Cheatography

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
```

Cheatography

Python 3 Grundlagen Cheat Sheet

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

Bedingte Ausführung

```
# Allgemein:
if <bedingung>:
    <anweisungen>
elif <andere_bedingung>:
    <andere_anweisungen>
else:
    <weitere_anweisungen>
# Beispiel:
x = 7
if x == 5:
    print('Die Zahl ist eine Fünf')
elif x == 7:
    print('Die Zahl ist eine Sieben')
else:
    print('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 == "geheim"`
<=	kleiner gleich	`5 <=7`
>=	größer gleich	`8 >= 5`

Bedingungen (cont)

!=	ungleich	`78 != 93` `"Peter" != "peter"`
in	(enthalten) in	`"b" in "Libelle"` `"peter" in userListe`
not in	nicht in	`"y" not in "Vogel"` `"peter" not in userListe`

Wiederholungen / Schleifen

```
# for-Schleife
for i in range(0,10):
    print('Durchlauf Nr.', i)
# while-Schleife
x = 5
while x > 0:
    print('x: {}'.format(x))
    x -= 1
# Funktionen zur Schleifensteuerung
for i in range(0,10):
    if i == 3:
        continue    # Nächster Durchlauf
    if i == 7:
        break       # Abbruch der Schleife
   print(i, new=', ')
>>> 0, 1, 2, 4, 5, 6,
```

Fehlerbehandlung

```
# Allgemein
try:
    # Anweisungen
except ExceptionType as err:
    # Anweisungen
    # Zugriff auf Fehlermeldung:
err
except:
    # Anweisungen

# Beispiel
def teilen(a, b):
    ergebnis = None
    try:
        ergebnis = a / b
    except ZeroDivisionError as err:
        print('Fehler: Division durch Null!')
        print('Meldung: {}'.format(err))
    return ergebnis
print(teilen(15, 0))
```

ExceptionType möglichst explizit angeben

Arbeiten mit Strings

`len(myStr)`	Anzahl der Zeichen von myStr
`myStr.lower()` `myStr.upper()`	Umwandlung zu Klein-/Großbuchstaben
`myStr.replace(old,new)`	Ersetzt in myStr 'old' durch 'new'
`myStr.split(char)`	Teilt an 'char' auf, erzeugt Liste

Cheatography

Python 3 Grundlagen Cheat Sheet

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

Arbeiten mit Strings (cont)

`myStr.strip()`	Entfernt Leerzeichen (white-space) am Anfang und Ende von myStr
`myStr[1:5]`	Zeichen 1-5 ausschneiden

Arbeiten mit Listen

`len(myList)`	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'.join(myList)`	Verbindet Elemente der Liste zu einem String, getrennt durch c

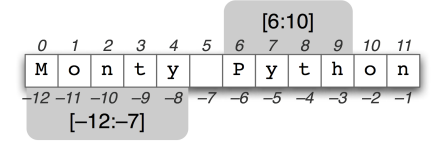
Iterieren über Listen *(ohne Index)*

```
for element in myList:
    print('Wert:', element)
```

Iterieren über Listen *(mit Index)*

```
for i, element in enumerate(myList):
    print('Position:', i)
    print('Wert:', element)
```

Slicing von Listen (incl. Strings)



Beispiele

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

Arbeiten mit Dictionaries

`len(myDict)`	Anzahl der Einträge
`del myDict[myKey]`	Löscht Schlüssel myKey
`list(myDict.keys())`	Liste aller Schlüssel
`list(myDict.values())`	Liste aller Werte
`list(myDict.items())`	Liste von Tupeln
`myKey in myDict`	True, wenn myKey vorhanden

Iterieren über Dictionaries:

```
myDict = {'name':'Peter', 'alter':'19'}
for key, value in myDict.items():
    print('Schlüssel:', key)
    print('Wert:', value)
```