Dein Pi benötigt natürlich auch eine Internetverbindung. Am leichtesten ist es, wenn du deinen Pi einfach über ein Ethernet-Kabel an das Internet anschliesst. Möchtest du deinen Pi kabellos mit dem Internet verbinden, findest du eine genaue Erklärung auf unserer Online-Plattform.

Was ist Python?

Python ist englisch und bedeutet auf deutsch übersetzt so viel wie Python (man spricht es halt anders aus). Die Python ist eine Schlange aus der Überfamilie der Pythonoidea. Python (jetzt bitte in englisch aussprechen. In etwa so: „Pei@%Son") ist aber auch eine super dufte Programmiersprache. Python ist nämlich sehr leicht zu lernen und doch sehr funktionsreich. Sie eignet sich also prima für Anfänger und kann später trotzdem noch für umfangreiche und professionelle Projekte eingesetzt werden.

Es ist wichtig, einige Elemente z.B. des Aufbaues zu verstehen, bevor man anfängt mit seinem Programmcode die Tastatur heißlaufen zu lassen. Betrachen wir also mal eine Zeile Python-Code in freier Wildbahn:

```
01| print(„Hallo. Ich bin Python")
```

Aha. So sieht also eine Code-Zeile aus. Ach guck mal, da gesellt sich noch eine Zeile hinzu:

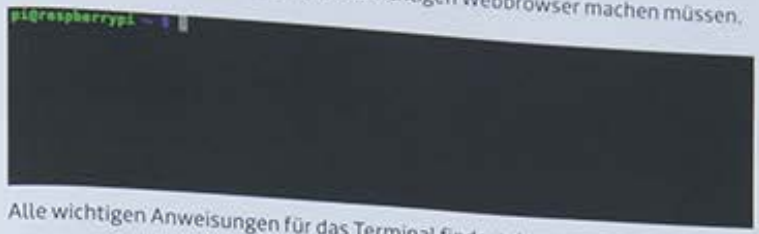
```
01| print(„Hallo. Ich bin Python")
02| lieblingsEssen = „Pommes und Mayo"
```

Da Programmierer sich das Leben gerne etwas leichter machen, achten sie nicht immer auf korrekte Rechtschreibung. Wörter werden zum Beispiel am Anfang mal klein geschrieben und an diese wird dann ohne Leertaste etwas Neues dran gehängt. Wundere dich also nicht, wenn dir die Rechtschreibung im Programmcode mal komisch vorkommt.

Achte bitte auch darauf, dass ein Programm von oben nach unten durchläuft. Also von der ersten Zeile bis zur letzen. Wenn du mehr darüber erfahren möchtest, wie der Python-Interpreter funktioniert, schau auf unserer Lernplattform vorbei. Viele Zeilen Programmcode ergeben am Ende ein tolles Programm. Was wohl das zweizeilige Programm von dieser Seite alles bewerkstelligen kann? Das lernst du in den folgenden Kapiteln. Viel Spaß!

Wie orientiere ich mich auf dem Pi?

Wie alle coolen Gangster aus Filmen werden wir auch hier das sogenannte Terminal benutzen. Das ist dieses Ding, das alles in Form von Text anzeigt und das alle Eingaben auch nur in Textform versteht. Im Grunde ist es das Betriebssystem, das du von normalen PCs kennst, nur ohne die graphische Benutzeroberfläche. Das machen wir, damit wir uns ganz auf unseren Programmcode und unsere bösen Welteroberungspläne konzentrieren können und uns keine Sorgen über Themen wie die Wahl des richtigen Webbrowser machen müssen.



Alle wichtigen Anweisungen für das Terminal findest du auf dem Cheat-Sheet, das dem Heft beiliegt. Aber für den Anfang werden wir die wichtigsten der wichtigen Anweisungen einmal erläutern und ausprobieren.

