

Sensorik und Bibliotheken

Vorbereitungsworkshop für das Messsystem LoRaMINT

Matthias Ruf

Sensorik und Bibliotheken • Ablauf Workshoptag 3

SFZ Südwürttemberg • Standort Ochsenhausen

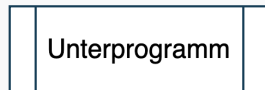
Workshoptag 3

1

Was eine Funktion wirklich ist

Funktionen – Ihr benutzt sie längst

- Wir haben bereits `print`, `round`, `len`, `max`, usw. verwendet, ohne zu hinterfragen, was das eigentlich ist.
 - Gestern stand es bereits im Ablaufplan, als Kasten mit den zwei Strichen.
- ⇒ heute wollen wir uns **Funktionen** genauer ansehen.



Eine Funktion aufrufen

Funktionsaufruf – was ist das eigentlich?

```
gerundet = round(21.4736, 1)
```

Der Funktionsname

round bezeichnet den Namen der Funktion, damit ruft man die Funktion auf.

Parameter

21.4736 und 1 sind die Werte, mit denen die Funktion arbeiten soll.

Der Rückgabewert

21.5 wird zurückgegeben. Er landet in der Variable gerundet.

Eine Funktion ist ein Art Auftrag: Du gibst etwas hinein, du bekommst etwas zurück. Was im Detail passiert, wird hinter dem Funktionsaufruf versteckt.

print ist anders

`print` *gibt aus*. Einen Rückgabewert hat sie nicht.

```
ergebnis = print("Hallo")  
print(ergebnis)
```

Auf dem Bildschirm steht Hallo und darunter None.

Ausgeben ist nicht Zurückgeben

`round` liefert einen Wert, der gespeichert werden kann. `print` gibt etwas aus, hat aber keinen Rückgabewert.

`None` heißt nichts und beschreibt eine leere Variable.

Beides nebeneinander

```
1 # Zwei Funktionen, die ihr kennt - und der Unterschied zwischen ihnen.
2
3 temperatur = 21.4736
4
5 # round bekommt zwei Werte und gibt einen zurueck.
6 gerundet = round(temperatur, 1)
7 print("Temperatur:", gerundet, "Grad")
8
9 # Dasselbe in einem Schritt: von innen nach aussen gelesen.
10 print("Temperatur:", round(temperatur, 1), "Grad")
11
12 # print gibt aus - zurueck gibt es nichts.
13 ergebnis = print("Hallo")
14 print(ergebnis)
```

Jetzt ihr

Einzelarbeit am Arbeitsblatt

Arbeitsblatt 3 • Aufgabe 1

Seht euch die Funktionen von Tag 1 noch einmal an, diesmal mit der Frage: Was geht hinein, was kommt heraus?

Der Trick beim Lesen

Bei `round(sum(werte), 1)` zuerst die innerste Klammer. Von innen nach außen.

Wenn ihr fertig seid

Die Expertenstufe wartet.

2

Funktionen von anderen

Woher die Funktionen kommen

`print` und `round` sind einfach da: Es handelt sich um **built-in functions**, die Python/MicroPython zur Verfügung stellt.

Alle anderen bereitgestellten Funktionen müssen **vor** dem Aufruf importieren werden.

`math` ist eine **Bibliothek**, also eine Sammlung fertiger Funktionen, die dir bereitgestellt wird.

Warum überhaupt

Was geladen ist, kostet Speicher. Es ist sinnvoll etwas nur zu importieren wenn es gebraucht wird.

```
import math

print(math.sqrt(49))
```

Der Punkt dazwischen

```
time.sleep(1)  
math.sqrt(144)
```

Vor dem Punkt

steht, **woher** die Funktion kommt.

Nach dem Punkt

steht, **welche** es ist.

...und dann wie immer die Klammern mit den Parametern

Drei Sorten

Eingebaut

`print, round, len`

Die **built-in functions**.

Kein `import` nötig.

Mitgeliefert

`time, math, machine`

Die

Standardbibliothek:

schon auf dem Board,
ein `import` genügt.

Dazugelegt

die Sensorbibliothek

Eine **externe**

Bibliothek: Sie muss als
Datei auf dem Board
liegen, sonst gibt es sie
nicht.

Woran ihr das merkt

`ImportError: no module named 'bme280'` heißt nicht, dass ihr euch vertippt habt. Es heißt: Diese Bibliothek liegt nicht auf dem Board.

Jetzt ihr Einzelarbeit am Arbeitsblatt

Arbeitsblatt 3 • Aufgabe 2

Holt euch zwei Bibliotheken ins Programm und probiert aus, was darin steckt.

Gleich danach
wird es die dritte Bibliothek sein, die
den Sensor ausliest.

Wenn ihr fertig seid
Findet heraus, was in einer Bibliothek
überhaupt alles drin ist.

