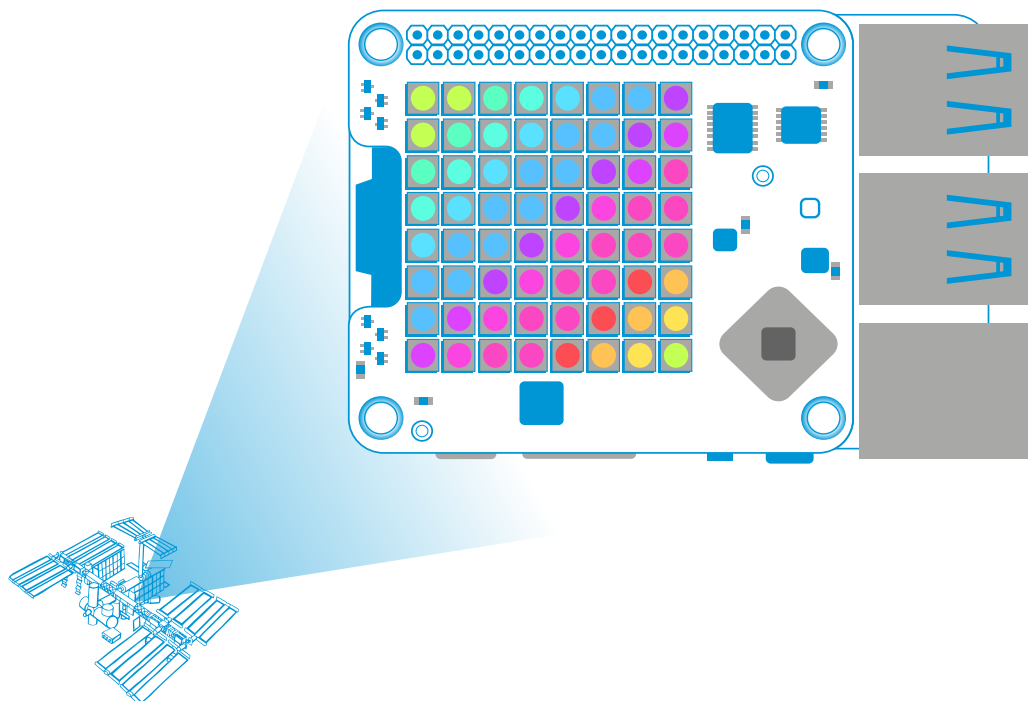


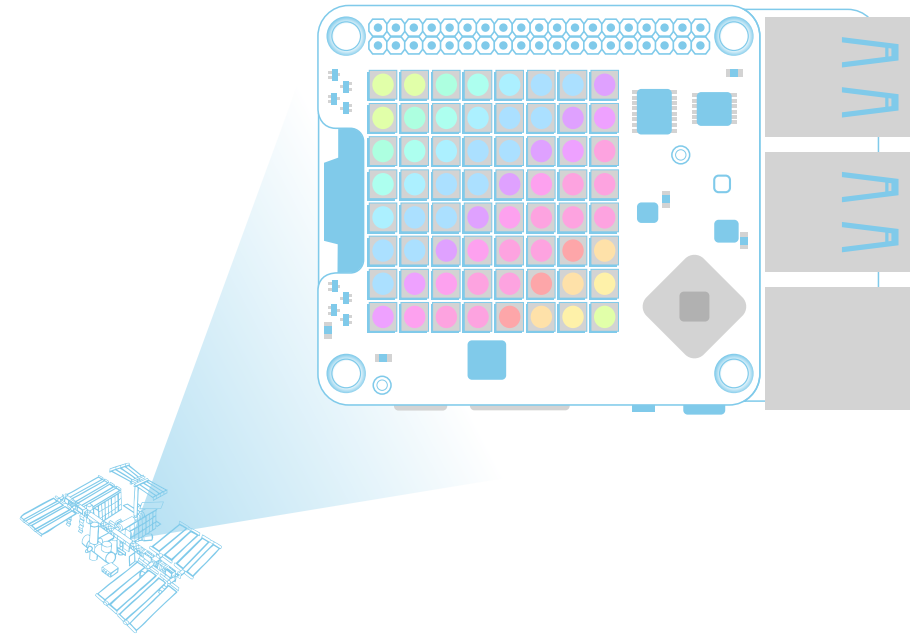


# teach with space

## → DER SENSE HAT

Anzeige von Text und Bildern auf der LED-Matrix des Sense HAT





Aufgabe 1 – Sense HAT montieren

seite 4

Aufgabe 2 – Hallo, hier ist die Erde!

seite 5

Aufgabe 3 – Wie funktionieren Farb-Displays?

seite 6

Aufgabe 4 – Bilder anzeigen

seite 9

Aufgabe 5 – Ausrichtung steuern

seite 12

## → DER SENSE HAT

### Anzeige von Text und Bildern auf der LED-Matrix des Sense HAT

Der Sense HAT ist eine Platine zum Nachrüsten des Raspberry Pi und wurde speziell für den Astro-Pi-Wettbewerb entwickelt. Sie ist ein grundlegender Bestandteil eurer Astro-Pi-Mission. Die Platine kann verschiedenste Informationen erfassen und über eine LED-Matrix anzeigen.

