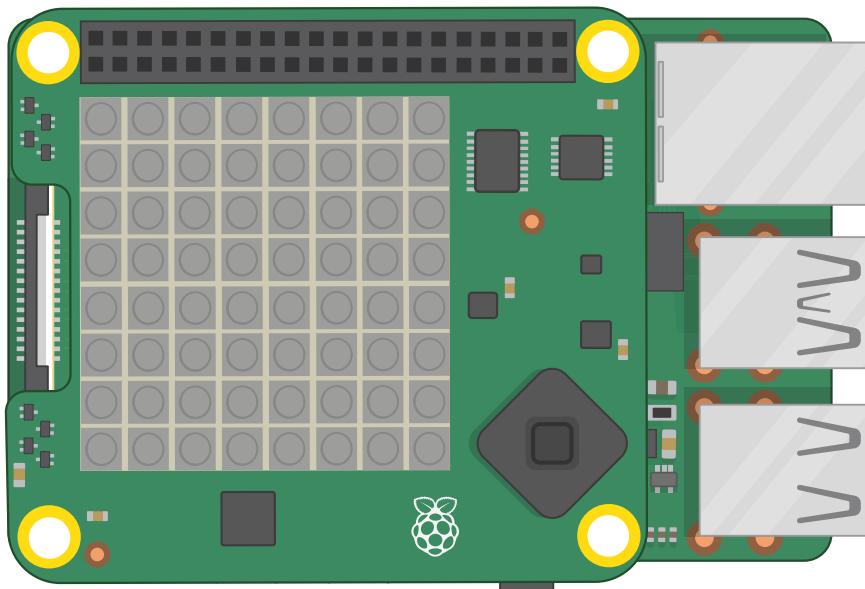


▲ Abbildung 7-13: Die LED-Matrix leuchtet hellgrün

Um die LEDs auszublenden, müssen Sie die RGB-Werte für Schwarz verwenden: 0 rot, 0 blau und 0 grün. Es gibt aber einen noch einfacheren Weg. Bearbeiten Sie diese Zeile Ihres Programms, damit sie so aussieht:

```
sense.clear()
```

Der Sense HAT wird sich verdunkeln, da die Funktion `sense.clear()`, bei der nichts zwischen den Klammern steht, gleichbedeutend ist mit der Anweisung, alle LEDs auf Schwarz zu schalten – d. h. sie auszuschalten (**Abbildung 7-14**). Wenn Ihre Programme die LEDsvollständig ausblenden müssen, ist das dafür die beste Funktion.



▲ Abbildung 7-14: Verwenden Sie die Funktion `sense.clear()` zum Ausschalten aller LEDs

Um Ihre eigene Version der oben in diesem Kapitel abgebildeten LED-Matrix mit zwei gezielt ausgewählten LEDs, die rot und blau leuchten, zu erstellen, fügen Sie Ihrem Programm nach `sense.clear()` folgende Zeilen hinzu:

```
sense.set_pixel(0, 2, (0, 0, 255))
sense.set_pixel(7, 4, (255, 0, 0))
```

Das erste Zahlenpaar ist die Position des Pixels auf der Matrix, X-Achse (waagerecht) gefolgt von der Y-Achse (senkrecht). Das zweite, in einem eigenen Satz von Klammern, sind die RGB-Werte für die Farbe des Pixels. Klicken Sie auf den Button „Run“ und Sie sehen den Effekt: zwei LEDs auf Ihrem Sense HAT leuchten auf, genau wie in **Abbildung 7-10** auf Seite 165.

Löschen Sie diese beiden Zeilen und geben Sie Folgendes ein:

```
sense.set_pixel(2, 2, (0, 0, 255))
sense.set_pixel(4, 2, (0, 0, 255))
sense.set_pixel(3, 4, (100, 0, 0))
sense.set_pixel(1, 5, (255, 0, 0))
sense.set_pixel(2, 6, (255, 0, 0))
sense.set_pixel(3, 6, (255, 0, 0))
sense.set_pixel(4, 6, (255, 0, 0))
sense.set_pixel(5, 5, (255, 0, 0))
```

Bevor Sie auf „Run“ klicken, sehen Sie sich die Koordinaten an und vergleichen Sie sie mit der Matrix: Können Sie erraten, welches Bild diese Anweisungen zeichnen werden? Klicken Sie auf „Run“, um herauszufinden, ob Sie Recht haben!

Ein detailliertes Bild mit einzelnen `set_pixel()`-Funktionen zu zeichnen ist allerdings langsam. Um die Dinge zu beschleunigen, können Sie mehrere Pixel gleichzeitig ändern. Löschen Sie alle Ihre `set_pixel()`-Zeilen und geben Sie Folgendes ein:

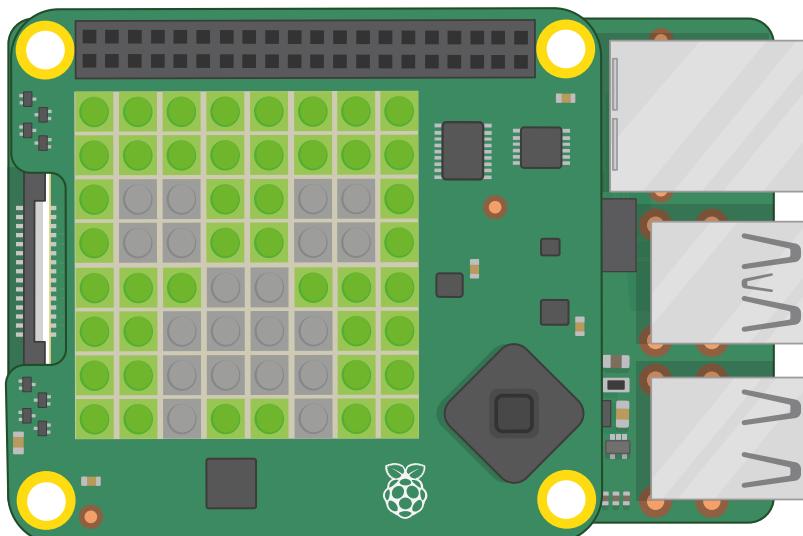
```
g = (0, 255, 0)
b = (0, 0, 0)

creeper_pixels = [
    g, g, g, g, g, g, g, g,
    g, g, g, g, g, g, g, g,
    g, b, b, g, g, b, b, g,
    g, b, b, g, g, b, b, g,
    g, g, g, b, b, g, g, g,
    g, g, b, b, b, g, g, g,
    g, g, b, b, b, b, g, g,
    g, g, b, g, b, g, g
]
sense.set_pixels(creeper_pixels)
```

Das war jetzt ganz schön viel, aber klicken Sie jetzt auf „Run“, um zu sehen, ob Sie einen kleinen KriecherCreeper erkennen können. Die ersten beiden Zeilen erzeugen zwei Variablen, die für die Farben stehen: Grün und Schwarz. Um den Code für die Zeichnung leichter schreib- und lesbar zu machen, sind die Variablen einzelne Buchstaben: g für grün und b für schwarz (black).

Der nächste Code-Block erzeugt eine Variable, die Farbwerte für alle 64 Pixel auf der LED-Matrix enthält, durch Kommas getrennt und zwischen eckigen Klammern eingeschlossen. Anstelle von Zahlen verwendet er jedoch die Farbvariablen, die Sie zuvor erstellt haben: Schauen Sie genau hin, denken Sie daran, dass g für Grün und b für Schwarz steht, und Sie können bereits das Bild sehen, das erscheinen wird (**Abbildung 7-15**).

Schließlich nimmt `sense.set_pixels(creeper_pixels)` diese Variable und verwendet die `sense.set_pixels()`-Funktion, um die gesamte Matrix auf einmal zu zeichnen. Viel einfacher als Pixel für Pixel zu zeichnen!



► Abbildung 7-15: Anzeigen eines Bildes auf der Matrix

Sie können Bilder auch drehen und spiegeln, entweder um sie in der richtigen Position zu zeigen, wenn Ihr Sense HAT umgedreht wird, oder um einfache Animationen aus einem einzelnen asymmetrischen Bild zu erstellen.

Beginnen Sie mit der Bearbeitung Ihrer Variablen `creeper_pixel`, um sein linkes Auge zu schließen, indem die dazugehörigen vier 'b'-Pixel durch 'g' ersetzt werden, beginnend mit den ersten beiden auf der dritten Zeile und dann die ersten beiden auf der vierten Zeile:

```
creeper_pixels = [
    g, g, g, g, g, g, g, g,
    g, g, g, g, g, g, g, g,
    g, g, g, g, g, b, b, g,
    g, g, g, g, g, b, b, g,
    g, g, g, b, b, g, g, g,
    g, g, b, b, b, b, g, g,
    g, g, b, b, b, b, g, g,
    g, g, b, g, g, b, g, g]
```

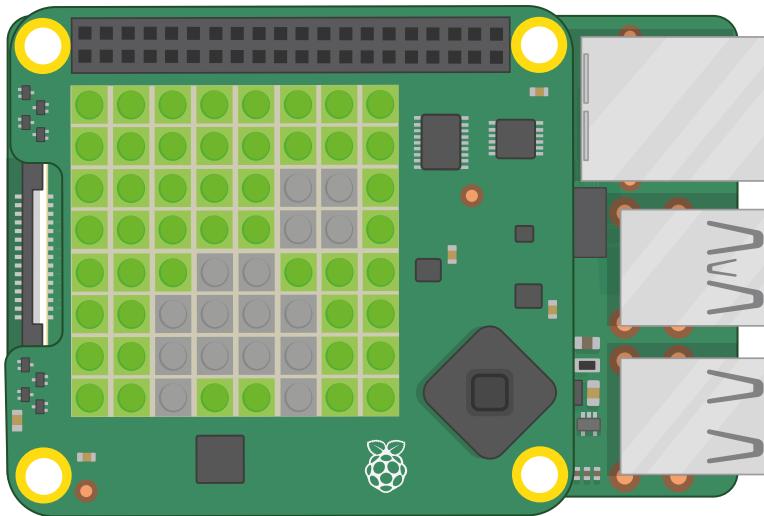
Klicken Sie auf „Run“, und Sie sehen, dass das linke Auge deses KriechersCreepers jetzt geschlossen ist (**Abbildung 7-16** (umseitig)). Um eine Animation zu erstellen, gehen Sie zum Anfang Ihres Programms und fügen Sie folgende Zeile hinzu:

```
from time import sleep
```

Gehen Sie dann nach unten und geben Sie Folgendes ein:

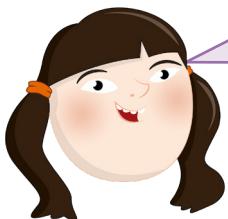
```
while True:  
    sleep(1)  
    sense.flip_h()
```

Klicken Sie auf „Run“ und beobachten Sie den KriecherCreeper, während er die Augen schließt und öffnet, eines nach dem anderen!



▲ Abbildung 7-16: Anzeigen einer einfachen Zwei-Bild-Animation

Die Funktion `sense.flip_h()` spiegelt ein Bild auf der horizontalen Achse. Wenn Sie ein Bild auf seiner vertikalen Achse spiegeln wollen, ersetzen Sie `sense.flip_h()` durch `sense.flip_v()`. Sie können ein Bild auch um 0, 90, 180 oder 270 Grad drehen, indem Sie `sense.set_rotation(90)` verwenden und die der Funktion übergebene Zahl ändern, je nachdem, um wie viele Grad Sie das Bild drehen möchten. Versuchen Sie, dies zu nutzen, damit sich der KriecherCreeper dreht anstatt zu blinzeln!



HERAUSFORDERUNG: NEUE ENTWÜRFE



Können Sie mehr Bilder und Animationen entwerfen? Besorgen Sie sich Millimeter- oder Gitterpapier und planen Sie Ihr Bild damit zunächst von Hand, um das Schreiben der Variable zu erleichtern. Können Sie ein Bild zeichnen und dann die Farben ändern? Tipp: Sie können Variablen ändern, nachdem Sie sie bereits einmal verwendet haben.

Die Welt um Sie herum wahrnehmen

Die wahre Stärke des Sense HAT liegt in den verschiedenen Sensoren, über die er verfügt. Diese ermöglichen es Ihnen, alle möglichen Messwerte, angefangen von der Temperatur bis hin zur Beschleunigung zu erfassen und in Ihren Programmen beliebig zu verwenden.



EMULATION DER SENSOREN



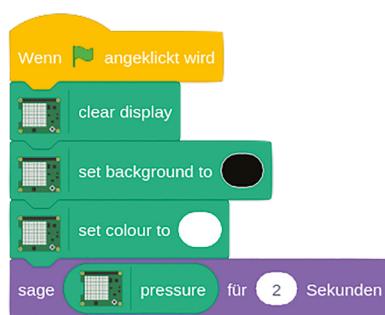
Wenn Sie den Sense HAT-Emulator verwenden, müssen Sie die Trägheits- und Umgebungssensorsimulation aktivieren: Klicken Sie im Emulator auf „Edit“, dann auf „Preferences“ und aktivieren Sie sie. Wählen Sie im gleichen Menü unter „Orientierungsskala“ die Option „180°..360°|0°..180°“, um sicherzustellen, dass die Zahlen im Emulator mit den von Scratch und Python gemeldeten Zahlen übereinstimmen, und klicken Sie dann auf den Button „Close“.

Umweltsensorik

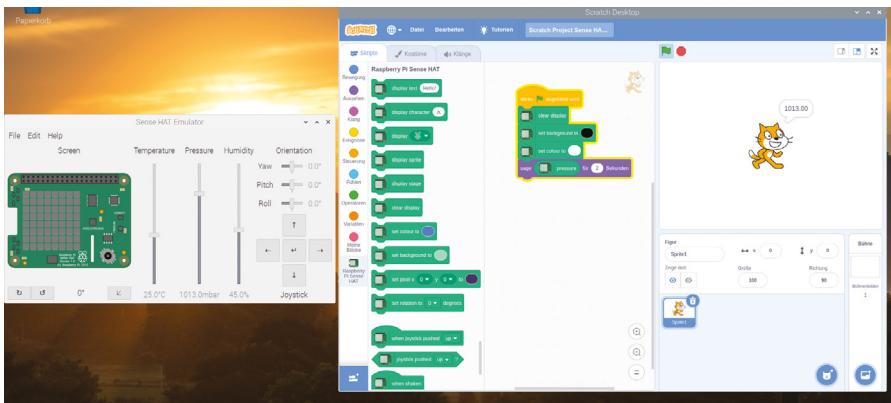
Der barometrische Drucksensor, der Luftfeuchtigkeitssensor und der Temperatursensor sind Umweltsensoren. Sie nehmen Messungen aus der Umgebung des Sense HAT vor.

Umweltsensorik in Scratch

Starten Sie in Scratch ein neues Programm, speichern Sie vorher Ihr altes, falls Sie dies wünschen, und fügen Sie die Raspberry Pi Sense HAT-Erweiterung hinzu, falls sie noch nicht geladen ist. Ziehen Sie einen „Ereignisse“-Block **Wenn grüne Flagge angeklickt wird** auf den Codebereich, dann einen Block **clear display** und dann einen Block **set background to schwarz** darunter. Fügen Sie einen Block **set colour to weiß** hinzu. Verwenden Sie die Schieberegler Helligkeit und Sättigung, um die richtige Farbe auszuwählen. Es ist immer eine gute Idee, dies zu Beginn Ihrer Programme zu tun, da es sicherstellt, dass der Sense HAT nichts aus einem alten Programm anzeigt und die richtigen Farben einsetzt. Ziehen Sie einen „Aussehen“-Block **sage Hallo! für 2 Sekunden** direkt unter Ihre vorhandenen Blöcke. Um einen Messwert vom Drucksensor auszulesen, suchen Sie den Block **pressure** in der Raspberry Pi Sense HAT-Kategorie und ziehen Sie ihn auf das Wort „Hallo!“ in Ihrem Block **sage Hallo! für 2 Sekunden**.



Klicken Sie auf die grüne Flagge und die Scratch-Katze zeigt Ihnen den aktuellen Messwert des Drucksensors in *Millibar*. Nach zwei Sekunden verschwindet die Meldung; versuchen Sie, auf den Sense HAT zu blasen (oder den Druckschieberegler im Emulator nach oben zu bewegen) und klicken Sie auf die grüne Flagge, um das Programm erneut zu starten. Diesmal sollten Sie einen höheren Messwert sehen (**Abbildung 7-17** umseitig).



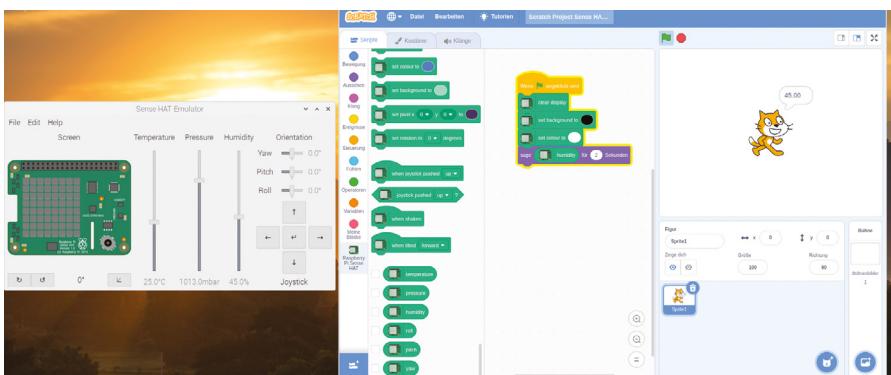
▲ Abbildung 7-17: Anzeigen des Drucksensorwerts



WERTE ÄNDERN

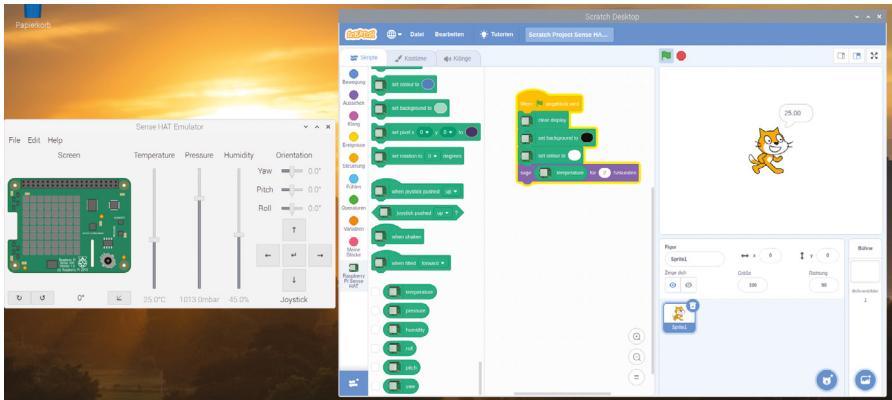
Wenn Sie den Sense HAT-Emulator verwenden, können Sie die von jedem der emulierten Sensoren gemeldeten Werte mithilfe seiner Schieberegler und Buttons ändern. Versuchen Sie, die Drucksensoreinstellung nach unten zu schieben und klicken Sie dann erneut auf die grüne Flagge.

Um auf den Luftfeuchtigkeitssensor umzuschalten, löschen Sie den Block **pressure** und ersetzen Sie ihn durch **humidity**. Starten Sie Ihr Programm erneut. Die aktuelle relative Luftfeuchtigkeit Ihres Raums wird angezeigt. Auch hier können Sie versuchen, das Programm erneut auszuführen, während Sie auf den Sense HAT blasen (oder den Luftfeuchtigkeitsregler des Emulators nach oben schieben), um den Messwert zu ändern (Abbildung 7-18) – Ihr Atem ist überraschend feucht!



▲ Abbildung 7-18: Anzeigen des Messwerts vom Luftfeuchtigkeitssensor

Der Um den Wert des Temperatursensors zu bekommen, löschten Sie einfach den **humidity** Block, ersetzten Sie ihn durch **temperature** und führten Sie dann Ihr Programm erneut aus. Sie sehen eine Temperatur in Grad Celsius (**Abbildung 7-19**). Es kann jedoch sein, dass dies nicht die tatsächliche Temperatur Ihres Raums ist: Der Raspberry Pi erzeugt während des Betriebs Wärme, was den Sense HAT und seine Sensoren ebenfalls erwärmt.



▲ Abbildung 7-19: Anzeigen des Messwerts des Temperatursensors



HERAUSFORDERUNG: BILDLAUF UND SCHLEIFEN



Können Sie Ihr Programm so ändern, dass es von jedem der Sensoren der Reihe nach eine Messung abliest und die Messwerte dann über die LED-Matrix scrollen lässt, anstatt sie im Bühnenbereich auszugeben? Können Sie Ihre Programmschleife so gestalten, dass sie ständig die aktuellen Umgebungsbedingungen ausgibt?

Umweltsensorik in Python

Um mit der Messung von Sensoren zu beginnen, erstellen Sie ein neues Programm in Thonny und speichern Sie es unter **Sense HAT Sensors**. Geben Sie Folgendes in den Skriptbereich ein, wie immer, wenn Sie den Sense HAT verwenden möchten. Benutzen Sie dabei, **sense_emu**, wenn Sie den Emulator verwenden:

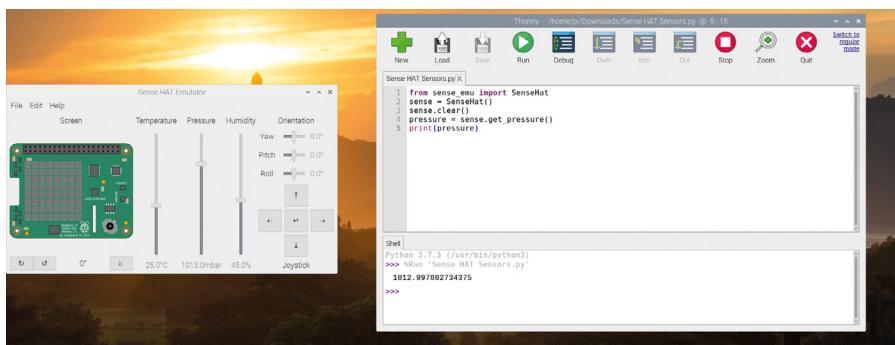
```
from sense_hat import SenseHat
sense = SenseHat()
sense.clear()
```

Es ist immer eine gute Idee **sense.clear()** am Anfang Ihrer Programme einzufügen, nur für den Fall, dass auf dem Display der Sense HAT noch etwas vom zuletzt ausgeführten Programm angezeigt wird.

Um einen Messwert vom Drucksensor auszulesen, geben Sie Folgendes ein:

```
pressure = sense.get_pressure()  
print(pressure)
```

Klicken Sie auf „Run“. Sie sehen wie eine Zahl in der Python-Shell am unteren Rand des Thonny-Fensters ausgegeben wird. Dies ist der vom Sensor für den barometrischen Druck ermittelte Luftdruckwert in *Millibar* (**Abbildung 7-20**). Versuchen Sie, auf den Sense HAT zu blasen (oder den Druck-Schieberegler im Emulator nach oben zu bewegen), während Sie erneut auf das Ausführen-Symbol klicken: Diesmal sollte die Zahl höher sein.



▲ Abbildung 7-20: Ausgabe eines Luftdruckwerts vom Sense HAT



WERTE ÄNDERN

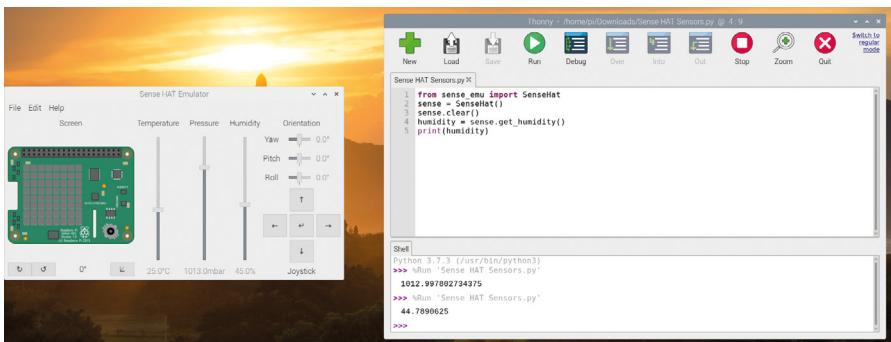


Wenn Sie den Sense HAT-Emulator verwenden, können Sie die von jedem der emulierten Sensoren gemeldeten Werte mithilfe seiner Schieberegler und Buttons ändern. Versuchen Sie, die Drucksensoreinstellung nach unten zu schieben, und klicken Sie dann erneut auf „Run“.

Um auf den Luftfeuchtigkeitssensor umzuschalten, entfernen Sie die letzten beiden Zeilen des Codes und ersetzen Sie sie durch:

```
humidity = sense.get_humidity()  
print(humidity)
```

Klicken Sie auf „Run“ und Sie sehen die Ausgabe einer weiteren Zahl in der Python-Shell. Dieses Mal ist es die aktuelle relative Luftfeuchtigkeit Ihres Raums in Prozent. Auch hier können Sie auf den Sense HAT blasen (oder den Luftfeuchtigkeitsschieberegler des Emulators nach oben schieben) und beobachten, wie der Wert steigt, wenn Sie Ihr Programm erneut ausführen (**Abbildung 7-21**) – Ihr Atem ist überraschend feucht!

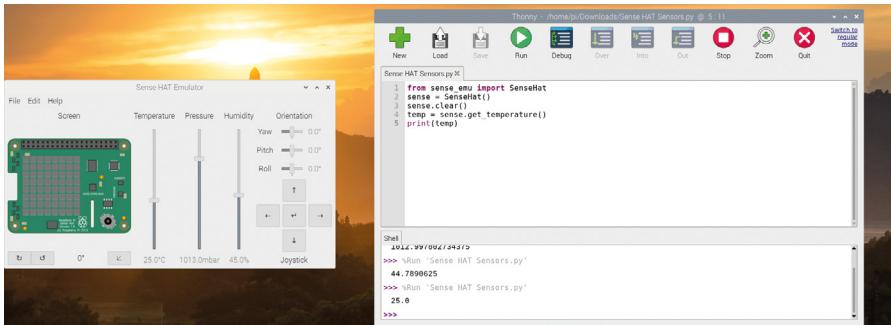


▲ Abbildung 7-21: Anzeigen des Messwerts des Luftfeuchtigkeitssensors

Für den Temperatursensor entfernen Sie die letzten beiden Zeilen Ihres Programms und ersetzen Sie sie durch folgende:

```
temp = sense.get_temperature()
print(temp)
```

Klicken Sie noch einmal auf „Run“ und Sie sehen eine Temperatur in Grad Celsius (Abbildung 7-22). Es kann jedoch sein, dass dies nicht die tatsächliche Temperatur Ihres Raums ist: Der Raspberry Pi erzeugt während des Betriebs Wärme, was den Sense HAT und seine Sensoren ebenfalls erwärmt.



