



## Tag 2b - Python

### Aufgabe 1: Integer vs. Float

- (a) Beantworte die folgenden Fragen, ohne den Python-Interpreter zu nutzen. Überprüfe dein Ergebnis anschließend, in dem du die Terme vom Interpreter berechnen lässt.
1. Ist  $(3 * 12) + 15$  in Python dasselbe wie  $3 * 12 + 15$ ?
  2. Ist  $21/7$  in Python dasselbe wie  $21.0/7.0$ ?
  3. Ist  $(20/3) * 3 + 1 == 21$ ?
  4. Wie ist es mit  $(20//3) * 3.0 + 1.0 == 21$ ? Warum?
- (b) Gegeben seien zwei Integer-Variablen  $x$  und  $y$ . Überlege dir, wie man mit den sechs Grundrechenarten herausfinden kann, ob sich  $x/y$  von bei der Berechnung von  $x/y$  unterscheidet.
- (c)
1. Berechne den Wahrheitswert von `(True or False) == not (not True and not False)`.
  2. Ersetze auf beiden Seiten des Vergleichs `True` durch `False`.
  3. Ersetze auf beiden Seiten des Vergleichs `False` durch `True`.
  4. Überlege, wieso das Ergebnis so sein könnte, wie es ist. Die Vorlesung von gestern könnte dabei hilfreich sein.
- (d)
1. Lege eine Variable `test` an, und weise ihr den Wert 11 zu.
  2. Teile `test` durch 2, und weise das Ergebnis wiederum `test` zu.
  3. Führe diesen Schritt einige (6-7) Male durch und überprüfe den Wert von `test`. Macht das Ergebnis Sinn?
  4. Wiederhole die bisherigen Schritte, nachdem du `test` den Wert 11.0 zugewiesen hast.

### Aufgabe 2: Zahlentrick

Es gibt einen Zahlentrick, der folgendermaßen funktioniert:

Man bittet jemanden, sich eine einstellige Zahl auszudenken. Dann soll er zu dieser Zahl noch einmal dieselbe Zahl addieren. Danach soll die 10 hinzuaddiert werden und durch 2 geteilt werden. Schließlich wird noch die gedachte Zahl abgezogen.

Egal, welche Zahl ausgedacht wurde - das Ergebnis ist immer die 5.

Um dies zu überprüfen, schreibe ein Programm, das eine einstellige Zahl als Eingabe bekommt und führe genau diese Rechenschritte mit dieser Zahl aus. Gib das Ergebnis aus.

### Aufgabe 3: Chiffrierung

Zeichen (Buchstaben, Ziffern, Interpunktionszeichen, etc) werden rechnerintern mithilfe einer Tabelle verwaltet, die jedem Zeichen eine Zahl zuordnet. Eine Möglichkeit der Textverschlüsselung ist, für jedes Zeichen des Textes das um  $n$  Tabelleneinträge verschobene Zeichen zu schreiben. Falls beim Verschieben das Ende der Tabelle erreicht wird, so wird beim Anfang weitergemacht. (Der %-Operator ist hier hilfreich).

#### Beispiel

Bei einem Verschiebungsoffset von  $n = 1$  wird z. B. aus 'HAL' der String 'IBM'.

- (a) Schreibe ein Programm, welches den Benutzer zunächst auffordert einen Wert einzugeben, der den Verschiebungsoffset  $n$  festlegt. Dann soll der Benutzer aufgefordert werden den zu verschlüsselnden Text einzugeben. Nach Eingabe des Textes soll der chiffrierte Text auf dem Bildschirm ausgegeben werden.

- (b) Schreibe ein zweites Programm, das einen chiffrierten Text als Eingabe erwartet und den de-chiffrierten Text auf dem Monitor ausgibt.

#### Aufgabe 4: Schleifen

- (a) Schreibe ein Programm, das alle ungeraden Zahlen von 1 bis 100 ausgibt.
1. Mithilfe einer for-Schleife
  2. Mithilfe einer while-Schleife
- (b) Schreibe ein Programm, das den Benutzer auffordert 5 Zahlen einzugeben, und dann den Mittelwert der Zahlen berechnet und ausgibt.  
Verwende dabei eine Schleife.
- (c) **schwierig** Schreibe ein Programm, das den Mittelwert beliebig vieler Zahlen berechnet und ausgibt.  
Tipp: Überlege dir, woran das Programm erkennen soll, dass alle Zahlen eingegeben sind und der Mittelwert nun berechnet werden soll.
- (d) Schreibe ein Programm, das die Zahlen von 1 bis  $n = 4$  paarweise miteinander multipliziert und das Ergebnis ausgibt.  
Die Ausgabe könnte z. B. so aussehen:
- ```
1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
...
4 x 4 = 16
```
- (e) Erweitere Dein Programm um eine Abfrage nach der Zahl  $n$ , bis zu der die Produkte berechnet werden sollen (d.h.  $n$  soll nun nicht mehr auf 4 festgelegt sein).
- (f) Schreibe ein Programm, das eine Multiplikationstabelle erzeugt und ausgibt.  
Hinweis: Eine *Multiplikationstabelle* ist eine Tabelle mit natürlichen Zeilen- und Spaltenindizes, die in Feld  $i, j$  die Zahl  $i \cdot j$  enthält.

#### Aufgabe 5: Programmieraufgaben

- (a) Schreibe ein Programm, welches ein Passwort verlangt. Gibt der Benutzer das richtige Passwort ein, soll das Programm „Authentifizierung erfolgreich“ ausgeben. Wird das Passwort jedoch dreimal hintereinander falsch eingegeben, soll die Ausgabe „Zugriff verweigert“ lauten.
- (b) Schreibe ein Programm, welches eine Zahl als Eingabe anfordert und alle Primzahlen bis zu dieser Zahl ausgibt. Eventuell ist es hilfreich, sich über das Thema „Primfaktorzerlegung“ und den Modulo- (%) Operator zu informieren.
- (c) Schreibe ein Programm, welches den Benutzer auffordert, sich eine Zahl zwischen 1 und 100 ausdenken. Das Programm soll dann sukzessive Fragen der Form „Ist die Zahl größer/kleiner als...“ stellen und so die ausgedachte Zahl mit möglichst wenigen Fragen erraten. Eventuell ist es hilfreich, das Thema „Binärsuche“ zu recherchieren.

Viel Erfolg!