In dieser Lektion lernt ihr die Sense-HAT-Hardware mit der zugehörigen Python-Bibliothek kennen. Außerdem erfahrt ihr, wie man die LED-Matrix steuert und Ausgabewerte visuell anzeigt.

Abbildung A1



[↑ Astro Pi logo](#)

## Versuchsmaterial

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

## → AUFGABE 1 – SENSE HAT MONTIEREN

Zwei erweiterte Raspberry-Pi-Computer, die zur Internationalen Raumstation ISS geschickt wurden, sind jeweils mit einer Sense-HAT-Platine (HAT = Hardware Attached on Top, dt. aufgesetzte Hardware) ausgestattet, ähnlich der in eurem Bausatz. Durch diese Platine kann der Raspberry Pi verschiedene Dinge wie die Temperatur oder Bewegungen wahrnehmen und Informationen über ein spezielles Display ausgeben – eine Matrix mit 8 x 8 LEDs. An Bord der ISS kann ein Raspberry Pi nicht an einen Monitor angeschlossen werden, deshalb ist die Matrix die einzige visuelle Ausgabemöglichkeit.

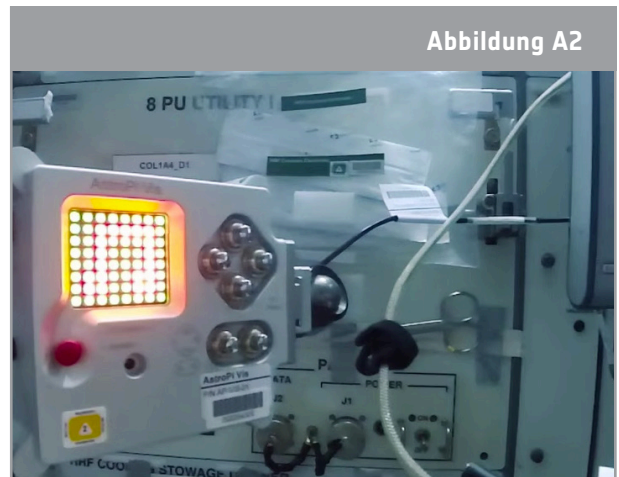


Abbildung A2

Bei dieser Aufgabe werdet ihr den Sense HAT montieren, um später seine Funktionen nutzen zu können.

↑ [Astro Pi Ed auf der ISS](#)

### Übung

1. Der Sense HAT ist in einer silbernen, antistatischen Schutzhülle verpackt und wird mit folgenden Befestigungselementen geliefert:
  - 4 sechseckigen Abstandsbolzen (beidseitig mit Innengewinde)
  - 8 Schrauben M2.5

Prüft den Packungsinhalt auf Vollständigkeit, bevor ihr fortfahrt.

2. Nun montiert ihr den Sense HAT auf den Raspberry Pi. Für die Montage muss der Raspberry Pi heruntergefahren sein und das Stromkabel sowie alle sonstigen Kabel müssen vom Computer getrennt sein.
3. Schraubt die sechseckigen Abstandsbolzen an den Raspberry Pi. Steckt dazu die Schrauben von unten durch die Bohrlöcher im Raspberry Pi und dreht die sechseckigen Abstandsbolzen mit Daumen und Zeigefinger auf die Schrauben (siehe Abbildung 2).

