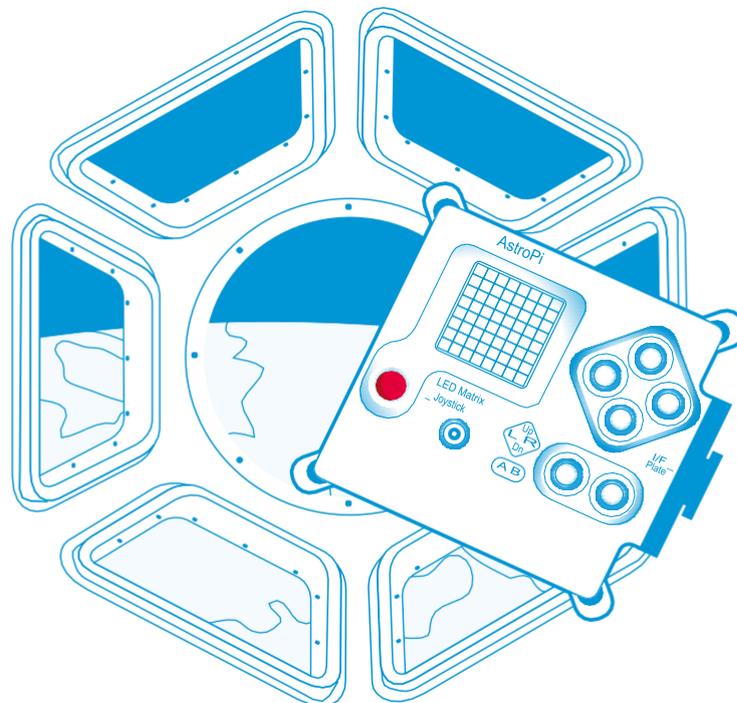
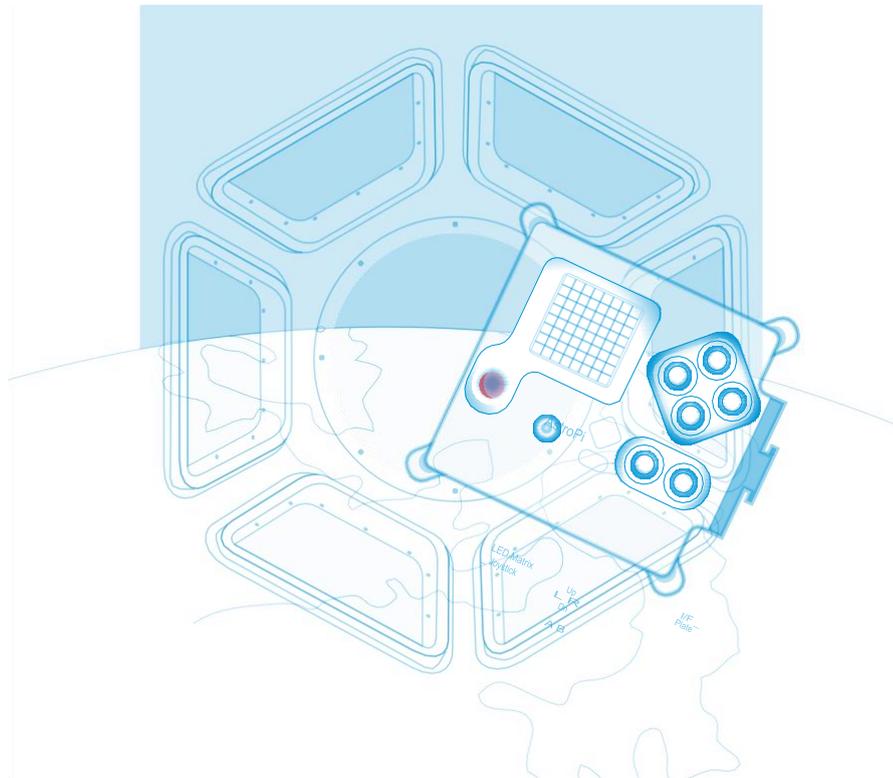