▲ Abbildung 7-22: Anzeigen der aktuellen Temperaturmessung

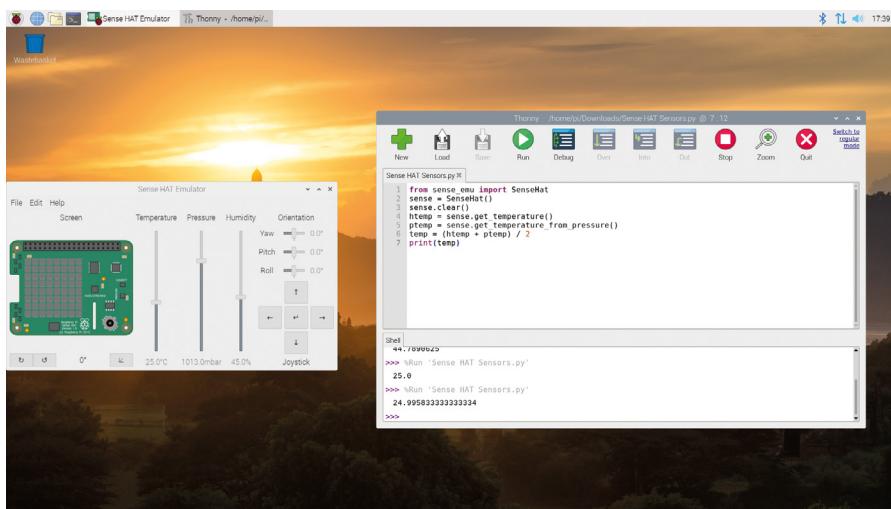
Normalerweise meldet der Sense HAT die Temperatur basierend auf einem Messwert des im Luftfeuchtigkeitssensor eingebauten Temperatursensors. Wenn Sie stattdessen den Messwert des Drucksensors verwenden möchten, sollten Sie `sense.get_temperature_from_pressure()` verwenden. Es ist auch möglich, die beiden Messwerte zu kombinieren, um einen Mittelwert zu erhalten, der unter Umständen genauer ist als die Verwendung eines der beiden Sensoren allein. Löschen Sie die letzten beiden Zeilen Ihres Programms und geben Sie Folgendes ein:

```

htemp = sense.get_temperature()
ptemp = sense.get_temperature_from_pressure()
temp = (htemp + ptemp) / 2
print(temp)

```

Klicken Sie auf „Run“. Sie sehen eine Zahl, die auf der Python-Konsole ausgegeben wird (**Abbildung 7-23**). Diesmal basiert sie auf den Messwerten beider Sensoren, die Sie addiert und durch zwei (die Anzahl der Messwerte) geteilt haben, um einen Mittelwert aus beiden zu erhalten. Wenn Sie den Emulator verwenden, zeigen alle drei Methoden – Feuchtigkeit, Druck und Durchschnitt – die gleiche Zahl an.



▲ Abbildung 7-23: Eine auf den Messwerten beider Sensoren basierende Temperatur



HERAUSFORDERUNG: BILDLAUF UND SCHLEIFEN

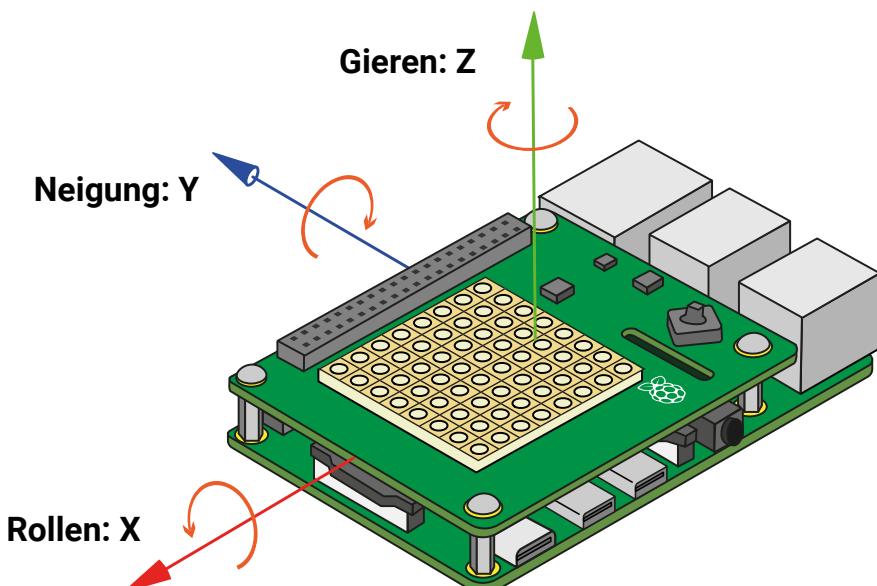


Können Sie Ihr Programm so ändern, dass Sie von jedem der Sensoren der Reihe nach eine Messung vornehmen und diese dann über die LED-Matrix scrollen, anstatt sie in der Shell auszugeben? Können Sie Ihre Programmschleife so gestalten, dass sie ständig die aktuellen Umgebungsbedingungen ausgibt?

Trägheitssensorik

Der Gyroskopsensor, der Beschleunigungsmesser und das Magnetometer bilden zusammen eine sogenannte *inertiale Messeinheit* (*Inertial Measurement Unit, IMU*). Während diese Sensoren technisch gesehen genau wie die Umgebungssensoren Messungen aus der Umgebung vornehmen (das Magnetometer beispielsweise misst die Magnetfeldstärke), werden sie normalerweise für Daten über die Bewegung des Sense HAT selbst verwendet. Die IMU ist die Summe mehrerer Sensoren. Manche Programmiersprachen ermöglichen es Ihnen, die Messwerte von jedem Sensor unabhängig voneinander zu erfassen, während andere Ihnen nur einen kombinierten Messwert liefern.

Bevor Sie sich jedoch einen Überblick über die IMU verschaffen können, müssen Sie verstehen, wie Bewegungen erfolgen. Der Sense HAT und Ihr Raspberry Pi, an dem er befestigt ist, können sich entlang dreier räumlicher Achsen bewegen: seitwärts auf der X-Achse, vorwärts und rückwärts auf der Y-Achse und aufwärts und abwärts auf der Z-Achse (**Abbildung 7-24**). Sie kann sich auch entlang der gleichen drei Achsen drehen, wobei die Bewegungen unterschiedliche Namen haben: Die Drehung um die X-Achse heißt *roll* (Rollen), die Drehung um die Y-Achse heißt *pitch* (Nicken) und die Drehung um die Z-Achse heißt *yaw* (Gieren). Wenn Sie den Sense HAT entlang seiner kurzen Achse drehen, ändern Sie die „Nicken“ (die Neigung). Die Drehung entlang seiner langen Achse ist das „Rollen“. Ein Drehen im Kreis, während er flach auf dem Tisch liegt, wird als „Gieren“ bezeichnet. Stellen Sie sich diese Bewegungen wie bei einem Flugzeug vor: Beim Abheben vergrößert das Flugzeug seine Neigung, um zu steigen. Wenn es eine Victory-Rolle vollführt, dreht es sich buchstäblich um seine Rollachse. Wenn es sich mit Hilfe seines Seitenruders wie ein Auto dreht, ohne zu rollen, ist das „Gieren“.



▲ Abbildung 7-24: Die Raumachsen der IMU der Sense HAT

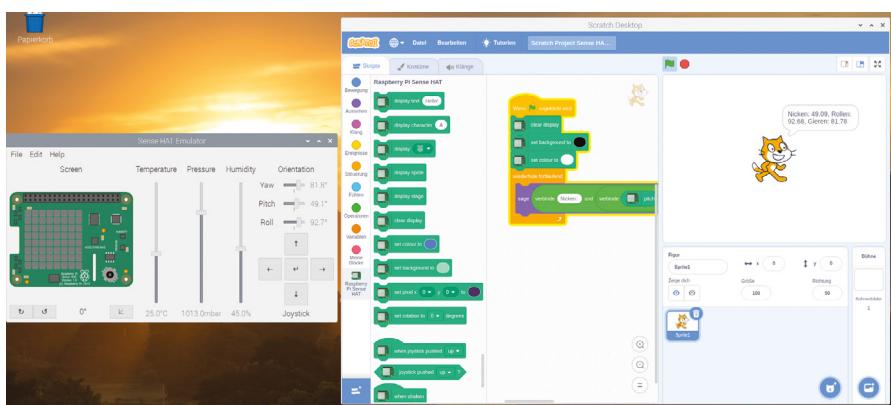
Trägheitssensorik in Scratch

Starten Sie ein neues Programm in Scratch und laden Sie die Raspberry Pi Sense HAT-Erweiterung, falls sie nicht bereits geladen ist. Starten Sie Ihr Programm auf die gleiche Weise wie zuvor: Ziehen Sie einen „Ereignisse“-Block **Wenn grüne Flagge angeklickt wird** auf Ihren Code-Bereich, dann ziehen Sie einen Block **clear display** darunter, gefolgt vom Ziehen und Bearbeiten eines Blocks **set background to schwarz** und **set colour to weiß**.

Als Nächstes ziehen Sie einen Block **wiederhole fortlaufend** an den unteren Rand Ihrer bestehenden Blöcke und füllen ihn mit einem Block **sage Hallo!**. Um einen Messwert für jede der drei Achsen der IMU – Nicken, Rollen und Gieren – anzuzeigen, müssen Sie Operator-Blöcke **verbinde** plus die entsprechenden Sense HAT-Blöcke hinzufügen. Denken Sie daran, Leerzeichen und Kommas einzufügen, damit die Ausgabe gut lesbar ist.



Klicken Sie auf die grüne Flagge, um Ihr Programm zu starten, und versuchen Sie, den Sense HAT und Raspberry Pi zu bewegen – achten Sie darauf, keine Kabel vom Raspberry Pi zu lösen! Wenn Sie den Sense HAT entlang seiner drei Achsen kippen, werden Sie sehen, wie sich die Werte für Nicken (Neigung), Rollen und Gieren entsprechend ändern (**Abbildung 7-25**).



▲ Abbildung 7-25: Anzeige der Nick-, Roll- und Gierwerte

Trägheitssensorik in Python

Starten Sie ein neues Programm in Thonny und speichern Sie es als **Sense HAT Movement**.

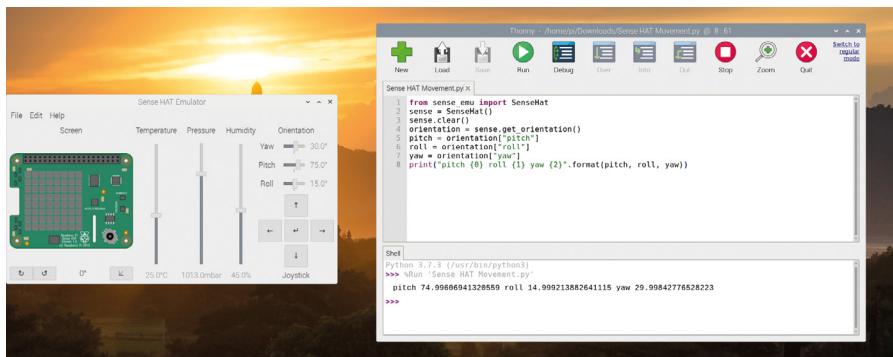
Leiten Sie Ihr Programm mit den üblichen Anfangszeilen ein und denken Sie daran, **sense_emu** zu verwenden, wenn Sie den Sense HAT-Emulator verwenden:

```
from sense_hat import SenseHat
sense = SenseHat()
sense.clear()
```

Geben Sie Folgendes ein, um anhand der Informationen von der IMU die aktuelle Ausrichtung des Sense HAT auf seinen drei Achsen zu ermitteln:

```
orientation = sense.get_orientation()
pitch = orientation["pitch"]
roll = orientation["roll"]
yaw = orientation["yaw"]
print("Nicken {0} Rollen {1} Gieren {2}".format(pitch, roll, yaw))
```

Klicken Sie auf „Run“ und Sie sehen die Messwerte für die Ausrichtung des Sense HAT, aufgeteilt auf die drei Achsen (**Abbildung 7-26**). Versuchen Sie, den Sense HAT zu drehen und erneut auf „Run“ zu klicken. Sie sollten sehen, dass die Zahlen sich ändern, um die neue Ausrichtung widerzuspiegeln.



▲ Abbildung 7-26: Anzeige der Nick-, Roll- und Gierwerte des Sense HAT

Die IMU kann jedoch mehr als nur die Orientierung messen: Sie kann auch Bewegungen erkennen. Um genaue Messwerte für Bewegungen zu erhalten, muss die IMU wiederholt und in kurzen Zeitabständen mit Hilfe einer Schleife ausgelesen werden: Anders als bei einer Orientierung erhält man mit einer einzigen Auslesung keine nützlichen Informationen, wenn es darum geht, Bewegungen zu erkennen. Löschen Sie alles nach **sense.clear()**, und geben Sie dann den folgenden Code ein:

```
while True:  
    acceleration = sense.get_accelerometer_raw()  
    x = acceleration["x"]  
    y = acceleration["y"]  
    z = acceleration["z"]
```

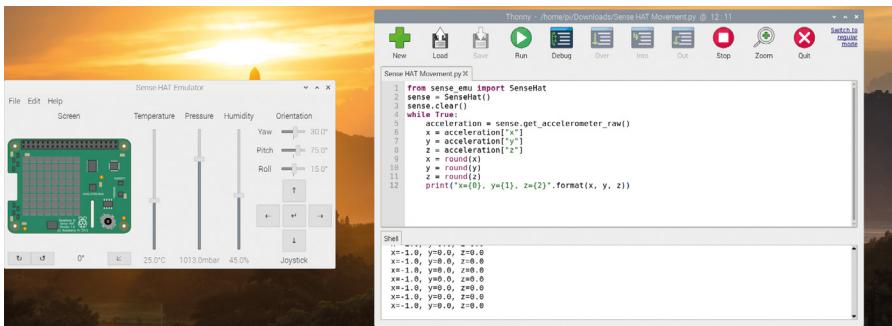
Sie verfügen nun über Variablen, die die aktuellen Beschleunigungsmesswerte für die drei Raumachsen enthalten: X, oder links und rechts, Y, oder vorwärts und rückwärts, und Z, oder oben und unten. Die Zahlen des Beschleunigungssensors sind sehr lange und daher schwer zu lesen. Geben Sie also Folgendes ein, um sie durch Runden auf die nächste ganze Zahl leichter verständlich zu machen:

```
x = round(x)  
y = round(y)  
z = round(z)
```

Zeigen Sie schließlich die drei Werte an, indem Sie die folgende Zeile eingeben:

```
print("x={0}, y={1}, z={2}".format(x, y, z))
```

Klicken Sie auf „Run“. Sie sehen Werte vom Beschleunigungsmesser, die in dem Bereich der Python-Shell ausgegeben werden (**Abbildung 7-27**). Im Gegensatz zu Ihrem vorherigen Programm werden diese kontinuierlich ausgegeben. Zum Stoppen der Ausgabe klicken Sie auf das rote Stop-Symbol, um das Programm zu beenden.



► Abbildung 7-27: Ablesungen für den Beschleunigungsmesser auf die nächste ganze Zahl gerundet

Vielleicht haben Sie bemerkt, dass der Beschleunigungsmesser Ihnen meldet, dass eine der Achsen – die Z-Achse, wenn Ihr Raspberry Pi flach auf dem Tisch liegt – einen Beschleunigungswert von 1,0 Erdbeschleunigungen (1G) hat, obwohl der Sense HAT sich nicht bewegt. Das liegt daran, dass sie die Anziehungschaft der Erde erkannt hat – die Kraft.

die den Sense HAT zum Erdmittelpunkt hinunterzieht. Diese Kraft ist auch der Grund dafür dass etwas zu Boden fällt, wenn Sie es versehentlich vom Schreibtisch stoßen.

Versuchen Sie bei laufendem Programm, den Sense HAT und den Raspberry Pi vorsichtig aufzuheben und hin und her zu drehen – aber achten Sie darauf, keines der angeschlossenen Kabel zu lösen! Wenn die Netzwerk- und USB-Anschlüsse des Raspberry Pi auf den Boden zeigen, werden Sie sehen, wie sich die Werte ändern, sodass auf der Z-Achse 0G und auf der X-Achse jetzt 1G angezeigt wird. Drehen Sie sie erneut, sodass die HDMI- und Stromanschlüsse auf den Boden zeigen, und jetzt ist es die Y-Achse, die 1G anzeigt. Wenn Sie im Gegenteil den HDMI-Anschluss zur Decke zeigen lassen, sehen Sie stattdessen -1G auf der Y-Achse.

Wenn man weiß, dass die Schwerkraft der Erde etwa 1G beträgt, und mit den räumlichen Achsen vertraut ist, kann man mit den Messwerten des Beschleunigungsmessers herausfinden, was unten und was oben ist. Auch die Bewegungserkennung ist damit möglich. Versuchen Sie, den Sense HAT und den Raspberry Pi vorsichtig zu schütteln. Beobachten Sie die Zahlen: Je stärker Sie schütteln, desto größer ist die Beschleunigung.

Wenn Sie `sense.get_accelerometer_raw()` verwenden, weisen Sie den Sense HAT an, die beiden anderen Sensoren in der IMU – den gyroskopischen Sensor und das Magnetometer – auszuschalten und nur Daten vom Beschleunigungsmesser zurückzugeben. Natürlich können Sie dasselbe auch mit den anderen Sensoren tun.

Suchen Sie die Zeile `acceleration = sense.get_accelerometer_raw()` und ändern Sie sie in Folgendes:

```
orientation = sense.get_gyroscope_raw()
```

Ändern Sie das Wort **acceleration** (Beschleunigung) in allen drei Zeilen darunter auf **orientation** (Ausrichtung). Klicken Sie auf „Run“ und Sie sehen die Ausrichtung des Sense HAT für alle drei Achsen, gerundet auf die nächste ganze Zahl. Anders als beim letzten Mal, als Sie die Orientierung ausgelesen haben, kommen die Daten diesmal jedoch ausschließlich vom Gyroskop, ohne Verwendung des Beschleunigungs- oder Magnetometers. Dies kann nützlich sein, wenn Sie z. B. die Ausrichtung eines sich auf dem Rücken eines Roboters bewegenden Sense HAT wissen wollen, ohne dass die Bewegung für Verwirrung sorgt. Die Verwirrung zu erzeugen, oder wenn Sie den Sense HAT in der Nähe eines starken Magnetfeldes verwenden.

Beenden Sie Ihr Programm, indem Sie auf die rote Stopptaste klicken. Um das Magnetometer zu verwenden, löschen Sie alles aus Ihrem Programm mit Ausnahme der ersten vier Zeilen und geben Sie dann Folgendes unter der Zeile **while True** ein:

```
north = sense.get_compass()
print(north)
```

Wenn Sie Ihr Programm ausführen, sehen Sie wie die Richtung des magnetischen Nordens mehrmals wiederholt im Bereich der Python-Shell angezeigt wird. Drehen Sie den Sense HAT

vorsichtig. Sie sehen, wie die angezeigte Richtung variiert, wenn sich die Orientierung des Sense HAT relativ zum Norden ändert: Sie haben einen Kompass gebaut! Wenn Sie einen Magneten haben – ein Kühlschrankmagnet reicht aus – versuchen Sie, ihn um den Sense HAT herum zu bewegen, um zu sehen, wie sich das auf die Magnetometeranzeige auswirkt.



HERAUSFORDERUNG: AUTOMATISCHE DREHUNG



Können Sie mit dem, was Sie über die LED-Matrix und die Sensoren der Trägheitsmesseinheit gelernt haben, ein Programm schreiben, das ein Bild in Abhängigkeit von der Position des Sense HAT dreht?

Joystick-Steuerung

Der Joystick der Sense HAT, unten rechts in der Ecke, ist zwar klein, aber überraschend vielseitig: Neben der Fähigkeit, Eingaben in vier Richtungen – oben, unten, links und rechts – zu erkennen, verfügt er auch über einen fünften Eingang, den man benutzt, indem man ihn wie einen Drucktaster nach unten (zum Boden hin) drückt.



WANRUNG!

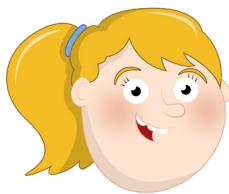


Der Sense HAT-Joystick sollte nur verwendet werden, wenn Sie die Abstandshalter wie zu Beginn dieses Kapitels beschrieben angebracht haben. Ohne die Abstandshalter kann ein Druck auf den Joystick die Sense HAT-Platine verbiegen und sowohl den Sense HAT als auch die GPIO-Stifteleiste des Raspberry Pi beschädigen.

Joystick-Steuerung in Scratch

Starten Sie ein neues Programm in Scratch. Dabei sollte die Raspberry Pi Sense HAT-Erweiterung geladen sein. Ziehen Sie wie zuvor einen „Ereignisse“-Block **Wenn [grün] angeklickt wird** auf Ihren Skriptbereich, dann ziehen Sie einen Block **clear display** darunter, gefolgt vom Ziehen und Bearbeiten eines Blocks **set background to schwarz** und eines Blocks **set colour to weiß**.

In Scratch werden die Eingaben des Joysticks vom Sense HAT den Cursortasten auf der Tastatur zugeordnet: Ein Drücken des Joysticks nach oben entspricht dem Drücken der Pfeiltaste nach oben, ein Drücken nach unten entspricht dem Drücken der Pfeiltaste nach unten, ein Drücken nach links entspricht dem Drücken der Pfeiltaste nach links und ein Drücken nach rechts entspricht dem Drücken der Pfeiltaste nach rechts; ein Drücken des Joysticks nach unten (zum Boden, centre) wie ein Drucktaster entspricht hingegen dem Drücken der **EINGABE**-Taste.



WARNUNG!

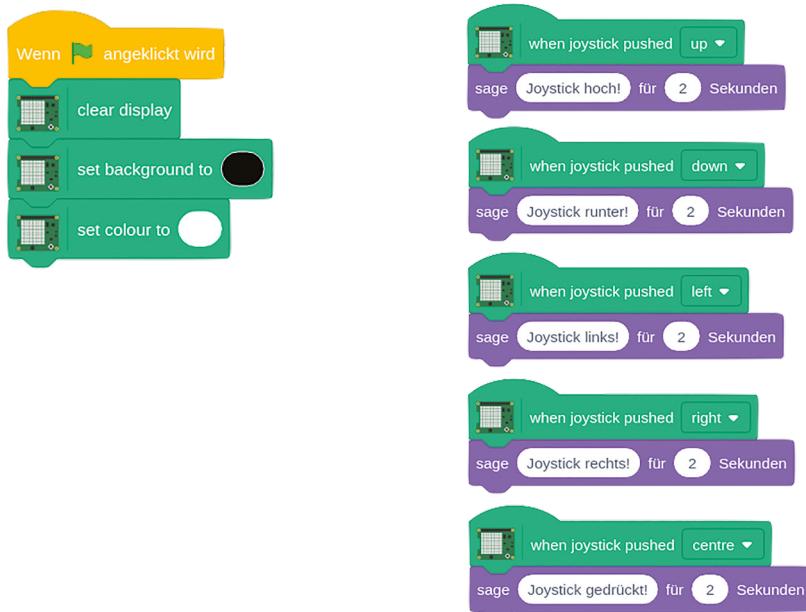
Die Joystick-Steuerung ist nur auf der physischen Sense HAT verfügbar. Wenn Sie den Sense HAT-Emulator verwenden, verwenden Sie stattdessen die entsprechenden Tasten auf Ihrer Tastatur, um das Drücken des Joysticks nachzuahmen.



Ziehen Sie einen Block **when joystick pushed up** auf Ihren Code-Bereich. Damit er etwas zu tun hat ziehen Sie einen Block **sage Hallo! für 2 Sekunden** darunter.



Drücken Sie den Joystick nach oben. Sie sehen, wie die Scratch-Katze fröhlich „Hallo!“ sagt. Als Nächstes bearbeiten Sie den Block **sage Hallo! für 2 Sekunden** zu einem **sage Joystick hoch! für 2 Sekunden** Block und fügen weiterhin „Ereignisse“- und „Aussehen“-Blöcke hinzu, bis Sie etwas für jede der fünf Arten, auf die der Joystick gedrückt werden kann, haben.



Versuchen Sie, den Joystick in verschiedene Richtungen zu drücken, damit Ihre Nachrichten erscheinen!



LETZTE HERAUSFORDERUNG



Können Sie mit dem Joystick des Sense HAT ein Scratch-Sprite auf dem Bühnenbereich steuern? Können Sie es so einrichten, dass die LEDs des Sense HAT einen Glückwunsch anzeigen, wenn das Sprite ein anderes Sprite, das ein Objekt darstellt, einsammelt?

Joystick-Steuerung in Python

Starten Sie ein neues Programm in Thonny und speichern Sie es als Sense HAT Joystick. Beginnen Sie mit den üblichen drei Zeilen, die den Sense HAT einrichten und die LED-Matrix löschen:

```
from sense_hat import SenseHat  
sense = SenseHat()  
sense.clear()
```

Als Nächstes richten Sie eine Endlosschleife ein:

```
while True:
```

Weisen Sie Python dann an, auf Eingaben des Sense-HAT-Joysticks mit der folgenden Zeile zu warten, die Thonny automatisch für Sie einrücken wird:

```
for event in sense.stick.get_events():
```

Fügen Sie am Schluss noch die folgende Zeile hinzu (sie wird von Thonny eingerückt) damit etwas passiert, wenn ein Joystickdruck erkannt wird.

```
    print(event.direction, event.action)
```

Klicken Sie auf „Run“ und versuchen Sie, den Joystick in verschiedene Richtungen zu drücken. Die gewählte Richtung wird im Python-Shell-Bereich ausgegeben: aufwärtsup (oben), abwärtsdown (unten), left (links), right (rechts) und middle (Mitte), wenn Sie den Joystick wie einen Drucktaster nach unten gedrückt haben.

Für jedes mal wenn Sie den Joystick drücken, werden zwei Ereignisse ausgelöst: das Ereignis **pressed** als Erstes, wenn Sie den Joystick in eine Richtung drücken, und danach das andere Ereignis **released**, wenn der Joystick in die Mitte zurückkehrt. Sie können dies in Ihren Programmen verwenden: Stellen Sie sich eine Figur in einem Spiel vor, die dazu gebracht werden könnte, sich zu bewegen, wenn der Joystick in eine Richtung gedrückt wird, und anzuhalten, sobald er losgelassen wird.