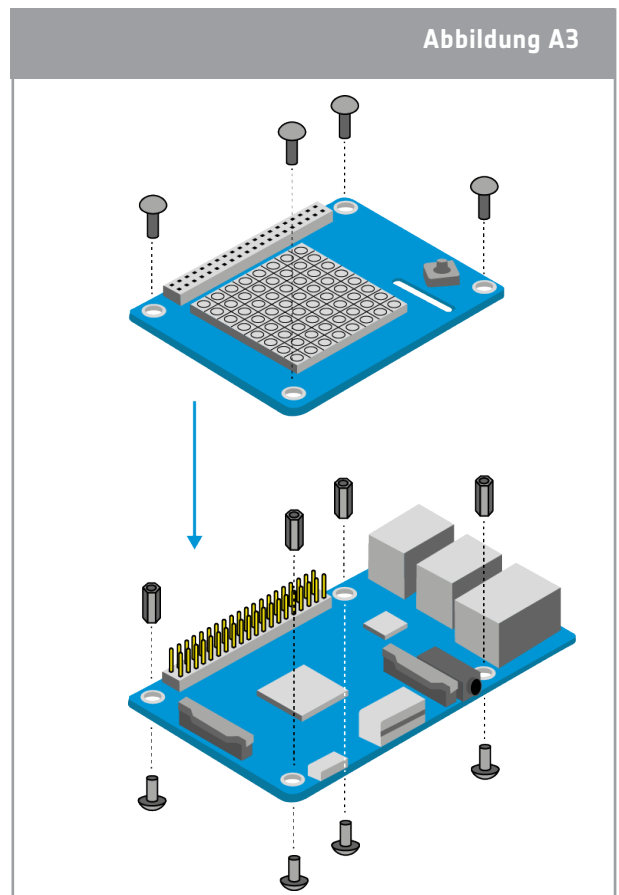


Abbildung A3

↑ [Montage des Sense HAT am Raspberry Pi](#)

4. Setzt dann den Sense HAT in den GPIO-Steckplatz (GPIO = General Purpose Input Output, programmierbare Ein- und Ausgänge für allgemeine Zwecke). Die Bohrlöcher in den Ecken müssen über den Abstandsbolzen liegen.
5. Schraubt abschließend die übrigen Schrauben von oben in die Abstandsbolzen.
6. Zieht die einzelnen Schrauben in den Abstandsbolzen mit einem kleinen Schraubendreher an. Sie müssen nicht extrem fest sitzen, nur so, dass der Sense HAT sich nicht löst.
7. Nun könnt ihr euren Astro Pi einschalten und starten.

## → AUFGABE 2 – HALLO, HIER IST DIE ERDE!

Die LED-Matrix des Sense HAT dient zur Anzeige von Formen, Symbolen und Nachrichten an die Besatzung der ISS. Bei dieser Aufgabe werdet ihr euer erstes Programm mit dem Sense HAT ausführen und eine Nachricht an die Astronauten an Bord der ISS senden.

### Übung

1. Öffnet Python 3, indem ihr oben am Bildschirm auf das Himbeer-Symbol klickt. Dadurch öffnet sich das Menü. Wählt Programming > Python 3 [Programmierung > Python 3]. Daraufhin wird das „Python Shell“-Fenster angezeigt. Klickt auf File > New File [Datei > Neue Datei] und schreibt den nachstehenden Code in das neue Fenster.

```
File Edit Format Run Options Windows Help
#to allow the program to use the Sense HAT hardware
from sense_hat import SenseHat

#to create a sense object which represents the Sense HAT
sense = SenseHat()

#to make the Sense HAT show the text
sense.show_message("Hello, this is Earth!")
```

2. Wählt File > Save As [Datei > Speichern als] und gebt eurem Programm einen Dateinamen. Klickt dann auf Run > Run module [Ausführen > Modul ausführen]. Eure Nachricht sollte dann als weißer Text über die LED-Matrix laufen.
3. Ihr könnt auch einen Parameter für die Laufgeschwindigkeit hinzufügen. Übernehmt folgenden Code und führt ihn aus:

```
File Edit Format Run Options Windows Help
#to allow the program to use the Sense HAT hardware
from sense_hat import SenseHat

#to create a sense object which represents the Sense HAT
sense = SenseHat()

#to make the Sense HAT show the text
sense.show_message("Hello, this is Earth!", scroll_speed=0.05)
```

4. Mithilfe des Befehls „sense.show\_letter“ lassen sich auf der LED-Matrix anstelle von ganzen Nachrichten auch einzelne Zeichen darstellen. Öffnet eine neue Datei und gebt folgenden Code ein:
5. Speichert den Code und führt ihn aus. Worum geht es dabei?

---



---

```
File Edit Format Run Options Windows Help
from sense_hat import SenseHat

#to allow the program to use the time module
import time

sense = SenseHat()

sense.show_letter("H")
time.sleep(0.5)
sense.show_letter("I")
time.sleep(0.5)

#to reset the LEDs to off
sense.clear()
```

# → AUFGABE 3 – WIE FUNKTIONIEREN FARB-DISPLAYS?

Die LED-Matrix des Sense HAT besteht aus 64 mehrfarbigen LEDs. Jede der 64 LEDs besteht aus drei kleineren LEDs: eine rote, eine grüne und eine blaue, wie bei einem Fernseher oder dem Display eines Smartphones.

Bei dieser Aufgabe lernt ihr, wie elektronische Farb-Displays funktionieren und wie ihr mit den farbigen LEDs bunte Nachrichten senden könnt.