3

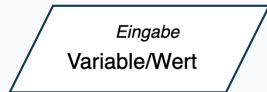
Der Sensor kommt dazu

Das Symbol von gestern

Im Ablaufplan gab es ein Symbol, bei dem stand: *später der Sensor*.

Bisher kamen die Messwerte aus eurem Kopf – ihr habt sie euch ausgedacht und in eine Liste geschrieben.

Ab jetzt kommen sie von außen. Aus der Luft in diesem Raum.



Jetzt der Sensor

Vier Drähte

Angeleiteter Aufbau

Strom

3V3 und GND. Damit lebt der Sensor.

Daten

SDA und SCL. Darüber redet er mit dem Board.

Mehr ist es nicht. Vier Drähte, und die Beschriftung steht auf beiden Platinen.

Welcher Pin wohin

Steht auf **Arbeitsblatt 3, Aufgabe 3** – als Bild und als Text. Sucht euch aus, womit ihr besser zurechtkommt.

Die eine Regel

Erst stecken, dann einschalten. Nie umgekehrt.

4

Echte Messwerte

Der erste eigene Messwert

Einzelarbeit am Arbeitsblatt

Arbeitsblatt 3 • Aufgabe 4

Drei Zeilen, und auf dem Bildschirm steht die Temperatur dieses Raumes. Legt danach die Hand auf den Sensor und misst noch einmal.

Wenn ihr feststeckt

Erst das Kärtchen, dann die Nachbarin, dann Hand heben.

Drei Größen

Der Sensor misst nicht nur Temperatur.

Wenn ihr fertig seid

In der Box liegt ein zweiter Sensor.

5

Messwerte mit Namen

Was die Liste nicht kann

Der Sensor liefert drei Werte. In einer Liste sieht das so aus:

```
messung = [21.5, 48.2, 1013.0]
```

Und morgen früh: Was war noch mal `messung[1]`?

Die Reihenfolge steht nirgends

Sie steht nur in eurem Kopf – und in dem Programm, das ihr geschrieben habt. Wer die Zahlen bekommt, sieht ihnen nicht an, was sie bedeuten.

Eine Liste ordnet Werte. Sie benennt sie nicht.

Ein Name statt einer Nummer

```
messung = {"temperatur": 21.5, "feuchte": 48.2, "druck": 1013.0}

print(messung["temperatur"])
```

Geschweifte Klammern
machen ein **Dictionary**
– eckige machen eine
Liste.

Immer paarweise
Schlüssel,
Doppelpunkt, **Wert**. Der
Schlüssel ist der Name.

Zugriff
Eckige Klammern wie bei
der Liste – nur steht
jetzt der Name darin,
nicht die Nummer.

Eine Messung mit Namen

```
1 # Eine Messung, wie sie der Sensor liefert: drei Werte, jeder mit Namen.
2
3 messung = {"temperatur": 21.5, "feuchte": 48.2, "druck": 1013.0}
4
5 print(messung["temperatur"])
6
7 # Ein Wert laesst sich einzeln ersetzen, ohne die anderen anzufassen.
8 messung["temperatur"] = 22.1
9
10 # Und mit der Schleife von gestern kommt alles beschriftet heraus.
11 for name in messung:
12     print(name, messung[name])
```

Jetzt ihr

Einzelarbeit am Arbeitsblatt

Arbeitsblatt 3 • Aufgabe 5

Legt eure drei echten Messwerte aus Aufgabe 4 in ein Dictionary – und gebt sie beschriftet aus.

Warum das die Mühe wert ist

An Tag 4 wird jeder Wert einzeln verschickt. Dann müsst ihr sagen, welcher es ist.

Wenn ihr fertig seid

Die Schleife von gestern läuft auch über ein Dictionary.

6

Euer Projekt

Wiederholt messen und warnen

Einzel- oder Partnerarbeit

Projektauftrag

Baut ein Programm, das immer wieder misst – und Alarm gibt, wenn ein Wert aus dem Rahmen fällt.

Alles kommt zusammen

Die Kontrollstrukturen von gestern – Schleife und Abfrage – mit dem Sensor und dem Dictionary von heute.

Erst die Grundversion

Sie steht auf dem Auftrag und ist für alle gleich. Danach sucht ihr euch eine Ausbaustufe aus.

Am Ende

führt ihr euer Programm kurz vor. Zwei Minuten, kein Vortrag.

7

Was hängengeblieben ist

Kahoot 4

Lernstandskontrolle

So läuft es

Noch einmal **kahoot.it** und die PIN von der Wand. Es geht um heute – Funktionen, Bibliotheken, der Sensor und das Dictionary.

Was ihr heute geschafft habt

Ihr habt einen Sensor angeschlossen, der vorher nur ein Bauteil in einer Box war. Euer Programm misst jetzt die echte Luft in diesem Raum, wiederholt das von allein und meldet sich, wenn etwas nicht stimmt.