1 Quellcode

1.1 hello

(Listing 1 hello).

```
# chmod a+x *.py
# python3 ./script.py
# % ju -- https://bw1.eu -- 3-11-18
# update:
#==================================

print("Hello, World")
```

Listing 1: Quellcode in Python, hello

1.2 programmierleitfaden

(Listing 2 programmierleitfaden).

```
# Shell-Befehle

# ssh
# ip addr
# Linux => pi@raspi:
# ssh ip-adresse -l pi
# git clone https://github.com/simonmonk/make_action.git
# cd /home/pi/make_action
# git pull
# python test.py
# python3 test.py

# Programmierleitfaden

# Bibliothek RPi.GPIO
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)# Pin-Benennung von Broadcom verwenden

# python3 hello.py
print("Hello, World")

# Variablen
a = 123.45
b = "message"
print(a, b)

# if, while
zahl = 10
if zahl > 10:
  print(zahl, " is big!")
else:
  print(zahl, " is small")

# Digitale Ausgänge
```

```
# GPIO-Pin 18 als digitalen Ausgang festlegen und auf high setzen
ledPin = 18
print(ledPin)
GPIO.setup(ledPin, GPIO.OUT)
GPIO.output(ledPin, True)

# Digitale Eingänge
buttonPin = 18
print(buttonPin)
GPIO.setup(buttonPin, GPIO.IN)
value = GPIO.input(buttonPin)
print(value)

# interner Pullup-Widerstand
switchPin = 18
print(switchPin)
GPIO.setup(switchPin, GPIO.IN, pull_up_down=GPIO.PUD_UP)

# Analoge Ausgänge
```