## Übung

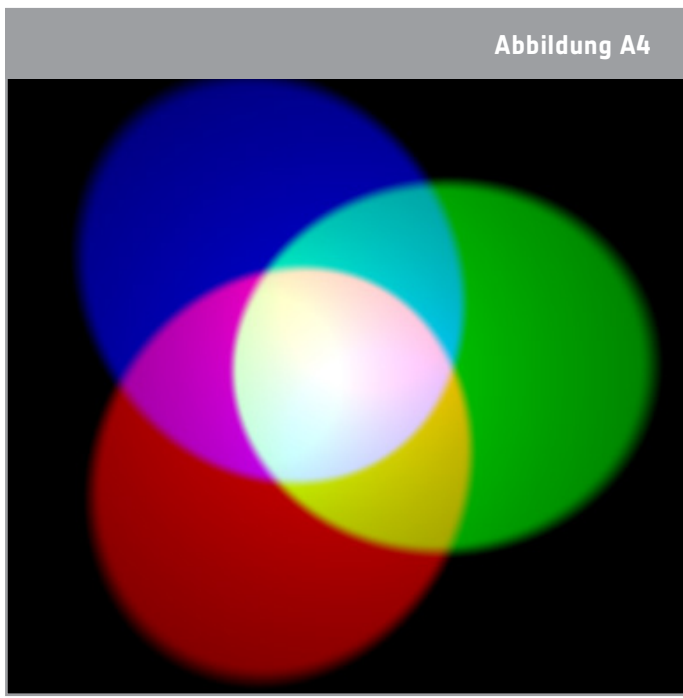
1. Bei der additiven Farbmischung werden die Farben Rot, Grün und Blau zu anderen Farben vermischt. Diese drei Farben werden als additive Grundfarben bezeichnet. In Abbildung A4 seht ihr drei Schweinwerferstrahlen gleicher Helligkeit, einen für jede Farbe. Wenn keine Farbe vorhanden ist, so ist das Ergebnis Schwarz.

a) Durch die Kombination von zwei Grundfarben entsteht eine dritte Farbe. Schreibe auf, welche Farben bei den einzelnen Kombinationen der Grundfarben entstehen.

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

b) Wie sieht das Ergebnis aus, wenn alle drei Grundfarben miteinander kombiniert werden?

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_



↑ Additive Farbmischung

2. Wenn man die Helligkeit der drei Grundfarben variiert, lassen sich sogar noch mehr Farben erzeugen als die, die ihr in dem abgebildeten Farbkreis seht. Öffnet ein neues Python-3-Fenster und gebt nachstehenden Code ein.

```
File Edit Format Run Options Windows Help
from sense_hat import SenseHat

#get the sense hat library
sense = SenseHat()

#set r, g and b brightness to maximum
r = 255
g = 255
b = 255

#clear whatever was there before, and then set all pixels to white
sense.clear((r, g, b))
```

Wählt File > Save As [Datei > Speichern als] und gebt eurem Programm einen Dateinamen. Klickt dann auf Run > Run module [Ausführen > Modul ausführen]. Welche Farbe wurde auf der SenseHAT-Matrix angezeigt?

---

3. Die Variablen „r“, „g“ und „b“ stehen für die Farben Rot, Grün und Blau. Die dahinter stehenden Zahlen geben an, wie hell jede Farbe sein soll. Die Helligkeitswerte können zwischen 0 und 255 liegen. Im oben stehenden Code wurde der Höchstwert 255 für jede Farbe verwendet. Ändert die Werte folgendermaßen: 255 für rot, aber 0 für grün und 0 für blau. Führt danach den Code erneut aus. Welche anderen Farben könnt ihr darstellen?
- 

4. Dieses Farbmischsystem wird im gesamten Astro-Pi-Programmiermodul verwendet. Einen besonderen Effekt haben unterschiedliche Farben bei der Programmierung von Lauftext. Gebt den folgenden Code in die neue Datei ein:

```
File Edit Format Run Options Windows Help
from sense_hat import SenseHat

sense = SenseHat()

#use show_message function and set text to red
sense.show_message("Hello, this is Earth!", text_colour=(255,0,0))
```