# Lehren mit dem All

## → ERSTE SCHRITTE MIT ASTRO PI

Programmiersprache mithilfe von Raspberry Pi  
kennnenlernen





European Astro Pi Challenge

Seite 3

Aufgabe 1 – Einführung zu Raspberry Pi

Seite 4

Aufgabe 2 – Einführung zu Python

Seite 6

Aufgabe 3 – Ein Quiz mit Python

Seite 8

Aufgabe 4 – Geometrische Formen mit Python zeichnen

Seite 10

Lehren mit dem All – Erste Schritte mit Astro Pi | T05.1b  
[www.esa.int/education](http://www.esa.int/education)

An ESA Education production  
Copyright2017 © European Space Agency

Überarbeitet von ESERO Germany

## Willkommen zur European Astro Pi Challenge!

Astro Pi ist der Name eines kleinen Computers, der von der Raspberry Pi Foundation in Zusammenarbeit mit der britischen Raumfahrtbehörde (UK Space Agency; UKSA) und der Europäischen Weltraumorganisation (ESA) entwickelt wurde.

Es gibt zwei ganz besondere Astro-Pi-Computer. Sie heißen Ed und Izzy und wurden extra für einen Flug ins Weltall gebaut. Beide befinden sich nun auf der Internationalen Raumstation ISS und stehen Schülerinnen und Schülern wie euch zur Verfügung.

In dieser ersten Lektion werdet ihr euren Raspberry Pi zum ersten Mal zusammenbauen und erste Befehle in der Programmiersprache lernen, die ihr zur Programmierung eures Astro Pi benötigt. So könnt ihr selbst programmieren und ein wissenschaftliches Experiment entwickeln, das dann vielleicht auf der ISS durchgeführt wird.

### Versuchsmaterial

- Astro-Pi-Bausatz
- Monitor
- USB-Tastatur
- USB-Maus

Abbildung A1



↑ Astro Pi logo

## Aufgabe 1 – Einführung zu Raspberry Pi

Raspberry Pi ist ein Computer, welcher den Computern die ihr bereits kennt, ähnelt, aber viel kleiner und kompakter ist. Mit einem Raspberry Pi könnt ihr im Internet surfen oder eine E-Mail schicken oder ihr könnt ihn in einen MP3-Player verwandeln, damit die Astronauten auf der ISS etwas entspannen können. Doch das war noch längst nicht alles! Mit dem Raspberry Pi kann man auch wissenschaftliche Experimente durchführen. Dabei werden mit diesem Computer Daten erfasst, die ihr später auswerten könnt – wie richtige Weltraumforscher!

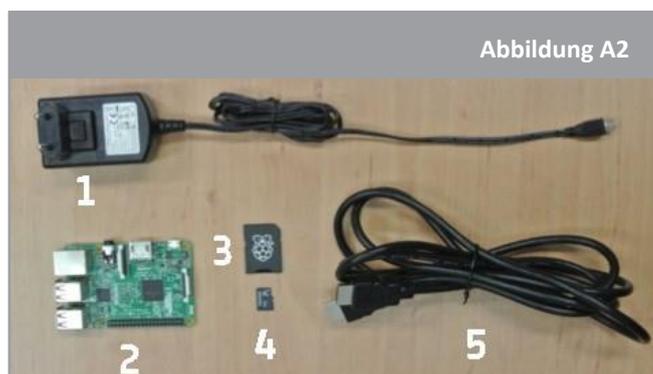
In dieser Aufgabe baut ihr zunächst euren Raspberry Pi zusammen, dann könnt ihr ausprobieren, was man damit alles machen kann.

### Übung

1. Zu Beginn eurer Astro-Pi-Mission benötigt ihr spezielle Komponenten, die ihr in eurem Astro-Pi-Bausatz findet. Außerdem benötigt ihr einen Monitor, eine Tastatur und eine Maus. In Abbildung A2 seht ihr, welche Komponenten ihr benötigt.