Listing 2: Quellcode in Python, programmierleitfaden

1.3 python-grundlagen

(Listing 3 python-grundlagen).

```
'''
  chmod a+x *.py
  python3 ./script.py
  % ju -- https://bw1.eu -- 3-11-18
  update:
  ================================
'''
import os
import copy
from datetime import datetime


## Variablen

a = 123
b = 12.34
c = "Hello"
d = 'Hello'
e = True

## Werte ausgeben

print("Werte ausgeben: ", a)

## Benutzereingaben einlesen

x = int(input("Eingabe Zahl: "))
print("Potenz: ", x ** 2)

## Arithmetik

# Temperatur in Celsius in Fahrenheit umgewandeln
tempC = int(input("Temperatur in C: "))
tempF = (tempC * 9) / 5 + 32
```

```
print(tempF)

## Strings

# Strings erzeugen
s = "abc def"
print("Strings erzeugen: ", s)
# Strings verketten (verbinden)
s1 = "abc"
s2 = str(23)
s = s1 + s2
print("Strings verketten: ", s)
# Strings in Zahlen umwandeln
zahl = int("-123")
kommazahl = float("00123.45")
bin_in_dez = int("1001", 2)# Binaerwert in eine Dezimalzahl
hex_in_int = int("AFF0", 16)# Hexadezimalzahl in einen Integerwert
print(zahl, kommazahl, bin_in_dez, hex_in_int)
# Laenge eines Strings
laenge = int(len("abcdef"))
print("Laenge eines Strings: ", laenge)
# Position in einem String suchen
s = "abcdefghi"
var = s.find("def")
print("Position in einem String suchen: ", var)
# Teilstring extrahieren
var = s[1:5]
print("Teilstring extrahieren: ", var)
# Teilstring durch einen anderen ersetzen
s = "It was the best of X. It was the worst of X."
var = s.replace("X", "times")# suchen, ersetzen
print("Teilstring durch einen anderen ersetzen: ", var)
# String in Gross- oder Kleinbuchstaben umwandeln
s = "aBcDe"
var = s.upper()
print("String in Grossbuchstaben umwandeln: ", var)
var = s.lower()
```

```
print("String in Kleinbuchstaben umwandeln: ", var)

## Befehle bedingt ausfuehren

x = 101
if x > 100:
  print("x is big")

x = 90
if x > 100:
  print("x is big")
elif x < 10:
  print("x is small")
else:
  print("x is medium")

## Vergleichsoperatoren

'''
  <, >, <=, >=, == oder !=
   < kleiner als
   > grösser als
  <= kleiner oder gleich
  >= grösser oder gleich
  == gleich
  != ungleich
'''

## Logische Operatoren

# and, or und not
x = 17
if x >= 10 and x <= 20:
  print('x is in the middle')

## Anweisungen genau x-mal ausfuehren
```

```
for i in range(1, 11):# 10x
  print(i)

## Befehle wiederholen, bis sich eine Bedingung aendert

antwort = ''
while antwort != 'X':
  antwort = input('Eingabe Antwort: ')
# Aus einer Schleife ausbrechen
while True:
  antwort = input('Eingabe Antwort: ')
  if antwort == 'X':
    break

## Funktion

def count_to_10():
  for i in range(1, 11):
    print(i)

count_to_10()

def count_to_n(n=10):
  for i in range(1, n + 1):
    print(i)

count_to_n()
count_to_n(5)

def count(from_num=1, to_num=10):
  for i in range(from_num, to_num + 1):
    print(i)

count()
count(4)
count(5, 12)
```

```
def make_bitte(satz):
  return satz + " bitte."

var = make_bitte("Gib den Kaese")
print(var)

## Listen

# Liste erzeugen
a = []# leere Liste
print("leere Liste: ", a)
a = [34, 'Fred', 12, False, 72.3]
print("Liste erzeugen: ", a)
# Auf Elemente einer Liste zugreifen
a = [34, 'Fred', 12, False, 72.3]
var = a[1]
print("Auf Elemente einer Liste zugreifen: ", var)
# Laenge einer Liste ermitteln
var = len(a)
print("Laenge einer Liste ermitteln: ", var)
# Elemente zu einer Liste hinzufuegen
a.append("new")
print("Elemente zu einer Liste hinzufuegen: ", a)
# Elemente aus einer Liste entfernen
a.pop()
print("Elemente aus einer Liste entfernen: ", a)
# Eine Liste durch Parsing eines Strings erzeugen
text = "abc; def; ghi"
var = text.split(';')
print("split-Befehl: ", var)
text = "abc -- def -- ghi"
var = text.split('--')
print("split-Befehl: ", var)
text = "abc, def, ghi"
var = text.split(',')
print("split-Befehl: ", var)
# Iteration ueber eine Liste
```

```
a = [34, 'Fred', 12, False, 72.3]
for x in a:
  print(x)
# Eine Liste durchzaehlen
for i in range(len(a)):
  print(i, a[i])
# Eine Liste sortieren
a = ["it", "was", "the", "best", "of", "times"]
a.sort()
print("Eine Liste sortieren: ", a)
# Eine Liste sortieren u. speichern
#import copy
a = ["it", "was", "the", "best", "of", "times"]
b = copy.copy(a)
b.sort()
print("Liste original: ", a)
print("Liste sortiert: ", b)
# Eine Liste zerlegen
l = ["a", "b", "c", "d"]
print("Liste: ", l)
liste = l[1:3]
print("l[1:3]: ", liste)
liste = l[:3]
print("Anfang l[:3]: ", liste)
liste = l[3:]
print("Ende l[3:]: ", liste)
liste = l[-2:]
print("Ende l[-2:]: ", liste)
liste = l[:-2]
print("Anfang l[:-2]: ", liste)
# Funktion auf eine Liste anwenden
l = ["abc", "def", "ghi", "ijk"]
print("Liste: ", l)
liste = [x.upper() for x in l]
print("Liste gross: ", liste)

## Dictionary
```

```
# speichert Schluessel-Wert-Paare ({'key':'value'})
phone_numbers = {'Simon':'01234 567899', 'Jane':'01234 666666'}
print("Dictionary: ", phone_numbers)
# Auf ein Dictionary zugreifen
var = phone_numbers['Simon']
print("Auf ein Dictionary zugreifen: ", var)
# Elemente aus einem Dictionary entfernen
var = phone_numbers.pop('Jane')
print("Elemente aus einem Dictionary entfernen: ", var)
