

Einführung in Python — Übung 1

Rebecca Breu, Bastian Tweddell

Oktober 2008

Login: XXXloginXXX
Passwort: XXXpasswordXXX
Bitte das Passwort ändern (`passwd`)!

Datentypen I

Aufgabe 1 (Erste Schritte, Zahlen) Starten Sie den Python-Interpreter.

- Benutzen Sie die Online-Hilfe: `help()`, `help(complex)`, `dir(complex)`
- Machen Sie sich mit den Datentypen `int`, `float`, `complex` und den grundlegenden Rechenoperationen vertraut.
- Was passiert, wenn Sie in einer Rechnung verschiedene Zahlen-Typen kombinieren?
- Geben Sie $1/3$ als Dezimalzahl `0.333333...` aus.
- Was ergibt `1/0`?
- Arbeiten Sie mit Variablenzuweisungen. Was passiert, wenn Sie einer Variable nacheinander Objekte unterschiedlicher Datentypen zuweisen?

Aufgabe 2 (Strings)

2.1 Schreiben Sie ein Programm, welches nacheinander den Vornamen und den Nachnamen des Benutzers abfragt. Anschließend soll der vollständige Name zusammen mit einer Begrüßung in einer Zeile ausgegeben werden, z.B.:

```
Dein Vorname? Jim
Dein Nachname? Kirk
Langes Leben und Friede, Jim Kirk
```

2.2 Erweitern Sie das obige Programm, sodass es zusätzlich folgendes ausgibt:

- Die Buchstaben 2 bis 5 vom Vornamen und den vorletzen Buchstaben des Nachnamens
- Den vollständigen Namen in Großbuchstaben
- Ob der Nachname auf „mann“ endet
- ...? Probieren Sie noch weitere String-Methoden aus:
<http://docs.python.org/lib/string-methods.html>

Aufgabe 3 (Listen) Starten Sie den Python-Interpreter.

- Legen Sie eine Liste mit mehreren Elementen an und greifen Sie auf verschiedene Listenelemente zu: Das zweite, die vierten bis sechten, das letzte. Was passiert, wenn Sie einen ungültigen Listenindex angeben?
- Fügen Sie der Liste nachträglich Elemente hinzu, löschen Sie einzelne Elemente. Wie geht Python mit verschiedenen Datentypen in der Liste um?
- Fügen Sie zwei Listen zusammen
- Kann man eine Liste als Element in eine Liste einfügen?

Statements

Aufgabe 4 (Schleifen und Verzweigungen) Schreiben Sie ein Programm, welches zehn Zahlen von der Tastatur einliest und anschließend deren Summe ausgibt.

Hinweis: `int()` und `float()` können auch auf Strings angewendet werden.

4.1 Schreiben Sie das Programm so um, dass es abbricht, falls die Zwischensumme größer als 42 ist.

4.2 Schreiben Sie das obige Programm so um, dass es negative Zahleneingaben ignoriert.

4.3 Schreiben Sie das obige Programm so um, dass es solange Zahlen einliest, bis der Benutzer `ende` eintippt.

Aufgabe 5 Schreiben Sie ein Programm, welches zehn Strings von der Tastatur einliest und in einer Liste speichert. Anschließend sollen in allen Strings der Buchstabe `a` durch ein `e` ersetzt werden und das Ergebnis ausgegeben werden.

Hinweis: Benutzen Sie Listen, aber greifen Sie nicht mit Indices `liste[i]` auf die einzelnen Elemente zu!

Aufgabe 6 Die `ord`-Funktion liefert die ASCII-Darstellung eines Zeichens zurück. Schreiben Sie ein Programm, welches einen String in ASCII-Darstellung ausgibt.

Hinweis: Auch hier werden keine Indices benötigt!

Funktionen

Aufgabe 7 (Fakultät) Schreiben Sie eine Funktion, welche die Fakultät $n!$ einer Zahl n zurückgibt. ($n! = 1 \cdot 2 \cdot \dots \cdot n$)

Aufgabe 8 (Ulam-Folge) Schreiben Sie eine Funktion, welche die Ulam-Folge zu einer Zahl a_0 als Liste zurückgibt. Es gilt:

$$a_{n+1} = \begin{cases} a_n/2, & \text{falls } a_n \text{ gerade,} \\ 3a_n + 1, & \text{falls } a_n \text{ ungerade,} \end{cases}$$

bei $a_n = 1$ breche die Folge ab. Beispiel:

```
>>> print ulam_folge(6)
[6, 3, 10, 5, 16, 8, 4, 2, 1]
```

Aufgabe 9 (Palindrom) Schreiben Sie eine Funktion, die testet, ob ein String ein Palindrom ist, also ein Wort, das vorwärts und rückwärts gelesen gleich ist (gehen Sie davon aus, dass der String keine gemischte Groß-/Kleinschreibung enthält). Was passiert, wenn der Funktion statt eines Strings eine Liste übergeben wird?