**Achtet:** auf die Syntax (also die Schreibfolge) `text_colour=(255, 0, 0)` – Kommas nicht vergessen!

5. Ihr könnt außerdem die Hintergrundfarbe der Nachricht verändern, und zwar so:

```
File Edit Format Run Options Windows Help
from sense_hat import SenseHat

sense = SenseHat()

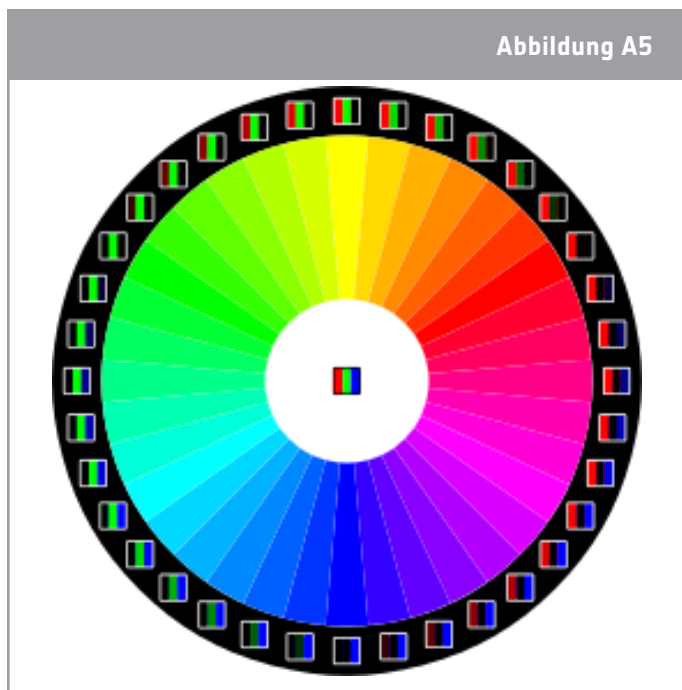
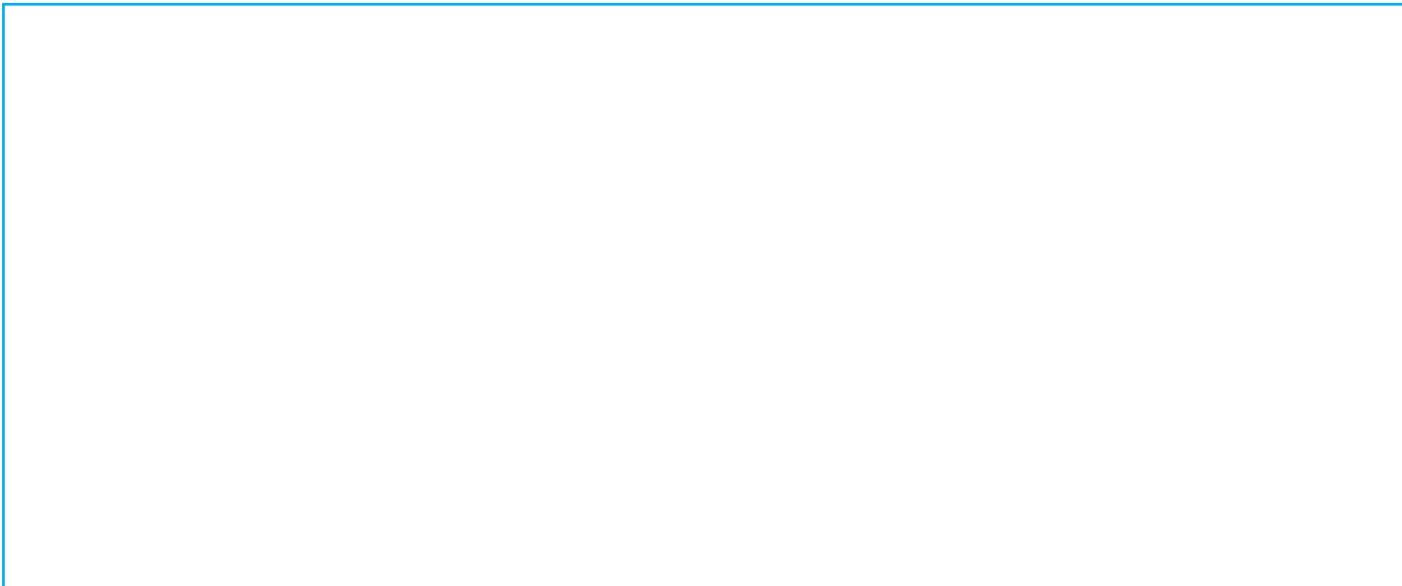
sense.show_message("Hello, this is Earth!", text_colour=(255,0,0), back_colour=(0,0,255))

sense.clear()
```

**Hinweis:** Die Kommas sind wichtig, nicht vergessen!

6. Schreibt eine Nachricht in Farben von gegenüberliegenden Seiten des Farbkreises, sodass sie für die viel beschäftigten Astronauten gut lesbar ist. Wie lässt sich der Text dauerhaft anzeigen? Schreibt euren Code in nachstehendes Feld.