Starte also deinen Pi und melde dich an, damit ich dir alles wirklich Wichtige für den Anfang erzählen kann... Fertig? Cool! Der standardmäßige **Benutzername** lautet dabei „pi" und das **Passwort** „raspberry".

Direkt nachdem du dich angemeldet hast, befindest du dich im Home-Verzeichnis des Pi. Das kannst du dir wie einen großen Ordner vorstellen, in dem du unter anderem Dateien erstellen kannst, die deinen Programmcode oder deinen Essensplan für die nächste Woche enthalten. Und damit du auch immer Ordnung halten kannst - so wie Mama und Papa sich das von deinem Zimmer auch wünschen - kannst du hier auch Unterordner erstellen, in denen du dann wiederum Dateien speichern kannst.

Ich zeige dir mal, wie du einen neuen Ordner erstellst, in dem du all deine Programme aus diesem Heft speichern kannst. Das geht ganz leicht, gib einfach folgende Zeile ein:

```
$ mkdir testProgramme
```

Und wenn du jetzt...

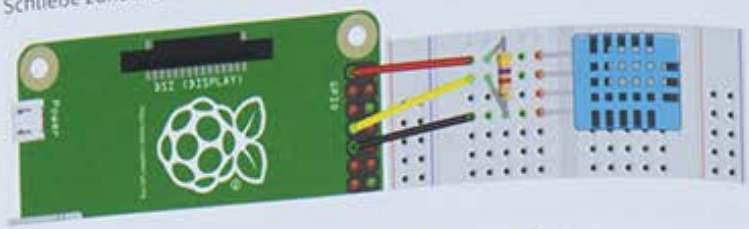
```
$ cd testProgramme
```

Temperatur und Luftfeuchtigkeit messen

Bis jetzt hast du schon LEDs zum Leuchten gebracht, Schleifen mit Programmcode gebunden, Daten in Variablen gespeichert. Wir wollten hier noch einmal anmerken, wie cool du bist und wie viel praktische und tolle Dinge du schon gelernt hast.

In der Zeit, in der du dieses Buch höchst konzentriert durch gearbeitet hast, hat dein Kopf bestimmt ein wenig begonnen zu dampfen. Das Resultat eines dampfenden Kopfes ist eine erhöhte Umgebungstemperatur. Diese Umgebungstemperatur wirst du im folgenden Kapitel zu messen lernen, damit du rechtzeitig Pause machen kannst, bevor es zu warm wird.

Schließe zunächst, wie immer, alles richtig an den Pi an.



Anschlüsse am Pi	Anschlüsse am Sensor
3,3 V	VDD (Das linke Bein)
GPIO zu 4,7 Kohm (gelb|lila|rot)	Data (Das Bein in der Mitte)
GND (Ground)	GND (Das rechte Bein)

Da nicht jeder Menschen mit seinem Raspberry Pi die Temperatur messen möchte, ist die Bibliothek und die benötigte Software leider nicht vorinstalliert. Du musst diese Bibliothek also selber aus dem Internet laden. Wenn dein Pi schon eine Internetverbindung hat, dann mach hier einfach weiter. Wenn dein Pi noch keine Internetverbindung hat, kannst du den Pi einfach über ein Lan-Kabel an den Router ins Netz bringen. Mehr wie immer online!

Alles im Lot? Mit der folgenden Anweisung kannst du dir die benötigte Software aus dem Internet laden:

```
$ sudo apt-get update
$ sudo apt-get install build-essential python3-dev
```

Und mit dem Befehl lädst du dir die Bibliothek runter:

```
$ git clone https://github.com/coding-world/Python_DHT.git
$ cd Python_DHT
$ sudo python3 setup.py install
$ cd
```

Jetzt ist alles fertig eingerichtet und du kannst mit dem Programmieren anfangen. Wie immer einfach eine neue Datei anlegen und los:

```
$ sudo nano dht11_einfach.py
```