Sie können den Joystick auch zum Auslösen von Funktionen verwenden, anstatt nur auf die Verwendung einer „for“-Schleife beschränkt zu sein. Löschen Sie alles unter `sense.clear()` und geben Sie Folgendes ein:

```
def rot():
    sense.clear(255, 0, 0)

def blau():
    sense.clear(0, 0, 255)

def grün():
    sense.clear(0, 255, 0)

def gelb():
    sense.clear(255, 255, 0)
```

Diese Funktionen ändern die gesamte Sense HAT LED-Matrix in eine einzige Farbe: Rot, Blau, Grün oder Gelb. Damit können Sie ganz einfach sehen, ob Ihr Programm funktioniert! Um den Effekt tatsächlich auszulösen, müssen Sie Python mitteilen, welche Funktion mit welcher Joystick-Eingabe einhergeht. Geben Sie Folgendes ein:

```
sense.stick.direction_up = rot
sense.stick.direction_down = blau
sense.stick.direction_left = grün
sense.stick.direction_right = gelb
sense.stick.direction_middle = sense.clear
```

Schließlich benötigt das Programm eine Endlosschleife (bekannt als die *main*-Schleife), um weiterzulaufen und weiterhin auf Joystick-Eingaben zu achten, statt Ihren Code einfach nur einmal den Code zu durchlaufen und sich dann zu beenden. Geben Sie die folgenden Zeilen ein:

```
while True:
    pass
```

Klicken Sie auf „Run“ und versuchen Sie, den Joystick zu bewegen: Sie werden sehen, wie die LEDs in tollenprächtigen Farben aufleuchten! Um die LEDs auszuschalten, drücken Sie den Joystick wie einen Drucktaster: die `middle`-Richtung ist so eingestellt, dass die Funktion `sense.clear()` verwendet wird, um sie alle auszuschalten. Herzlichen Glückwunsch: Sie können Eingaben, die über den Joystick erfolgen, erfassen!



LETZTE HERAUSFORDERUNG



Können Sie mit dem, was Sie gelernt haben, ein Bild auf den Bildschirm zeichnen und es dann in die Richtung drehen, in die der Joystick gedrückt wird? Können Sie die "middle" Joystick-Eingabe (wie Drucktaster) zwischen mehr als einem Bild wechseln lassen?

Scratch-Projekt: Sense HAT-Wunderkerze

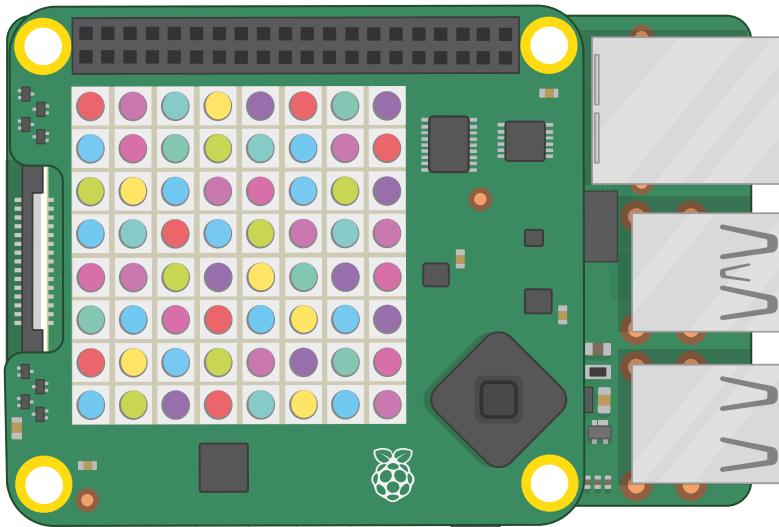
Jetzt, wo Sie sich mit dem Sense HAT auskennen, ist es an der Zeit, alles, was Sie gelernt haben, zusammenzufügen, um eine wärmeempfindliche Wunderkerze zu bauen – ein Gerät, das am besten funktioniert, wenn es kalt ist, und mit zunehmender Erhitzungswärmung immer langsamer wird. Starten Sie ein neues Scratch-Projekt und fügen Sie die Raspberry Pi Sense HAT-Erweiterung hinzu, falls sie noch nicht geladen ist. Beginnen Sie wie immer mit vier Blöcken: **Wenn grüne Flagge angeklickt wird**, **clear display**, **set background to schwarz** und **set colour to weiß**. Denken Sie daran, dass Sie die Farben von den Standardeinstellungen ändern müssen.

Beginnen Sie mit der Erstellung einer einfachen, aber kunstvollen Wunderkerze. Beginnen Sie mit dem Ziehen eines „Ereignisse“-Blocks **wiederhole fortlaufend** auf den Codebereich und füllen Sie ihn mit einem Block **set pixel x 0 y 0 to Farbe**. Statt der Verwendung von Set-Nummern füllen Sie jeden der x-, y- und Farbschnitte dieses Blocks mit einem „Operatoren“-Block **Zufallszahl 1 bis 10** aus.

Die Werte 1 bis 10 passen noch nicht, sodass Sie einige Änderungen vornehmen müssen. Die ersten beiden Zahlen im Block **set pixel** sind die X- und Y-Koordinaten des Pixels auf der LED-Matrix, d. h. sie sollten eine Zahl zwischen 0 und 7 sein. Ändern Sie also die ersten beiden Blöcke, damit sie so aussehen: **Zufallszahl 0 bis 7**. Der nächste Abschnitt ist die Farbe, auf die das Pixel gesetzt werden soll. Wenn Sie den Farbwähler verwenden, wird die von Ihnen gewählte Farbe direkt im Skriptbereich angezeigt. Intern werden die Farben jedoch durch eine Zahl dargestellt und Sie können die Zahl auch direkt verwenden. Bearbeiten Sie den letzten Zufallszahl-Block, sodass er wie folgt aussieht: **Zufallszahl 0 bis 16777215**.



Klicken Sie auf die grüne Flagge. Sie werden sehen, dass die LEDs auf dem Sense HAT in zufälligen Farben leuchten (**Abbildung 7-28**). Herzlichen Glückwunsch: Sie haben eine elektronische Wunderkerze erstellt!



▲ Abbildung 7-28: Leuchten der Pixel in zufälligen Farben

Die Wunderkerze ist nicht besonders interaktiv. Um dies zu ändern, beginnen Sie damit, einen Block **warte 1 Sekunde** so zu ziehen, dass er sich unter dem Block **set pixel**, aber noch innerhalb des Blocks **wiederhole fortlaufend** befindet. Ziehen Sie einen „Operatoren“-Block **○ / ○** auf die 1 und geben Sie dann 10 in das zweite Feld ein. Ziehen Sie abschließend einen Block **temperature** über den ersten Raum im Block „Divisions-Operator“..



Wenn Sie auf die grüne Flagge klicken, werden Sie feststellen, dass die Wunderkerze wesentlich langsamer ist als vorher, es sei denn, Sie wohnen an einem sehr kalten Ort. Das liegt daran, dass Sie eine temperaturabhängige Verzögerung erzeugt haben: Das Programm wartet jetzt eine Anzahl von *die aktuelle Temperatur geteilt durch 10* Sekunden vor jeder Schleife. Wenn die Temperatur in Ihrem Raum 20 °C beträgt, wartet das Programm 2 Sekunden, bevor es eine Schleife durchläuft. Wenn die Temperatur 10 °C beträgt, wartet es 1 Sekunde. Wenn sie unter 10 °C liegt, wartet es weniger als eine Sekunde.

Wenn Ihr Sense HAT eine negative Temperatur unter 0 °C, dem Gefrierpunkt von Wasser, misst, wird Ihr Programm versuchen, weniger als 0 Sekunden zu warten. Da das nicht möglich ist, zumindest ohne Zeitreisen zu erfinden, werden Sie den gleichen Effekt sehen, als ob siees 0 Sekunden warten würde. Herzlichen Glückwunsch: Sie können nun die Integration der verschiedenen Funktionen des Sense HAT in Ihre eigenen Programme in Angriff nehmen!

Python-Projekt: Sense-HAT-Tricorder

Jetzt, wo Sie sich mit dem Sense HAT auskennen, ist es an der Zeit, alles, was Sie gelernt haben, zusammenzufügen und einen Tricorder zu bauen – ein Gerät, das den Fans eines bestimmten Science-Fiction-Franchise sofort vertraut ist, da es in der Lage ist, die Werte verschiedener Sensoren auszugeben, die darin eingebaut sind.

Starten Sie ein neues Programm in Thonny und speichern Sie es als **Tricorder**. Dann beginnen Sie mit den traditionellen Zeilen, die Sie benötigen, um ein Sense HAT-Programm zu erstellen:

```
from sense_hat import SenseHat  
sense = SenseHat()  
sense.clear()
```

Als Nächstes sollten Sie damit beginnen, Funktionen für jeden der verschiedenen Sensoren des Sense HAT zu definieren. Beginnen Sie mit der inertialen Messeinheit (IMU)und geben Sie Folgendes ein:

```
def orientation():  
    orientation = sense.get_orientation()  
    pitch = orientation["pitch"]  
    roll = orientation["roll"]  
    yaw = orientation["yaw"]
```

Da Sie die Ergebnisse vom Sensor über die LEDs laufen lassen werden, ist es sinnvoll, sie zu runden, damit Sie nicht auf Dutzende von Dezimalstellen warten müssen. Runden Sie diese jedoch nicht auf ganze Zahlen, sondern auf eine Dezimalstelle, indem Sie Folgendes eingeben:

```
pitch = round(pitch, 1)  
roll = round(roll, 1)  
yaw = round(yaw, 1)
```

Schließlich müssen Sie Python anweisen, die Ergebnisse auf den LEDs scrollen zu lassen, sodass der Tricorder als Handgerät funktioniert, ohne dass er an einen Monitor oder Fernseher angeschlossen sein muss:

```
sense.show_message("Nicken {0}, Rollen {1}, Gieren {2}".
format(pitch, roll, yaw))
```

Nun, da Sie eine vollständige Funktion zum Ablesen und Anzeigen der Orientierung von der IMU haben, müssen Sie ähnliche Funktionen für jeden der anderen Sensoren erstellen. Beginnen Sie mit dem Temperatursensor:

```
def temperature():
    temp = sense.get_temperature()
    temp = round(temp, 1)
    sense.show_message("Temperatur: %s Grad Celsius" % temp)
```

Schauen Sie genau auf die Zeile, die das Ergebnis auf die LEDs ausgibt: `%s` ist als Platzhalter bekannt und wird durch den Inhalt der Variablen `temp` ersetzt. Auf diese Weise können Sie die Ausgabe mit der Beschriftung „Temperatur:“ und einer Maßeinheit „Grad Celsius“ schön formatieren, was Ihr Programm wesentlich freundlicher macht.

Als Nächstes definieren Sie eine Funktion für den Luftfeuchtigkeitssensor:

```
def humidity():
    humidity = sense.get_humidity()
    humidity = round(humidity, 1)
    sense.show_message("Feuchtigkeit: %s Prozent" % humidity)
```

Dann der Drucksensor:

```
def pressure():
    pressure = sense.get_pressure()
    pressure = round(pressure, 1)
    sense.show_message("Druck: %s Millibars" % pressure)
```

Und schließlich die Kompassanzeige des Magnetometers:

```
def compass():
    for i in range(0, 10):
        north = sense.get_compass()
        north = round(north, 1)
        sense.show_message("Norden: %s Grad" % north)
```

Die kurze **for**-Schleife in dieser Funktion nimmt zehn Messwerte vom Magnetometer auf, um sicherzustellen, dass genügend Daten vorhanden sind um Ihnen ein genaues Ergebnis zu liefern. Wenn Sie feststellen, dass sich der gemeldete Wert ständig ändert, versuchen Sie, den Code auf 20, 30 oder sogar 100 Wiederholungen der Schleife abzuändern, um die Genauigkeit weiter zu verbessern.

Ihr Programm hat jetzt fünf Funktionen, von denen jede einen Messwert von einem der Sensoren des Sense HAT überausliest und über die LEDs scrollt. Es muss jedoch eine Möglichkeit gefunden werden, den Sensor auszuwählen, den Sie verwenden möchten, und der Joystick ist dafür perfekt geeignet.

Geben Sie Folgendes ein:

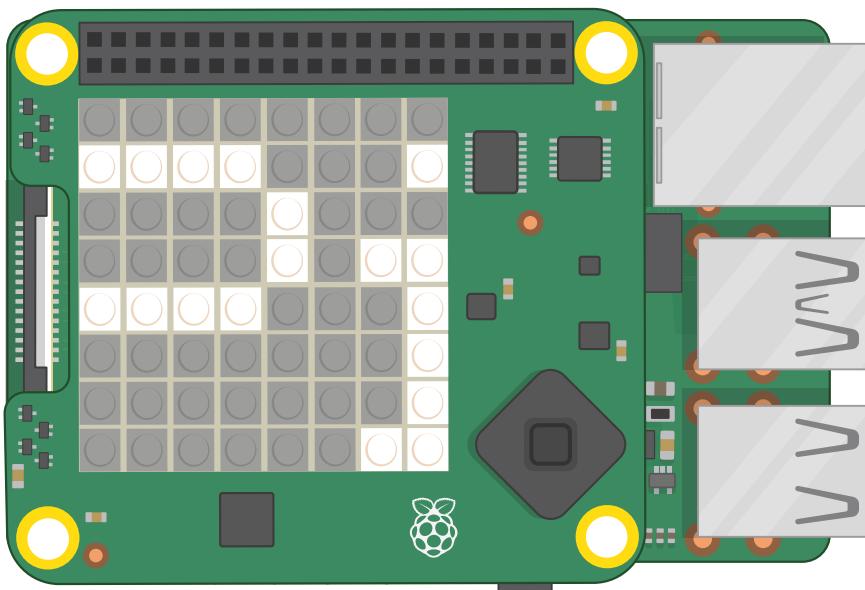
```
sense.stick.direction_up = orientation  
sense.stick.direction_right = temperature  
sense.stick.direction_down = compass  
sense.stick.direction_left = humidity  
sense.stick.direction_middle = pressure
```

Diese Zeilen ordnen jeder der fünf möglichen Richtungen auf dem Joystick einen Sensor zu: Nach oben wird der Orientierungssensor, nach unten das Magnetometer, nach links der Luftfeuchtigkeitssensor, nach rechts der Temperatursensor und durch Drücken des Joysticks in der Mitte wird der Drucksensor ausgelesen.

Schließlich benötigen Sie noch eine Hauptschleife, damit das Programm kontinuierlich auf die Joystick-Steuerung reagiert und sich nicht einfach sofort beendet. Geben Sie ganz unten in Ihrem Programm Folgendes ein:

```
while True:  
    pass
```

Klicken Sie auf „Run“ und bewegen Sie den Joystick, um einen Messwert von einem der Sensoren zu erhalten (**Abbildung 7-29**). Wenn das Ergebnis durch das LED-Matrix Display durchgelaufen ist, drücken Sie eine andere Richtung. Herzlichen Glückwunsch: Sie haben einen Handheld-Tricorder gebaut, auf den die Föderation der Planeten stolz wäre!



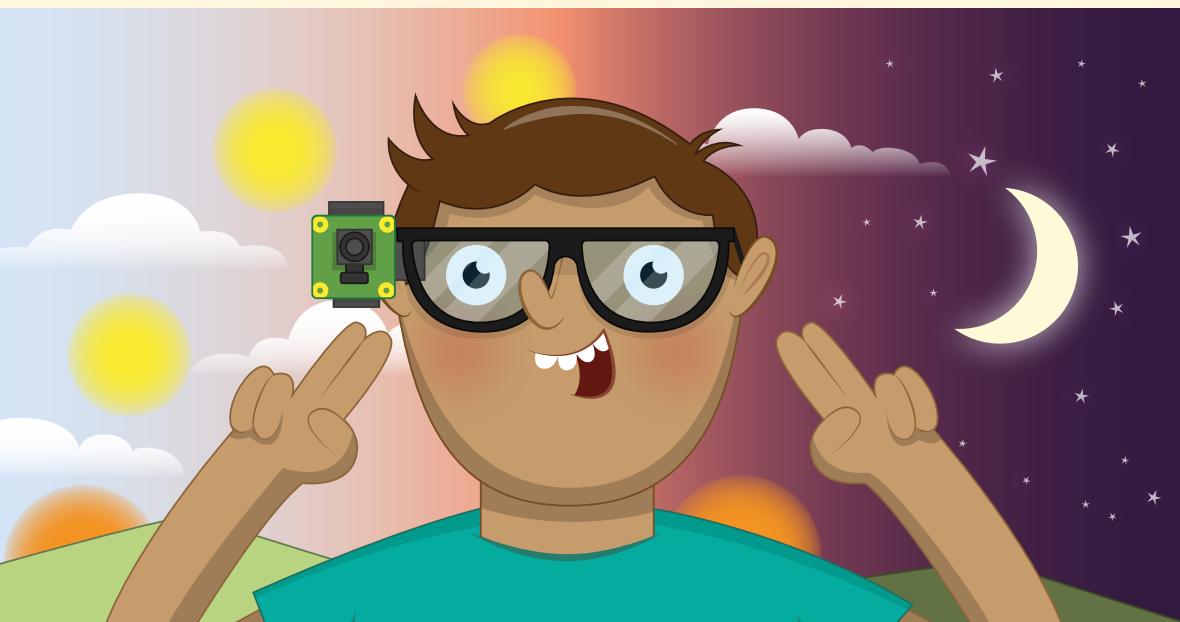
▲ Abbildung 7-29: Jeder Messwert scrollt über die Anzeige

Weitere Sense HAT-Projekte finden Sie unter den folgenden Links **Anhang D**,
Weiterführende Literatur.

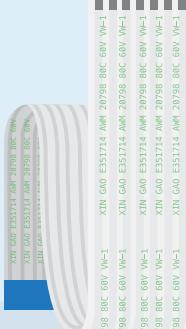
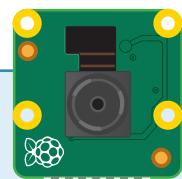
Kapitel 8

Raspberry Pi Kamera

Wenn Sie ein Camera Module oder eine HQ Camera an den Raspberry Pi anschließen, können Sie hochauflösende Fotos und Videos aufnehmen und beeindruckende Computer Vision-Projekte erstellen



Wenn Sie schon immer etwas bauen wollten, das sehen kann – in der Robotik bekannt als *Computer Vision* – dann ist das optionale Camera Module (Kameramodul) des Raspberry Pi oder die neue High Quality (HQ) Camera ein wunderbarer Ausgangspunkt. Das Camera Module und die HQ Camera sind kleine quadratische Platinen mit einem dünnen Flachbandkabel. Sie werden an den CSI-Anschluss (Camera Serial Interface) Ihres Raspberry Pi angeschlossen (nicht verfügbar beim Raspberry Pi 400) und liefern hochauflösende Standbilder und bewegte Videosignale, die selbstständig verwendet oder in Ihre Programme integriert werden können.



KAMERATYPEN

Es sind drei Arten von Raspberry Pi-Kameras erhältlich: das Standard Camera Module, die NoIR-Version und die High Quality (HQ) Camera. Wenn Sie normale Bilder und Videos in gut beleuchteten Umgebungen aufnehmen möchten, sollten Sie das Standard Camera Module verwenden. Wenn Sie spezielle Objektive verwenden möchten und auf beste Bildqualität Wert legen, verwenden Sie die HQ Camera. Das NoIR Camera Module – so genannt, weil es keinen Infrarot- oder IR-Filter besitzt – ist für den Einsatz mit Infrarot-Lichtquellen zur Aufnahme von Bildern und Videos in völliger Dunkelheit vorgesehen. Wenn Sie einen Nistkasten, eine Überwachungskamera oder ein anderes Projekt mit Nachtsicht bauen, bevorzugen Sie die NoIR-Version – aber denken Sie daran, gleichzeitig eine Infrarot-Lichtquelle zu kaufen!



Die Standard- und NoIR-Raspberry Pi Camera Modules basieren auf einem Sony IMX219-Bildsensor. Dies ist ein 8-Megapixel-Sensor, der das bedeutet, dass er Bilder mit bis zu 8 Millionen Pixeln aufnehmen kann. Es werden Bilder mit einer Breite von bis zu 3280 Pixel und einer Höhe von bis zu 2464 Pixel erfasst. Neben Standbildern kann das Camera Module auch Videomaterial in Full-HD-Auflösung (1920 x 1080 Pixel) mit einer Rate von bis zu 30 Bildern pro Sekunde aufnehmen (30 fps). Für flüssigere Bewegungen oder auch zur Erzeugung eines Zeitlupeneffekts kann die Kamera so eingestellt werden, dass sie mit reduzierter Auflösung aber dafür mit höherer Bildfrequenz aufnimmt: 60 fps für 720-p-Videomaterial (1296 x 730 Pixel) und bis zu 90 fps für 480-p-Videomaterial (VGA, 640 x 480 Pixel).

Die High Quality Camera verwendet einen 12,3-Megapixel-Sensor vom Typ Sony IMX477, der auch größer ist als der Sensor in den Standard- und NoIR Camera Modules. Das bedeutet, dass er mehr Licht sammeln kann und damit Bilder in höherer Qualität erzeugt. Im Gegensatz zu den Camera Modules hat die HQ Camera jedoch kein eingebautes Objektiv. Ohne ein solches kann sie keine Bilder oder Videos aufnehmen. Sie können ein beliebiges Objektiv mit einem C- oder CS-Anschluss verwenden. Andere Objektivanschlüsse erfordern einen entsprechenden C- oder CS-Anschlussadapter. Einzelheiten zur Montage eines Objektivs siehe **Anhang F: Einrichtung der High Quality Camera**.

RASPBERRY PI 400

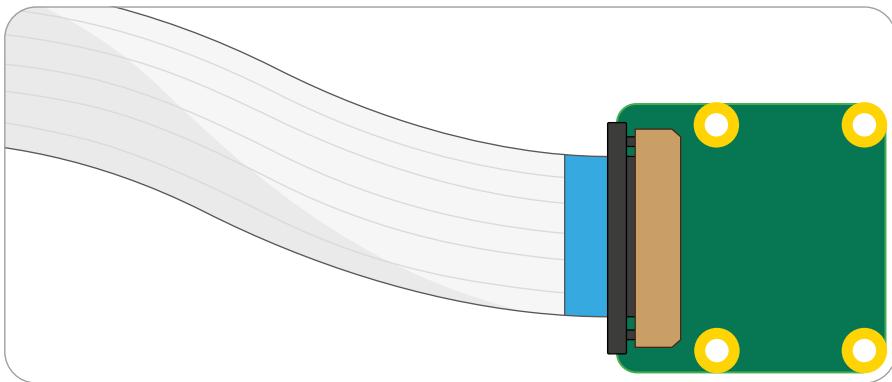
Leider sind die Raspberry Pi Camera Modules nicht mit dem Raspberry Pi 400 kompatibel. Sie können USB-Webkameras als Alternative verwenden, aber müssen in dem Fall auf den Einsatz der spezifischen Software-Tools für das Raspberry Pi Kameramodul Camera Module, die im Raspberry Pi OS enthalten sind, verzichten.



Installieren der Kamera

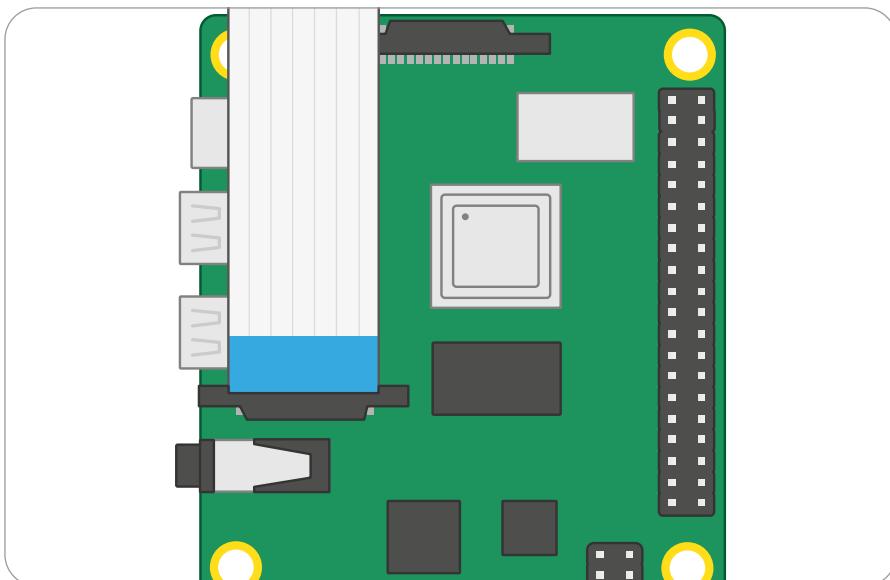
Wie jedes Hardware-Zusatzgerät sollte das Camera Module oder die HQ Camera nur dann an den Raspberry Pi angeschlossen oder vom Raspberry Pi getrennt werden, wenn der Strom abgeschaltet und das Netzkabel abgezogen ist. Wenn Ihr Raspberry Pi eingeschaltet ist, wählen Sie „Shutdown“ aus dem Raspberry-Menü, warten Sie, bis er sich abgeschaltet hat, und ziehen Sie den Netzstecker.

In den meisten Fällen ist das mitgelieferte Flachbandkabel bereits an das Camera Module oder die HQ Camera angeschlossen. Ist dies nicht der Fall, drehen Sie Ihre Kameraplatine um, sodass der Sensor auf der Unterseite liegt, und suchen Sie nach einem flachen Kunststoffstecker. Haken Sie Ihre Fingernägel vorsichtig um die abstehenden Kanten und ziehen Sie den Stecker nach außen, bis er teilweise herausgezogen ist. Schieben Sie das Flachbandkabel mit den silbernen Rändern nach unten und dem blauen Kunststoffband nach oben unter die soeben herausgezogene Lasche und schieben Sie diese dann behutsam wieder ein, bis ein Klick zu hören ist (**Abbildung 8-1**). Es spielt keine Rolle, welches Ende des Kabels Sie verwenden. Wenn das Kabel richtig installiert ist, ist es gerade positioniert und lässt sich durch leichtes Ziehen nicht herausnehmen. Ist es nicht richtig installiert, ziehen Sie die Lasche nochmals heraus und versuchen Sie es erneut.