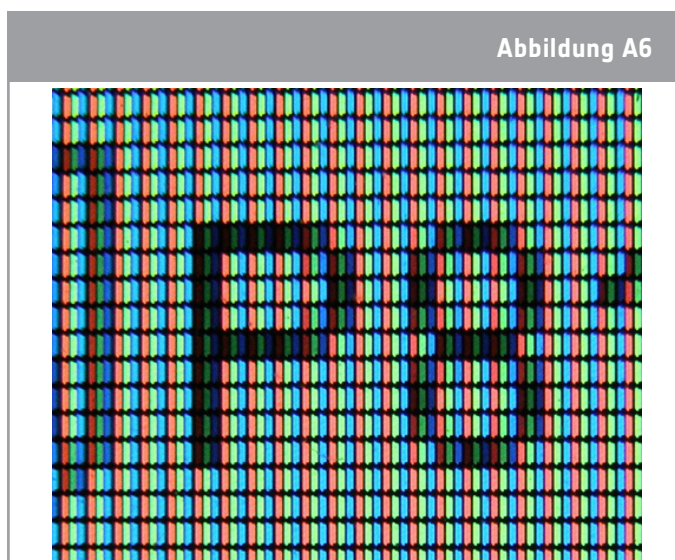
↑ Farbkreis

## → AUFGABE 4 – BILDER ANZEIGEN

Mit der LED-Matrix lässt sich aber mehr als nur Text darstellen. Ihr könnt auch ein Bild erzeugen! Bei der Vergrößerung einer Digitalaufnahme werden Tausende kleine Quadrate sichtbar – und jedes hat nur eine Farbe. Aus der Kombination dieser Quadrate, der Pixel, entsteht das Bild. Bei dieser Aufgabe lernt ihr, was ein Pixel ist und wie ihr Bilder auf der LED-Matrix des Sense HAT anzeigen könnt.

### Übung

- Abbildung A6 zeigt die Pixel auf einem LCD-Bildschirm eines Notebooks. Ein Pixel ist ein kleines Element eines digitalen Bildes, wie die winzigen Quadrate, die ihr auf dem Display von Mobiltelefonen seht. Jedes Pixel auf eurem Bildschirm besteht aus drei Subpixeln (rot, grün und blau). Ihr seht, dass die Pixel zum Formen von Buchstaben und Zahlen ein- und ausgeschaltet werden. Alle Computer- und Smartphone-Bildschirme funktionieren auf diese Weise. Wenn ihr also Formen auf der LED-Matrix darstellen wollt, müsst ihr dasselbe tun.



↑ Digital pictures are composed of pixels

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

```
File Edit Format Run Options Windows Help
from sense_hat import SenseHat

sense = SenseHat()

sense.clear()

#set X and Y coordinates to 0,0
x = 0
y = 0

#set pixel 0,0 to bright white
sense.set_pixel(x, y, 255, 255, 255)
```

Wählt File > Save As [Datei > Speichern als] und gebt eurem Programm einen Dateinamen. Klickt dann auf Run > Run module [Ausführen > Modul ausführen]. Was passiert?

---



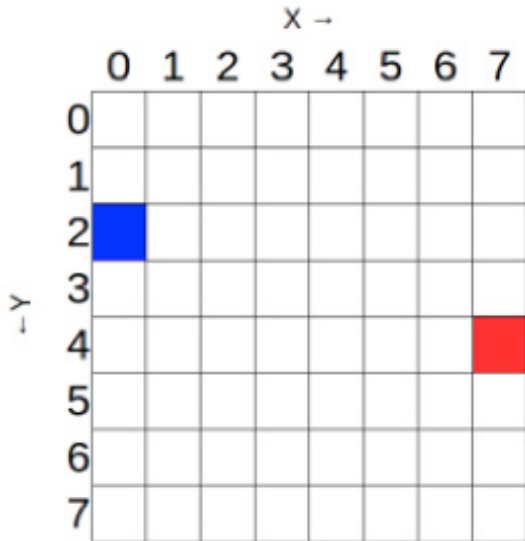
---

2. Mit den Variablen „x“ und „y“ lässt sich steuern, welche LED mit dem Befehl `set_pixel` verändert werden soll. X ist horizontal und beinhaltet die Werte von 0 auf der linken Seite bis 7 auf der rechten Seite. Y ist vertikal und beinhaltet die Werte von 0 oben bis 7 unten. Gebt den folgenden Code in ein neues Fenster ein:

```
File Edit Format Run Options Windows Help
from sense_hat import SenseHat

sense = SenseHat()

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



a) Malt in der Matrix die Kästchen farbig aus, wie sie eurer Meinung nach auf der Sense-HAT-Matrix angezeigt werden.

b) Speichert den Code und führt ihn aus. Ist das passiert, was ihr erwartet habt?

3. Wahrscheinlich werdet ihr in Versuchung geraten, immer neue Muster und Formen mit dem Befehl `set_pixel` in euren Code einzubinden. Es gibt aber auch den Befehl `set_pixels`, mit dem ihr alle 64 LEDs mit nur einer Zeile Code verändern könnt! Damit könntet ihr beispielsweise einen Minecraft Creeper auf der LED-Matrix darstellen. Und euch stehen viele Variablen zur Verfügung, mit denen ihr eure Farbpalette festlegen könnt. Führt die nachstehenden Beispiele aus:

