

EINFÜHRUNG IN DIE PROGRAMMIERUNG MIT PYTHON

Fingerübungen



Woche 02 – 18. Mai 2016

Datentyp bool

Boolesche Werte und Operatoren, Vergleichsoperatoren

(1) Typkonvertierung

Experimentieren Sie mit den Funktionen `int()`, `float()`, `str()` und `bool()`. Welche (Parameter-)Werte führen zu Fehlern? Überlegen Sie sich insbesondere, für welche Werte `bool()` zu *True* bzw. *False* auswertet.

Interessant sind z. B.:

- `int(2.8)`
- `float(True)`
- `bool(-1)`
- `int()`
- `int(False)`
- `bool("")`

(2) Boolesche Operationen

Nutzen Sie die Operatoren `==`, `>`, `<`, `<=`, `>=` und `!=` für einfache arithmetische Vergleiche wie bspw. `5 < 3` oder `a == b`. Verknüpfen Sie mehrere solcher (booleschen) Vergleiche mithilfe der Operatoren `and`, `or` und `not`.

Interessante Beispiele (zuerst überlegen, dann ausprobieren!):

- `not not True`
- `bool(2*7.7 - 5*3) and True`
- `"a" >= "A"`
- `not 5 > 2 or (2/2 == 1)`
- `"ABCD" >= "AD"`
- `5%2 < 1 or (3 != 4 and 2**3 < 7)`

if-Abfragen

Bedingte Anweisungen (`if-elif-else`): Fallunterscheidung mit `booleans`

(1) Sortieren (minimal)

Schreiben Sie ein Stück Code, das die Werte zweier Variablen `a` und `b` genau dann vertauscht, wenn der Wert von `a` größer ist als der Wert von `b`.

(2) Altersgruppen

Erstellen Sie ein Programm, das das Alter von dem/der Nutzer/in abfragt und in Abhängigkeit davon dessen „Altersgruppe“ ermittelt und ausgibt. Dabei soll Folgendes gelten:

Alter:	0	1 bis 13	14 bis 17	18 bis 130
Einteilung:	Säugling	Kind	Jugendliche(r)	Erwachsene(r)

Sie dürfen davon ausgehen, dass der Nutzer nur Zahlenwerte (d. h. Integer im Bereich $]-\infty, \infty[$) eingibt. Beachten Sie jedoch, dass `input()` immer Zeichenketten (`str`) zurückgibt und der entsprechende Wert ungültig sein kann (z. B. falls `alter > 130`).

if-Abfragen

Bedingte Anweisungen (if-elif-else): Fallunterscheidung mit booleans

(3) [Zusatzaufgabe] Intervallvergleiche

In Python gibt es die sog. Intervallvergleiche (engl. „*interval comparison*“), welche Sie nun in Ihrer Lösung aus Aufgabe (2) anstatt der „normalen“ Vergleiche benutzen sollen.

Beispiel: a soll im Bereich von 3 bis 17 liegen (jeweils inklusive)

- „normal“: $a \geq 3 \text{ and } a \leq 17$
- Intervallvergleich: $3 \leq a \leq 17$

while-Schleifen

while(-else)-Statement, Zählervariablen, break und continue

(1) Erste Versuche

Implementieren Sie eine Schleife, die eine Zählervariable (*i*) nutzt, um die Zahlen (*int*) von 0 bis 9 nacheinander auszugeben (mit `print()`).

(2) Fencepost-Problem

Erstellen Sie ein Programm, das alle (ganzen) Zahlen von 1 bis *n* in einem String (*str*) mit Kommas zusammenfügt und diesen ausgibt. *n* sei dabei eine Variable vom Typ *int*, wobei Sie davon ausgehen dürfen, dass gilt: $n \geq 1$. Es sollen sich also bspw. folgende Ausgaben ergeben:

n:	1	2	3	4
Ausgabe:	'1'	'1, 2'	'1, 2, 3'	'1, 2, 3, 4'

- (3) Könnte der `else`-Teil nicht einfach hinter (d. h. unterhalb) der gesamten Schleife ($\hat{=}$ `while`-Teil) stehen? Falls nicht: Wo liegt der Unterschied und wann sind die beiden „Versionen“ identisch bzw. verschieden?

while-Schleifen

while(-else)-Statement, Zählervariablen, break und continue

(4) [Zusatzaufgabe] Potenz a^b

Implementieren Sie die Funktionalität des Potenzoperators `**` (ohne ihn zu verwenden...) mithilfe einer `while`-Schleife. Die Variable `a` soll die Basis, also eine beliebige Zahl (*int/float*) besitzen; `b` sei der ganzzahlige (*int*) Exponent, wobei gilt: $b \geq 0$ (\rightarrow Exponent nicht negativ). Das Ergebnis (a^b) soll anschließend in `c` gespeichert sein.

Versuchen Sie nach Möglichkeit keine zusätzlichen Variablen zu verwenden (nur `a`, `b` und `c`).