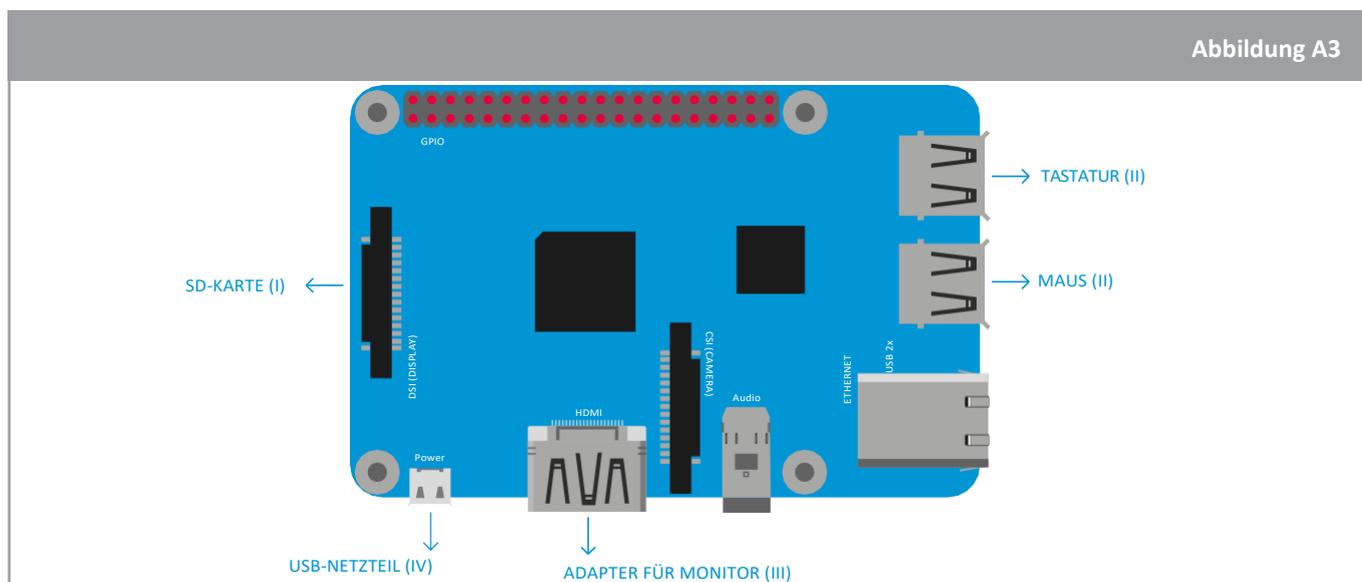
Diese Teile bilden die Hardware. Benennt die einzelnen Teile des in Abbildung 2 gezeigten Raspberry- Pi-Computers (MicroSD-Karte, SD-Kartenadapter, Raspberry Pi 3, Netzteil, HDMI-Kabel).

1. \_\_\_\_\_
2. \_\_\_\_\_
3. \_\_\_\_\_
4. \_\_\_\_\_
5. \_\_\_\_\_



↑ Bauteile des Raspberry Pi

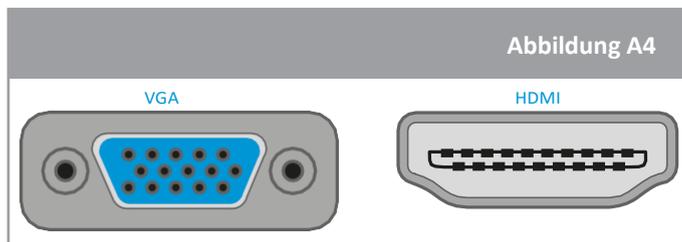
2. Nun baut ihr euren Raspberry Pi zusammen! Folgt dazu den Anweisungen I bis IV, die Abbildung A3 dient euch zur Orientierung.



↑ Übersichtsabbildung für den Bau des Raspberry Pi

- I. Nehmt die SD-Karte aus dem SD-Kartenadapter. Setzt sie in den SD-Kartensteckplatz am Raspberry Pi. Sie passt nur in einer Richtung hinein.
- II. Steckt die Stecker eurer Tastatur und Maus in die USB-Anschlüsse.
- III. Schließt das HDMI-Kabel vom Monitor (oder Fernseher) an den HDMI-Anschluss am Raspberry Pi an und schaltet den Monitor ein.

TIPP: Dies ist wahrscheinlich der schwierigste Schritt beim Zusammensetzen eures Raspberry Pi. Achtet darauf, den richtigen Eingang zu verwenden (z. B. HDMI, DVI usw.). Falls euer Monitor nur einen VGA-Anschluss hat (siehe Abbildung A4), müsst ihr den HDMI-auf-VGA-Adapter (aus dem Bausatz) verwenden.



