



Übungszettel 2a - Python

Aufgabe 1: Python starten

Teilaufgaben (a)-(c) beziehen sich auf das Arbeiten unter den Systemen, die auf den RBI-Rechnern installiert sind. Falls du **IDLE** auf deinem eigenen Laptop installiert hast, öffne das Programm und beginne bei Teilaufgabe (d)

- (a) Öffne ein Terminal (konsole, xterm, ...), gib `python` ein und drücke die Enter-Taste. Lies ab, welche python-Version vorliegt.
- (b) Gib danach `exit()` ein und drücke die Enter-Taste. Was passiert?
- (c) Wir versuchen nun, die Entwicklungsumgebung *idle* zu starten.
 1. Gib dazu in einem Terminal `idle` ein (drücke noch nicht die Enter-Taste).
 2. Drücke danach zweimal auf die *TAB*-Taste. Mit der *TAB*-Taste kannst du anzeigen lassen, ob es mehrere *idle*-Versionen gibt. Falls mehrere *idle*-Versionen erscheinen, tippe die höchste Version ein, gefolgt von einem `&` und drücke auf Enter. Beispiele:
Falls nur *idle* erscheint, gib ein: `idle &`
Falls *idle3* erscheint, gib ein: `idle3 &`
 3. Ein Fenster mit der Überschrift *Python Shell* öffnet sich. *python* befindet sich jetzt im *interaktiven Modus*, welchen du schnell durch die drei Zeichen `>>>` erkennst.
- (d) Gib nun in der python-Shell `print('Hello World!')` ein.
- (e) Gib in der python-Shell (d. h. im interaktiven Modus) ein:

```
>>> a = 123
>>> b = 456
>>> a*b
```
- (f) Gib in der python-Shell (d. h. im interaktiven Modus) ein:

```
>>> c = 2
>>> d = 32
>>> c**d
```
- (g) Am Anfang hast du *Hello World* ausgegeben. Danach hast du $a \cdot b$ und c^d berechnet. Diese Anweisungen lassen sich in einem Programm speichern, indem du eine Datei wie folgt anlegst:
 1. In dem Fenster *Python-Shell*, klicke links oben auf *File* → *New File*. Ein Fenster mit dem Titel **Untitled** öffnen sich.
 2. Gib in dem Fenster folgendes ein.

```
print('Hello World!')

a = 123
b = 456
a*b

c = 2
d = 32
c**d
```

3. Im Fenster **Untitled**, klicke links oben auf *File* → *SaveAs*. Es öffnet sich ein Fenster, in welchem du den Dateinamen und den Speicherort auswählen kannst. Speichere die Datei als `hello.py` in einem Ordner deiner Wahl ab.
4. Das Fenster **Untitled** hat nun die Überschrift *hello.py*. Klicke in diesem Fenster oben auf *Run* → *Run Module* (Alternativ kannst du auch die Taste *F5* drücken). Nun wird das Programm in einer interaktiven python-Shell ausgeführt. Hier kannst du jetzt die eben deklarierten Variablen verwenden. Gib hierzu zum Beispiel `print(a)` oder nur `a` ein und bestätige mit der Enter-Taste.
5. Falls du später die Datei `hello.py` oder eine andere python-Datei öffnen willst, starte IDLE mit dem Befehl `idle hello.py`. Falls IDLE schon gestartet wurde, führt folgende Prozedur zum Ziel: Im Fenster *Python Shell*, klicke oben auf *File* → *Open* und wähle durch Klicken deine Datei aus. Klicke dann auf *Open*.
6. Möchtest du den Inhalt dieser Datei auch in einem anderen Programm verwenden, so kannst du

```
from hello import *
print(a+b)
```

in eine weitere Datei schreiben und diese Ausführen.

Solution: Die Studierenden sollen einfach die Anweisungen ausführen, um sich mit idle vertraut zu machen.

Der Codevorschlag mit `from hello import *` ist nicht so richtig guter Stiel, wird auch im Skript von abgeraten. Allerdings ist das Beispiel hier so klein, dass ein versehentliches Überschreiben von Variablen unwahrscheinlich ist.

Mit `exit()` wird der interaktive Modus von Python verlassen.