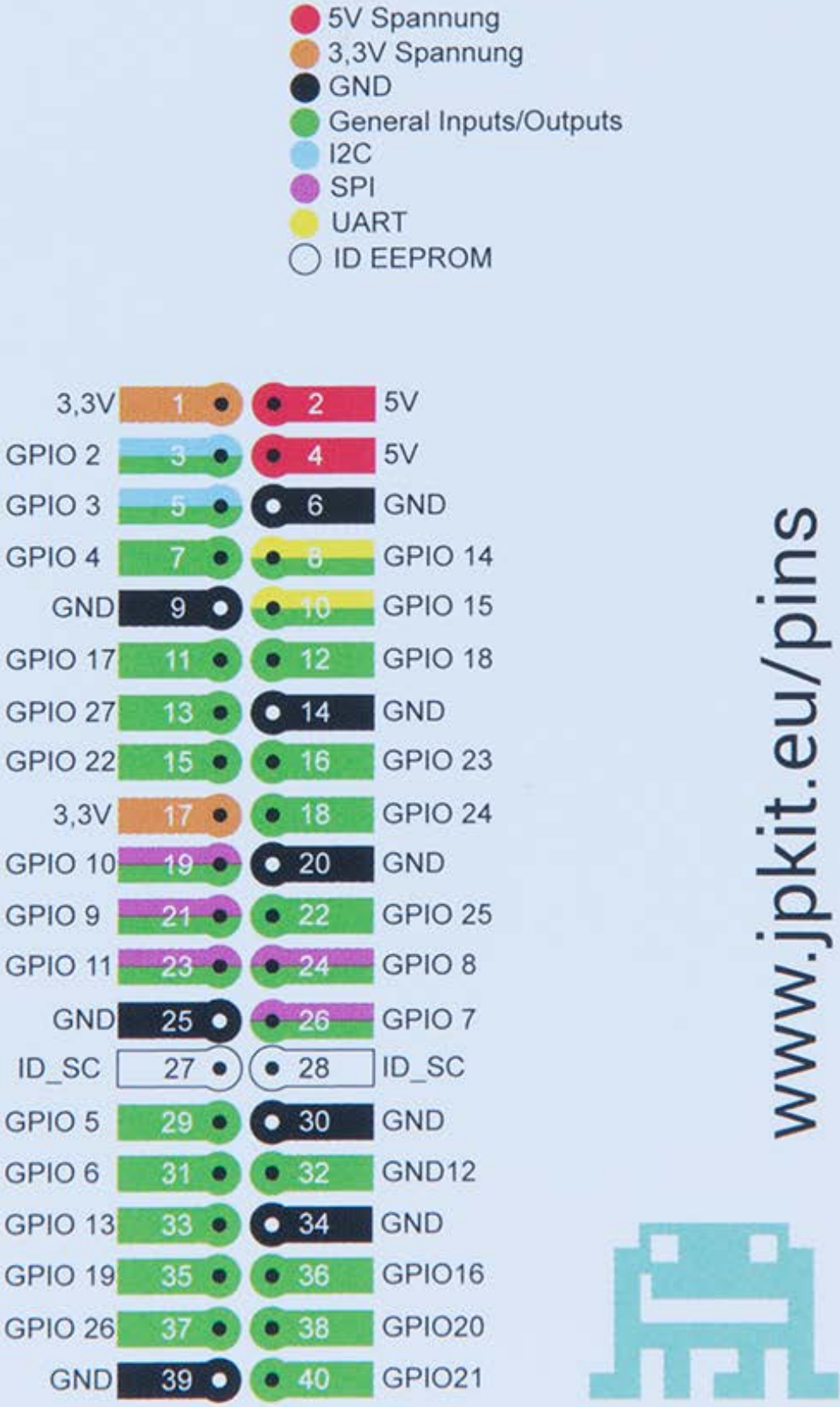
Das unbeschriebene Blatt mit Code füllen...

```
01| import Python_DHT
02|
03| sensor = Python_DHT.DHT11
04| pin = 4
05| feuchtigkeit, temperatur = Python_DHT.read_retry(sensor, pin)
06| print("Temperatur = "+str(temperatur)+ "C Feuchtigkeit = "+str( feuchtigkeit)+"%")
```