↑ VGA- und HDMI-Anschlüsse

- IV. Schließt das Micro-USB-Stromkabel an.

Dadurch wird der Raspberry Pi eingeschaltet und hochgefahren. Dies kann bis zu fünf Minuten dauern. Der Raspberry Pi sollte aufleuchten und auf dem Monitor sollte Text erscheinen. Nachdem der Raspberry Pi hochgefahren ist und kein Text mehr über den Bildschirm flimmert, solltet ihr die grafische Oberfläche aus Abbildung A5 sehen.



↑ Die Desktop-Umgebung bzw. grafische Benutzeroberfläche

3. Nun könnt ihr mit der Raspberry-Pi- Software arbeiten. Ihr könnt zum Beispiel einen Webbrowser öffnen und im Internet surfen. Klickt dazu oben am Bildschirm auf das Hauptmenü (das Himbeer-Symbol) und wählt Internet > Chromium Web Browser (für den Internetzugang benötigt ihr WLAN). Versucht, im Internet die aktuelle Position der ISS zu finden. Notiert sie euch.

4. Macht euch mit der Raspberry-Pi-Software vertraut, indem ihr verschiedene Optionen aus dem Programmiermenü auswählt.

Könnt ihr die Unterschiede zwischen Hardware und Software erklären?

---



---



---

**Hinweis:** Zieht zum Ausschalten des Raspberry Pi nicht einfach den Stecker aus der Steckdose. Wenn ihr mit eurer Arbeit am Raspberry Pi fertig seid, müsst ihr ihn herunterfahren. Klickt dazu im Hauptmenü auf die entsprechende Schaltfläche.

### Weiterführende Aufgabe

Nicht alles ist so, wie es scheint! Erstellt eine Liste mit digitalen Geräten, die eurer Meinung nach Computer sind und die ihr im Alltag benutzt. Vergleicht die Liste mit der eurer Mitschülerinnen und Mitschüler und diskutiert, was einen Computer ausmacht.

## Aufgabe 2 – Einführung zu Python

Damit ihr über den Raspberry Pi mit den Astronauten auf der ISS kommunizieren könnt, müsst ihr ihm die nötigen Anweisungen geben. Wie jeder Computer tut er nur das, was ihr ihm beibringt oder wofür ihr ihn programmiert. Dazu müsst ihr eine Programmiersprache lernen (genauso wie ihr eure eigene Sprache lernen musstet, um mit den Menschen in eurem Umfeld kommunizieren zu können). In dieser Aufgabe lernt ihr die grundlegenden Vokabeln der Python-Sprache und schreibt euer erstes Programm.

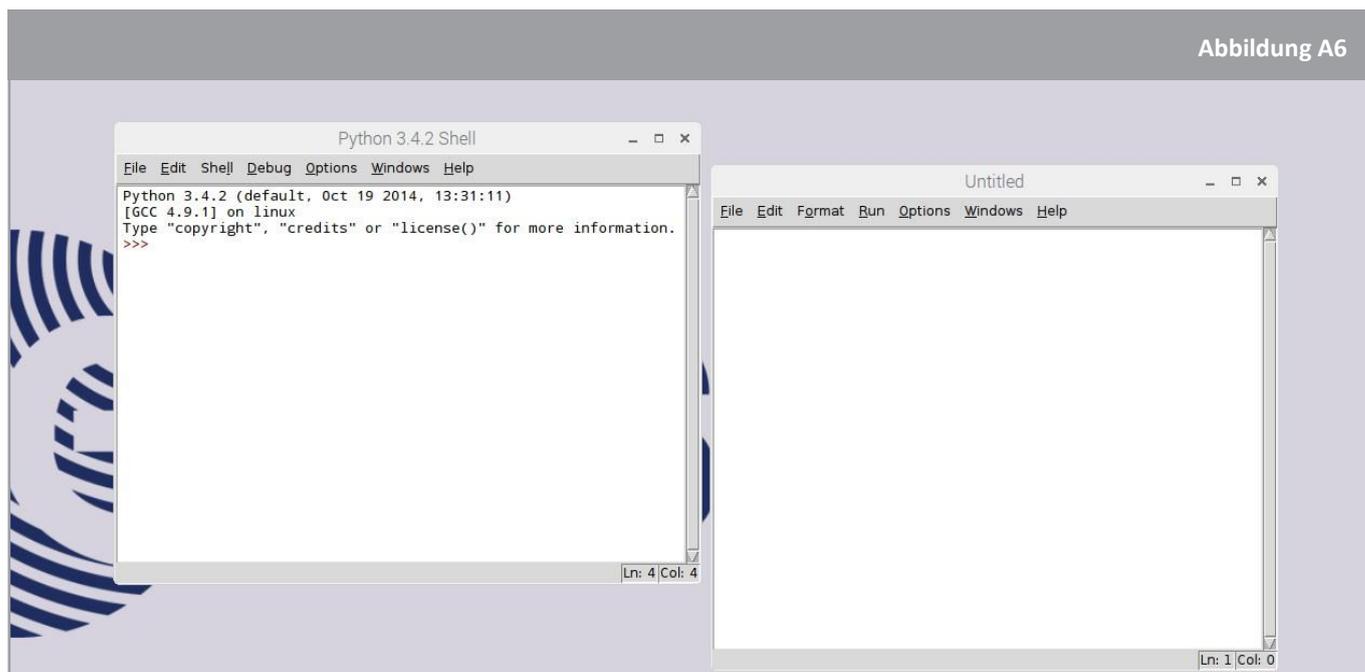
### Schon gewusst?

