Kontrollstrukturen

Philipp Hanisch, Valentin Roland

6. Oktober 2022

Python-Grundlagen

Gliederung

1. Was sind Kontrollstrukturen?
2. Verzweigungen ("If-Statements")
3. While-Schleife
4. Intermezzo: Listen
5. For-Schleife

Was sind Kontrollstrukturen?

Wozu?

- Strukturieren Programmverhalten
- für alle imperativen Sprachen ähnlich
- z.B.: Verzweigung, Zählschleife

Kontrollstrukturen in Python

Python kennt:

- Ein- / Zweiseitige Verzweigung: `if .. : / if .. : .. else: ..`
- Zählschleife: `for .. in .. :`
- Kopfgesteuerte Schleife: `while .. :`

Verzweigungen ("If-Statements")

Einseitige Verzweigungen

```
1     if a == b:
2         print ("a is equal to b!")
3     print ("this is printed every time. bye.")
4
```

Zweiseitige Verzweigungen

```
1     if a == b:
2         print ("a is equal to b!")
3     else:
4         print ("a is not equal to b!")
5         # be careful: 1 == 1, but 1 != "1" !
6
7     print ("this is printed every time. bye.")
8
```

Verschachtelte Verzweigungen

```
1     if a < b:
2         print ("a is less than b!")
3     else:
4         if a == b:
5             print ("a == b!")
6         print ("a >= b!")
7         # is also executed when a == b!
8
```

Kurzform elif Verzweigungen

Oder mit `elif`:

```
1     if a < b:
2         print ("a is less than b!")
3     elif a == b:
4         print ("a == b!")
5     else:
6         print ("a > b!")
7         # is not if when a == b!
8
```

While-Schleife

While-Schleife

→ Wiederholt, solange Bedingung erfüllt ist:

```
1     a = 10
2     while a > -1:
3         a -= 1
4
```

Vorzeitiges Abbrechen

```
1    a = 10
2    while a > -1:
3        a -= 1
4        if a > 3:
5            continue # jump to next interation
6        print ("countdown:", a) # only printed for 3,2,1
7        if a == 1:
8            break # break out of loop immediately
9
```

Intermezzo: Listen

Was sind Listen?

Wie alltägliche Listen. In Python:

- `[]`, bzw. `[1,2,3]`
- Methoden: `insert()`, `append()`, `pop()`, `remove()`, `index()`
- siehe auch `help([])`

Wie verwende ich Listen

```
1     a = [1,5,3,2] # create new list
2     a.sort() # a is now sorted
3     print (a[3]) # 4th element of sorted list -> 5
4     highest = a.pop() # removes last element (5) and returns it
5     highest += 1
6     a.insert(0, highest) # add 6 at the start
7     print (a) # -> [6,1,2,3]
8
```

For-Schleife

For-Schleife

Klassische Zählschleife gibt es nicht in Python

↪ `for` durchläuft Listen

```
1    a = [1,2,3]
2    for elem in a:
3        print (elem)
4
```

Wie zählen?

```
1    for index in range(10):
2        print (index)
3    # prints numbers 0 to 9
4
```

```
1    a = [1,2,3]
2    for index, elem in enumerate(a):
3        print (index, elem)
4
```