

Programmieren I, WS 2012/13

Tutorium Übungsblatt

Die Aufgaben sollen auf Papier gelöst werden.

Achtung: Dies ist keine offizielle Probeklausur!

Aufgabe 1

Welche Werte und Typen geben die folgenden Anweisungen zurück, wenn man sie in den Interpreter eingibt? Wird jedes Mal etwas zurückgegeben?

Achtung: Mindestens ein Ausdruck führt zu einem Fehler. Welche(r)?

1. `a = 14/3; a*4`
3. `4*"tz"`
4. `smartphone = 199,95`
5. `einkaufsliste = ["Gurken", "Tomaten", "Rotwein"]`
6. `'40' = antwort; print antwort + 2`
7. `"dum"+"di"*2 == "dumdidumdi"`

Aufgabe 2

Gegeben sei die folgende Funktion:

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

def distance(wordA, wordB):

    listA = list(wordA)
    listB = list(wordB)
    count = ''

    for char in range(len(listA)):
        if listA[char] != listB[char]:
            count += 1

    print('Die Differenz betraegt ' + count + '!')
    return count
```

- a) Wofür ist die Funktion gut, und wie geht sie vor? Beschreibe in deinen eigenen Worten!
- b) Drei Fehler haben sich eingeschlichen. Welche sind das? Wie kannst du den zweiten Fehler mit einem try/except-Befehl abfangen? Trage ihn oben in die Funktion ein!

Aufgabe 3

Wenn wir in der Computerlinguistik mit Text arbeiten, dann wollen wir die Wörter für gewöhnlich in ihrer normalisierten Form, das heißt ohne anhängende Satzzeichen oder Unterschiede in der Groß- und Kleinschreibung, haben. Wir müssen also alle möglichen Eingabearten auf eine einheitliche Schreibweise reduzieren (das Wort „ist“ könnte zum Beispiel als „Ist“, „Ist“, „ist“, „ist“, oder „ist!“ im Text vorkommen). Ein einfaches „split(" ")“ reicht da meist nicht aus.

Wir wollen eine Funktion schreiben, die als ersten Eingabeparameter einen Satz bzw. Text nimmt, und diesen normalisiert. Dazu soll der Satz zeichenweise mit einer Menge erlaubter Zeichen (welche wir als zweiten Eingabeparameter mitgeben) abgeglichen werden. Wenn das untersuchte Zeichen unter den erlaubten Zeichen ist, soll es zu einem neuen String `cleanedText` zugefügt werden. Zurückgegeben wird dann der normalisierte Satz bzw. Text.

Die Signatur der Funktion ist die folgende:

```
normalizeWords(word, allowedChars) -> str
```

- a) Wie gehst du vor? Zeichne ein Flussdiagramm!
- b) Jetzt schreibe die Funktion und versieh sie mit den Kommentaren aus deinem Diagramm!
- c) Ändere nun das Ausgabeformat: Was musst du tun, um die normalisierten Wörter direkt in einer Liste zurück zu geben? Wie kannst du das in einer Zeile erreichen? Gib die neue Signatur an!
- d) Wenn du die Funktion mit einer for-Schleife implementiert hast: Ändere sie in eine while-Schleife! Hast du eine while-Schleife verwendet, dann ändere sie in eine for-Schleife!