Aufgabe 2: Primitive Datentypen

(a) Im interaktiven Modus von Python:

1. Erstelle zwei logische Variablen `a` und `b` mit unterschiedlichen Belegungen
2. Gib das Ergebnis der Konjunktion dieser Variablen aus
3. Gib das Ergebnis der Disjunktion dieser Variablen aus
4. Belege `a` mit ihrem negierten Wert
5. Gib das Ergebnis der Konjunktion der beiden Variablen aus
6. Gib das Ergebnis der Disjunktion der beiden Variablen aus

Solution: Die Definition von Konjunktion und Disjunktion steht im Skript S. 81.

```
1 >>> a = True
2 >>> b = False
3 >>> a and b
4 False
5 >>> a or b
6 True
7 >>> a = not a
8 >>> a and b
9 False
10 >>> a or b
11 False
```

(b) Im interaktiven Modus von Python berechne: $(2 + 3 \cdot 4) \cdot 365$, wobei:

- nur Integer-Werte benutzt werden dürfen
- arithmetische Operationen nur für Variablen benutzt werden dürfen
- in jeder Zeile auf einmal nur eine arithmetische Operation berechnet werden darf

Solution:

```
1 >>> a = 2
2 >>> b = 3
3 >>> c = 4
4 >>> d = 365
5 >>> e = b * c
6 >>> f = a + e
7 >>> g = f * d
8 >>> g
9 5110
```

(c) Im interaktiven Modus von Python berechne: $\frac{250 \cdot 48 - 6}{35 \cdot 2^3}$, wobei:

- arithmetische Operationen nur für Variablen benutzt werden dürfen
- in jeder Zeile auf einmal nur eine arithmetische Operation berechnet werden darf
- 1. nur Float-Werte benutzt werden dürfen
- 2. nur Float-Werte benutzt werden dürfen aber ein gerundetes Ergebnis erwartet wird
- 3. nur Integer-Werte benutzt werden dürfen

Solution:

```
1 # 1. nur Floats:
2 >>> a = 250.0
3 >>> b = 48.0
4 >>> c = 6.0
5 >>> d = 35.0
6 >>> e = 2.0
7 >>> f = 3.0
8 >>> h = a * b
9 >>> i = h - c
10 >>> j = e ** f
11 >>> k = d * j
12 >>> l = i / k
13 >>> l
14 42.83571428571429
15
16 # 2. zum Runden:
17 >>> l = round(l)
18 >>> l
19 43.0
20
21 # 3. nur Integers:
22 >>> a = 250
23 >>> b = 48
24 >>> c = 6
25 >>> d = 35
26 >>> e = 2
27 >>> f = 3
28 >>> h = a * b
29 >>> i = h - c
30 >>> j = e ** f
31 >>> k = d * j
32 >>> l = int(i / k) # muss in Integer gecastet werden, da Rückgabe der
    Division immer float
33 >>> l
34 42
```

Aufgabe 3: Boolesche Ausdrücke

Schreibe das Programm unten in eine Datei, in der du die Wahrheitswerte der Variablen ändern kannst. Dann kannst du das Programm mit unterschiedlichen Wahrheitswerten ausführen. Informationen in Aufgabe 1(g) könnten hilfreich sein.

Weise den Variablen im folgenden Python-Programm Werte zu, sodass folgende Sätze auf der Konsole ausgegeben werden:

- (a) Seid ihr der König? Ich hab' euch nicht gewählt!
- (b) Seid ihr der König? Wir haben eine Hexe gefunden. Dürfen wir sie verbrennen?

```
1 bauer=True
2 koenig=False
3 gewaehlt=False
4 streichholz=False
5
6 if bauer and koenig:
7     print ("Seid ihr der König?")
8 if bauer and koenig and not gewaehlt:
9     print("Ich hab' euch nicht gewählt!")
10 else:
11     if bauer and streichholz:
12         print("Wir haben eine Hexe gefunden.")
13         print("Dürfen wir sie verbrennen?")
14     else:
15         if bauer:
16             print("Hilfe hilf ich werde unterdrückt!")
```

Solution:

- (a) bauer = True, koenig = True, gewaehlt = False, streichholz = True oder False
- (b) bauer = True, koenig = True, gewaehlt = True, streichholz = True

Aufgabe 4: Zahlenraten

Schreibe ein Programm das die Nutzer*innen eine Zahl zwischen 0 und 100 raten läßt.