▲ Abbildung 8-1: Anschließen des Flachbandkabels an das Camera Module

Installieren Sie das andere Ende des Kabels auf die gleiche Weise. Suchen Sie den Kamera- (oder CSI-) Anschluss am Raspberry Pi und ziehen Sie dessen Lasche vorsichtig nach oben. Wenn Ihr Raspberry Pi in einem Gehäuse installiert ist, erleichtern Sie sich das Arbeiten, indem Sie das Gehäuse zuerst abnehmen. Wenn der Raspberry Pi so positioniert ist, dass der HDMI-Anschluss (bzw. die HDMI-Anschlüsse) Ihnen zugewandt ist, schieben Sie das Flachbandkabel so ein, dass die silbernen Ränder zu Ihrer Linken und das blaue Kunststoffband zu Ihrer Rechten liegen (**Abbildung 8-2**). Dann schieben Sie die Lasche vorsichtig wieder an ihren Platz zurück. Wenn das Kabel richtig installiert ist, ist es gerade positioniert und lässt sich nicht abziehen, wenn Sie leicht daran ziehen. Ist es nicht richtig installiert, ziehen Sie die Lasche heraus und versuchen Sie es noch einmal.



▲ Abbildung 8-2: Anschließen des Flachbandkabels an den Kamera/CSI-Anschluss des Raspberry Pi

Das Camera Module wird mit einem Stückchen blauem Kunststoff als Abdeckung des Objektivs geliefert, um es während der Herstellung, des Versands und der Installation vor Kratzern zu schützen. Suchen Sie diese kleine Plastiklasche und ziehen Sie sie vorsichtig vom Objektiv ab, um die Kamera einsatzbereit zu machen.

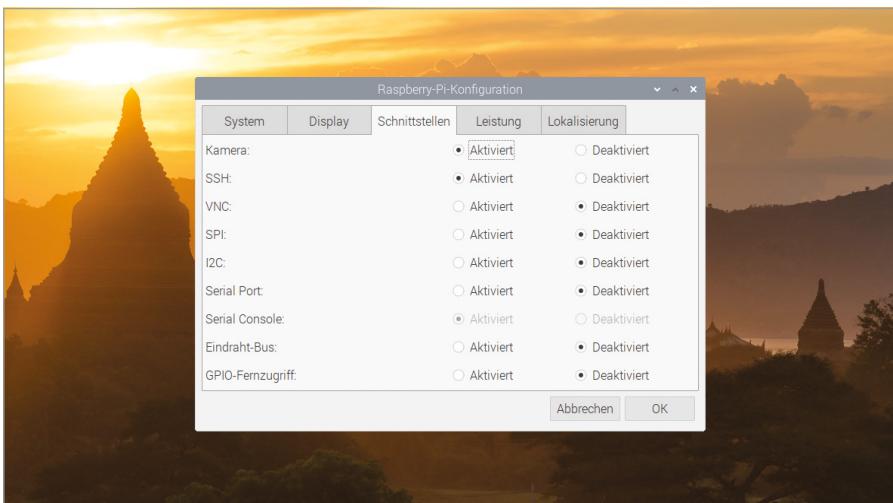


FOKUS EINSTELLEN

Das Camera Module wird in der Regel mit einem kleinen Kunststoffrad zur Einstellung des Fokus des Objektivs geliefert. Während der werkseitig eingestellte Fokus in der Regel perfekt ist, können Sie, wenn Sie Ihre Kamera für sehr nahe Nahaufnahmen verwenden, das Rad über das Objektiv schieben und es vorsichtig drehen, um den Fokus manuell einzustellen. Zur Fokussierung der HQ Camera siehe [Anhang F](#).



Schließen Sie das Netzteil wieder an den Raspberry Pi an und warten Sie, während das Raspberry Pi OS geladen wird. Bevor Sie die Kamera verwenden können, müssen Sie den Raspberry Pi darauf hinweisen: Öffnen Sie das Menü mit dem Raspberry-Symbol, wählen Sie die Kategorie „Einstellungen“ und klicken Sie auf „Raspberry-Pi-Konfiguration“. Wenn das Tool geladen ist, klicken Sie auf die Registerkarte „Schnittstellen“, suchen Sie den Eintrag „Kamera“ in der Liste und klicken Sie auf das runde Optionsfeld links neben „Aktiviert“, um es einzuschalten (**Abbildung 8-3**, umseitig). Klicken Sie auf OK. Das Tool fordert Sie auf, Ihren Raspberry Pi neu zu starten. Tun Sie dies und Ihre Kamera ist einsatzbereit!

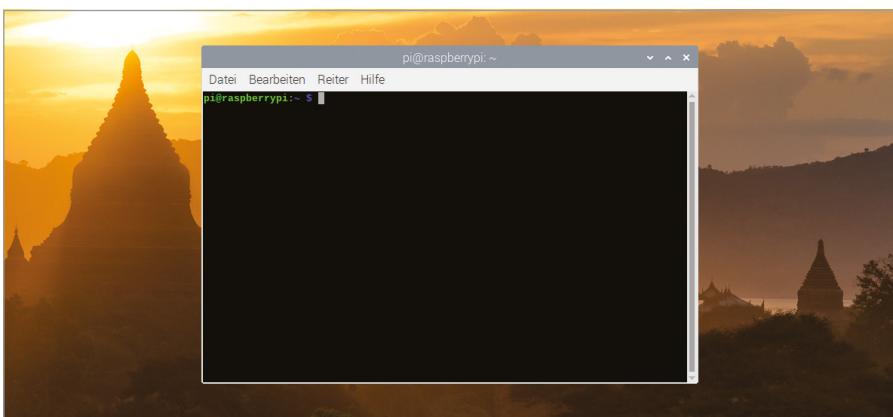


▲ Abbildung 8-3: Sie müssen die Kamera in der Raspberry-Pi-Konfiguration aktivieren

Testen der Kamera

Um zu bestätigen, dass Ihr Camera Module oder Ihre HQ Camera ordnungsgemäß installiert ist und Sie die Schnittstelle im Raspberry-Pi-Konfigurationstool aktiviert haben, können Sie das Tool **raspistill** verwenden. Zusammen mit **raspivid** für Videos ist dieses Tool darauf ausgelegt, Bilder von der Kamera mithilfe des Raspberry Pi *Befehlszeilenschnittstelle (CLI)* aufzunehmen.

Im Gegensatz zu den Programmen, die Sie bisher benutzt haben, werden Sie raspistill nicht im Menü finden. Klicken Sie stattdessen auf das Raspberry-Symbol, um das Menü zu öffnen, wählen Sie die Kategorie „Zubehör“ und klicken Sie dann auf „LXTerminal“. Es erscheint ein schwarzes Fenster, das Schriftzeichen in grün und blau enthält (**Abbildung 8-4**). Dies ist das *Terminal*, mit dem Sie auf die Befehlszeilenschnittstelle (CLI) zugreifen können.

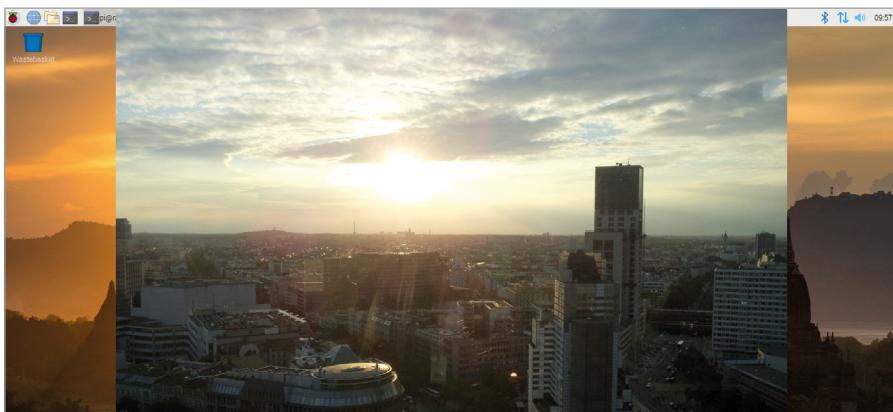


▲ Abbildung 8-4: Öffnen eines Terminal-Fensters zur Eingabe von Befehlen

Um die Kamera zu testen, geben Sie Folgendes in das Terminal ein:

```
raspistill -o test.jpg
```

Sobald Sie die **EINGABETASTE** drücken, sehen Sie ein großes Bild von dem, was die Kamera sieht, auf dem Bildschirm (**Abbildung 8-5**). Dies wird als *Live-Vorschau* bezeichnet. Sofern Sie dem raspistill-Tool keine andere Anweisung geben, dauert diese Vorschau 5 Sekunden. Nachdem diese 5 Sekunden abgelaufen sind, nimmt die Kamera ein einzelnes Standbild auf und speichert es in Ihrem Benutzerordner unter dem Namen **test.jpg**. Wenn Sie ein weiteres Bild erfassen möchten, geben Sie denselben Befehl erneut ein. Ändern Sie den Namen der Ausgabedatei nach dem **-o**. Andernfalls wird das erste Bild beim Speichern überschrieben.



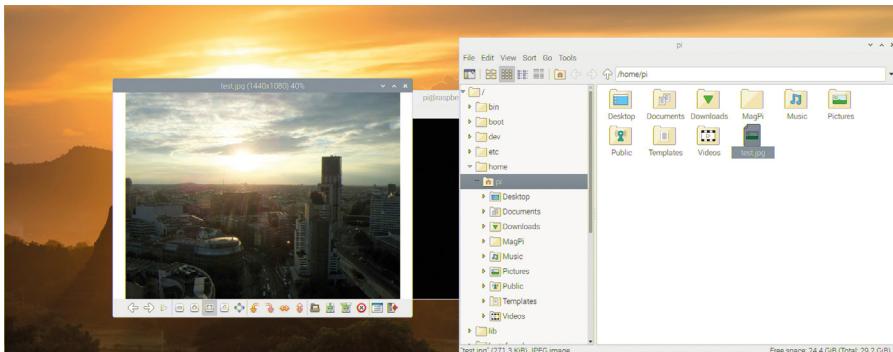
▲ Abbildung 8-5: Die Live-Vorschau von der Kamera

Wenn das Bild in der Live-Vorschau auf dem Kopf stand, müssen Sie raspistill mitteilen, dass die Kamera verkehrt herum steht. Das Camera Module ist so konstruiert, dass das Flachbandkabel an seiner unteren Kante (und damit des Blickfeldes) herausgeführt wird. Wenn es, von dem Blickfeld aus gesehen, an den Seiten oder oben herausgeführt wird, wie es bei einigen Kamerahalterungen von Drittanbietern der Fall ist, können Sie das Bild mit dem **-rot**-Parameter um 90, 180 oder 270 Grad drehen. Bei einer Kamera, die so montiert ist, dass das Kabel oben herauskommt, verwenden Sie einfach den folgenden Befehl:

```
raspistill -rot 180 -o test.jpg
```

Wenn das Flachbandkabel am rechten Rand herauskommt, verwenden Sie einen Rotationswert von 90 Grad. Beim linken Rand 270 Grad. Wenn Ihre ursprüngliche Aufnahme die falsche Orientierung hatte, versuchen Sie es nochmal, indem Sie die Ausrichtung mit dem **-rot**-Parameter korrigieren.

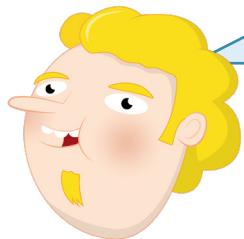
Um Ihr Bild zu sehen, öffnen Sie den Dateimanager aus der Kategorie „Zubehör“ des Raspberry-Menüs: Das von Ihnen aufgenommene Bild mit dem Namen **test.jpg** wird in Ihrem **home/pi**-Ordner angezeigt. Suchen Sie es in der Liste der Dateien und doppelklicken Sie dann darauf, um es in einem Bildbetrachter zu laden (**Abbildung 8-6**). Sie können das Bild auch an E-Mails anhängen, es mit Hilfe des Browsers auf Webseiten hochladen, oder auf ein externes Speichergerät ziehen.



▲ Abbildung 8-6: Öffnen des aufgenommenen Bildes

Vorstellung von picamera

Die flexibelste Art, das Camera Modul oder die -HQ Camera zu steuern, ist unter Verwendung von Python mit der praktischen picamera-Bibliothek. Damit haben Sie die volle Kontrolle über die Vorschau-, Bild- und Videoaufnahmefähigkeiten der Kamera und können diese in Ihre eigenen Programme integrieren – sogar in Kombination mit Programmen, die das GPIO-Modul über die GPIO Zero-Bibliothek nutzen!



PYTHON-PROGRAMMIERUNG

Die Projekte in diesem Kapitel setzen Erfahrungen mit der Programmiersprache Python, der integrierten Entwicklungsumgebung Thonny und den GPIO-Stiften des Raspberry Pi voraus. Falls Sie dies noch nicht getan haben, arbeiten Sie bitte zuerst die Projekte in **Kapitel 5, Programmieren mit Python** und **Kapitel 6, Physical Computing mit Scratch und Python** durch.



Schließen Sie das Terminal, falls es noch geöffnet ist, indem Sie auf das X oben rechts im Fenster klicken. Dann laden Sie Thonny aus der Kategorie „Entwicklung“ des -Raspberry-Menüs. Speichern Sie Ihr neues Projekt unter **Camera** und beginnen Sie dann mit dem Import der Bibliotheken, die Ihr Programm benötigt, indem Sie Folgendes in den Skriptbereich eingeben:

```
from picamera import PiCamera
from time import sleep
camera = PiCamera()
```

Die letzte Zeile ermöglicht Ihnen die Steuerung des Camera Module oder der HQ Camera über die **camera**-Funktion. Geben Sie zu Beginn Folgendes ein:

```
camera.start_preview()
sleep(10)
camera.stop_preview()
```

Klicken Sie auf „Run“. Ihr Desktop verschwindet und an seiner Stelle sehen Sie eine Vollbildvorschau dessen, was die Kamera sehen kann (**Abbildung 8-7**). Versuchen Sie, die Kamera zu bewegen oder mit der Hand vor der Linse zu winken. Sie werden sehen, wie sich das Bild auf dem Bildschirm entsprechend ändert. Nach 10 Sekunden wird die Vorschau geschlossen und Ihr Programm beendet – aber im Gegensatz zur Vorschau von raspistill wird kein Bild gespeichert.



▲ Abbildung 8-7: Eine bildschirmfüllende Live-Vorschau der Kameraansicht

Wenn Ihre Vorschau verkehrt herum angezeigt wird, drehen Sie das Bild.berichtigen Sie dies durch Drehen des Bildes: Direkt unter der Zeile **camera = PiCamera()** geben Sie ein:

```
camera.rotation = 180
```

Dieser Befehl korrigiert die Ausrichtung. Wie bei raspistill, können Sie das Bild mit **camera.rotation** um 90, 180 oder 270 Grad drehen, je nachdem, ob das Kabel auf der rechten, oberen oder linken Seite des Kameramodul/Camera Module herausgeführt wird. Denken Sie daran, **camera.rotation** beim Start jedes von Ihnen geschriebenen Programms zu verwenden, um zu vermeiden, dass Bilder oder Videos verkehrt herum aufgenommen werden!

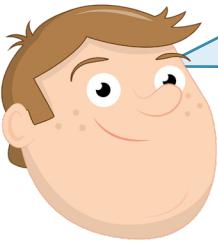
Aufnahme von Standbildern

Um ein Bild aufzunehmen und nicht nur eine Live-Vorschau zu zeigen, müssen Sie Änderungen an Ihrem Programm vornehmen. Beginnen Sie damit, die Verzögerung für die Vorschau zu verringern: Suchen Sie die Zeile `sleep(10)` und ändern Sie sie wie folgt:

`sleep(5)`

Fügen Sie direkt unter dieser Zeile Folgendes hinzu:

`camera.capture('/home/pi/Desktop/image.jpg')`

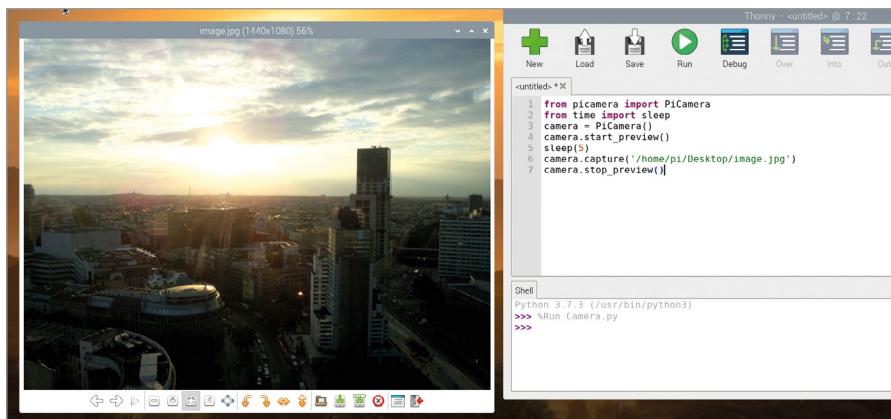


ZEIT ZUM EINSTELLEN



Wenn sich die Kamera im Vorschaumodus befindet, analysiert sie das Video, um festzustellen, ob sie ihre Einstellungen anpassen muss, um die beste Qualität zu gewährleisten. Sie sehen dies, wenn Sie sich in einer sehr dunklen oder sehr hellen Umgebung befinden, wobei die Vorschau zunächst nicht deutlich zu sehen ist und dann schnell klarer wird. Um der Kamera Zeit zur Anpassung zu geben, fügen Sie Ihrem Programm immer mindestens 2 Sekunden Vorschauzeit hinzu, bevor Sie ein Bild aufnehmen.

Die Funktion `camera.capture` weist Python an, ein Standbild zu speichern. Sie müssen dem Bild dabei gleich einen Namen zuweisen und auch den Ordner angeben, in dem es gespeichert werden soll. In diesem Beispiel speichern Sie es auf dem Desktop – suchen Sie danach direkt unterhalb dems Papierkorbs. Wenn das Thonny-Fenster im Weg ist, klicken Sie einfach auf seine Titelleiste und verschieben es. Doppelklicken Sie auf die Datei, um das von Ihnen aufgenommene Bild zu sehen (**Abbildung 8-8**). Herzlichen Glückwunsch: Sie haben eine Kamera programmiert!



▲ Abbildung 8-8: Öffnen des aufgenommenen Bildes

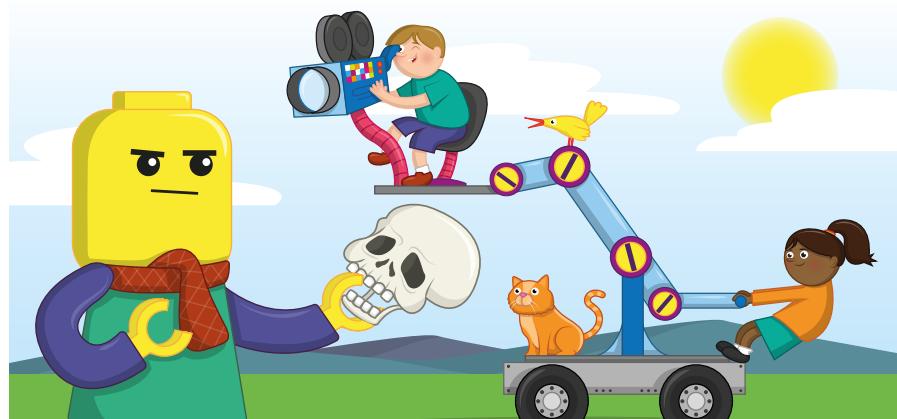
Aufnehmen von Videos

Sie können nicht nur Standbilder aufnehmen, sondern auch Videos. Löschen Sie alles zwischen den Zeilen `camera.start_preview()` und `camera.stop_preview()`. Dann geben Sie unter `camera.start_preview()` Folgendes ein:

```
camera.start_recording('/home/pi/Desktop/video.h264')
sleep(10)
camera.stop_recording()
```

Die Kameravorschau erscheint wie bisher, diesmal wird sie jedoch auch in einer Datei auf dem Desktop aufgezeichnet. Warten Sie die 10 Sekunden Schlafpause (`sleep`) von Python ab – wie wär's mit einem Tanz oder etwas Gymnastik vor der Kamera, um das Video interessanter zu machen? Wenn die Vorschau sich schließt, finden Sie Ihre Videodatei auf dem Desktop.

Zum Abspielen des Videos doppelklicken Sie einfach auf die Datei `video.h264` auf Ihrem Desktop. Das Video wird abgespielt – und wenn Sie getanzt haben, sehen Sie, wie gut Ihre Moves waren! Falls Ihr Video mit VLC media player abgespielt wurde, können Sie das VLC Fenster danach einfach beenden falls Sie kein weiteres Video anschauen möchten. Bei Wiedergabe mit dem omxplayer wird nach Beendigung des Videos die Player-Software mit m Glückwunscheiner freundlichen Nachricht im Terminal beendet. Herzlichen Glückwunsch: Sie können jetzt Videoaufnahmen mit Ihrem Raspberry Pi Kameramodul Camera Module oder Ihrer HQ Camera machen!



Stop-Motion-Animation auf Tastendruck

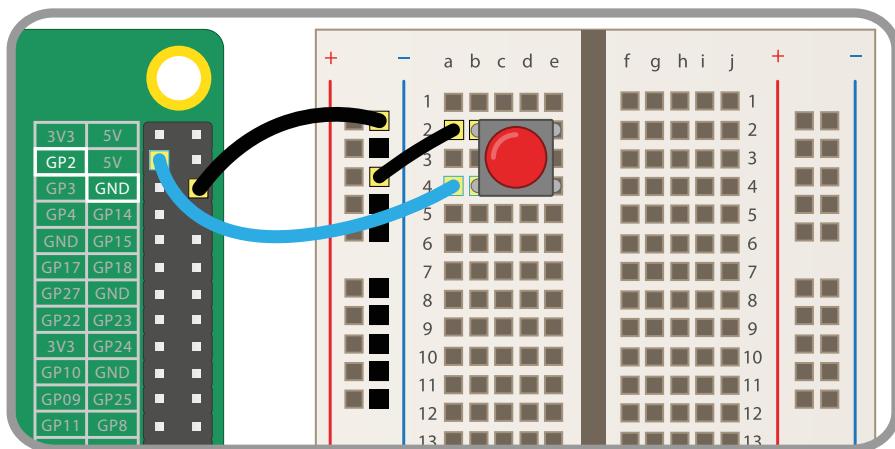
Mit dem, was Sie in diesem Kapitel gelernt haben, und Ihrem Wissen aus **Kapitel 6, Physical Computing mit Scratch und Python** darüber, wie man Hardware mit der GPIO-Stiftleiste des Raspberry Pi verbindet, können Sie jetzt Ihr eigenes Stop-Motion-Animationsstudio bauen.

Bei der Stop-Motion-Animation werden ganz viele einzelne Bilder von reglosen Objekten wie Modellautos oder Trickfilmfiguren, nacheinander aufgenommen, wobei die Objekte vor jeder Aufnahme ganz leicht verändert werden. Die Objekte bewegen sich auf keinem der Bilder, aber

beim Abspielen sieht es für das menschliche Auge so aus, als würden sie sich bewegen – so schnell oder langsam, wie Sie das wollen (wie beim Daumenkino).

Für dieses Projekt benötigen Sie eine Steckplatine, männliche-zu-männlichen-Jumperkabel (M2M) und männliche-zu-weiblichen-Jumperkabel (M2F) sowie einen Drucktaster. Wenn Sie keine Steckplatine haben, können Sie den Schalter stattdessen mit zwei weiblichen-zu-weiblichen Kabeln (F2F) anschließen, aber es wird schwieriger sein, ihn zu drücken. Wenn Sie sich Informationen zu diesen Komponenten in Erinnerung rufen möchten, sehen Sie sich **Kapitel 6, Physical Computing mit Scratch und Python** an. Natürlich benötigen Sie auch Objekte zum Animieren: Das kann alles sein, von einem Tonklumpen bis hin zu einem Spielzeugauto oder einer Action-Figur.

Beginnen Sie mit der Erstellung der Schaltung: Fügen Sie den Drucktaster zur Steckplatine hinzu und verbinden Sie dann die Erdungsschiene der Steckplatine mit einem Massestift auf dem Raspberry Pi (gekennzeichnet mit GND auf **Abbildung 8-9**) über ein männliches-zu-weibliches-Jumperkabel. Verwenden Sie ein männliches-zu-männliches-Jumperkabel, um ein Beinchen des Tasters mit der Erdungsschiene auf der Steckplatine zu verbinden, dann ein männliches-zu-weibliches-Jumperkabel, um das andere Beinchen des Schalters mit GPIO-Stift 2 zu verbinden (gekennzeichnet mit GP2 auf **Abbildung 8-9**).



▲ Abbildung 8-9: Schaltplan für den Anschluss eines Drucktasters an die GPIO-Stifte

Starten Sie ein neues Projekt in Thonny und speichern Sie es als **Stop Motion**. Beginnen Sie mit dem Importieren und Einrichten der Bibliotheken, die Sie für die Verwendung der Kamera und des GPIO-Anschlusses benötigen:

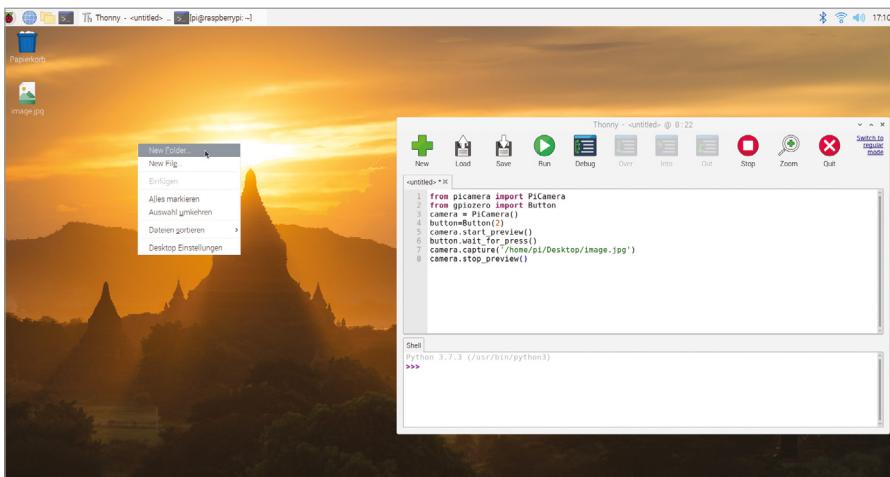
```
from picamera import PiCamera
from gpiozero import Button
camera = PiCamera()
button = Button(2)
```

Geben Sie Folgendes ein:

```
camera.start_preview()
button.wait_for_press()
camera.capture('/home/pi/Desktop/image.jpg')
camera.stop_preview()
```

Klicken Sie auf „Run“ und Sie sehen eine Vorschau von dem, worauf Ihre Kamera zeigt. Die Vorschau bleibt auf dem Bildschirm, bis Sie den Drucktaster drücken werden: Drücken Sie ihn jetzt und die Vorschau wird geschlossen, nachdem Ihr Programm ein Bild auf dem Desktop gespeichert hat. Finden Sie das Bild mit dem Namen **image.jpg** und doppelklicken Sie darauf, um es zu öffnen und zu bestätigen, dass das Programm funktioniert.