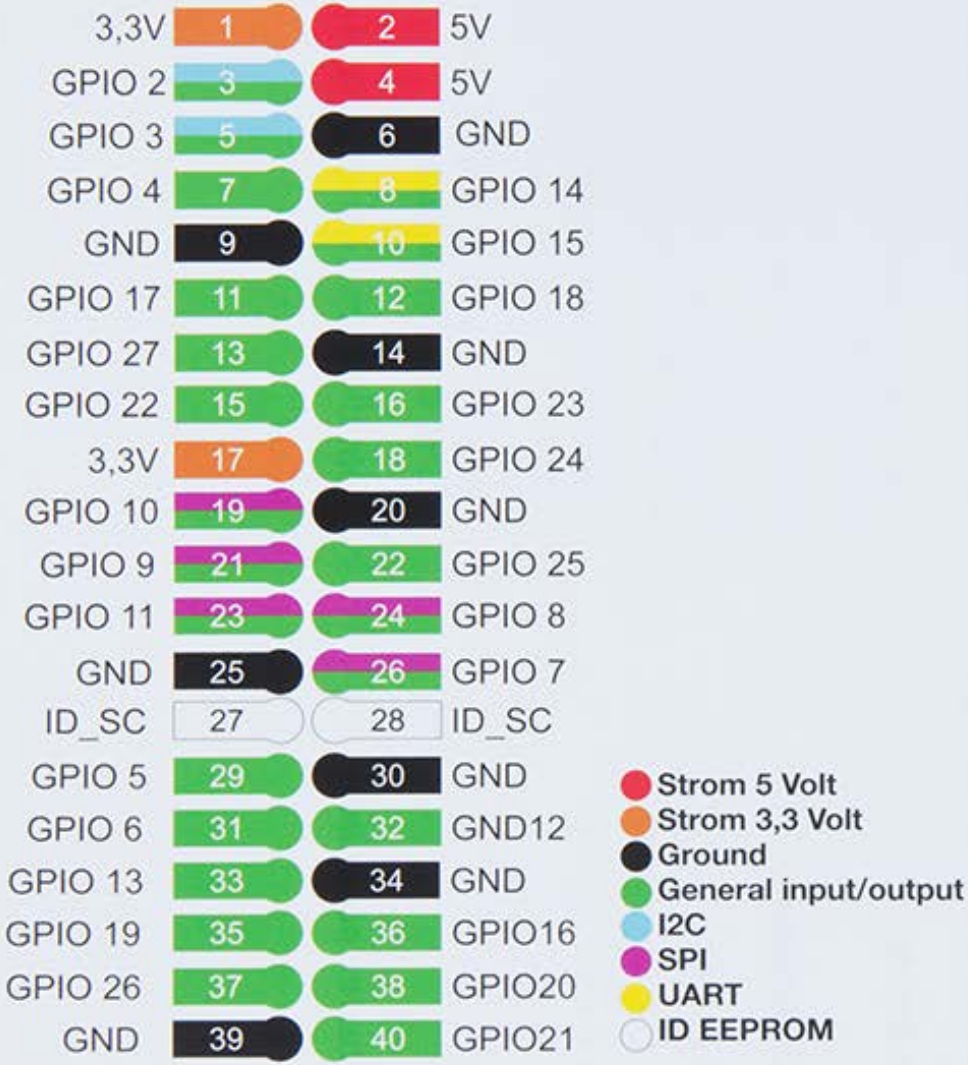
Am Anfang binden wir wie immer die benötigte Bibliothek ein. In **Zeile 3** erstellen wir eine Variable namens **sensor** und speichern in ihr einfach nur die Art des Sensors, den wir auslesen möchten. In unserem Fall ist das der der DHT11.