print("Dictionary: ", phone_numbers)
# Iteration ueber Dictionaries
phone_numbers = {'Simon':'01234 567899', 'Jane':'01234 666666'}
for name in phone_numbers:
  print(name)
# oder
for name, num in phone_numbers.items():
  print(name + " " + num)

## Zahlen formatieren

x = 1.2345678
print("Zahl unformatiert: ", x)
var = "x = {:.2f}".format(x)
print("Zahlen formatieren: ", var)
var = "x = {:7.2f}".format(x)
print("Zahlen formatieren: ", var)
# Temperatur in Celsius und Fahrenheit ausgeben
c = 20.5           # Celsius
f = c * 9 / 5 + 32 # Fahrenheit
out = "Temperature {:5.2f} Grad C, {:5.2f} Grad F.".format(c, f)
print("Zahlen formatieren: ", out)

## Datum & Uhrzeit formatieren

#from datetime import datetime
d = datetime.now()
```

```
out = "{:%Y-%m-%d_%H:%M:%S}".format(d) # 2018-11-03_16:50:05
print("Datum & Uhrzeit formatieren: ", out)
out = "{:%Y-%b-%d}".format(d) # 2018-Nov-03
print("Datum & Uhrzeit formatieren: ", out)
out = "{:%d-%m-%Y}".format(d) # 03-11-2018
print("Datum & Uhrzeit formatieren: ", out)
out = "{:%H:%M}".format(d)    # 16:54
print("Datum & Uhrzeit formatieren: ", out)
out = "{:%d-%b-%y}".format(d) # 03-Nov-18
print("Datum & Uhrzeit formatieren: ", out)

## Funktion  - Mehrere Werte zurueckliefern

# eine Temperatur in Kelvin sowohl in Fahrenheit
# als auch in Celsius umwandeln
def calculate_temperatures(kelvin):
  celsius = kelvin - 273
  fahrenheit = celsius * 9 / 5 + 32
  return celsius, fahrenheit

c, f = calculate_temperatures(0) # kelvin
print("Funktion  - Mehrere Werte zurueckliefern: ", c, f)

## Eine Klasse definieren

'''
  OOP = Wiederverwendbarkeit, Datenkapselung
  Klasse = Bauplan fuer Objekte
  Person = Bezeichner wird gross geschrieben
  Objekte = Eigenschaften, Attribute, Zustand
            und Methoden, Fuktionen, Verhalten, Operationen
  self = this Objektreferenz, verweist auf das akt. Objekt
  pass = leere Anweisung
  obj = Person() # Zuweisung einer Variablen, Instanzieren
  Konstruktor = Objekt erzeugen
'''
```

```
# Klasse fuer ein Adressbuch
class Person:
  '''Diese Klasse repraesentiert ein Person-Objekt'''
  def __init__(self, name, tel):#
    '''
      Variable namens name, die fuer jedes Mitglied (Member)
      der Klasse Person zugaenglich ist.
      Diese Variable wird mit dem Wert initialisiert,
      der uebergeben wird, wenn eine neue Instanz erzeugt wird.
    '''
    # Member-Variablen
    self.name = name
    self.tel = tel
  # Eine Methode definieren
  def ausgeben(self):
    return self.name + " " + self.tel


# Person-Objekt obj
# Instanzieren
obj1 = Person("Simon", "01234567")
obj2 = Person("Willi", "01334567")
# Zugriff auf Eigenschaften
print("Zugriff auf Eigenschaften: ", obj1.name, obj1.tel)
print("Zugriff auf Eigenschaften: ", obj2.name, obj2.tel)
# Zugriff auf Methoden
print("Zugriff auf Methoden: ", obj1.ausgeben())
print("Zugriff auf Methoden: ", obj2.ausgeben())


class Mensch:
  '''Diese Klasse repraesentiert ein Person-Objekt'''
  def __init__(self, first_name, surname, tel):
    self.first_name = first_name
    self.surname = surname
    self.tel = tel
  def full_name(self):
```

```
    return self.first_name + " " + self.surname

# Vererbung von Superklasse Mensch
class Mitarbeiter(Mensch):
  def __init__(self, first_name, surname, tel, gehalt):
    super().__init__(first_name, surname, tel)
    self.gehalt = gehalt

  def erhoehen(self, menge):
    self.gehalt = self.gehalt + menge

# Instanzieren
obj3 = Mensch("Ali", "Simon", "01234567")
obj4 = Mitarbeiter("Ali", "will", "012554567", 2100.0)
obj5 = Mitarbeiter("Olek", "Karl", "012754567", 2700.0)
# Zugriff auf Eigenschaften
print("Zugriff auf Eigenschaften: ", obj3.first_name, obj3.surname , obj3.
    tel)
print("Zugriff auf Eigenschaften: ", obj4.first_name, obj4.surname , obj4.
    tel, obj4.gehalt)
print("Zugriff auf Eigenschaften: ", obj5.first_name, obj5.surname , obj5.
    tel, obj5.gehalt)
# Zugriff auf Methoden
print("Zugriff auf Methoden: ", obj3.full_name())
obj4.erhoehen(200)
print("Zugriff auf Eigenschaften: ", obj4.gehalt)
summe = obj4.gehalt + obj5.gehalt
print("Gehalt gesamt: ", summe)

## In eine Datei schreiben

'''
  # Dateimodi
  Modus Beschreibung
  r Lesen (Read)
  w Schreiben (Write)
  a Anhaengen (Append) - an das Ende einer vorhandenen Datei
```

```
  b Binaermodus
  t Textmodus (Standard)
  + Kuerzel fuer r+w
'''

f = open('datei.txt', 'w')
f.write('This file is not empty')
f.close()

## Aus einer Datei lesen

try:
  f = open('datei.txt')
  s = f.read()
  f.close()
except IOError:
  print("Cannot open the file")
```

Listing 3: Quellcode in Python, python-grundlagen

1.4 test

(Listing 4 test).

```
# test.py

import time
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)

ledPin = 18
sek = 2
GPIO.setup(ledPin, GPIO.OUT)


try:
    while True:
        GPIO.output(ledPin, False)
        print("led aus")
        time.sleep(sek)
        GPIO.output(ledPin, True)
        print("led ein")
        time.sleep(sek)

finally:
    print("Cleaning up")
    GPIO.cleanup()
```

Listing 4: Quellcode in Python, test