```
File Edit Format Run Options Windows
from sense_hat import SenseHat

sense = SenseHat()
0= (0, 255, 0)
X= (0, 0, 0)

creeper_pixels = [
0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0,
0, X, X, 0, 0, X, X, 0,
0, X, X, 0, 0, X, X, 0,
0, 0, 0, X, X, 0, 0, 0,
0, 0, X, X, X, X, 0, 0,
0, 0, X, X, X, X, 0, 0,
0, 0, X, 0, 0, X, 0, 0,
]

sense.set_pixels(creeper_pixels)
```

```
File Edit Format Run Options Windows Help
from sense_hat import SenseHat

sense = SenseHat()

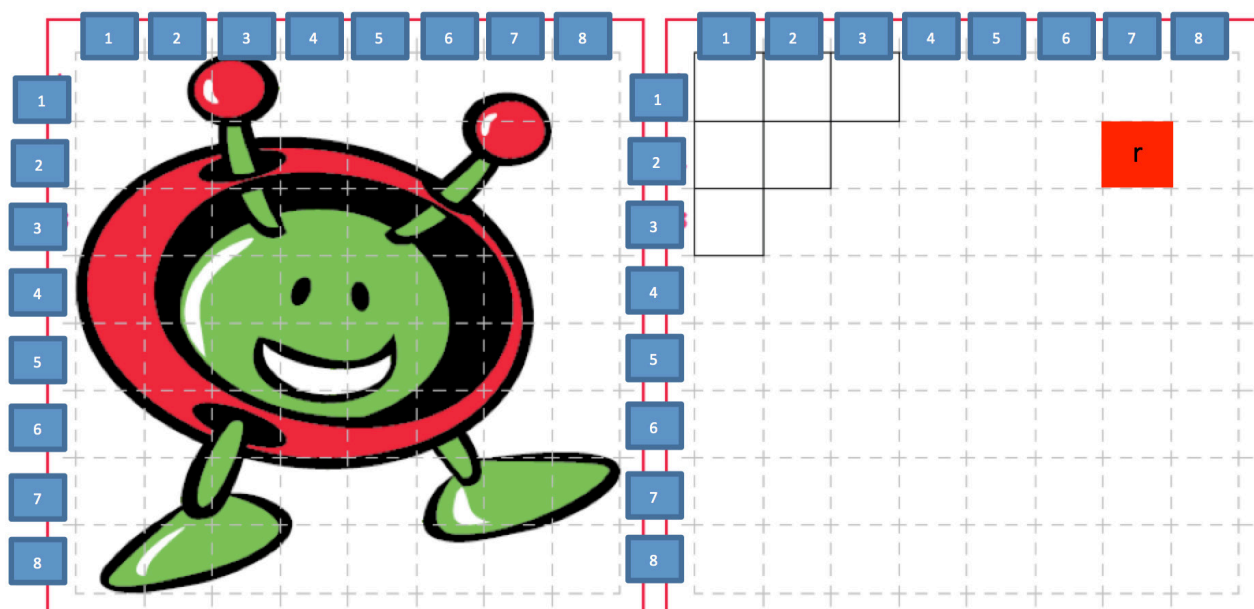
w = [255, 255, 255]
r = [255, 0, 0]
o = [255, 127, 0]
y = [255, 255, 0]
g = [0, 255, 0]
b = [0, 0, 255]
i = [75, 0, 130]
v = [159, 0, 255]
e = [0, 0, 0] # e stands for empty/black

image = [
e,e,e,e,e,e,e,e,
e,e,e,r,r,e,e,e,
e,r,r,o,o,r,r,e,
r,o,o,y,y,o,o,r,
o,y,y,g,g,y,y,o,
y,g,g,b,b,g,g,y,
b,b,b,i,i,b,b,b,
b,i,i,v,v,i,i,b
]

