Python 3 Cheat-Sheet

Version 2021–07–08_1521Z

Funktionsdefinition (hier mit 2 Parametern)

```
def funktionsname(parameter1, parameter2):
    Programmcode  oder  pass
```

Funktionsaufruf

```
funktionsname()
funktionsname(Wert1, Wert2)
```

Zuweisung (falls innerhalb einer Funktion, dann automatische Deklaration als lokale Variable)

```
variablenname = Wert
```

Globale Variablen innerhalb von Funktionen

```
def funktionsname(parameter):
    global variablenname
    …
    variablenname = Wert
```

Kontrollstrukturen / Ablaufsteuerung

```
if Bedingung:
    Programmcode (für Bedingung wahr)
elif Bedingung2:
    Programmcode (für 1. Bed. falsch, aber 2. wahr)
else:
    Programmcode (für Bedingungen falsch)

while Bedingung:
    Programmcode
    (wird ausgeführt/wiederholt solange Bedingung wahr)

for variablenname in range(Endwert):
    Programmcode
    (wird wiederholt solange Variable kleiner als Endwert)

for variablenname in range(Startwert, Endwert):
    Programmcode
    (wird wiederholt solange Variable kleiner als Endwert)

for variablenname in Liste:
    Programmcode
    (wird für alle Elemente in der Liste je einmal ausgeführt)

break  (vorzeitiger Abbruch einer Schleife)
return  (Funktion beenden, ohne Rückgabewert)
return Wert  (Funktion beenden, mit Rückgabewert)
raise Fehlerklasse("Meldung")  (Fehler „werfen“)
raise SystemExit(Wert)  (Programm beenden)
```

Wichtige Funktionen (ggf. mit Rückgabe)

```
print(String)  → keine Rückgabe bzw. None
input(String)  → Eingabe als String
str(Wert)  → Wert in String umgewandelt
int(Wert)  → Wert in Ganzzahl umgewandelt
float(Wert)  → Wert in Gleitkommazahl umgewandelt
```

Wahrheitswerte

```
True  (immer wahr)
False  (immer falsch)
```

Das „Nichts“

```
None
```

Operatoren

```
Wert1 == Wert2  → wahr wenn Werte gleich
Wert1 != Wert2  → wahr wenn Werte ungleich
Wert1 is Wert2  → wahr wenn Werte dasselbe
Wert1 is not Wert2  → wahr wenn nicht dasselbe
not Wert  → wahr wenn falsch und falsch wenn wahr
Wert1 and Wert2  → nur wahr falls beide wahr
Wert1 or Wert2  → wahr falls mindestens ein Wert wahr
Zahl1 + Zahl2  → Summe der Zahlen
String1 + String2  → beide Texte hintereinandergefügt
Zahl1 / Zahl2  → Division mit Gleitkomma
Zahl1 // Zahl2  → Ganzzahldivision (mit Abrundung)
Zahl1 % Zahl2  → Modulo-Operator bzw. Divisionsrest
Zahl1 ** Zahl2  → Zahl1 hoch Zahl2
```

Arbeiten mit Listen

```
[]  → neue, leere Liste
[Wert1, Wert2, Wert3]  → neue Liste mit Werten
len(Liste)  → Anzahl der Elemente in der Liste
Liste[Position]  → Wert an Position
Liste[Position] = Wert  (existierenden Wert überschreiben)
Liste.append(Wert)  → keine Rückg., aber Liste verläng.
Liste.clear()  → keine Rückgabe, aber Liste geleert
Liste.copy()  → neue Liste mit gleichem Inhalt
Wert in Liste  → wahr falls Wert in Liste enthalten
Wert not in Liste  → wahr falls Wert nicht in Liste
```

Exceptionhandling / Fehlerbehandlung

```
try:
    Programmcode (wird bei Fehler abgebrochen)
except Fehlerklasse:
    Programmcode (für Fehlerfall)
else:
    Programmcode (für Erfolgsfall)
finally:
    Programmcode (wird immer ausgeführt)
```

Python 3 Cheat-Sheet

Version 2021–07–08_1521Z

Online-Dokumentation: `https://docs.python.org/`

- „Tutorial“ Einführung in die Programmiersprache
- „Library Reference“ Beschreibung aller Funktionen, Datentypen, Module, usw.

Shebang / Hash-Bang am Programmanfang

```
#!/usr/bin/env python3
```

Kommentare

```
# Zeilen die mit einem Hashzeichen
# (Doppelkreuz, Rautezeichen) beginnen
# sind Kommentare und werden bei der
# Programmausführung ignoriert
```

Modul (Programmbibliothek) importieren und Funktion nutzen

`import` *modulname*
…
modulname`.`*funktionsname*`(`…`)`

Einzelne Funktion importieren und nutzen

`from` *modulname* **`import`** *funktionsname*
…
funktionsname`(`…`)`

Modul importieren ohne beim Zugriff den Modulnamen nutzen zu müssen

`from` *modulname* **`import *`**
…
funktionsname1`(`…`)`
funktionsname2`(`…`)`

Module für Mathematik und Zufall

```
import math
math.sqrt(9) → 3
math.cos(math.pi) → -1

import random
random.random() → Zahl ≥ 0 und < 1
random.randrange(Startwert, Endwert)
                → Ganzzahl ≥ Startwert und < Endwert
random.randint(Startwert, Endwert)
                → Ganzzahl ≥ Startwert und ≤ Endwert
```

Objektorientierung

`class` *Klassenname*`:`
    Klassendefinition oder **`pass`**

`class` *Klassenname*`(`*Basisklassenname*`):`
    Klassendefinition oder **`pass`**

```
class Spieler:

    def __init__(self, name):
        self.name = name

    def gewonnen(self):
        print(self.name + " gewinnt")

    def verloren(self):
        print(self.name + " verliert")

spieler1 = Spieler("Alice")
spieler2 = Spieler("Bob")
spieler1.gewonnen()
```

Wichtige Fehlerklassen

```
SystemExit (Programm- oder Thread-Beendigung)
Exception (alle normalen Ausnahmezust. und Fehler)
 ValueError (unerwarter Wert, z.B. bei Typ-Umw.)
 ArithmeticError (mathematischer Fehler)
  OverflowError (Zahlenwert zu groß)
  ZeroDivisionError (Division durch Null)
 LookupError (Index oder Schlüssel ungültig)
  IndexError (num. Index ungültig, z.B. bei Liste)
  KeyError (Schlüssel ungültig, z.B. bei Wörterbuch)
 AssertionError (irrige Annahme)
 MemoryError (nicht genügend Arbeitsspeicher)
 RuntimeError (allgemeiner Laufzeitfehler)
  NotImplementedError (nicht impl. Funktion)
  RecursionError (zu tiefe Rekursion)
```

Hacken Craften Funken e.V.
www.hacrafu.de/cheat-sheets