- (a) Nutzer*innen haben einen Versuch. Folgende Codeschnipsel können dir dabei helfen.

- else:
- secretNumber = 87
- print('Herzlichen Glückwunsch! Sie haben die Zahl erraten.')
- print('Schade! Das war nicht die gesuchte Zahl.')
- if(int(a)==secretNumber):
- a = input('Geben Sie eine Zahl ein: ')

- (b) Die Nutzer*innen haben so viele Versuche, bis die Zahl erraten ist. Die Anzahl der benötigten Versuche wird am Ende ausgegeben. Das Programm soll den Nutzer*innen Hinweise geben, ob die gesuchte Zahl größer oder kleiner als die eingegebene Zahl ist. Überlege was überprüft werden muss, und was in welchem Fall geschehen soll. Falls nötig, skizziere ein Kontrollflussdiagramm (Kapitel 2.5.1).

Folgende Codeschnipsel können dir weiterhelfen.

- secretNumber = 87
- print('Herzlichen Glückwunsch! Sie haben die Zahl erraten! Sie haben dafür',count,'Versuche gebraucht.')
- else:
- if(int(a)==secretNumber):
- print('Die Zahl ist zu groß. Raten Sie nochmal.')
- break

- a = input('Geben Sei eine Zahl ein: ')
- elif(int(a) < secretNumber):
- count += 1
- count = 1
- print('Die Zahl ist zu klein. Raten Sie nochmal.')
- while True:

Solution:

```

(a) # Das Programm lässt die Nutzer*innen eine Zahl raten
2 # Nutzer*innen haben einen Versuch
3 #
4
5 secretNumber = 87
6 a = input("Geben Sie eine Zahl ein: ")
7 if(int(a)==secretNumber): # Zahl wurde erraten
8     print("Herzlichen Glückwunsch! Sie haben die Zahl erraten!")
9 else: # Zahl wurde nicht erraten
10    print("Schade! Das war nicht die gesuchte Zahl.")

(b) # Das Programm lässt die Nutzer*innen eine Zahl raten
2 # Nutzer*innen haben beliebig viele Versuche und erhalten
3 # Hinweise, ob die geratene Zahl zu gröss oder zu klein war
4
5
6 secretNumber = 87 # zu erratende Zahl
7 count = 1 # Zähler, um Versuche zu zählen
8 while True:
9     a = input('Geben Sie eine Zahl ein: ')
10    if(int(a)==secretNumber): # Zahl wurde erraten
11        print('Herzlichen Glückwunsch! Sie haben die Zahl erraten!
12 Sie haben dafür', count, 'Versuche gebraucht.')
13        break # Programm abbrechen
14    elif(int(a) < secretNumber): # eingegebene Zahl ist zu klein
15        print('Die Zahl ist zu klein. Raten Sie nochmal.')
16    else:
17        print('Die Zahl ist zu gröss. Raten Sie nochmal.')
18    count +=1 # Zähler erhöhen

```

Aufgabe 5: Verzweigungen

Schreibe ein Programm, das sich folgendermaßen verhält:

1. Nutzer*innen müssen eine natürliche Zahl eingeben
2. Ist die Eingabe ungültig, muss sie wiederholt werden
3. Bei den Eingaben 1, 2, 3 oder 5 wird die Zahl verdoppelt und anschließend ausgegeben
4. Bei den Eingaben 4, 6 oder 7 wird die Zahl vervierfacht und anschließend ausgegeben
5. Andere Zahlen werden unverändert ausgegeben

Hinweis: Beachte die Datentypen beim Vergleich!

Solution:

```

1 #def conditional_if():
2 while(True):
3     number = input("Enter a number: ")

```

```
4     if number.isdigit():
5         break
6 number = int(number) # for future comparisons
7 if number in (1, 2, 3, 5):
8     number *= 2
9 elif number in (4, 6, 7):
10    number *= 4
11 print(number)
```

Viel Erfolg!