Astronauten müssen auch weitere Sprachen lernen, bevor sie ins All fliegen dürfen. So müssen beispielsweise alle Astronauten Russisch lernen, weil die Steuerhebel und -tasten der Sojus-Raumkapsel russisch beschriftet sind. Derzeit ist die Sojus, das einzige Raumfahrzeug, mit dem Menschen zur ISS und zurück zur Erde fliegen können. Das russische Wort Sojus bedeutet „Union“. Obwohl die Sojus bereits zu Beginn des Raumfahrtzeitalters von der Sowjetunion entwickelt wurde, wird sie noch heute gebaut, jedoch mit einigen Änderungen.



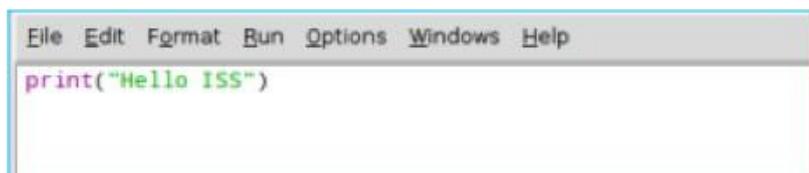
### Übung

- Um Python im Raspberry Pi verwenden zu können, klickt auf das Hauptmenü oben am Bildschirm (Himbeer-Symbol) und wählt Programming > Open Python 3(IDLE) (Programmierung > Python 3(IDLE) öffnen]). Dadurch wird ein Ausgabefenster namens „Python Shell“ geöffnet (siehe Abbildung A6). Ihr könnt euer Programm direkt in dieses Shell-Fenster eingeben, aber es ist einfacher, in einem Bearbeitungsfenster (Editor) zu arbeiten. Wählt zum Öffnen eines neuen Fensters File> New File[Datei> Neue Datei]. Nun solltet ihr zwei Fenster sehen. Eines ist zum Schreiben eures Programms (das Editor-Fenster) und ein zweites ist für die Ausgabe, in ihm werden z. B. Ergebnisse und Fehlermeldungen angezeigt (Python Shell). Ordnet die Fenster nebeneinander an.



↑ [Python Shell \(links\)](#) und [Editor-Fenster \(rechts\)](#)

2. Ihr seid nun bereit, in Python zu programmieren. Gebt den folgenden Code in das Editor-Fenster ein.



```
File Edit Format Run Options Windows Help
print("Hello ISS")
```

