Python 3 Grundlagen Cheat Sheet

by [deleted] via cheatography.com/37514/cs/11784/

Nützliche Links

Offizielle Python 3.6 Dokumentation
https://docs.python.org/3.6/

Stackoverflow mit Tag für Python 3
https://stackoverflow.com/questions/tagged/python-3.x

Häufige Fehler

Beschreibung der Fehlerursache	falsch *statt* richtig
Groß-/Kleinschreibung inkorrekt	`wahr = true` *statt* `wahr = True`
Leere Klammer fehlt nach Funktionsname	`sprite.down` *statt* `sprite.down()`
Gleitkommazahl in Komma-Schreibweise	`meinFloat = 1,37` *statt* `meinFloat = 1.37`

Datentypen

Typ	Beispielwerte `Deklaration`
Integer	-23, 42, 5 `ganzzahl = 1`
Float	-1.72, 3.141 `gleitzahl = 1.3`
String	"Hi", "Hey du!" `text = 'Hallo'`

Datentypen (cont)

Boolean	True, False `wahrheit = False`
Tupel	(1, 2, 3), ('Eins', 2, 3.45) `unveraenderbar = (1, 2, 3)`
Liste	[1, 2, 3], ['Eins', 2, 3.45] `veraenderbar = [1, 2, 3]`
Dictionary	{name:'hans', alter: 7,...} `person = {name: 'Hans', alter: 7}`
Iterator	string, range, andere Objekte `irgendwas = range(7)`

Ausgabe

`print( 'Hello World')`	Einzelner String
`print( 'Hi', end='')`	Ohne Zeilenumbruch
`print()`	Leere Zeile
`name = 'John'` `print( 'Hello', name)`	String und Variable

Formatiert: *(z.B. zwei Nachkommastellen)*

```
waehrung = 'Euro'
kosten = 3.1572
print( 'Betrag: {:.2f} {}'.format(kosten, waehrung))
>>> Preis: 3.15 Euro
```

Eingabe

```
eingabe = input('Eingabe: ')
try:
        eingabe = int(eingabe)
except ValueError as err:
        print( 'Fehler! Keine Ganzzahl!')
        print( 'Meldung: {}'.format(err))
```

Der Datentyp von **input()** ist immer String, daher ggf. in anderen Datentyp umwandeln und dabei Fehlerbehandlung nutzen.
Nicht die Funktion **eval()** zur Umwandlung nutzen!

Operator

`x+y`	Addition
`x-y`	Subtraktion
`x*y`	Multiplikation
`x/y`	Division
`x**y`	Exponentiation
`x%y`	Modulo (Rest bei ganzahliger Division)

Funktionen deklarieren

```
# Allgemein:
def <Funktionsname>(<Parameter>):
    <Anweisungen>
        return <Rückgabewert>

# Beispiel:
def addiere(x, y):
        ergebnis = x + y
        return ergebnis
```


Sponsored by **Readable.com**
Measure your website readability!
https://readable.com

Bedingte Ausführung

```
# Allgemein:
if <be din gun:g>
    <an wei sun gen>
elif <an der e_b edi ngu ng>
    <an der e_a nwe isu nge n>
else:
    <we ite re_ anw eis ung en>
# Beispiel:
x = 7
if x == 5:
        pri nt('Die Zahl ist
eine Fünf')
elif x == 7:
        pri nt('Die Zahl ist
eine Sieben')
else:
        pri nt('Die Zahl ist
weder 5 noch 7')
```

Der ***elif***-Block ist optional und kann beliebig oft verwendet werden.