Bei Stop-Motion-Animationen werden viele Standbilder erstellt, um den Eindruck von Bewegung entstehen zu lassen, wenn sie aneinandergereiht werden. Diese ganzen Bilder sollten natürlich nicht lose auf Ihrem Desktop herumliegen. Legen Sie einen Ordner an, um sie zu speichern. Klicken Sie mit der rechten Maustaste irgendwo auf eine freie Stelle auf dem Desktop und wählen Sie „New Folder“ (neuer Ordner) (**Abbildung 8-10**). Nennen Sie den Ordner **animation** (alles in Kleinbuchstaben) und klicken Sie dann auf den Button OK.



▲ Abbildung 8-10: Erstellen eines neuen Ordners für Ihre aufgenommenen Bilder

Ihr Programm nach jeder Aufnahme neu starten zu müssen, wäre sehr mühsam. Ändern Sie es so, dass es in einer Schleife läuft. Im Gegensatz zu den Schleifen, die Sie früher schon erstellt haben, muss diese Schleife jedoch eine Möglichkeit haben, kontrolliert beendet zu werden. Andernfalls würden Sie den Desktop nicht mehr sehen, wenn Sie das Programm stoppen, während die Kameravorschau gerade angezeigt wird! Dazu müssen Sie zwei spezielle Anweisungen verwenden, **try** und **except**.

Beginnen Sie damit, alles nach `camera.start_preview()`, zu löschen, und geben Sie dann ein:

```
frame = 1
```

Dadurch wird die neue Variable `frame` erstellt, die Ihr Programm zur Speicherung der aktuellen Bildnummer verwenden wird. Sie werden sie in Kürze verwenden, um sicherzustellen, dass jedes Mal eine neue Datei gespeichert wird. Ohne diese Variable würden Sie mit jedem Tastendruck das vorherige Bild überschreiben!

Als Nächstes richten Sie Ihre Schleife ein:

```
while True:  
    try:
```

Die neue Anweisung `try` weist Python an, den Code auszuführen, der darin enthalten ist. In diesem Fall der Code für die Aufnahme von Bildern. Geben Sie ein:

```
button.wait_for_press()  
camera.capture('/home/pi/Desktop/animation/frame%03d.jpg' % frame)  
frame += 1
```

In diesen drei Zeilen Programmcode finden Sie ein paar clevere Tricks. Der erste betrifft den Dateinamen für die Erfassung: mit `%03d` wird Python angewiesen, eine Zahl zu nehmen und so viele führende Nullen hinzuzufügen, wie nötig sind, um sie dreistellig zu machen. So wird '1' zu '001', '2' zu '002' und '10' zu '010'. Dies ist erforderlich, um die Dateien in der richtigen Reihenfolge zu halten und sicherzustellen, dass bereits gespeicherte Dateien nicht überschrieben werden.

Die Eingabe von `% frame` am Ende dieser Zeile weist Python an, die Nummer der „frame“-Variablen im Dateinamen zu verwenden. Um sicherzustellen, dass jede Datei unverwechselbareinen eindeutigen Namen hat, inkrementiert die letzte Zeile – `frame += 1` – die Variable `frame` jedes Mal um 1. Wenn Sie die Taste zum ersten Mal drücken, wird `frame` von 1 auf 2 erhöht. Das nächste Mal von 2 auf 3, und so weiter.

Im Moment wird Ihr Programmcode jedoch nicht sauber beendet, wenn Sie mit dem Fotografieren fertig sind. Um dies zu ermöglichen, benötigen Sie einen `except`-Teil für Ihren `try`-Code. Geben Sie Folgendes ein und denken Sie daran, eine Einrückungsebene in der ersten Zeile zu entfernen, damit Python weiß, dass dieser Code nicht Teil `destroy`-Abschnitts ist:

```
except KeyboardInterrupt:  
    camera.stop_preview()  
    break
```

Ihr fertiges Programm sieht so aus:

```

from picamera import PiCamera
from time import sleep
from gpiozero import Button
camera = PiCamera()
button = Button(2)
camera.start_preview()
frame = 1
while True:
    try:
        button.wait_for_press()
        camera.capture('/home/pi/Desktop/animation/frame%03d.jpg'
% frame)
        frame += 1
    except KeyboardInterrupt:
        camera.stop_preview()
        break

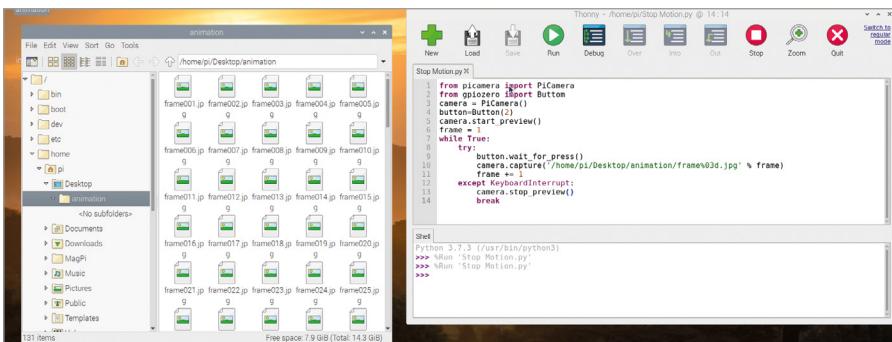
```

Versuchen Sie, auf „Run“ zu klicken, aber statt den Drucktaster zu betätigen, drücken Sie **STRG** und **C** auf der Tastatur. Sie brauchen nicht beide Tasten gleichzeitig zu drücken: Drücken und halten Sie einfach die **STRG**-Taste, drücken Sie dann **C** und lassen Sie anschließend **STRG** wieder los. Diese beiden Tasten wirken als Unterbrechung (engl. Interrupt), die Python anweist, mit dem laufenden Vorgang aufzuhören. Ohne die Zeile **except KeyboardInterrupt**: würde Python sofort beendet werden und die Kameravorschau würde Ihren Bildschirm blockieren. Mit dieser Zeile führt Python jedoch jeden darin befindlichen Code aus – in diesem Fall den Code zum Stoppen der Kameravorschau und zum sauberen Beenden.

Jetzt sind Sie soweit, mit der Aufnahme Ihrer Stop-Motion-Animation zu beginnen! Positionieren Sie das Camera Module oder die HQ-Camera dort, wo die zu animierenden Objekte erfasst werden können. Stellen Sie sicher, dass die Kamera sich dabei nicht bewegt. Wird die Kamera bewegt, verdirbt dies den Effekt. Platzieren Sie Ihre Objekte an ihren Startpositionen und klicken Sie auf „Run“, um Ihr Programm zu starten. Prüfen Sie, ob in der Vorschau alles gut aussieht, und drücken Sie auf den Drucktaster, um Ihr erstes Bild aufzunehmen.

Bewegen Sie die Objekte leicht – je geringer die Veränderung zwischen den Einzelbildern, desto flüssiger wird die fertige Animation sein. Drücken Sie den Drucktaster erneut, um ein weiteres Einzelbild aufzunehmen. Machen Sie so weiter, bis Ihre Animation fertig ist. Je mehr Einzelbilder Sie aufnehmen, desto länger wird Ihre Animation.

Wenn Sie fertig sind, drücken Sie **STRG+C**, um Ihr Programm zu schließen. Doppelklicken Sie dann auf den Ordner **animation** auf dem Desktop, um die aufgenommenen Bilder zu sehen (**Abbildung 8-11**, siehe umseitig). Doppelklicken Sie auf ein beliebiges Bild, um es zu öffnen und im Detail zu sehen!



▲ Abbildung 8-11: Die aufgenommenen Bilder im Ordner

Im Moment haben Sie bloß einen Ordner voller Standbilder. Um eine Animation zu erstellen, müssen Sie diese in ein Video umwandeln. Dafür klicken Sie auf das Raspberry-Symbol, um das Menü zu öffnen, wählen „Zubehör“ und klicken dann auf „LXTerminal“. Dies öffnet eine **Befehlszeilenschnittstelle**, die in **Anhang C** ausführlicher besprochen wird, damit Sie Befehle für den Raspberry Pi eingeben können. Wenn das Terminal geöffnet ist, wechseln Sie zunächst in den Ordner, indem Sie Folgendes eingeben:

cd Desktop/animation

Es ist wichtig, dass das „D“ von „Desktop“ in Großbuchstaben geschrieben wird. Das Raspberry Pi OS unterscheidet *Groß-/Kleinschreibung*. Wenn Sie daher einen Befehl oder Ordnernamen nicht genau so eingeben, wie er ursprünglich geschrieben wurde, funktioniert die Eingabe nicht! Nach dem Wechsel des Ordners geben Sie Folgendes ein:

ffmpeg -i frame%03d.jpg -r 10 animation.h264

Dabei wird ein Programm namens **ffmpeg** genutzt, um die Standbilder im Ordner zu nehmen und sie in ein Video mit dem Namen **animation.h264** zu konvertieren. (Hinweis: Wenn ffmpeg nicht verfügbar ist, können Sie es mit **sudo apt-get install ffmpeg** installieren.) Je nachdem, wie viele Standbilder Sie aufgenommen haben, kann dieser Vorgang einige Minuten dauern. Sie wissen, dass er beendet ist, wenn die Eingabeaufforderung des Terminals wieder erscheint. Um das Video abzuspielen, suchen Sie die Datei **animation.h264** im Ordner **animation** und doppelklicken Sie darauf, um sie zu öffnen. Alternativ können Sie sie vom Terminal aus abspielen, indem Sie Folgendes eingeben:

omxplayer animation.h264

Sobald das Video geladen ist, sehen Sie, wie Ihre Stop-Motion-Animation zum Leben erwacht. Herzlichen Glückwunsch: Sie haben Ihren Raspberry Pi in ein leistungsfähiges Animationsstudio verwandelt!

Wenn sich Ihre Animation zu schnell oder zu langsam bewegt, ändern Sie den Parameter **-r 10** des Befehls **ffmpeg** auf eine niedrigere oder höhere Zahl. Dies ist die Bildfrequenz

oder die Anzahl der Standbilder pro Sekunde Video. Eine niedrigere Zahl lässt Ihre Animation langsamer laufen und sieht ruckeliger aus. Eine höhere Zahl sieht flüssiger aus, lässt die Animation aber schneller ablaufen.

Wenn Sie Ihr Video speichern möchten, verschieben Sie es in Ihren Video-Ordner. Andernfalls wird dieses beim nächsten Programmstart überschrieben!

Erweiterte Kameraeinstellungen

Wenn Sie mehr Kontrolle über das Raspberry Pi Camera Module oder die HQ Camera benötigen, können Sie die Python-Bibliothek picamera verwenden, um auf verschiedene Einstellungen zuzugreifen. Diese Einstellungen werden zusammen mit ihren Standardwerten im Folgenden detailliert aufgeführt, damit Sie sie bei Bedarf in Ihre eigenen Programme einschließen können.

`camera.awb_mode = 'auto'`

Dies stellt den automatischen Weißabgleich der Kamera ein und kann auf einen der folgenden Modi eingestellt werden: **off**, **auto**, **sunlight**, **cloudy**, **shade**, **tungsten**, **fluorescent**, **incandescent**, **flash** oder **horizon**. Wenn Sie denken, dass Ihre Bilder und Videos einen Blau- oder Gelbstich haben, versuchen Sie einen anderen Modus.

`camera.brightness = 50`

Hiermit wird die Helligkeit des Kamerabildes eingestellt, von der dunkelsten bei 0 bis zur hellsten bei 100.

`camera.color_effects = None`

Dadurch ändert sich der Farbeffekt, den die Kamera verwendet. Normalerweise sollten Sie diese Einstellung unverändert lassen, aber wenn Sie ein Zahlenpaar angeben, können Sie die Art und Weise ändern, wie die Kamera Farbe aufzeichnet: Versuchen Sie **(128, 128)**, um ein Schwarz-Weiß-Bild zu erstellen.

`camera.contrast = 0`

Hier wird der Kontrast des Bildes eingestellt. Eine höhere Zahl lässt die Dinge dramatischer und schroffer erscheinen; eine niedrigere Zahl lässt die Dinge verwaschener erscheinen. Sie können eine beliebige Zahl zwischen -100 für minimalen Kontrast und 100 für maximalen Kontrast verwenden.

`camera.crop = (0.0, 0.0, 1.0, 1.0)`

Auf diese Weise können Sie das Bild zuschneiden und Teile an den Seiten und oben abschneiden, um nur den relevanten Bildausschnitt zu erfassen. Die Zahlen stehen für X-Koordinate, Y-Koordinate, Breite und Höhe und erfassen standardmäßig das gesamte Bild. Versuchen Sie, die letzten beiden Zahlen zu reduzieren – 0.5 und 0.5 sind ein guter Ausgangspunkt – und beobachten Sie, welche Wirkung diese Einstellung hat.

camera.exposure_compensation = 0

Dadurch wird die *Belichtungskompensation* der Kamera eingestellt. Sie können damit manuell steuern, wie viel Licht für jedes Bild erfasst wird. Im Gegensatz zur Änderung der Helligkeitseinstellung wird damit tatsächlich die Kamera selbst gesteuert. Gültige Werte reichen von -25 für ein sehr dunkles Bild bis 25 für ein sehr helles Bild.

camera.exposure_mode = 'auto'

Dies setzt den *Belichtungsmodus* oder die Logik, die das Camera Module bzw. die HQ Camera verwendet, um zu entscheiden, wie ein Bild belichtet werden soll. Mögliche Modi sind: **off**, **auto**, **night**, **backlight**, **spotlight**, **sports**, **snow**, **beach**, **verylong**, **fixedfps**, **antishake** und **fireworks**.

camera.framerate = 30

Hier wird die Anzahl der Bilder festgelegt, die zur Erstellung eines Videos pro Sekunde erfasst werden, oder die *Bildfrequenz*. Eine höhere Bildfrequenz erzeugt ein flüssigeres Video, belegt aber mehr Speicherplatz. Höhere Bildraten erfordern eine geringere Auflösung, die Sie über **camera.resolution** einstellen können.

camera.hflip = False

Dadurch wird das Kamerabild entlang der horizontalen (h) oder X-Achse gespiegelt, wenn die Einstellung auf **True** steht.

camera.image_effect = 'none'

Dies wendet einen von vielen Bildeffekten auf den Videostream an. Der Effekt ist dann sowohl in der Vorschau als auch in den gespeicherten Bildern und Videos sichtbar. Mögliche Effekte sind: **blur**, **cartoon**, **colorbalance**, **colorpoint**, **colorswap**, **deinterlace1**, **deinterlace2**, **denoise**, **emboss**, **film**, **gpen**, **hatch**, **negative**, **none**, **oilpaint**, **pastel**, **posterise**, **saturation**, **sketch**, **solarize**, **washedout** und **watercolor**.

camera.ISO = 0

Dadurch ändert sich die ISO-Einstellung der Kamera, was sich auf die Lichtempfindlichkeit der Kamera auswirkt. Standardmäßig stellt die Kamera dies automatisch in Abhängigkeit vom verfügbaren Licht ein. Sie können den ISO-Wert selbst auf einen der folgenden Werte einstellen: 100, 200, 320, 400, 500, 640, 800. Je höher der ISO-Wert, desto besser ist die Leistung der Kamera in schwach beleuchteten Umgebungen, aber desto körniger ist das Bild oder Video.

camera.meter_mode = 'average'

Diese Anweisung steuert, wie die Kamera bei der Einstellung der Belichtung die Menge des verfügbaren Lichts bestimmt. Die Standardeinstellung mittelt die verfügbare Lichtmenge über das gesamte Bild. Andere mögliche Modi sind: **backlit**, **matrix** und **spot**.

camera.resolution = (1920, 1080)

Hier wird die Auflösung des aufgenommenen Bildes oder Videos festgelegt, dargestellt durch zwei Werte für Breite und Höhe. Niedrigere Auflösungen beanspruchen weniger Speicherplatz und ermöglichen eine höhere Bildfrequenz. Höhere Auflösungen haben eine bessere Qualität, belegen aber mehr Speicherplatz.

camera.rotation = 0

Hiermit wird die Drehung des Bildes von 0 Grad über 90, 180 bis hin zu 270 Grad gesteuert. Verwenden Sie diesen Parameter, wenn Sie die Kamera nicht so positionieren können, dass das Flachbandkabel nach unten herauskommt.

camera.saturation = 0

Damit wird die Sättigung des Bildes oder die Lebendigkeit der Farben gesteuert. Mögliche Werte reichen von -100 bis 100.

camera.sharpness = 0

Hiermit wird die Schärfe des Bildes gesteuert. Mögliche Werte reichen von -100 bis 100.

camera.shutter_speed = 0

Dieser Parameter steuert, wie schnell sich der Verschluss bei der Aufnahme von Bildern und Videos öffnet und schließt. Sie können die Verschlusszeit manuell in Mikrosekunden einstellen, wobei längere Verschlusszeiten besser bei geringerem Licht und kürzere besser bei hellerem Licht funktionieren. Die Einstellung sollte normalerweise auf ihrer automatischen Standardvorgabe belassen werden.

camera.vflip = False

Dadurch wird das Kamerabild entlang der vertikalen oder Y-Achse gespiegelt, wenn die Einstellung auf **True** steht.

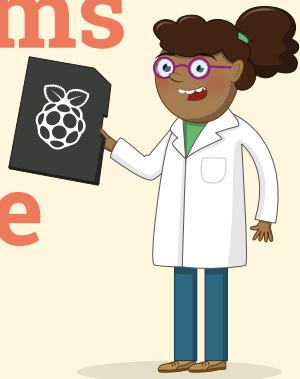
camera.video_stabilization = False

Bei Einstellung auf **True** wird die Videostabilisierung aktiviert. Dies ist nur erforderlich, wenn das Camera Modul oder die -K HQ Camera sich während der Aufnahme bewegt, z. B. wenn sie an einem Roboter befestigt sind oder getragen werden, um das Verwackeln des aufgenommenen Videos zu reduzieren.

Weitere Informationen zu diesen Einstellungen sowie weitere Einstellungen, die hier nicht dokumentiert sind, finden Sie unter picamera.readthedocs.io.

Anhang A

Installieren eines Betriebssystems auf einer microSD-Karte



Sie können microSD-Karten, auf denen NOOBS (New Out of the Box Software) vorinstalliert ist, bei allen guten Raspberry Pi-Händlern kaufen. Mit diesen microSD-Karten können Sie problemlos Raspberry Pi OS (früher bekannt als Raspbian) für Ihren Raspberry Pi installieren. Alternativ können Sie mit Hilfe der folgenden Anweisungen den Raspberry Pi Imager zur manuellen Installation eines Betriebssystems auf Ihrer eigenen leeren (oder wiederverwendeten) microSD-Karte benutzen.

WARNUNG!



Wenn Sie eine microSD-Karte mit bereits vorinstalliertem NOOBS erworben haben, brauchen Sie nichts weiter zu tun, als diese in Ihren Raspberry Pi einzustecken. Diese Anweisungen sind für leere microSD-Karten oder für bereits benutzte Karten, auf denen Sie ein neues Betriebssystem installieren möchten. Wenn Sie diese Anweisungen auf einer microSD-Karte mit darauf vorhandenen Dateien ausführen, gehen diese Dateien verloren. Stellen Sie also sicher, dass Sie vorher eine Sicherungskopie erstellt haben!

Raspberry Pi Imager herunterladen

Raspberry Pi OS basiert auf Debian und ist das offizielle Betriebssystem für den Raspberry Pi. Der einfachste Weg, Raspberry Pi OS auf einer microSD-Karte für Ihren Raspberry Pi zu installieren, ist das Tool Raspberry Pi Imager zu verwenden, das Sie unter rpf.io/downloads finden. Beachten Sie, dass diese Methode die Installation Ihres Betriebssystems mit Hilfe von NOOBS ersetzt, obwohl letzteres immer noch auf derselben Download-Seite verfügbar ist.

Die Anwendung Raspberry Pi Imager ist für Windows-, macOS- und Ubuntu-Linux-Computer verfügbar. Wählen Sie bitte die für Ihr System passende Version.

Auf macOS-Rechnern klicken Sie doppelt auf die heruntergeladene DMG-Datei.

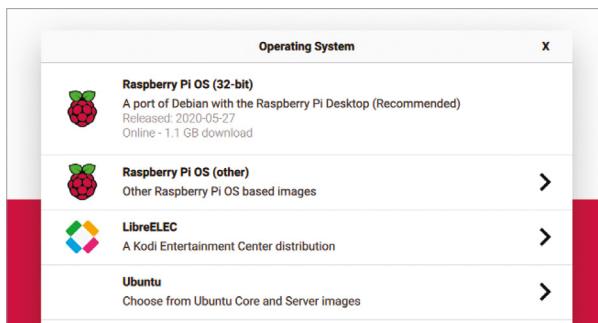
Möglicherweise müssen Sie Ihre Sicherheits- und Datenschutzeinstellungen so ändern, dass „Apps-Download erlauben von“ auf die Option „App Store und verifizierte Entwickler“ eingestellt ist, damit die Anwendung starten kann. Sie können dann einfach das Raspberry Pi Imager-Symbol in den Ordner „Programme“ ziehen.

Auf einem Windows-PC klicken Sie doppelt auf die heruntergeladene EXE-Datei. Wenn Sie dazu aufgefordert werden, wählen Sie die Schaltfläche „Ja“, um die Ausführung zu ermöglichen. Klicken Sie dann auf die Schaltfläche „Install“, um die Installation zu starten..

Schreiben Sie das Betriebssystem auf die microSD-Karte

Stecken Sie Ihre microSD-Karte in Ihren PC oder Mac ein: Sie benötigen dazu einen USB-Adapter für microSD-Karten, es sei denn, es ist ein Kartenleser eingebaut. Beachten Sie, dass die Karte nicht vorformatiert werden braucht.

Starten Sie die Anwendung Raspberry Pi Imager. Klicken Sie auf die Schaltfläche 'Choose OS', um das Betriebssystem auszuwählen, das Sie installieren möchten. Die erste Option ist die Standardversion von Raspberry Pi OS. Wenn Sie die abgespeckte Lite-Version oder die Vollversion (mit allen empfohlenen Programmen vorinstalliert) bevorzugen, wählen Sie „Raspberry Pi OS (other)“. Es gibt auch Optionen zur Installation von LibreELEC (wählen Sie die passende Version für Ihr Raspberry Pi-Modell) und Ubuntu Core oder Server.



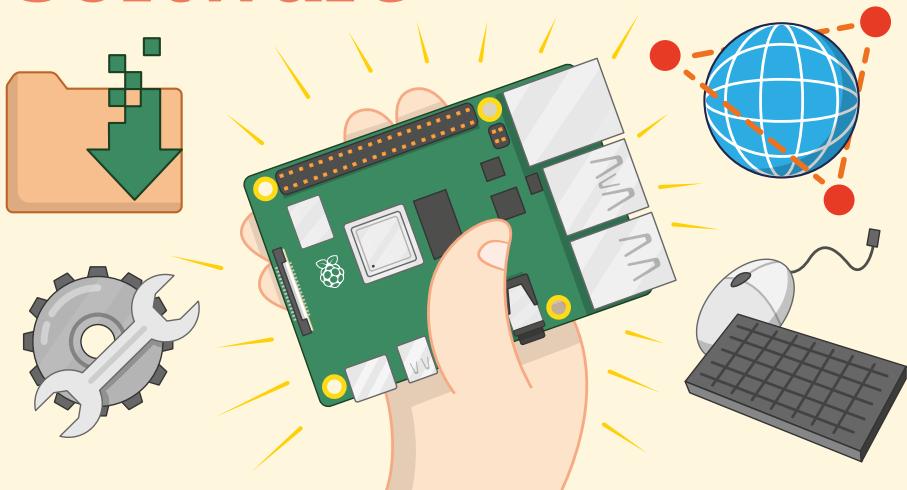
Anmerkung: Wenn Sie ein anderes Betriebssystem, wie z. B. Lakka, installieren möchten, laden Sie einfach dessen Image-Datei von der entsprechenden Website herunter und wählen Sie dann die Option „Use Custom“ in Raspberry Pi Imager.

Wenn ein Betriebssystem ausgewählt ist, klicken Sie auf die Schaltfläche „Choose SD card“ und wählen Sie Ihre microSD-Karte aus (normalerweise gibt es nur eine Option).

Klicken Sie abschließend auf die Schaltfläche „Write“ und warten Sie, während das Tool das ausgewählte Betriebssystem auf Ihre Karte schreibt und es dann überprüft. Wenn es fertig ist, können Sie die microSD-Karte entfernen. Sie können sie dann in Ihren Raspberry Pi einsetzen und in das Betriebssystem booten, das Sie gerade installiert haben.

Anhang B

Installieren und Deinstallieren von Software



Raspberry Pi OS wird mit einer Auswahl populärer Softwarepakete geliefert, die von der Raspberry Pi Foundation handverlesen wurden, aber dies sind nicht die einzigen Softwarepakete, die auf einem Raspberry Pi laufen können. Mit Hilfe der folgenden Anweisungen können Sie zusätzliche Software finden, installieren und wieder deinstallieren – und so die Fähigkeiten Ihres Raspberry Pi erweitern.

Die Anweisungen in diesem Anhang sind ein Zusatz zu den Anweisungen in **Kapitel 3, Verwendung Ihres Raspberry Pi**. Falls Sie das noch nicht gelesen haben, tun Sie dies, bevor Sie die in diesem Anhang beschriebenen Methoden anwenden.



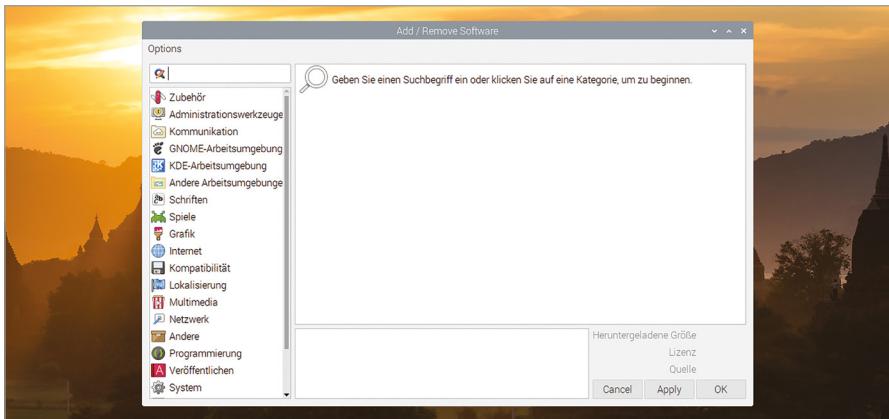
KARTENKAPAZITÄT

Das Hinzufügen weiterer Software zu Ihrem Raspberry Pi nimmt Platz auf Ihrer microSD-Karte in Anspruch. Mit einer Karte mit 16 GB oder mehr können Sie mehr Software installieren. Um zu prüfen, ob die Karte, die Sie verwenden möchten, mit dem Raspberry Pi kompatibel ist, besuchen Sie rpf.io/sdcardlist.