In **Zeile 4** legen wir den Pin am Pi fest, den wir nutzen möchten.

In **Zeile 5** lernst du wieder etwas Neues. Man kann nämlich auch mehrere Variablen durch ein Komma getrennt hintereinander erzeugen und ihnen hinter dem Gleichzeichen wieder durch Kommata getrennt verschiedene Werte zuweisen. In diesem Fall geben wir mit der Funktion **Python_DHT.read_retry()** Bescheid, welchen Sensor und welchen Pin wir nutzen wollen.


5V Spannung
3,3V Spannung
GND
General Inputs/Outputs
I2C
SPI
UART
ID EEPROM
3,3V 1 2 5V
GPIO 2 3 4 5V
GPIO 3 5 6 GND
GPIO 4 7 8 GPIO 14
GND 9 10 GPIO 15
GPIO 17 11 12 GPIO 18
GPIO 27 13 14 GND
GPIO 22 15 16 GPIO 23
3,3V 17 18 GPIO 24
GPIO 10 19 20 GND
GPIO 9 21 22 GPIO 25
GPIO 11 23 24 GPIO 8
GND 25 26 GPIO 7
ID_SC 27 28 ID_SC
GPIO 5 29 30 GND
GPIO 6 31 32 GND12
GPIO 13 33 34 GND
GPIO 19 35 36 GPIO16
GPIO 26 37 38 GPIO20
GND 39 40 GPIO21
www.jpkit.eu/pins

Die Pinbelegung des Raspberry Pi



Fehlerbehebung für Anfänger

Funktioniert ein Programm mal nicht, dann ist irgendetwas schief gelaufen. Für Neulinge ist es manchmal schwer einen Fehler zu finden, denn bei all dem Programmcode weiss man gar nicht wo man anfangen soll zu suchen. Aber solange die Programmfehler ihren Ursprung nicht in einem gewaltigen Knick in der Logik haben, kann man den Schaden meist sehr leicht beheben. Halte dich an unseren Plan für Fehlerbehebung und alles wird reibungslos funktionieren.

1. Hast du alles richtig geschrieben? Prüfe alle Variablen- und Funktionsnamen auf ihre Richtigkeit. Beachte dabei die Groß- und Kleinschreibung. Der Python-Interpreter ist nämlich genau so streng wie ein Deutschlehrer

2. Nach jeder **if-Anweisung** und **while-Schleife** kommt immer ein Doppelpunkt!

3. Ist der Code nach einer **if-Anweisung** oder **while-Schleife** immer richtig mit der Tabulatortaste eingerückt?

3. Hast du alle Klammern und Kommata richtig gesetzt?

4. Sind alle wichtigen Bibliotheken eingebunden?

Die wichtigsten Terminal Befehle

Befehl	Bedeutung
ls	Listet alle Inhalte in einem Ordner auf
cd ***Ordnername***	Wechselt in den angegebenen Ordner
cd ..	Wechselt in den übergeordneten Ordner
rm ***DateiName***	Löscht die angegebenen Datei
cp ***alteDatei.py neueDatei.py***	Kopiert eine alte Datei in eine neue
mkdir ***Ordnername***	Erstellt einen Ordner mit angegeben Namen
sudo nano ***DateiName.py***	Erstellt eine Datei mit angegebenen Namen und öffnet den Nano-Editor
sudo python3 ***DateiName.py***	Führt die angegebene Python Datei aus
sudo ***Befehl***	Führt den angegebenen Befehl mir Root-Rechten aus
cat ***DateiName***	Zeigt den Inhalt einer Datei an
man ***Befehl***	Zeigt Informationen zu dem angegebenen Befehl an
Str + C oder ctrl + C	Beendet ein Programm oder Befehl im Terminal
clear	Löscht die letzen Terminal Befehle und räumt das Terminal auf
Pfeiltasten ***oben/unten***	Zeigt die letzen benutzen Befehle an