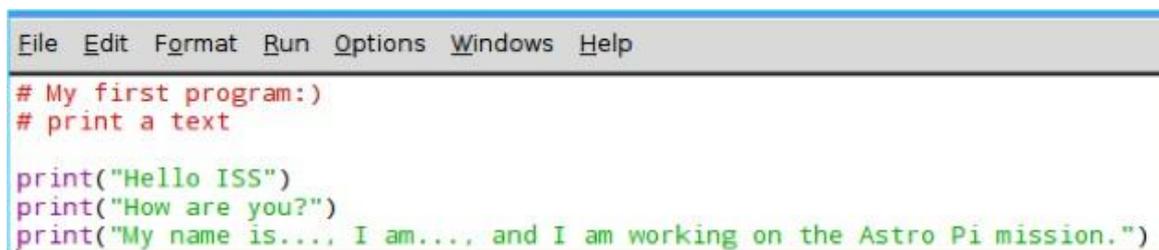
3. Um die Ergebnisse eurer Programmierung zu sehen, müsst ihr Run > Run Module [Ausführen > Modul ausführen] wählen. Das Programm fordert euch auf, euren programmierten Code zu speichern. Gebt eurer Datei einen Namen und führt sie aus. Was passiert?

---

---

**Hinweis:** Vergesst nicht, eure Arbeit regelmäßig zu speichern! Um die Datei, an der ihr gerade arbeitet, zu speichern, klickt auf „File->Save“ [Datei > Speichern].

4. Nun wird es Zeit, dass ihr euch bei den Astronauten auf der ISS vorstellt. Übernehmt den nachstehenden Code in eure Datei und schreibt eine Nachricht an die Astronauten. Herzlichen Glückwunsch! Ihr habt soeben euer erstes Programm in der Python-Sprache geschrieben.



```
File Edit Format Run Options Windows Help
# My first program:)
# print a text

print("Hello ISS")
print("How are you?")
print("My name is..., I am..., and I am working on the Astro Pi mission.")
```

**Hinweis:** Der Computer versucht, alles, was ihr in das Fenster schreibt, in Aktionen umzusetzen. Wenn ihr nur eine kleine Anmerkung für euch selbst schreiben möchtet, dann müsst ihr das Raute-Zeichen (#) vor eure Anmerkung setzen. Dieser Vorgang nennt sich „Kommentieren“. Kommentare werden nicht in die Maschinensprache übersetzt. Der Computer ignoriert jeglichen Text, der mit # beginnt.

## Aufgabe 3 – Ein Quiz mit Python

Bisher habt ihr dem Computer einfache Anweisungen gegeben. Nun werdet ihr ihn dazu bringen, Fragen zu stellen und auf eure Antworten zu reagieren.

### Übung

1. Öffnet ein neues Python-3-Fenster und gebt nachstehenden Code ein.

```
File Edit Format Run Options Windows Help
# whenever a word or number follows the '=' symbol, it is called a variable.
# variables are used to store information that the computer must remember.

orbits = input (" How long does it take the ISS to orbith the Earth?")
```

2. Klickt auf Run>Run Module [Ausführen > Modul ausführen], um euren Code auszuführen. Was passiert?

---



---

3. Python nutzt die Funktion **input ()** [Eingabe], um euch Fragen zu stellen. Wisst ihr, wie lange die ISS braucht, um die Erde zu umrunden? Gebt die Frage in Python Shell ein.

4. Ihr könnt das Python-Shell-Fenster auch als Rechner verwenden. Hier ein paar Beispiele:

```
File Edit Shell Debug Options Windows Help
Python 3.4.2 (default, Oct 19 2014, 13:31:11)
[GCC 4.9.1] on linux
Type "copyright", "credits" or "license()" for more information.
>>> 12+12
24
>>> 12*2
24
>>> 12/2
6.0
>>> 12*2-12
12
```

Beantwortet nachstehende Frage unter Zuhilfenahme von Python Shell als Rechner. Erläutert euren Rechenweg.

Wie viele Sonnenuntergänge erleben die Astronauten auf der ISS pro Tag?