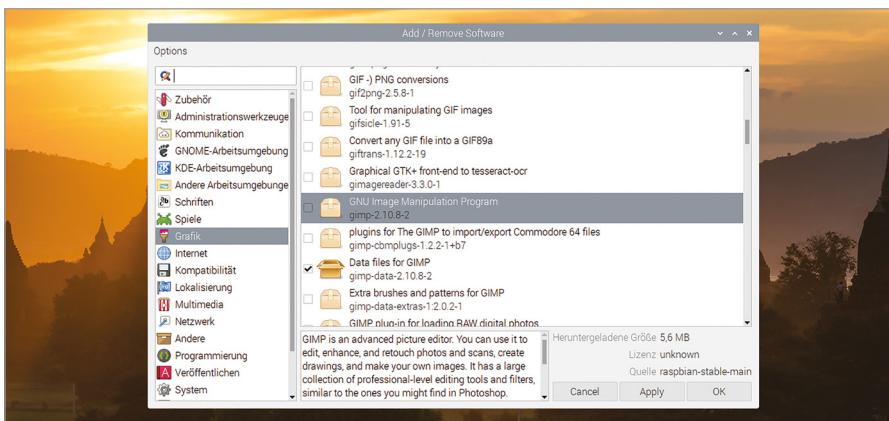


Verfügbare Software durchsuchen

Die Liste der für Raspberry Pi OS verfügbaren Softwarepakete stammt aus den sogenannten *software repositories* („Paketquellen“) des Betriebssystems. Sie können Sie wie folgt anzeigen und durchsuchen.: Klicken Sie auf das Raspberry-Symbol, um das Menü zu öffnen, wählen Sie die Kategorie „Einstellungen“ und klicken Sie dann auf Add/Remove Software. Nach einigen Sekunden erscheint das Fenster des Tools.



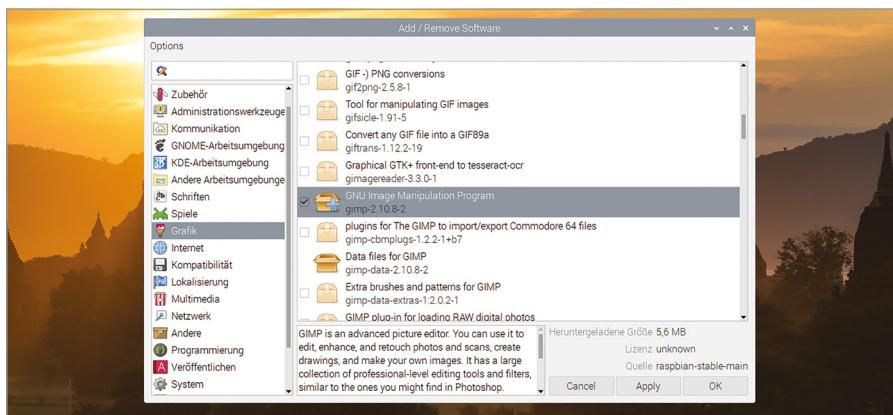
Die linke Seite des Software-Fensters enthält eine Liste von Kategorien – dieselben Kategorien, die Sie im Hauptmenü finden, wenn Sie auf das Raspberry-Symbol klicken. Wenn Sie auf eine davon klicken, erhalten Sie eine Liste der in dieser Kategorie verfügbaren Software. Sie können auch einen Suchbegriff in das Feld oben links im Fenster eingeben, z. B. „Texteditor“ oder „Games“, und erhalten eine Liste mit dazu passenden Softwarepaketen aus allen Kategorien. Drücken Sie nach Eingabe des Suchbegriffs die Eingabetaste um die Suche zu starten. Wenn Sie auf ein beliebiges Paket klicken, werden im unteren Bereich des Fensters zusätzliche Informationen zu diesem Paket angezeigt.



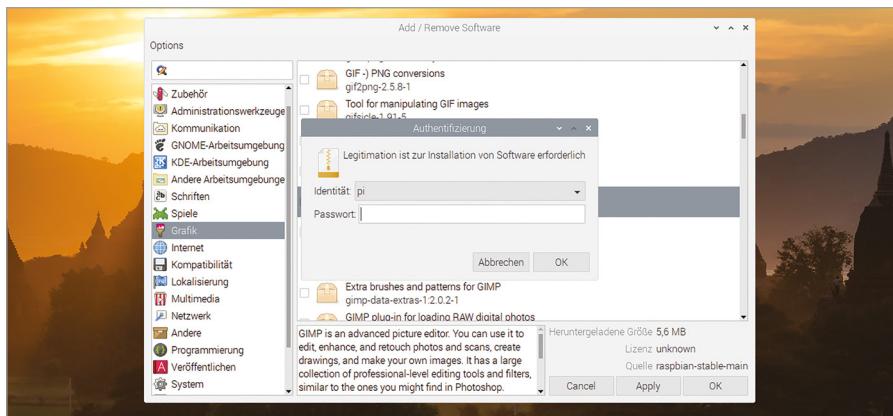
Wenn in der von Ihnen gewählten Kategorie viele Softwarepakete verfügbar sind, kann es einige Zeit dauern, bis das Add/Remove Software Tool die Liste vollständig vorbereitet hat und für Sie anzeigt.

Software installieren

Um ein Paket zur Installation auszuwählen, markieren Sie das Kästchen daneben, indem Sie darauf klicken. Sie können mehr als ein Paket auf einmal installieren: Klicken Sie einfach auf weitere Kästchen, um weitere Pakete hinzuzufügen. Das Symbol neben dem Paketnamen ändert sich in einen offenen Karton mit einem „+“-Symbol, um zu bestätigen, dass es installiert werden wird.

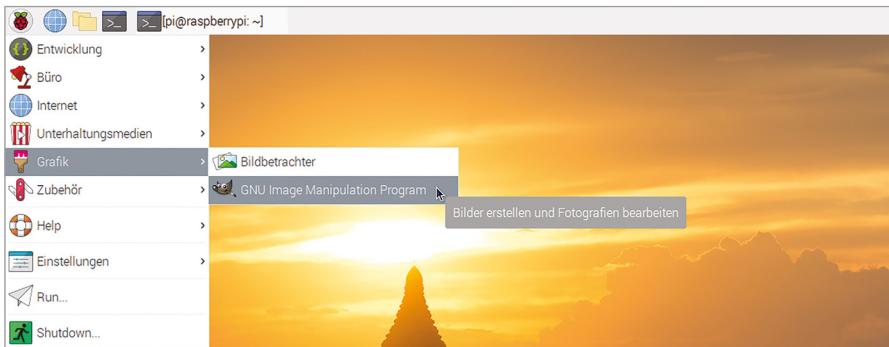


Wenn Sie mit Ihrer Auswahl zufrieden sind, klicken Sie entweder auf OK oder auf die Schaltfläche „Apply“. Der einzige Unterschied ist, dass OK das Softwaretool schließt, wenn Ihre Software installiert ist, während die Schaltfläche „Apply“ es danach geöffnet lässt. Sie werden gebeten, Ihr Passwort einzugeben, um Ihre Identität zu bestätigen – Sie möchten schließlich nicht, dass einfach irgendjemand Software zu Ihrem Raspberry Pi hinzufügen oder entfernen kann!



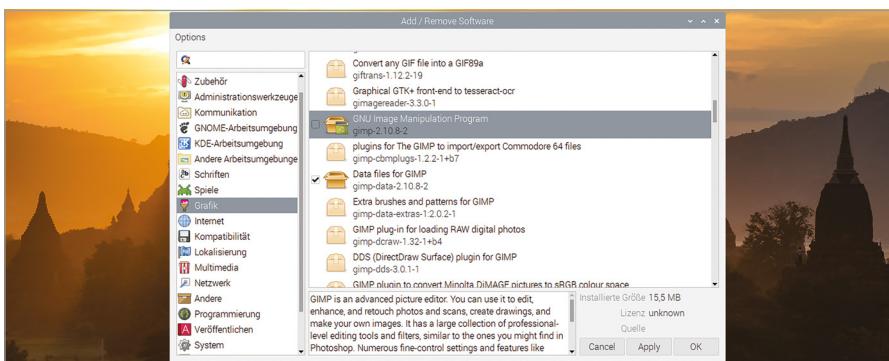
Wenn Sie ein einzelnes Paket installieren, kann es passieren, dass andere Pakete mitinstalliert werden, die als *Dependencies* (Abhängigkeiten) bezeichnet werden. Diese Pakete benötigen die von Ihnen zur Installation ausgewählte Software, um zu funktionieren – zum Beispiel Soundeffekt-Bundles für ein Spiel oder eine Datenbank, die zu einem Webserver gehört.

Sobald die Software installiert ist, sollten Sie sie finden können, indem Sie auf das Raspberry-Symbol klicken, um das Menü zu öffnen und darin die richtige Kategorie des Softwarepaketes finden. Beachten Sie, dass die Menükategorie nicht immer die gleiche ist wie die Kategorie aus dem Add/Remove Software Tool. Einige Programme haben überhaupt keinen Eintrag im Menü. Diese Programme sind als *Kommandozeilen-Software* bekannt und müssen über die Kommandozeile im Terminal ausgeführt werden. Weitere Informationen über die Kommandozeile und das Terminal finden Sie unter **Anhang C, Die Kommandozeile**.



Software deinstallieren

Um ein Paket zum Entfernen oder zur *Deinstallation* auszuwählen, suchen Sie es in der Liste der Pakete. Die Suchfunktion ist besonders praktisch dazu. Entfernen Sie das Häkchen in dem Kästchen daneben, indem Sie es anklicken. Sie können mehr als ein Paket auf einmal deinstallieren: Entfernen Sie weitere Häkchen, um zusätzliche Pakete zu entfernen. Das Symbol neben dem Paketnamen ändert sich in einen offenen Karton neben einem kleinen Papierkorb, um zu bestätigen, dass das Paket deinstalliert werden wird.



Wie zuvor können Sie auf OK oder „Apply“ klicken, um mit der Deinstallation der ausgewählten Softwarepakete zu beginnen. Sie werden gebeten, Ihr Passwort zu bestätigen, es sei denn, Sie haben dies innerhalb der letzten paar Minuten getan, und Sie werden möglicherweise zusätzlich aufgefordert, zu bestätigen, dass Sie auch alle Abhängigkeiten in Bezug auf Ihr Softwarepaket entfernen möchten. Wenn die Deinstallation abgeschlossen ist, verschwindet das Programm aus dem Raspberry-Symbol-Menü, aber Dateien, die Sie mit der Software selbst erstellt haben – z. B. Bilder bei einem Grafikpaket oder gespeicherte Spielstände bei einem Spiel – werden nicht entfernt.

WARNUNG!



Alle in Raspberry Pi OS installierten Programme erscheinen unter „Add/Remove Software“, einschließlich der für den Betrieb Ihres Raspberry Pi erforderlichen Software. Falls Sie versehentlich die falschen Pakete entfernen, kann es dazu kommen, dass die Desktop-Umgebung nicht mehr lädt. Um dies zu vermeiden, deinstallieren Sie Dinge nur dann, wenn Sie sicher sind, dass Sie sie nicht mehr benötigen. Wenn es für diesen Rat bereits zu spät ist, installieren Sie Raspberry Pi OS einfach neu, indem Sie den Anweisungen in **Kapitel 2, Erste Schritte mit Ihrem Raspberry Pi** folgen, oder lesen Sie **Anhang A** für eine ausführlichere Anleitung zur Installation des Betriebssystems..

Anhang C

Die Kommandozeile



Während Sie den Großteil der Software auf einem Raspberry Pi über die Desktop-Umgebung verwalten können, kann auf einige Tools nur über einen textbasierten Modus zugegriffen werden, der als *Command-Line Interface (CLI, Kommandozeile)* bezeichnet wird. Dies erfolgt in einer Anwendung namens Terminal. Die meisten Benutzer werden die CLI nie benutzen müssen, aber für diejenigen, die mehr erfahren möchten, bietet dieser Anhang eine grundlegende Einführung.



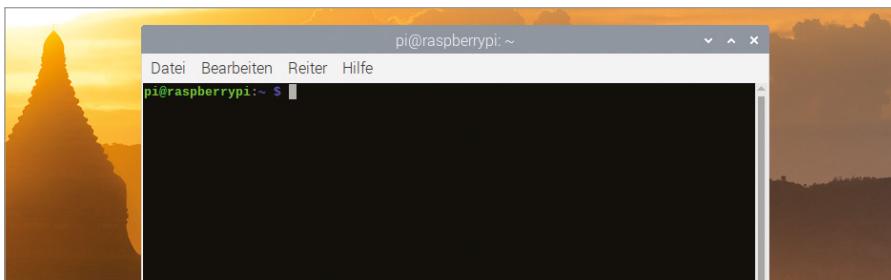
MEHR INFO

Dieser Anhang ist nicht als erschöpfende Anleitung für die Linux-Kommandozeilenschnittstelle gedacht. Für eine ausführlichere Beschreibung der Verwendung der CLI, öffnen Sie den Link rpf.io/terminal in einem Webbrowser.



Das Terminal öffnen

Der Zugriff auf die CLI erfolgt über das Terminal, ein Softwarepaket, das technisch als *virtual teletype (VTY (VTY))* (virtuelles Fernschreibterminal) bezeichnet wird. Der Name geht auf die Anfänge der Computer zurück, als die Benutzer Ihre Befehle noch über eine große elektromechanische Schreibmaschine und nicht über Tastatur und Monitor eingaben. Um das Terminal-Paket zu starten, klicken Sie auf das Raspberry-Symbol, um das Menü zu öffnen, wählen Sie die Kategorie „Zubehör“ und klicken Sie dann auf Terminal („LXTerminal“).



Das Terminal-Fenster kann wie jedes andere Fenster über den Desktop gezogen, in der Größe verändert, maximiert und minimiert werden. Sie können die Schrift darin auch größer machen, wenn sie schwer zu sehen ist, oder kleiner, wenn mehr in das Fenster passen soll: Klicken Sie auf das Menü „Bearbeiten“ und wählen Sie „Zoom In“ oder „Zoom Out“, oder drücken und halten Sie die **STRG**-Taste auf der Tastatur gefolgt von + oder -.

Die Eingabeaufforderung

Das Erste, was Sie in einem Terminal sehen, ist die *Eingabeaufforderung* („Prompt“), die auf Ihre Anweisungen wartet. Die Eingabeaufforderung auf einem Raspberry Pi, auf dem Raspberry Pi OS läuft, sieht wie folgt aus:

pi@raspberrypi:~ \$

Der erste Teil der Aufforderung, **pi** ist Ihr Benutzername. Der zweite Teil nach dem **@** ist der Hostname des Computers, den Sie verwenden, und zwar standardmäßig **raspberrypi**. Auf den **:** folgt eine Tilde **~**. Dies ist eine Kurzform des Verweises auf Ihren Benutzerordner und steht für Ihr *aktuelles Arbeitsverzeichnis* (*current working directory, CWD*). Schließlich zeigt das Symbol **\$**, dass Ihr Benutzer ein *Benutzer ohne Privilegien* ist. Das bedeutet, dass Sie ein Passwort benötigen, um Aufgaben wie das Hinzufügen oder Entfernen von Software durchzuführen.

Im Dateisystem navigieren

Versuchen Sie, Folgendes einzugeben, und drücken Sie dann die **EINGABE**-Taste:

cd Desktop

Wie Sie sehen, wechselt die Eingabeaufforderung zu:

```
pi@raspberrypi:~/Desktop $
```

Damit wird angezeigt, dass sich Ihr aktuelles Arbeitsverzeichnis geändert hat: Sie befanden sich vorher in Ihrem Benutzerordner, wie durch die Tilde ~ angezeigt. Jetzt befinden Sie sich im **Desktop**-Unterverzeichnis unterhalb Ihres Benutzerordners. Dazu haben Sie den Befehl **cd** – „change directory“ (Verzeichnis wechseln) – verwendet.



GROß- UND KLEINSCHREIBUNG



Die Kommandozeilenschnittstelle von Raspberry Pi OS unterscheidet zwischen Groß- und Kleinschreibung, d. h. es ist wichtig, ob Befehle oder Namen Groß- und Kleinbuchstaben haben. Wenn Sie beim Versuch, Verzeichnisse zu wechseln, die Meldung erhalten, dass keine Datei oder kein Verzeichnis vorhanden ist, überprüfen Sie, ob Sie Desktop mit einem großen D geschrieben haben.

Es gibt vier Möglichkeiten, zu Ihrem Benutzerordner zurückzukehren: Versuchen Sie sie der Reihe nach und wechseln Sie dazwischen jedes Mal zurück in das **Desktop**-Unterverzeichnis. Die erste ist:

```
cd ..
```

Das Symbol .. ist eine weitere Abkürzung, diesmal für „Verzeichnis über diesem“, auch bekannt als *übergeordnetes Verzeichnis*. Weil das übergeordnete Verzeichnis zu **Desktop** Ihr Benutzerordner ist, kehren Sie dorthin zurück. Wechseln Sie zurück in das **Desktop**-Unterverzeichnis und versuchen Sie die zweite Methode:

```
cd ~
```

Dieser Befehl verwendet das Symbol ~ und das bedeutet „in meinen Benutzerordner wechseln“. Im Gegensatz zum Befehl **cd ..**, der Sie einfach in das übergeordnete Verzeichnis des Verzeichnisses bringt, in dem Sie sich gerade befinden, funktioniert dieser Befehl von überall her – aber es gibt einen noch einfacheren Weg:

```
cd
```

Wenn kein Verzeichnisname eingegeben wird, geht **cd** einfach standardmäßig zurück zu Ihrem Benutzerordner. Der letzte Weg, um wieder in Ihren Benutzerordner zu gelangen, ist die folgende Eingabe:

```
cd /home/pi
```

Dabei wird ein sogenannter *absoluter Pfad* eingegeben, der unabhängig vom aktuellen Arbeitsverzeichnis funktioniert. Wie mit **cd** für sich allein oder **cd ~** kehren Sie von überall her zu Ihrem Benutzerordner zurück. Im Gegensatz zu den anderen Methoden müssen Sie jedoch Ihren Benutzernamen kennen.

Umgang mit Dateien

Um den Umgang mit Dateien zu üben, wechseln Sie in das **Desktop**-Verzeichnis und geben Sie Folgendes ein:

touch Test

Sie sehen jetzt eine neue Datei namens **Test** auf Ihrem Desktop. Der Befehl **touch** wird normalerweise verwendet, um die Datums- und Zeitangaben in einer Datei zu aktualisieren, aber wenn – wie in diesem Fall – die Datei nicht existiert, wird sie erstellt.

Geben Sie Folgendes ein:

cp Test Test2

Sie sehen nun eine weitere Datei, **Test2** auf dem Desktop. Dies ist eine *Kopie* der Originaldatei, die in jeder Hinsicht identisch ist. Löschen Sie sie durch folgenden Befehl:

rm Test2

Das entfernt die Datei und Sie werden sehen, wie sie verschwindet.

WARNUNG!

Im Gegensatz zum Löschen von Dateien mit dem grafischen Dateimanager, der sie für den späteren Abruf im Papierkorb speichert, sind Dateien, die mit **rm** gelöscht wurden, endgültig gelöscht. Achten Sie daher darauf Ihre Dateinamen sorgfältig einzugeben!



Versuchen Sie als Nächstes:

mv Test Test2

Dieser Befehl verschiebt die Datei und Sie sehen, wie Ihre Original **Test**-Datei verschwindet und durch **Test2** ersetzt wird. Der Befehl zum Verschieben, **mv**, kann so verwendet werden, um Dateien umzubenennen.

Wenn Sie nicht im Desktop-Verzeichnis sind, müssen Sie trotzdem in der Lage sein, zu sehen, welche Dateien sich in einem Verzeichnis befinden. Geben Sie ein:

ls

Dieser Befehl **listet** den Inhalt des aktuellen Verzeichnisses oder jedes anderen Verzeichnisses auf, das Sie dem Kommando angeben. Weitere Details, beispielsweise eine Auflistung aller versteckten Dateien und der Dateigrößen, können Sie mithilfe zusätzlicher Parameter abfragen:

ls -larth

Diese Parameter steuern den **ls**-Befehl: **l** ändert die Ausgabe in eine lange vertikale Liste. **a** weist das System an, alle Dateien und Verzeichnisse anzuzeigen, auch solche, die normalerweise verborgen wären. **r** kehrt die normale Sortierreihenfolge um. **t** sortiert nach Änderungszeitpunkt. In Kombination mit **r** werden die ältesten Dateien am Anfang der Liste und die neuesten Dateien am Ende ausgegeben. **h** verwendet menschenlesbare Dateigrößen, sodass die Liste leichter verständlich wird.

Ausführen von Programmen

Einige Programme können nur von der Kommandozeile aus ausgeführt werden, während andere sowohl grafische als auch Kommandozeilen-Schnittstellen haben. Ein Beispiel für Letzteres ist das Raspberry Pi-Konfigurationswerkzeug, das Sie normalerweise über das Menü mit dem Raspberry-Symbol starten.

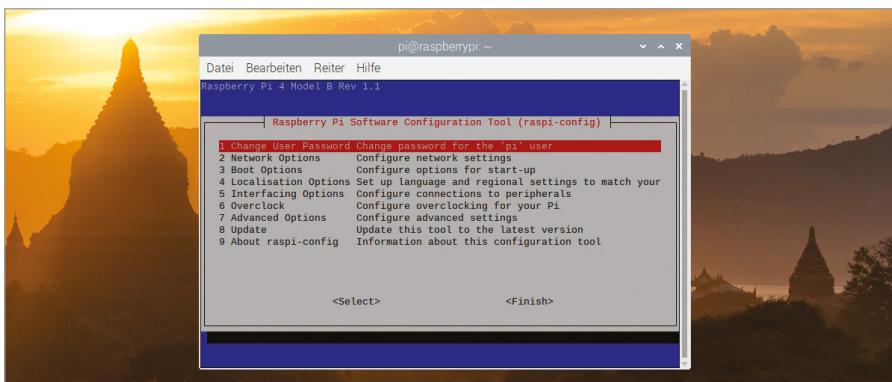
Geben Sie ein:

raspi-config

Sie erhalten eine Fehlermeldung, die Ihnen mitteilt, dass die Software nur als *root* ausgeführt werden kann, also mit dem Superuser-Konto auf Ihrem Raspberry Pi. Sie erfahren auch, wie Sie dies tun können, nämlich mit folgendem Befehl:

sudo raspi-config

Der Teil **sudo** des Befehls ist eine Abkürzung von *switch-user do* (Benutzerwechsel ausführen) und weist Raspberry Pi OS an, den Befehl als root-Benutzer auszuführen.



Sie benötigen **sudo** nur dann, wenn ein Programm höhere Rechte benötigt, z. B. bei der Installation oder Deinstallation von Software oder bei der Anpassung von Systemeinstellungen. Ein Spiel sollte zum Beispiel niemals mit **sudo** aufgerufen werden.

Drücken Sie die Taste **TAB** (Tabulator) zweimal, um „Finish“ zu wählen. Drücken Sie die **EINGABETASTE**, um das Raspberry Pi Software Configuration Tool zu beenden und zur Kommandozeilenschnittstelle zurückzukehren. Abschließend geben Sie ein:

exit

Dadurch wird Ihre Sitzung mit der Kommandozeilenschnittstelle beendet und die Terminal-Anwendung geschlossen.

Verwendung der TTYS

Die Terminal-Anwendung ist nicht die einzige Möglichkeit, die Kommandozeilenschnittstelle zu nutzen: Sie können auch zu einem der bereits laufenden Terminals wechseln, die als *ttypes* oder *TTYS* bekannt sind. Halten Sie die **STRG**- und **ALT**-Tasten auf Ihrer Tastatur und drücken Sie die **F2**-Taste zum Umschalten auf 'tty2'

```
Raspbian GNU/Linux 9 raspberrypi tty2
raspberrypi login:
```

Sie müssen sich erneut mit Ihrem Benutzernamen und Passwort anmelden. Danach können Sie die Kommandozeilenschnittstelle wie im Terminal benutzen. Die Verwendung dieser TTYS ist praktisch, wenn aus welchem Grund auch immer die Desktop-Schnittstelle nicht funktioniert.

Um vom TTY wegzuschalten, drücken und halten Sie **STRG+ALT** und drücken Sie dann **F7**: Der Desktop wird wieder angezeigt. Drücken Sie **STRG+ALT+F2** und wechseln Sie damit wieder zu „tty2“. Alles, was Sie darin bisher ausgeführt hatten, wird immer noch da sein.

Bevor Sie wieder wechseln, tippen Sie:

exit

Dann drücken Sie **STRG+ALT+F7**, um zurück zum Desktop zu gelangen. Der Grund für das Beenden der Sitzung vor dem Wechsel weg vom TTY besteht darin, dass jeder, der Zugang zur Tastatur hat, auf ein TTY wechseln kann. Wenn Sie darin noch angemeldet sind, können diese Personen auf Ihr Konto zugreifen, ohne Ihr Passwort zu kennen!

Herzlichen Glückwunsch: Sie haben Ihre ersten Schritteum die Kommandozeilenschnittstelle von Raspberry Pi OS beherrschen zu lernen, gemacht!