Aufgabe 10 (Freier Fall) Die Formel für den freien Fall lautet:

$$h(t) = h_0 - \frac{1}{2}gt^2,$$

wobei $h(t)$ die Fallhöhe zur Zeit t ist, h_0 die Ausgangshöhe zur Zeit 0 und g die Erdbeschleunigung. Implementieren eine Funktion `fallhoehe(h_0, t, g)`!

10.1 Rufen Sie die obige Funktion mit Keyword-Parametern auf. Was passiert, wenn Sie die Reihenfolge der Keyword-Parameter ändern?

10.2 Ändern Sie das Programm, sodass die Erdbeschleunigung per default 9.81 beträgt.

Aufgabe 11 Was macht folgender Code?

```
def nichts():
    print "Ich gebe nichts zurueck."

a = nichts()
print a
```

Input/Output

Aufgabe 12 (String-Formatierung) Starten Sie den Python-Interpreter.

- Geben Sie `float`-Zahlen mit festgelegter Anzahl an Vor- und Nachkommastellen aus und in Exponentialdarstellung.
- Geben Sie `int`-Zahlen in Hexadezimal- und Oktaldarstellung aus.
- Sie wissen über eine Person die Daten `vorname`, `nachname`, `wohntort`, `alter`. Geben Sie die Daten so aus, dass Sie z.B. folgende Zeile erhalten:

```
"James Kirk ist 35 Jahre alt und wohnt in Iowa."
```

- Probieren Sie weitere Formatierungen aus:
<http://docs.python.org/lib/typesseq-strings.html>

Aufgabe 13 (Kommandozeilenparameter) Schreiben Sie ein Programm, welches seinen Namen und alle ihm übergebenen Parameter ausgibt.

Aufgabe 14 (Dateien lesen) Schreiben Sie ein Programm, welches die Häufigkeit des Wortes „Spam“ in einer Datei zählt.

Aufgabe 15 (Dateien lesen und schreiben) Gegeben sei eine Datei, in der in jeder Zeile beliebig viele, durch Doppelpunkt getrennte Zahlen stehen:

```
13.0:14.5:17:0.03:-2
89:-0.2354:6666
```

Lesen Sie die Zahlen ein, addieren Sie alle Zahlen einer Reihe und geben Sie die Zahlen in einer neuen Datei aus, im selben Format wie oben, nur dass als letztes zusätzlich die Summe steht.

Hinweis: Schauen Sie sich die String-Methoden `strip`, `split` und `join` an.

Ausnahmen

Aufgabe 16 Lesen Sie vom Benutzer eine Zahl ein und geben Sie ihr Quadrat aus. Was passiert, wenn der Benutzer etwas anderes als eine Zahl eingibt? Geben Sie eine benutzerfreundliche Fehlermeldung aus!

Aufgabe 17 (Fakultät erweitert) Die Fakultät ist nur für natürliche Zahlen definiert. Wie reagiert die Funktion aus Aufgabe 7 auf negative Parameter? Lassen Sie die Funktion in diesem Fall einen `ValueError` auslösen.

Aufgabe 18 Schreiben Sie ein Programm, welches in einer Endlosschleife Eingaben vom Benutzer einliest und jeweils prüft, ob die Eingabe ein Palindrom ist (verwenden Sie die Funktion aus Aufgabe 9).

Was passiert, wenn der Benutzer `Ctrl-C` oder `Ctrl-D` drückt? Ändern Sie das Programm, sodass der Benutzer beim Drücken von `Ctrl-C` oder `Ctrl-D` gefragt wird, ob er das Programm beenden möchte.

Aufgabe 19 Mit dem Modul `readline` können für die Benutzereingaben mittels `raw_input` Editiermöglichkeiten wie in der Shell aktiviert werden. Dieses Modul ist standardmäßig nur unter *nix-Systemen verfügbar. Folgender Code importiert das Modul nicht auf Windows-Rechnern:

```
if not sys.platform.startswith("win"):
    import readline
    import rlcompleter
    readline.parse_and_bind("tab: complete")
```

Warum ist ein Ansatz mit Ausnahmen besser? Schreiben Sie den Code so um, dass er mit Ausnahmen arbeitet.

Zusätzliche Aufgaben

Aufgabe 20 (Zahlenraten) Programmieren Sie das Spiel Zahlenraten: Ein Spieler (hier der Computer) denkt sich eine Zahl zwischen 1 und 100 aus, der andere Spieler (hier der Benutzer) muss die Zahl raten. Nach jedem Versuch erfährt der Benutzer, ob er zu hoch, zu niedrig oder genau richtig lag. Eine Runde ist zuende, wenn die Zahl richtig geraten wurde.

Realisieren Sie das Programm in folgenden Schritten:

20.1 Das Programm lässt den Benutzer eine Runde spielen und gibt am Ende die Anzahl der benötigten Versuche aus.

Hinweis: Eine Zufallszahl zwischen 1 und 100 wird mit dem Befehl `random.randint(1, 100)` erzeugt.

20.2 Das Programm fragt nach jeder Runde, ob der Benutzer noch einmal spielen möchte.

20.3 Das Programm speichert eine Highscore-Liste in einer Datei.

War das nicht genug? Weitere Aufgaben:

<http://www.pythonchallenge.com>