sense.set_pixels(image)
```

**Achtet** auf die eckigen Klammern [] und die Syntax `sense.set_pixels(Name des Bilds)`.

4. Nun ist es an der Zeit, dass ihr euer eigenes Bild programmiert. Versucht doch einmal, das ESA-Maskottchen Paxi anhand des nachstehenden Rasters mit Pixeln darzustellen. Dabei sollte die Farbe eines Quadrates der Hauptfarbe im jeweiligen Rasterfeld des Bilds von Paxi entsprechen. Schreibt den Code auf, mit dem sich Paxis Bild auf der LED-Matrix anzeigen lässt.



5. Öffnet ein neues Python-3-Fenster und führt euren Code aus. Wird Paxi auf der LED-Matrix angezeigt?

## → AUFGABE 5 – AUSRICHTUNG STEuern

Bisher wurden eure Texte und Bilder in derselben Richtung angezeigt, unter der Annahme, dass die Seite mit dem HDMI-Anschluss die Unterseite ist. Das mag aber nicht immer der Fall sein (besonders auf der ISS). Bei dieser Aufgabe lernt ihr, wie ihr die Ausrichtung ändern, das heißt die Anzeige auf der Matrix drehen könnt.

### Übung

- Um die Ausrichtung der Matrix zu ändern, könnt ihr die Methode `sense.set_rotation()` verwenden und in den Klammern eins von vier Winkelmaßen eingeben (0, 90, 180, 270). Öffnet ein neues Python-3-Fenster und gebt nachstehenden Code ein. Wählt File > Save As [Datei > Speichern als] und gebt eurem Programm einen Dateinamen. Klickt dann auf Run > Run module [Ausführen > Modul ausführen].

```
File Edit Format Run Options Windows Help
from sense_hat import SenseHat

sense = SenseHat()

r = [255, 0, 0]
o = [255, 127, 0]
y = [255, 255, 0]
g = [0, 255, 0]
b = [0, 0, 255]
i = [75, 0, 130]
v = [159, 0, 255]
e = [0, 0, 0]

image = [
e,e,e,e,e,e,e,e,
e,e,e,r,r,e,e,e,
e,r,r,o,o,r,r,e,
r,o,o,y,y,o,o,r,
o,y,y,g,g,y,y,o,
y,g,g,b,b,g,g,y,
b,b,b,i,i,b,b,b,
b,i,i,v,v,i,i,b
]

sense.set_pixels(image)
sense.set_rotation(180)
```

- Ihr könnt auch Text erzeugen, der sich dreht. Verwendet dazu die „for“-Schleife. Öffnet ein neues Python-3-Fenster. Übernehmt nachstehenden Code und führt in aus. Was passiert?

---



---

```
File Edit Format Run Options Windows Help
from sense_hat import SenseHat
from time import sleep

sense = SenseHat()

sense.show_letter("J")

#create a list of angles to iterate through
angles = [0, 90, 180, 270, 0, 90, 180, 270]

#cycles through the angles, rotating
for r in angles:
    sense.set_rotation(r)
    sleep(0.5)
```

3. Das Bild lässt sich mit folgenden Codezeilen auch spiegeln, entweder horizontal oder vertikal:

```
sense.flip_h()
oder
sense.flip_v()
```

Mit nachstehendem Beispiel könnt ihr eine einfache Animation erzeugen, bei der das Bild immer wieder gespiegelt wird.

```
File Edit Format Run Options Windows Help
from sense_hat import SenseHat
from time import sleep

sense = SenseHat()

w = [150, 150, 150]
b = [0, 0, 255]
e = [0, 0, 0]

image = [
e,e,e,e,e,e,e,e,
e,e,e,e,e,e,e,e,
w,w,w,e,e,w,w,w,
w,w,b,e,e,w,w,b,
w,w,w,e,e,w,w,w,
e,e,e,e,e,e,e,e,
e,e,e,e,e,e,e,e,
e,e,e,e,e,e,e,e
]

sense.set_pixels(image)

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

### Weiterführende Aufgabe

Können ihr mithilfe einer der bereits beschriebenen Zeichentechniken und der Anweisung „sense.set\_rotation“ ein sich drehendes Bild erzeugen?

## DIE NÄCHSTEN SCHRITTE

Ausgabegeräte wie die Sense-HAT-Matrix sind eine gute Möglichkeit für euch, um mit eurem Astro Pi Text und Bilder anzuzeigen. Der Sense HAT des Raspberry Pi ist außerdem mit verschiedenen Sensoren ausgestattet, mit denen ihr eure Umgebung analysieren und spannende Experimente durchführen könnt – sowohl in eurem Klassenraum als auch auf der ISS! Bei den nächsten Aufgaben erkundet ihr, was ihr mit den Sense-HAT-Sensoren alles machen könnt.

**teach with space – der sense hat | T05.2b**  
[www.esa.int/education](http://www.esa.int/education)

**An ESA Education production**  
in collaboration with Raspberry Pi Foundation, ESERO Poland and ESERO UK

Copyright 2017 © European Space Agency