Anhang D

Weiterführende Literatur

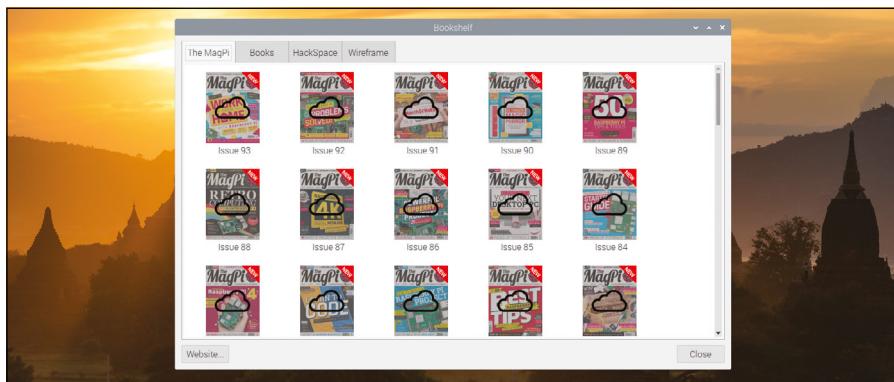


Das offizielle *Raspberry Pi-Handbuch für Anfänger* ist dazu gedacht, Ihnen den Einstieg in die Benutzung Ihres Raspberry Pi zu erleichtern, aber es ist keineswegs ein vollständiger Überblick über alles, was Sie tun können. Die Raspberry-Pi-Gemeinschaft ist weltumspannend und riesig. Menschen nutzen den Raspberry Pi für alles, von Spielen und Sensoranwendungen bis hin zu Robotik und künstlicher Intelligenz, und es gibt eine reiche Fülle von Inspirationen und Anregungen. Dieser Anhang hebt einige Quellen von Projektideen, Unterrichtsplänen und anderem Material hervor, die als ein großartiger nächster Schritt dienen, nachdem Sie sich das *Handbuch für Anfänger* durch gearbeitet haben.

Bookshelf (Bücherregal)

► Raspberry-Menü > Help > Bookshelf

Bookshelf ist eine Anwendung, die im Lieferumfang von Raspberry Pi OS enthalten ist und Ihnen beim Stöbern hilft, digitale Versionen der Veröffentlichungen von Raspberry Pi Press herunter zu laden und zu lesen – einschließlich dieses Raspberry Pi Anfängerhandbuchs. Starten Sie es einfach, indem Sie auf das Raspberry-Menüsymbol klicken, Help wählen und auf Bookshelf klicken; stöbern Sie dann in einer Reihe von Zeitschriften und Büchern, die Sie alle kostenlos herunterladen und in Ruhe lesen können.



Der Raspberry Pi-Blog

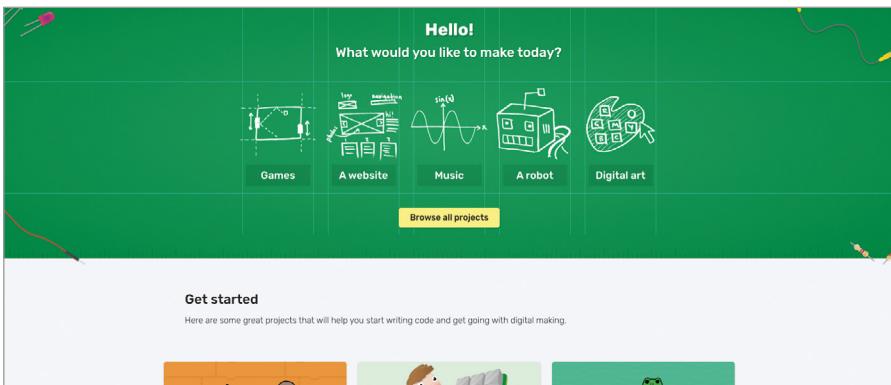
► rpf.io/blog

Ihr erster Anlaufpunkt für die neuesten Nachrichten über alles, rund um Raspberry Pi. Der offizielle Blog deckt alles ab, von der Einführung neuer Hardware und neuem Lehrmaterial bis hin zu Zusammenstellungen der besten Gemeinschaftsprojekte, -kampagnen und -initiativen. Wenn Sie über alles, was Raspberry Pi betrifft, auf dem Laufenden bleiben wollen, dann sind Sie hier genau richtig.

Raspberry Pi-Projekte

► rpfi.io/projects

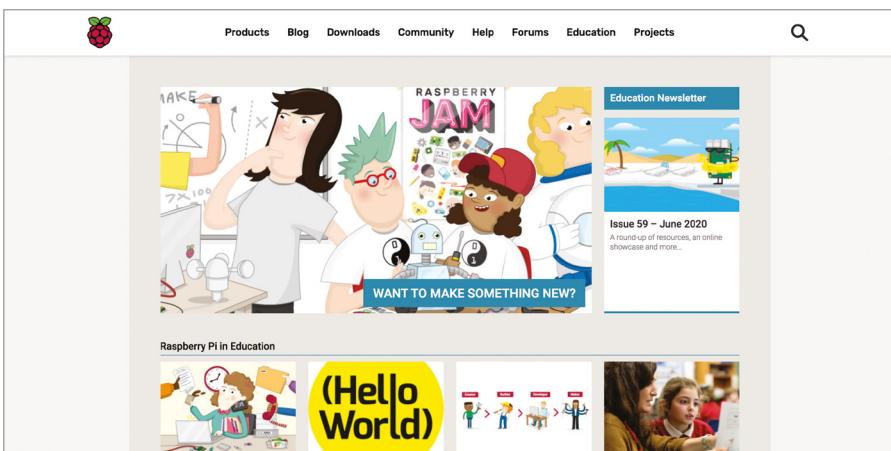
Die offizielle Raspberry Pi-Projekt-Webseite bietet Schritt-für-Schritt-Anleitungen für Projekte in einer Reihe von Kategorien, vom Spielen und Musizieren bis hin zur Erstellung Ihrer eigenen Website oder eines vom Raspberry Pi angetriebenen Roboters. Die meisten Projekte sind in einer Vielzahl von Sprachen verfügbar und decken eine Reihe von Schwierigkeitsgraden ab, die für jeden geeignet sind, vom absoluten Anfänger bis zum erfahrenen Maker.



Raspberry Pi Education

► rpfi.io/education

Die offizielle Website von Raspberry Pi Education bietet Newsletter, Online-Schulungen und Projekte, bei denen Pädagogen im Mittelpunkt stehen. Die Website enthält auch Links zu zusätzlichen Ressourcen, darunter das Schulungsprogramm der Picademy, die von Freiwilligen getragenen Organisationen Code Club und CoderDojo, in denen Programmierkenntnisse vermittelt werden, sowie weltweite Raspberry Jam-Veranstaltungen.



Die Raspberry Pi-Foren

► rpf.io/forums

Die Raspberry Pi-Foren sind der Ort, an dem Raspberry Pi-Fans zusammenkommen und sich über alles von Anfängerthemen bis hin zu tiefgehenden technischen Themen unterhalten können – und es gibt sogar einen 'Off-Topic'-Bereich für allgemeine Gespräche!

The screenshot shows the official Raspberry Pi forums website. At the top, there's a navigation bar with links for Products, Blog, Downloads, Community, Help, Forums, Education, and Projects. Below the navigation is a search bar and links for Register and Login. The main content area has a header 'Community' with columns for Topics, Posts, and Last post. It lists three topics: 'General discussion', 'Announcements', and 'Other languages'. Each topic row includes a small icon, the topic name, its statistics, and a brief description.

Topics	Posts	Last post
41299	328456	Re: Yesterday, another 3 Pi4B... by Jamesh Wed Jul 01, 2020 3:23 pm
5	5	Raspberry Pi OS (64 bit) beta... by gsh Thu May 28, 2020 6:29 am
18214	92312	Re: Hulp bij code by Baspiet Wed Jul 01, 2020 2:41 pm

The MagPi Magazine

► magpi.cc

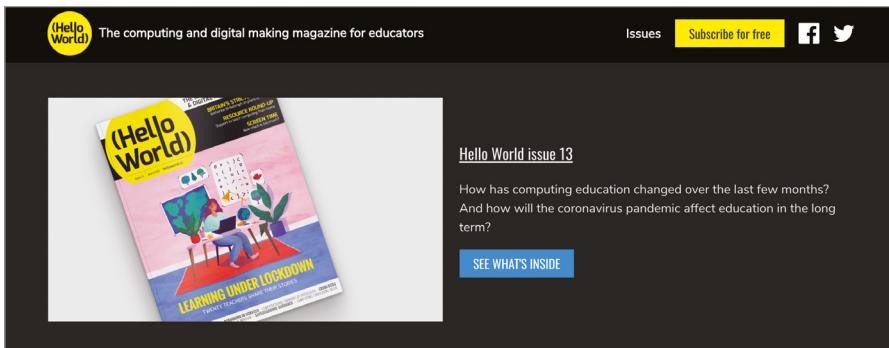
The MagPi ist die offizielle Raspberry Pi Zeitschrift. Es ist eine monatliche Hochglanzpublikation, die von Tutorials und Leitfäden bis hin zu Rezensionen und Nachrichten alles abdeckt. Sie wird dabei kräftig von der weltweiten Raspberry-Pi-Gemeinschaft unterstützt. Exemplare sind in allen guten Zeitschriftenläden und Supermärkten erhältlich und können auch kostenlos digital heruntergeladen werden, unter der Creative-Commons-Lizenz. The MagPi veröffentlicht auch Bücher und Bookazines zu einer Vielzahl von Themen, die in gedruckter Form gekauft oder kostenlos heruntergeladen werden können.

The screenshot shows the official The MagPi magazine website. At the top, there's a logo for 'The MagPi Magazine' and social media links for YouTube, Facebook, Twitter, and RSS. Below the header is a navigation bar with links for Articles, Tutorials, Reviews, Issues, Books, Subscribe, Buy now, and Advertise. The main content area features a large banner for 'THE OFFICIAL RASPBERRY PI MAGAZINE' and a 'FREE RASPBERRY PI ZERO KIT' offer. The kit is shown with a price of £20 and a 12-month print subscription. To the right of the banner is a box for 'The MagPi issue 95', which discusses building a classic games console. It includes links for 'Buy now' and 'Subscribe', as well as download links for the App Store and Google Play, and a link to 'Download Free PDF'.

Hello World-Magazine

► helloworld.cc

Hello World erscheint dreimal im Jahr und ist für in Großbritannien ansässige Lehrer, Freiwillige und Bibliothekare kostenlos erhältlich. Alle anderen können kostenlose digitale Kopien unter der Creative-Commons-Lizenz herunterladen. Alternativ sind auch bezahlte Abonnements für die Druckversion erhältlich.

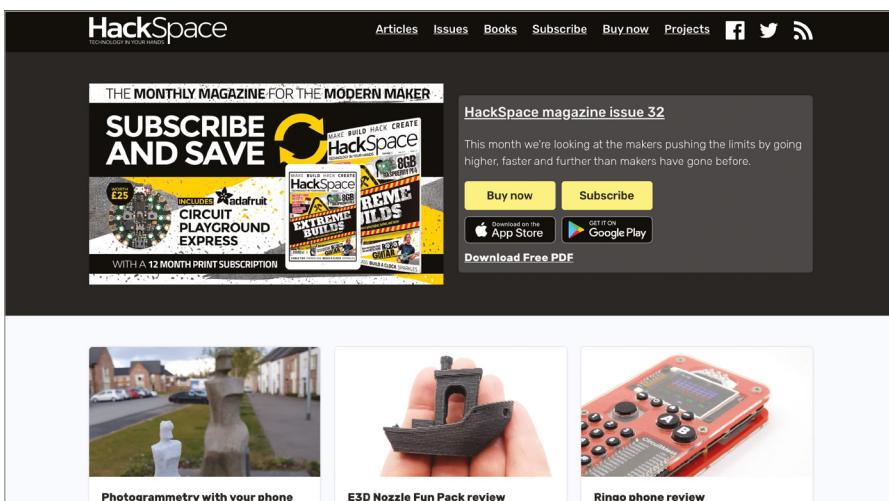


The screenshot shows the website for Hello World Magazine. At the top, there's a logo with the text '(Hello World)' and a yellow circle, followed by the tagline 'The computing and digital making magazine for educators'. To the right are links for 'Issues' and 'Subscribe for free', along with social media icons for Facebook and Twitter. Below this, a large image of 'Hello World issue 13' is displayed, featuring a person working at a desk with a computer and various electronic components. The cover has a pink and blue color scheme with the title 'Hello World' and the subtitle 'LEARNING UNDER LOCKDOWN'. To the right of the image, the text 'Hello World issue 13' is written, followed by a question about how computing education has changed due to the pandemic. A blue button labeled 'SEE WHAT'S INSIDE' is located below the text.

HackSpace-Magazine

► hsmag.cc

Das HackSpace Magazine, das sich an ein breiteres Publikum als The MagPi richtet, wirft mit Hardware- und Software-Rezensionen, Tutorials und Interviews einen Blick auf die Maker-Community. Wenn Sie daran interessiert sind, Ihren Horizont über den Raspberry Pi hinaus zu erweitern, ist das HackSpace Magazin ein großartiger Ausgangspunkt – es ist in gedruckter Form in Supermärkten und Zeitungsläden erhältlich oder kann kostenlos in digitaler Form heruntergeladen werden.



The screenshot shows the website for HackSpace Magazine. At the top, there's a logo with the text 'HackSpace' and a small note 'THE MONTHLY MAGAZINE FOR THE MODERN MAKER'. To the right are links for 'Articles', 'Issues', 'Books', 'Subscribe', 'Buy now', 'Projects', and social media icons for Facebook, Twitter, and RSS. Below this, a large banner for 'SUBSCRIBE AND SAVE' features a circular arrow icon and images of previous magazine covers. To the right, a box for 'HackSpace magazine issue 32' contains the text 'This month we're looking at the makers pushing the limits by going higher, faster and further than makers have gone before.' It includes buttons for 'Buy now', 'Subscribe', and links to download from the App Store and Google Play, as well as a 'Download Free PDF' link. At the bottom, there are three smaller images with captions: 'Photogrammetry with your phone', 'E3D Nozzle Fun Pack review', and 'Ringo phone review'.

Anhang E

Raspberry Pi-Konfigurationswerkzeug



Das Tool Raspberry-Pi-Konfiguration ist ein leistungsstarkes Paket zur Anpassung zahlreicher Einstellungen an Ihrem Raspberry Pi, von den für Programme verfügbaren Schnittstellen bis hin zur Steuerung über das Netzwerk. Für Anfänger kann es allerdings etwas überwältigend sein, deshalb wird dieser Anhang Sie der Reihe nach durch die einzelnen Einstellungen führen und deren Zweck erklären.

Sie können das Raspberry-Pi-Konfigurationswerkzeug aus dem Menü mit dem Raspberry-Symbol unter der Kategorie „Einstellungen“ starten. Es kann auch über die Befehlszeilenschnittstelle, zum Beispiel im Terminal, mit dem Befehl `raspi-config` ausgeführt werden. Die Layouts der Kommandozeilenversion und der grafischen Version sind unterschiedlich, wobei die Optionen in verschiedenen Kategorien erscheinen, je nachdem, welche der Versionen Sie verwenden; dieser Anhang basiert auf der grafischen Version.

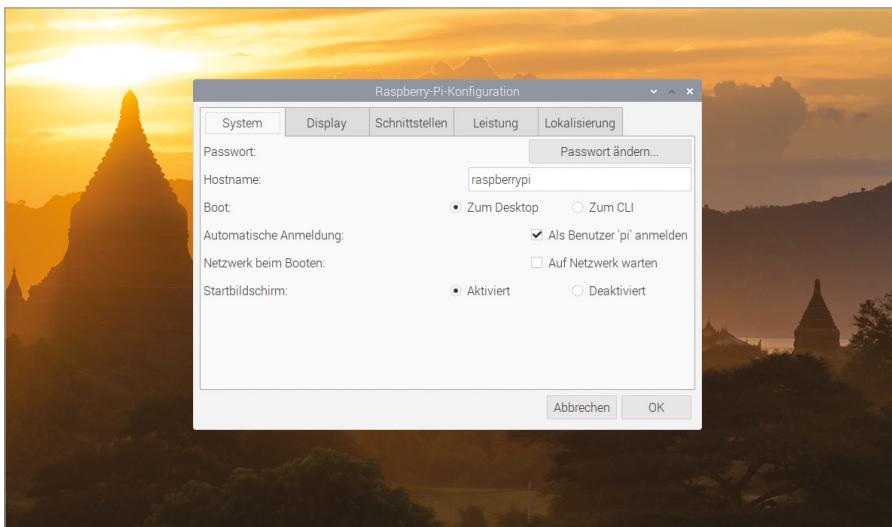
WARNUNG!

Wenn Sie nicht wissen, dass Sie eine bestimmte Einstellung ändern müssen, lassen Sie das Raspberry Pi-Konfigurationswerkzeug am besten in Ruhe. Wenn Sie neue Hardware zu Ihrem Raspberry Pi hinzufügen, wie z. B. eine Audio-HAT oder ein Camera Module (Kameramodul), erfahren Sie in der Anleitung, welche Einstellung Sie ändern müssen. Andernfalls sollten Sie die Standardeinstellungen im Allgemeinen unverändert lassen.



Registerkarte „System“

Die Registerkarte System enthält Optionen, die verschiedene Systemeinstellungen von Raspberry Pi OS steuern.



■ Passwort: Klicken Sie auf die Schaltfläche „Passwort ändern...“, um ein neues Kennwort für Ihr aktuelles Benutzerkonto festzulegen. Standardmäßig ist dies das 'pi'-Konto.

■ Hostname: Der Name, mit dem sich Ihr Raspberry Pi in Netzwerken identifiziert. Wenn Sie mehr als einen Raspberry Pi im selben Netzwerk haben, müssen sie jeweils einen eindeutigen Namen haben.

■ Boot: Wenn Sie diese Einstellung auf „Zum Desktop“ (die Voreinstellung) setzen, wird die vertraute Raspberry Pi OS-Arbeitsoberfläche geladen: Wenn Sie sie auf „Zum CLI“ setzen, wird die Befehlszeilenschnittstelle geladen, wie in **Anhang C, Die Kommandozeile** beschrieben.

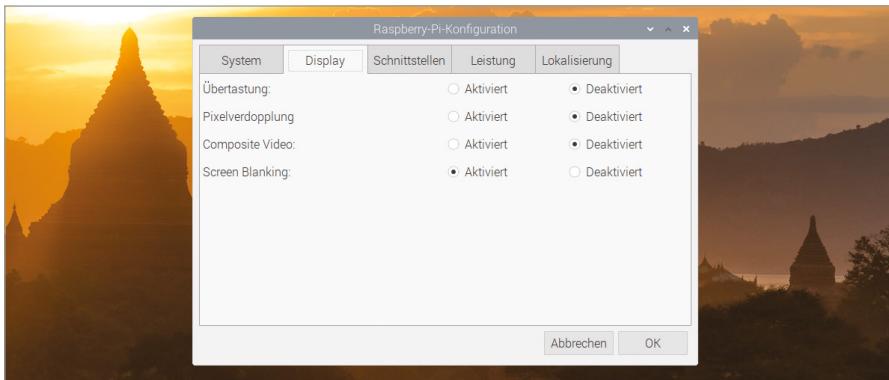
■ Automatische Anmeldung: Wenn „Als Benutzer 'pi' anmelden“ angekreuzt ist (Standardeinstellung), lädt Raspberry Pi OS die Arbeitsoberfläche, ohne dass Sie Ihren Benutzernamen und Ihr Kennwort eingeben müssen.

■ Netzwerk beim Booten: Wenn „Auf Netzwerk warten“ angekreuzt ist, wartet Raspberry Pi OS mit dem Starten, bis es eine funktionierende Netzwerkverbindung hat.

■ Startbildschirm: Wenn hier „Aktiviert“ (die Standardeinstellung) eingestellt ist, werden die Boot-Meldungen von Raspberry Pi OS hinter einem grafischen Startbildschirm versteckt.

Registerkarte „Display“

Die Registerkarte „Display“ enthält Einstellungen, die steuern, wie die Bildschirmausgabe angezeigt wird.



■ **Übertastung:** Diese Einstellung steuert, ob die Videoausgabe vom Raspberry Pi schwarze Balken an den Rändern aufweisen soll, um den Rahmen vieler Fernseher auszugleichen. Wenn Sie schwarze Balken sehen, stellen Sie hier „Deaktiviert“ ein. Wenn nicht, lassen Sie sie auf „Aktiviert“ stehen.

■ **Pixelverdopplung:** Wenn Sie einen hochauflösenden, aber kleinen Bildschirm verwenden, können Sie die Pixelverdopplung einschalten, um alles auf dem Bildschirm größer und damit besser sichtbar zu machen.

■ **Composite Video:** Hiermit wird die Composite Video-Ausgabe („FBAS“) gesteuert, die an der kombinierten Audio-Video (AV)-Buchse verfügbar ist, wenn sie mit einem TRRS-Adapter (Tip-Ring-Ring-Sleeve, vierpoliger Stecker) verwendet wird. Wenn Sie den Composite-Video-Ausgang anstelle von HDMI verwenden möchten, setzen Sie ihn auf „Aktiviert“; andernfalls lassen Sie ihn deaktiviert.

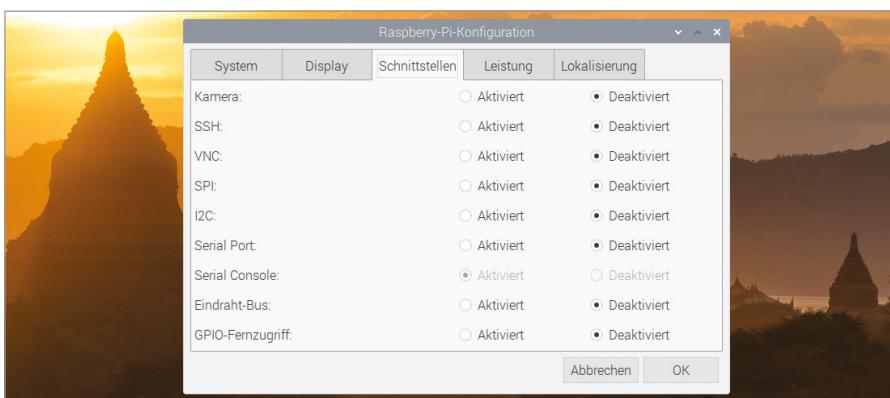
■ **Screen Blanking:** Mit dieser Option können Sie die Bildschirmausblendung (das Timeout, das die Anzeige nach einigen Minuten Inaktivität des Nutzers abschaltet) ein- und ausschalten.

Registerkarte „Schnittstellen“

Die Registerkarte „Schnittstellen“ enthält Einstellungen, die die auf dem Raspberry Pi verfügbaren Hardwareschnittstellen steuern.

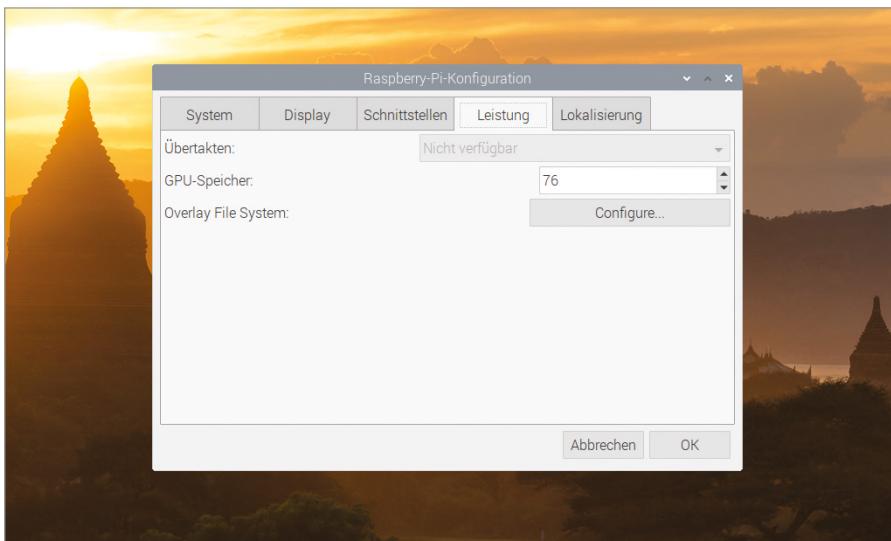
■ **Kamera:** Aktiviert oder deaktiviert das Camera Serial Interface (CSI) für die Verwendung mit einem Raspberry Pi-Kameramodul.

- **SSH:** Aktiviert/deaktiviert die Secure Shell(SSH)-Schnittstelle. Sie ermöglicht es Ihnen, mit Hilfe eines SSH-Clients eine Befehlszeilenschnittstelle auf Ihrem Raspberry Pi von einem anderen Computer in Ihrem Netzwerk aus zu öffnen.
- **VNC:** Aktiviert/deaktiviert die Virtual Network Computing (VNC)-Schnittstelle. Damit können Sie mit Hilfe eines VNC-Clients den Desktop auf Ihrem Raspberry Pi von einem anderen Computer in Ihrem Netzwerk aus sehen und steuern.
- **SPI:** Aktiviert oder deaktiviert das Serial Peripheral Interface (SPI), das zur Steuerung einiger Hardware-Add-ons verwendet wird, die an die GPIO-Pins angeschlossen werden können.
- **I2C:** Aktiviert oder deaktiviert die I²C-Schnittstelle (Inter-Integrated Circuit), die zur Steuerung einiger Hardware-Add-ons verwendet wird, die an die GPIO-Pins angeschlossen werden können.
- **Serial Port:** Aktiviert oder deaktiviert den seriellen Anschluss vom Raspberry Pi, der an den GPIO-Pins verfügbar ist.
- **Serial Console:** Aktiviert oder deaktiviert die serielle Konsole, eine am seriellen Anschluss verfügbare Befehlszeilenschnittstelle. Diese Option ist nur verfügbar, wenn die obige Einstellung für die serielle Schnittstelle auf „Aktiviert“ gesetzt ist.
- **Eindraht-Bus:** Aktiviert oder deaktiviert die Eindraht-Bus-Schnittstelle (1-Wire), die zur Steuerung einiger Hardware-Add-Ons verwendet wird, die an die GPIO-Pins angeschlossen werden können.
- **GPIO-Fernzugriff:** Aktiviert oder deaktiviert einen Netzwerkdienst, der es Ihnen ermöglicht, mithilfe der GPIO Zero-Bibliothek die GPIO-Pins Ihres Raspberry Pi von einem anderen Computer in Ihrem Netzwerk aus zu steuern. Weitere Informationen zu Remote-GPIO finden Sie unter gpiozero.readthedocs.io.



