Python Cheat Sheet

If, else if, else

```
foo = 3

if foo == 3:
  print('foo is 3')
elif foo == 2:
  print('foo is 2')
else:
  print('foo is neither 3 nor 2')
```

Basic Input / Output

```
# Print a message
print('Hello World')

# Print multiple values (of different types)
foo = 'cruel'
print('Goodbye', foo, 'World') # => Goodbye cruel World

name = input('What is your name?')
```

Imports / time

```
# Some Libraries
import time
import os
import random
import RPI.GPIO as GPIO

time.time() # current System time
time.sleep(s) # sleep in seconds
```

Loops

```
for i in range(10):
  print(i)

while True:
  print("Hello World")

break # Terminates current loop and resumes execution at the next statement

continue # skips the execution of the rest of the code inside the loop for the current iteration only
```

GPIO

```
# Use (Boardcom-) Chip-Nummer
GPIO.setmode(GPIO.BCM)

#Use Pin-Numbers
GPIO.setmode (Pin, GPIO.BOARD)

# Set GPIO Pins Type
GPIO.setup(Pin, GPIO.IN) # Input
GPIO.setup(Pin, GPIO.OUT) # Output

# Set GPIO Pins
GPIO.output(Pin, GPIO.HIGH) # 1, True => on
GPIO.output(Pin, GPIO.HIGH) # 0, False => off

#Eventhandler
GPIO.add_event_detect(TASTER_PIN, GPIO.RISING, callback=function, bouncetime=200) # GPIO.RISING, GPIO.FALLING, GPIO.BOTH

# Cleanup GPIO Pins (Set all Pins to input)
GPIO.cleanup()
```

Lists and Dictionaries

```
# Lists
my_list = []
len(my_list)
my_list[i]
my_list.append(x)

# Dictionarys
my_dict = {}
len(my_dict)
my_dict['key']
my_dict['key'] = 'value'
```

Try / except / finally

```
try:
  # Code Block
except EXCEPTION:
  # error handeling
except OTHER_EXCEPTION:
  # error handeling
else:
  # execute code when there is no error
finally:
  # is always executed

# Some exceptions: KeyboardInterrupt, EOFError, ValueError
```

Comparing Values

```
# Are two values equal?
foo == 3

# Are two values not equal?
foo != 3

# Less than another?
foo < 3

# Greater than another?
foo > 3

# Less or equal to?
foo <= 3

# Greater or equal to?
foo >= 3

# Boolean: True or False
```

Funktionen

```
def my_function():
  return 0

def my_function(args)
    return args
```