Bedingungen

<	kleiner als	`wert < 10`
>	größer als	`wert > 5`
==	gleich	`wert == 23` `passwort == " geh ei m "`
<=	kleiner gleich	`5 <=7`
>=	größer gleich	`8 >= 5`

Bedingungen (cont)

!=	ungleich	`78 != 93` `" Pet er" != " pet er"`
in	(entha-lten) in	`" b" in "Libelle"` `" pet er" in userListe`
not in	nicht in	`" y" not in "Vogel"` `" pet er" not in userLis te`

Wiederholungen / Schleifen

```
# for-Schleife
for i in range( 0,10):
        pri nt( 'Du rchlauf
Nr.', i)
# while- Sch leife
x = 5
while x > 0:
        pri nt('x: {}'.fo rma -
t(x))
       x -= 1
# Funktionen zur Schlei fen -
ste uerung
for i in range( 0,10):
       if i == 3:
         continue    # Nächster
Durchlauf
       if i == 7:
         break       # Abbruch
der Schleife
      pri nt(i, new=', ')
>>> 0, 1, 2, 4, 5, 6,
```

Fehlerbehandlung

```
# Allgemein
try:
        # Anweis ungen
except Except ionTypeas err:
        # Anweis ungen
        # Zugriff auf Fehler -
mel dung: err
except:
        # Anweis ungen

# Beispiel
def teilen(a, b):
        erg ebnis = None
       try:
                erg ebnis = a /
b
        except ZeroDi vis ion -
Error as err:
                pri nt( 'Fe -
hler: Division durch Null!')
                pri nt( 'Me -
ldung: {}'.fo rma t(err))
        return ergebnis
print( tei len(15, 0))
```

ExceptionType möglichst explizit angeben

Arbeiten mit Strings

`len(m  yStr)`	Anzahl der Zeichen von myStr
`m yStr.lo  wer()` `m yStr.up  per()`	Umwandlung zu Klein-/Gr-oßbuchs-taben
`m yStr.replace(old,new )`	Ersetzt in myStr 'old' durch 'new'
`m yStr.sp lit (char)`	Teilt an 'char' auf, erzeugt Liste

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com

Arbeiten mit Strings (cont)

`m yStr.st rip ()`	Entfernt Leerzeichen (whitespace) am Anfang und Ende von myStr
`m yStr [1:5]`	Zeichen 1-5 ausschneiden

Arbeiten mit Listen

`len(m  yList)`	Anzahl der Elemente
`x in myList`	True, wenn x ein Element in myList
`myList.append(x)`	x an myList anhängen
`myList[i] = x`	Element an Stelle i ersetzen durch x
`'c'.jo in( myLis t)`	Verbindet Elemente der Liste zu einem String, getrennt durch c

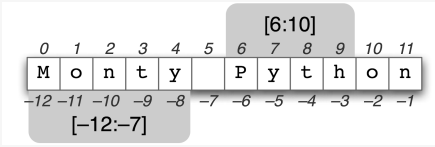
Iterieren über Listen *(ohne Index)*

```
for element in myList:
        pri nt( 'We rt:', elemen
t)
```

Iterieren über Listen *(mit Index)*

```
for i, element in enumer ate (my
 List):
        pri nt( 'Po sit ion:', i
)
        pri nt( 'We rt:', elemen
t)
```

Slicing von Listen (incl. Strings)



Beispiele

```
myList [-1 ]   # letztes Element
myList [-2 :]  # letzten 2 Eleme
nte
myList [:- 2]  # alles bis auf d
ie letzten zwei
myList [2: -1: 2] # Vom dritten
bis vorletzten Element, aber nur
 jedes zweite
```

Arbeiten mit Dictionaries

`len(m  yDict)`	Anzahl der Einträge
`del m yDic t[m yKey]`	Löscht Schlüssel myKey
`list(  m y Dic t.k e - ys())`	Liste aller Schlüssel
`list(m yDict.values() )`	Liste aller Werte
`list(m  yD ict.it ems ())`	Liste von Tupeln
`myKey in m yDict`	True, wenn myKey vorhanden

Iterieren über Dictionaries:

```
m yDict = {'name ':' Peter', 'al
ter ':'19'}
for key, value in m yDic t.i tem
s():
        pri nt( 'Sc hlü ssel:',
key)
        pri nt( 'We rt:', value)
```


Sponsored by **Readable.com**
Measure your website readability!
https://readable.com