Registerkarte „Leistung“

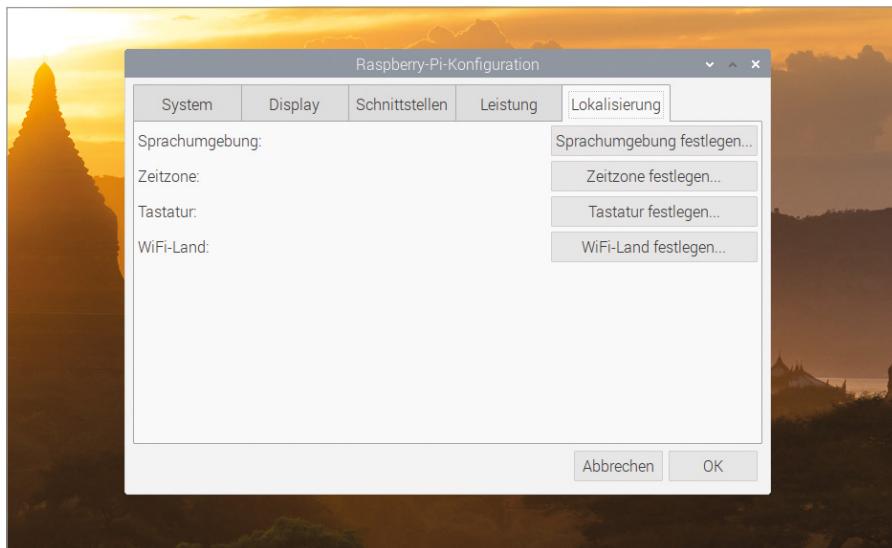
Die Registerkarte „Leistung“ enthält Einstellungen, die steuern, wie viel Speicher verfügbar ist und wie schnell der Prozessor vom Raspberry Pi läuft.



- **Übertakten:** Ermöglicht es Ihnen, aus einer Reihe von Einstellungen zu wählen, die die Leistung Ihres Raspberry Pi erhöhen - das allerdings auf Kosten von einem erhöhten Stromverbrauchs, höherer Wärmeerzeugung und einer möglicherweise verkürzten Gesamtlebensdauer. Nicht für alle Raspberry Pi Modelle verfügbar.
- **GPU-Speicher:** Ermöglicht es Ihnen, die Menge des für den Grafikprozessor vom Raspberry Pi reservierten Speichers einzustellen. Höhere Werte als der Standardwert können die Leistung für komplizierte 3D-Rendering- und GPGPU-Aufgaben (GPGPU = General Purpose GPU) auf Kosten der Reduktion des für das Raspberry Pi OS verfügbaren Speichers verbessern; niedrigere Werte können die Leistung für speicherintensive Aufgaben verbessern, allerdings zu beeinträchtiger Leistung oder Ausfall von 3D-Rendering-, Kamera- und einigen Videowiedergabefunktionen führen.
- **Overlay File System:** Ermöglicht es Ihnen, das Dateisystem von Raspberry Pi zu sperren, sodass Änderungen nur auf einer virtuellen RAM-Disk vorgenommen und nicht auf die microSD-Karte geschrieben werden. So kehren Sie bei jedem Neustart in einen saubereren Zustand zurück.

Registerkarte „Lokalisierung“

Die Registerkarte „Lokalisierung“ enthält Einstellungen, die steuern, in welcher Region Ihr Raspberry Pi betrieben werden soll, einschließlich Tastatur-Layout-Einstellungen.



- **Sprachumgebung:** Ermöglicht Ihnen die Auswahl Ihrer Sprachumgebung, einer Systemeinstellung, die Sprache, Land und Zeichensatz („Zeichenvorrat“) umfasst. Bitte beachten Sie, dass eine hier vorgenommene Änderung der Sprache die Anzeigesprache nur in Anwendungen ändert, für die eine Übersetzung verfügbar ist.
- **Zeitzone:** Ermöglicht Ihnen die Wahl Ihrer lokalen Zeitzone, indem Sie ein Gebiet der Welt gefolgt von der nächstgelegenen Stadt („Standort“) auswählen. Wenn Ihr Raspberry Pi an das Netzwerk angeschlossen ist, aber die Uhr eine falsche Zeit anzeigt, liegt das normalerweise daran, dass die falsche Zeitzone ausgewählt wurde.
- **Tastatur:** Ermöglicht Ihnen die Wahl des Tastaturtyps, der Sprache und des Layouts. Wenn Sie feststellen, dass Ihre Tastatur falsche Buchstaben oder Symbole tippt, können Sie dies hier korrigieren.
- **WiFi-Land:** Ermöglicht es Ihnen, Ihr Land für Zwecke der Rundfunkregulierung einzustellen. Vergewissern Sie sich, dass Sie das Land auswählen, in dem Ihr Raspberry Pi verwendet wird: Die Auswahl eines anderen Landes kann dazu führen, dass keine Verbindung zu nahe gelegenen WLAN-Zugangspunkten hergestellt werden kann. Darüber hinaus kann es auch einen Verstoß gegen das Rundfunkrecht darstellen. Bevor das eingebaute WLAN benutzt werden kann, muss ein Land eingestellt werden.

Anhang F

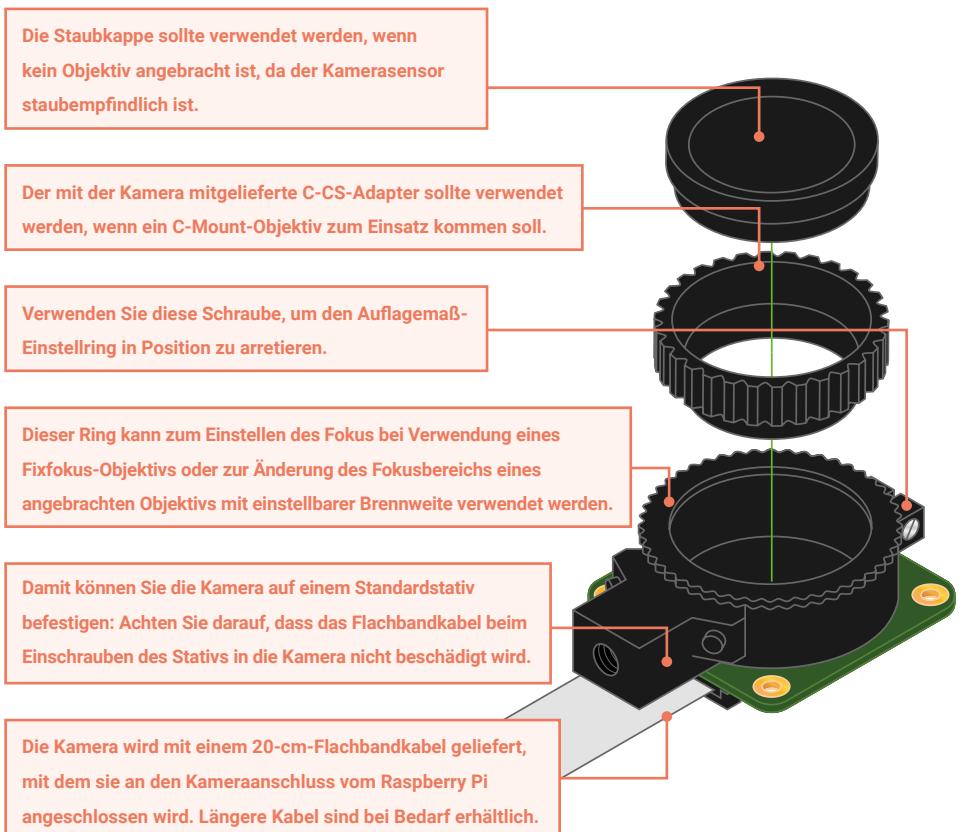
Einrichtung der High Quality Camera

Die High Quality Camera (hochauflösende Kamera, kurz HQ Camera) kann Bilder mit höherer Auflösung als das Standard- Camera Module aufnehmen. Im Gegensatz zu letzterem hat sie keine fest eingebaute Linse. Stattdessen kann sie mit jedem Standard-C- oder CS-Mount-Objektiv verwendet werden; 6-mm- und 16-mm-Objektive können zusammen mit der Kamera gekauft werden, um Ihnen den Einstieg zu erleichtern.

6-mm-CS-Mount-Objektiv

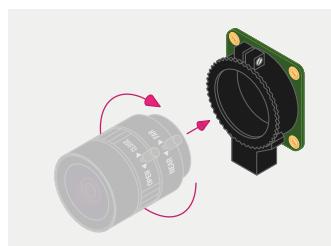
Für die HQ Camera ist ein kostengünstiges 6-mm-Objektiv erhältlich. Dieses Objektiv eignet sich für einfache Fotografie. Es kann auch für die Makrofotografie verwendet werden, da es Objekte auf sehr kurze Entfernungen fokussieren kann.





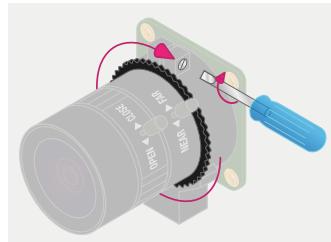
01 Montage des Objektivs

Das 6-mm-Objektiv nutzt den CS-Mount und benötigt daher keinen C-CS-Adapterring (siehe Abbildung oben). Es fokussiert nicht richtig, wenn der Adapter angebracht ist. Entfernen Sie diesen also, falls nötig. Drehen Sie dann das Objektiv im Uhrzeigersinn bis zum Anschlag in den Auflagemaß-Einstellring herein.



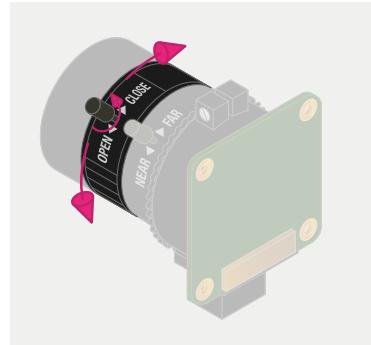
02 Auflagemaß-Einstellring und Feststellschraube

Der Auflagemaß-Einstellring sollte für eine möglichst kurze Auflagemaßlänge ganz eingeschraubt werden. Verwenden Sie die Auflagemaß-Feststellschraube, um sicherzustellen, dass er beim Einstellen der Blende oder des Fokus nicht aus dieser Position herausbewegt wird.



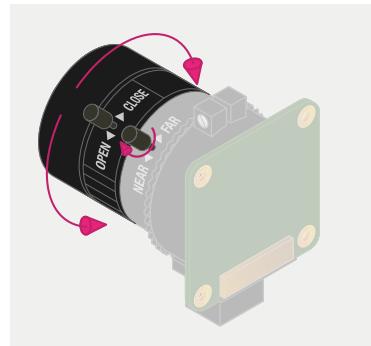
03 Blende

Zum Einstellen der Blende halten Sie die Kamera so, dass das Objektiv von Ihnen weg zeigt. Drehen Sie den mittleren Ring, während Sie den äußeren Ring, der am weitesten von der Kamera entfernt ist, still halten. Drehen Sie im Uhrzeigersinn, um die Blende zu schließen und die Bildhelligkeit zu verringern. Drehen Sie gegen den Uhrzeigersinn, um die Blende zu öffnen. Wenn Sie mit der Lichtstärke zufrieden sind, ziehen Sie die Schraube an der Seite des Objektivs an, um die Blende zu fixieren.



04 Fokus

Arretieren Sie zunächst den inneren Fokusring mit der Aufschrift „NEAR ▲ FAR“, indem Sie seine Schraube anziehen. Halten Sie nun die Kamera so, dass das Objektiv von Ihnen weg zeigt. Halten Sie die beiden äußeren Ringe des Objektivs fest und drehen Sie beide im Uhrzeigersinn, bis das Bild scharf wird – das braucht ca. vier oder fünf ganze Umdrehungen. Zum Einstellen der Schärfe drehen Sie die beiden äußeren Ringe im Uhrzeigersinn, um auf ein nahe gelegenes Objekt zu fokussieren. Drehen Sie sie gegen den Uhrzeigersinn, um auf ein entferntes Objekt zu fokussieren. Es kann sein, dass Sie danach die Blende erneut einstellen müssen.



16mm CS-Mount-Objektiv

Das 16-mm-Objektiv liefert ein qualitativ hochwertigeres Bild als das 6-mm-Objektiv. Es hat einen engen Blickwinkel, der sich besser für die Betrachtung entfernter Objekte eignet.



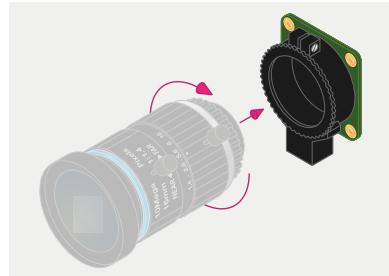
01 Montage des C-CS Adapters

Stellen Sie sicher, dass der C-CS-Adapter, der mit der HQ Camera geliefert wird, am 16-mm-Objektiv angebracht ist. Das Objektiv nutzt den C-Mount, hat also ein längeres Auflagemaß als das 6-mm-Objektiv und benötigt daher den Adapter.



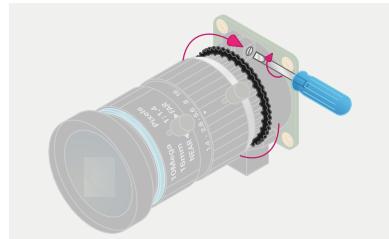
02 Montage des Objektivs an die Kamera

Drehen Sie dann das 16-mm-Objektiv im Uhrzeigersinn bis zum Anschlag in den Auflagemaß-Einstellring herein.



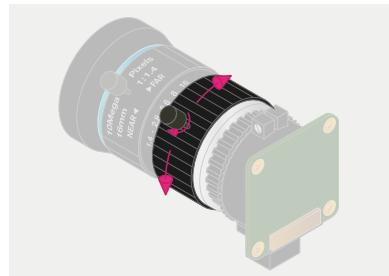
03 Auflagemaß-Einstellring und Feststellschraube

Der Auflagemaß-Einstellring sollte für eine möglichst kurze Auflagemaßlänge ganz eingeschraubt werden. Verwenden Sie die Auflagemaß-Feststellschraube, um sicherzustellen, dass er beim Einstellen der Blende oder des Fokus nicht aus dieser Position herausbewegt wird.



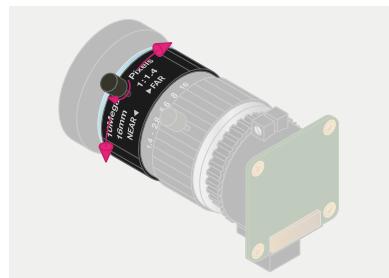
04 Blende

Zum Einstellen der Blende halten Sie die Kamera so, dass das Objektiv von Ihnen weg zeigt. Drehen Sie den Innenring, der der Kamera am nächsten liegt, während Sie die Kamera still halten. Drehen Sie im Uhrzeigersinn, um die Blende zu schließen und die Bildhelligkeit zu verringern. Drehen Sie gegen den Uhrzeigersinn, um die Blende zu öffnen. Wenn Sie mit der Lichtstärke zufrieden sind, ziehen Sie die Schraube an der Seite des Objektivs an, um die Blende zu fixieren.



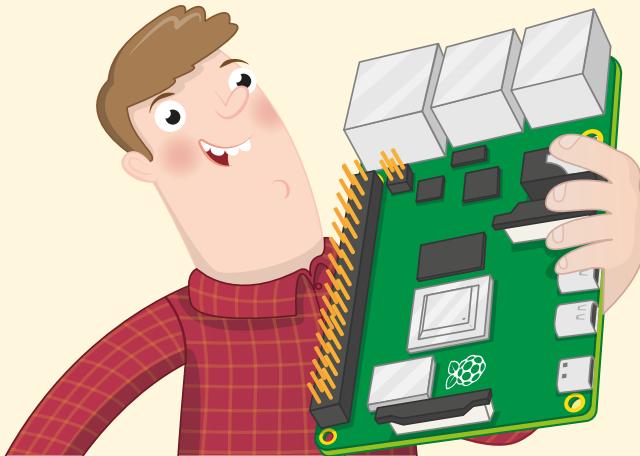
05 Fokus

Um den Fokus anzupassen, halten Sie die Kamera so, dass das Objektiv von Ihnen weg zeigt. Drehen Sie den Fokussierring mit der Aufschrift „NEAR ▲ FAR“ gegen den Uhrzeigersinn, um auf ein nahe gelegenes Objekt zu fokussieren. Drehen Sie ihn im Uhrzeigersinn, um auf ein entferntes Objekt zu fokussieren. Es kann sein, dass Sie danach die Blende erneut einstellen müssen.



Anhang G

Raspberry Pi-Spezifikationen

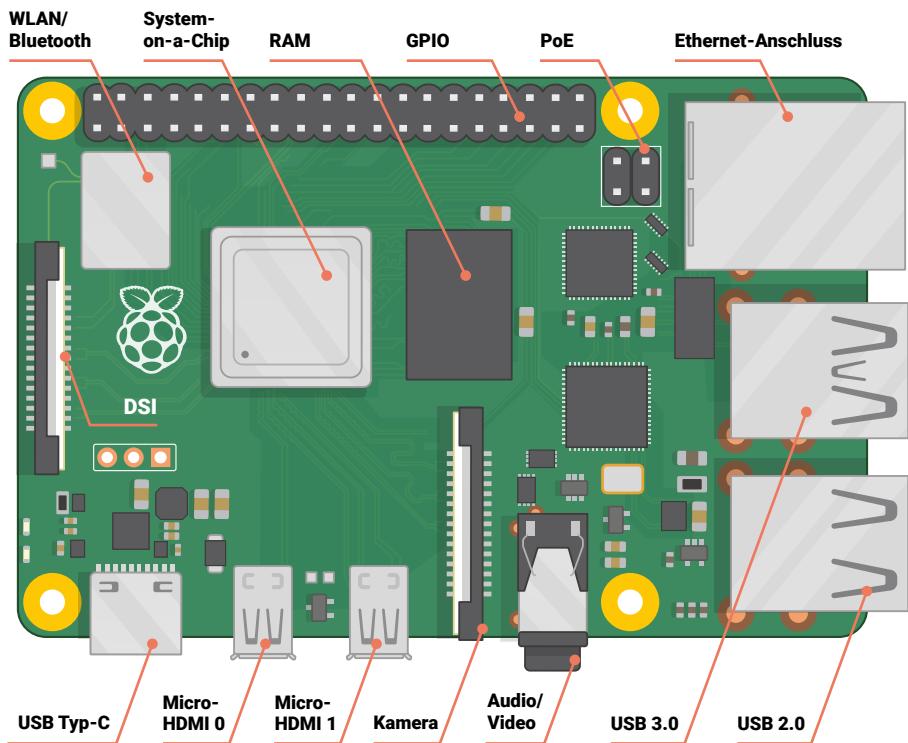


Die verschiedenen Komponenten und Merkmale eines Computers werden als seine Spezifikationen bezeichnet, und ein Blick auf die Spezifikationen gibt Ihnen die Informationen, die Sie zum Vergleich zweier Computer benötigen. Diese Spezifikationen können auf den ersten Blick verwirrend erscheinen. Sie sind sehr technisch und man muss sie nicht kennen, um einen Raspberry Pi zu verwenden, aber neugierige Leser finden sie hier.

Der System-on-a-Chip des Raspberry Pi 4 Modell B und Raspberry Pi 400 ist ein Broadcom BCM2711B0, was Sie auf dem Metaldeckel sehen können, wenn Sie genau genug hinsehen (auf dem Raspberry Pi 4). Dieser verfügt über vier 64-Bit ARM Cortex-A72 Central Processing Unit (CPU)-Kerne mit jeweils 1,5 GHz oder 1,8 GHz (1,5 oder 1,8 Milliarden Zyklen pro Sekunde) und eine Broadcom VideoCore VI (Six / sechs)-Grafikverarbeitungseinheit (GPU) mit 500 MHz (500 Millionen Zyklen pro Sekunde) für Video-Aufgaben und für 3D-Rendering-Aufgaben wie beispielsweise Spiele.

Der System-on-a-Chip ist mit 2 GB, 4 GB oder 8 GB (zwei, vier oder acht Milliarden Byte) – 4 GB bei Raspberry Pi 400 – LPDDR4 (Low-Power Double-Data-Rate 4) RAM (random-access memory, Speicher mit wahlfreiem Zugriff) verbunden, der mit 3200 MHz (3.200 Millionen Zyklen pro Sekunde) läuft. Dieser Speicher wird von dem Zentralprozessor und dem Grafikprozessor gemeinsam genutzt. Der microSD-Kartensteckplatz unterstützt bis zu 512 GB (512 Milliarden Bytes) Speicherplatz.

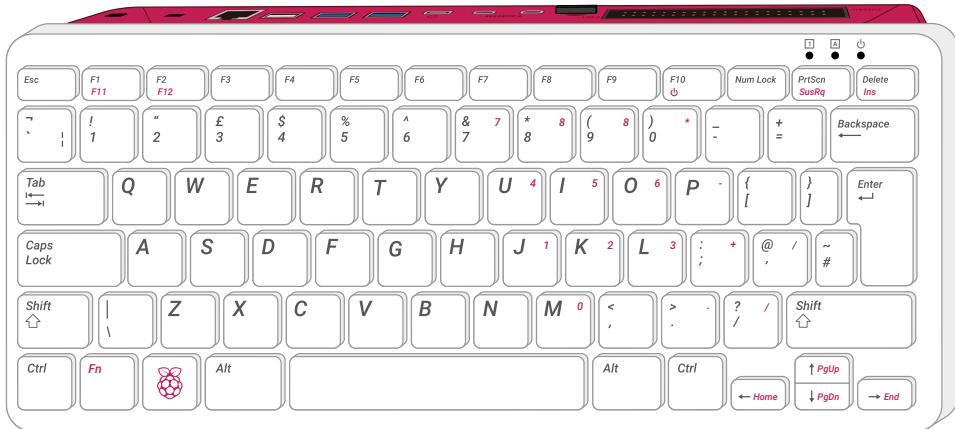
Der Ethernet-Anschluss unterstützt Verbindungsgeschwindigkeiten bis hin zu Gigabit (1000 Mbps, 1000-Base-T), während das WLAN/Bluetoothmodul 802.11ac-WLAN-Netzwerke



unterstützt, die auf den Frequenzbändern 2,4 GHz und 5 GHz funken, sowie Bluetooth 5.0 und Bluetooth Low Energy (BLE)-Verbindungen.

In eine Aufzählung von Aufzählungspunkten zerlegt, sehen die Spezifikationen vom Raspberry Pi 4 wie folgt aus:

- **CPU:** 64 Bit Quad-Core ARM Cortex-A72 mit 1,5 GHz
- **GPU:** VideoCore VI mit 500 MHz
- **RAM:** 2 GB, 4 GB oder 8 GB LPDDR4
- **Netzwerke:** Gigabit-Ethernet, Dual-Band 802.11ac, Bluetooth 5.0, Bluetooth Low Energy
- **Audio-/Video-Ausgänge:** 3.5 mm analoge AV-Buchse, 2 × Micro-HDMI 2.0
- **Peripherie:** 2 × USB 2.0-Anschlüsse, 2 × USB 3.0-Anschlüsse, Serielle Kamerabuchse (CSI), Serielle Anzeigeschnittstelle (DSI)
- **Speicher:** microSD, bis zu 512 GB
- **Stromversorgung:** 5 Volt mit 3 Ampere über USB Typ-C
- **Extras:** 40-polige GPIO-Stiftleiste, Power-over-Ethernet-Kompatibilität (benötigt zusätzliche Hardware)



Die Spezifikationen vom Raspberry Pi 400 sind:

- **CPU:** 64 Bit Quad-Core ARM Cortex-A72 mit 1,8 GHz
- **GPU:** VideoCore VI mit 500 MHz
- **RAM:** 4 GB LPDDR4
- **Netzwerke:** Gigabit-Ethernet, Dual-Band 802.11ac, Bluetooth 5.0, Bluetooth Low Energy
- **Audio-/Video-Ausgänge:** 2 × micro-HDMI 2.0
- **Peripherie:** 1 × USB 2.0 Anschluss, 2 × USB 3.0 Anschlüsse
- **Speicher:** microSD, bis zu 512 GB (16 GB im Lieferumfang)
- **Stromversorgung:** 5 Volt mit 3 Ampere über USB Typ-C
- **Extras:** 40-polige GPIO-Stifteleiste

Anhang H

Raspberry Pi Sicherheits- und Benutzerhandbuch



Raspberry Pi

Entworfen und vertrieben von
Raspberry Pi Trading Ltd
Maurice Wilkes Gebäude
Cowley Road
Cambridge
CB4 0DS
GROSSBRITANNIEN
www.raspberrypi.org

Raspberry Pi Informationen zur Einhaltung
gesetzlicher Vorschriften und zur Sicherheit

Raspberry Pi 4 Modell B
FCC-ID: 2ABC-B-RPI4B
IC-ID: 20953-RPI4B

Raspberry Pi 400
FCC-ID: 2ABC-B-RPI400
IC-ID: 20953-RPI400

WICHTIG: Lesen Sie vor dem Anschluss eines
Netzteils die Installationsanweisungen unter
www.raspberrypi.org/safety

 **WARNUNG:** Krebs und
Reproduktionsschäden –
www.P65Warnings.ca.gov.

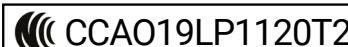
Alle Informationen zur Einhaltung gesetzlicher
Vorschriften und entsprechende Zertifikate finden
Sie unter www.raspberrypi.org/compliance



IFETEL: 2019LAB-ANCE4957
Zertifikatnummer für Raspberry Pi 4 Modell B.



TRA Registrierte Nr.
ER73381/19
Zertifikatnummer für Raspberry Pi 4 Modell B.



Zertifikatnummer für Raspberry Pi 4 Modell B.



NTC
Typ genehmigt:
Nr. ESD-GEC-1920098C
Zertifikatnummer für Raspberry Pi 4 Modell B.



TA-2019/750 GENEHMIGT
Zertifikatnummer für Raspberry Pi 4 Modell B.



Die übernommenen Marken HDMI, HDMI High-Definition
Multimedia Interface und das HDMI-Logo sind Marken oder
eingetragene Marken von HDMI Licensing Administrator, Inc.
in den Vereinigten Staaten und anderen Ländern

DAS OFFIZIELLE

Raspberry Pi

-Handbuch für Anfänger

Der Raspberry Pi ist ein kleiner, cleverer, in Großbritannien gebauter Computer, der voller Potenzial steckt. Der Raspberry Pi wird mit derselben Technologie hergestellt, die auch in Smartphones verbaut ist, und wurde entwickelt um Ihnen zu helfen, Programmieren zu lernen, zu entdecken, wie Computer funktionieren, und Ihre eigenen fantastischen Dinge zu bauen. Dieses Buch wurde geschrieben, um Ihnen zu zeigen, wie wirklich einfach es ist, anzufangen.

Geballtes praktisches Wissen für Sie:

- › Richten Sie Ihren Raspberry Pi ein, installieren Sie sein Betriebssystem und nehmen Sie diesen vielseitig einsetzbaren Computer in Betrieb.
- › Beginnen Sie eigene Programmierprojekte, mit Schritt-für-Schritt-Anleitungen unter Verwendung der Programmiersprachen Scratch 3 und Python.
- › Experimentieren Sie mit elektronischen Bauteilen und haben Sie Spaß daran, tolle Projekte zu erschaffen.

raspberrypi.org



9 781912 047956