---



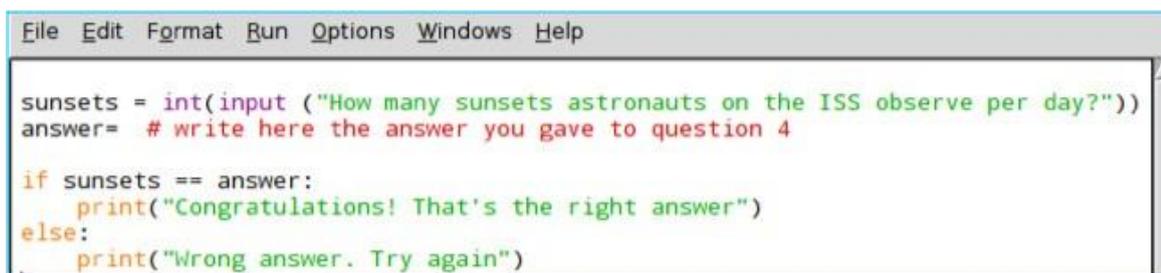
---



---

5. Nun macht ihr ein Quiz mit Raspberry Pi. Dazu müsst ihr zwischen richtigen und falschen Antworten wählen. In der Python-Sprache geht das mithilfe von **Auswahlanweisungen**, manchmal auch „**if**“-**Anweisungen** [„if“ = wenn] genannt. Öffnet eine neue Datei und schreibt nachstehenden Code. Ihr müsst den Code mit der richtigen Antwort vervollständigen. Was meint ihr, wird das Programm tun? Führt das Modul aus, dann seht ihr es.

**Hinweis:** Der Code „**int**“ steht vor dem Wort „**input**“, damit Python weiß, dass die Antwort eine ganze Zahl (ein Integer) ist, zum Beispiel 5 oder 10. Wenn die Information in Python als Zahlentyp behandelt werden soll, müsst ihr dem Computer diese Anweisung geben. Neben ganzen Zahlen können auch Gleitkommazahlen verwendet werden. Dabei handelt es sich um Dezimalzahlen, zum Beispiel 5,1 oder 10,99



```
File Edit Format Run Options Windows Help
sunsets = int(input ("How many sunsets astronauts on the ISS observe per day?"))
answer= # write here the answer you gave to question 4

if sunsets == answer:
    print("Congratulations! That's the right answer")
else:
    print("Wrong answer. Try again")
```

6. Ihr könnt beim Quiz noch etwas nachhelfen. Gebt vor der Zeile „else“ [sonst] den nachstehenden Befehl ein und führt das Modul erneut aus. Schreibt einen Kommentar, um zu erklären, was das Schlüsselwort „elif“ bewirkt.

#

---

```
elif sunsets >13 and sunsets <16:
    print("Almost there! Try again")
```

## Weiterführende Aufgabe

Mit „if“-Anweisungen lassen sich mehrere Dinge prüfen anstatt nur eins. Dazu müsst ihr einfach so viele „elif“-Anweisungen verwenden, wie ihr benötigt. Schreibt ein Programm, das den Benutzer auffordert, eine Zahl zwischen 45 und 60 einzugeben. Gebt eine Antwort ein, die dem Benutzer mitteilt, ob die Zahl „innerhalb des Bereichs“ liegt, „zu hoch“ ist oder „zu niedrig“.

## Aufgabe 4 – Geometrische Formen mit Python zeichnen

In Python gibt es viele Module, die ihr in euren Programmen verwenden könnt. Stellt euch ein Modul als eine Art Lehrbuch vor, das erklärt, wie man etwas macht. Diese Module sind in Python in einer Bibliothek hinterlegt. Bei dieser Aufgabe lernt ihr, wie ihr mit dem „Turtle“-Modul von Python geometrische Formen zeichnen könnt. Dabei lernt ihr weitere Elemente der Python-Sprache, die sehr nützlich für eure Mission sein werden.

### Übung

1. Öffnet ein neues Python-3-Fenster und gebt nachstehenden Code ein. Speichert ihn und führt ihn aus.

```
File Edit Format Run Options Windows Help
# to begin using Turtle you need to import the Turtle from the Python library
from turtle import Turtle, Screen

screen = Screen() # creates the window wich will display the turtle drawings
turtle = Turtle() # creates a turtle. The variable (ou noun) turtle is made to refer to it.

turtle.shape("turtle") # gives the turtle shape
```

2. Ihr habt soeben die Schildkröte (Turtle) erstellt, mit der ihr Bilder und Formen zeichnen könnt. Gebt eurer Schildkröte nun Anweisungen.

2.1. Fügt eurem Code folgende Zeile hinzu:

```
turtle.forward(100)
```

Was hat die Schildkröte gemacht?

