Python Cheat Sheet / Spickzettel

Eingabe und Ausgabe

```
print('Hallo Python!')

ndays = 365
print('There are', ndays, 'in a year')
print(f'There are' {ndays} in a year')

name = input('What is your name? ')
num = int(input('Enter a number: '))
```

String

```
'Hello' oder "Python"
text = 'Hello' + 'Python'
'Hello…' * 4
len('Hello Python')
int('365')
text[0]   # erstes Element
text[-1]  # letztes Element
text[0:3] # 'Hel'
```

String Methoden

```
text = "Hello World"
text.upper()
text.lower()
text.count('l')
text.strip()
text.split(' ')
text.replace('e', 'a')
text.replace('l','')
```

Vergleichsoperatoren

<	kleiner als
<=	kleiner oder gleich
>	größer als
>=	größer oder gleich
==	gleich
!=	nicht gleich

Logische Operatoren

and	logisches UND
or	logisches ODER
not	logisches NICHT

Schleifen

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

total = 0
for i in range(10):
    total = total + i
print(total)

text = 'Hallo'
for c in text:
    print(c)

msg = 'Ich liebe Python!'
for i in range(len(msg)):
    print(i, msg[i])

i = 0
While i < 10:
    print(f"Nummer {i}")
    i = i + 1

while True:
    print("Endlosschleife")
```

Bedingungen

```
x = 3
if x == 3:
    print('x is 3')

zahl = 80
if zahl >= 50:
    print('pass')
else:
    print('fail')

zahl = 80
if zahl >= 65:
    print('ok')
elif zahl >= 50:
    print('pass')
else:
    print('fail')

text = 'hallo'
if text == 'hallo':
    print('hey')
```

Datentypen

int	123	ganze Zahl
float	1.23	Kommazahl
bool	True/False	Logischer Wert
str	'Hello'	Zeichenkette
list	[0, 1]	Liste
dict	{"a": 1}	Wörterbuch
None		kein Wert

Module

```
import time
import time as ZEIT
from time import sleep
from time import sleep as WAIT
```

Variablen

```
celsius = int(input('Temperature: '))
fahrenheit = celsius * 9 / 5 + 32
print(fahrenheit, 'Fahrenheit')
```

Mitgliedschaftsoperatoren

in	Wert in Objekt
not in	Wert nicht in Objekt

Wörterbuch

```
d = {}       # leeres Wörterbuch
d = {"a": 1}
d.keys()     # Key Liste
d.values()   # Value Liste
d.items()    # Key, Value Liste
d["a"]       # Value
d["b"] = 2   # hinzufügen
```

Liste

```
l = []        # leere Liste
l = [1, 2, 3]
l.append(x)  # hinzufügen
l[0]      # erstes Element
l[-1]     # letztes Element
```

MicroPython Cheat Sheet

Digitaler Ausgang

```
from machine import Pin

led = Pin(2, Pin.OUT)
led.on()       # LED an
led.off()      # LED aus
led.value(1)   # LED an
led.value(0)   # LED aus
```

Onboard LED (Raspberry Pi Pico (W))

```
from machine import Pin

led_onboard = Pin("LED", Pin.OUT)
led_onboard.on()       # LED an
led_onboard off()      # LED aus
led_onboard.value(1)   # LED an
led_onboard.value(0)   # LED aus
```

Servo / PWM

```
from machine import Pin
from machine import PWM

servo_pin = Pin(0, Pin.OUT)
# 50 Hz -> 0.02 s = 20 ms = 20000 us = 20000000 ns
servo = PWM(servo_pin, freq=50)
# 5% (minimum)  -> 1000000 ns = 1000 us = 1 ms
servo.duty_ns(1000000)
# 7.5% (medium) -> 1500000 ns = 1500 us = 1.5 ms
servo.duty_ns(1500000)
# 10% (maximum) -> 2000000 ns = 2000 us = 2 ms
servo.duty_ns(2000000)
```

Digitaler Eingang

```
from machine import Pin

button = Pin(4, Pin.IN, Pin.PULL_UP)
button_state = button.value()
```

Zeit Bibliotheken

```
import time
import utime

time.sleep(1)    # 1 sek
time.sleep(0.5) # ½ sek

utime.sleep_us(10) # 10 µs
```

Analoger Ausgang / PWM

```
from machine import Pin
from machine import PWM

led_pin = Pin(2, Pin.OUT)
led_pwm = PWM(led_pin, freq=500, duty_u16=0)
led_pwm.duty_u16(0)      # LED aus
led_pwm.duty_u16(32767)  # LED halb hell
led_pwm.duty_u16(65535)  # LED ganz hell
```

Analoger Eingang / ADC

```
from machine import Pin
from machine import ADC

analog_pin = Pin(26)
adc = ADC(analog_pin)
adc_value = adc.read_u16()  # 16 bit -> 65535
adc_voltage = adc_value * 3.3 / 65535
```

Zufallszahl

```
from random import choice

choice([0, 1])

choice([0.25, 0.5, 1])
```

Temperatursensor onboard (Raspberry Pi Pico (W))

```
from machine import ADC

temp_adc = ADC(4)
adc_value = temp_adc.read_u16()
adc_voltage = adc_value * 3.3 / 65535
temperature = 27 - (adc_voltage - 0.706) / 0.001721
```

Pulslänge am Eingang messen

```
from machine import Pin, time_pulse_us

input_pin = Pin(10, mode=Pin.IN)
# High-Puls (1) messen
pulse_time = time_pulse_us(input_pin, 1, TIMEOUT_US)
```