---

2.2. Ergänzt eine weitere Anweisung:

```
turtle.right(90)
```

Ändert die in Klammern stehenden Zahlen. Ändert den Befehl „right“ (rechts) in „left“ (links). Was tut die Schildkröte durch diese Anweisungen?

---

**Hinweis:** Der Wert 90 in den Klammern beim Befehl `turtle.right(90)` bezieht sich auf die Einheit Grad.

2.3. Fügt darunter eine weitere Anweisung hinzu, um die Schildkröte weiter zu bewegen:

```
turtle.forward(100)
```

Was hat die Schildkröte gemacht?

---

- 2.4. Was müsst ihr noch in euren Code schreiben, um ein Quadrat zu vollenden? Schreibt den letzten Codebestandteil in das nachstehende Feld, damit eure Schildkröte ein Quadrat zeichnet.

3. Ihr könnt eurer Schildkröte viele verschiedene Anweisungen geben. Schaut euch folgenden Code an.

```
File Edit Format Run Options Windows Help
from turtle import Turtle, Screen

screen = Screen()
turtle = Turtle()

turtle.shape("turtle")

turtle.forward(100)
turtle.right(120)
turtle.forward(100)
turtle.right(120)
turtle.forward(100)
```

- 3.1. Welche geometrische Form wird die Schildkröte wohl damit zeichnen? Notiert die Antwort.
- 

- 3.2. Speichert den Code und führt ihn aus. War eure Vermutung richtig?

- 3.3. In dem oben stehenden Code, habt ihr einige Zeilen wiederholt. In der Informatik ist das nicht sehr effizient. Anstatt eine Codezeile mehrmals zu schreiben, kann man sie einmal schreiben und eine Anweisung hinzufügen, dass der Befehl wiederholt werden soll. In Python gibt es zwei Möglichkeiten, eine Anweisung zu wiederholen, und zwar mit sogenannten Schleifen (loops): der „while“-Schleife und der „for“-Schleife.

- 3.4. Öffnet ein neues Python-3-Fenster und gebt nachstehenden Code ein. Der Code innerhalb der Schleife (das heißt, der Code, der nach links eingerückt ist) beinhaltet die Anweisungen, die das Programm wiederholt. Speichert den Code und führt das Programm aus. Was passiert?

```
File Edit Format Run Options Windows Help
from turtle import Turtle, Screen

screen = Screen()
turtle = Turtle()

turtle.shape("turtle")

turtle.forward(100)
turtle.right(120)
turtle.forward(100)
turtle.right(120)
turtle.forward(100)
```

- 3.5. In diesem Beispiel wurde eine „for“-Schleife verwendet. Übernehmt nachstehenden Code und führt in aus, um zu sehen, was passiert.

```
File Edit Format Run Options Windows Help
from turtle import Turtle, Screen

screen = Screen()
turtle = Turtle()

turtle.shape("turtle")

for i in range (3):
    turtle.forward(100)
    turtle.right(120)
```

**Hinweis:** Ist euch aufgefallen, wie die Anweisungen unter den „while“/„for“-Schleifen verschachtelt waren? Dies nennt sich „Einzug“. Python nutzt den Einzug, um Code-Blöcke bei Schleifen oder „if“-Anweisungen zu begrenzen. Jeder Einzug besteht aus vier Leerzeichen, aber wenn ihr die Tabulatortaste drückt, werden die Leerzeichen automatisch eingefügt. Vergesst beim Kopieren und Einfügen von Code den Einzug nicht!

- 3.6. Welche Schleife würdet ihr für die Zeichnung verwenden? Begründet eure Wahl und erläutert die Unterschiede zwischen den beiden Schleifenarten.

## Weiterführende Aufgabe

Ihr könnt nach vorn, nach rechts und links gehen und diese Schritte mithilfe von Schleifen wiederholen. Lasst eurer Kreativität freien Lauf! Könnt ihr das ESA-Maskottchen Paxi zeichnen? Welche geometrischen Formen erkennt ihr in Paxis Bild?



## DIE NÄCHSTEN SCHRITTE

Ihr habt nun euren Raspberry Pi zusammengebaut und kennt die grundlegende Programmierstruktur von Python. Fahrt nun mit Lektion 2 fort, um zu erfahren, wie ihr über die LED-Matrix des Sense HAT mit den Astronauten auf der ISS kommunizieren könnt.