

## Computergestützte Mathematik zur linearen Algebra – 2. Übungsblatt

### Aufgabe 5: (Listenmanipulation)

*Befehle:* `print`, `len`

Wenn `variable` ein Objekt ist, so kann man im Editor von SPYDER durch tippen von `variable`. (ggf. <Tab> drücken) alle Funktionen sehen die das Objekt hat. Benutzen Sie diese Eigenschaft um die folgende Aufgabe zu lösen.

Erstellen Sie ein PYTHON-Skript *listenspielereien.py* mit folgenden Inhalten:

- Erstellen Sie das *Tupel* `tupe1` mit den Einträgen 1, 7, 'Gustav Gans' und 7.
- Geben Sie mit Hilfe des Befehls `len` die Anzahl der Elemente in `tupe1` aus.
- Geben Sie aus wie oft die Zahl 7 in `tupe1` vorkommt.
- Geben Sie das dritte Element von `tupe1` aus.
- Erstellen Sie die *Liste* `liste1` mit den Einträgen 'Tick' und 'Track'.
- Fügen Sie in `liste1` zwischen 'Tick' und 'Track' den Eintrag 'Trick' hinzu und geben Sie `liste1` aus.
- Wieso können Sie in `tupe1` keine neuen Einträge hinzufügen?
- Erstellen Sie die *Liste* `liste2` mit den Einträgen 1, 'Dagobert', 2, 3,  $-4$ , `liste1`, 3, 14 und  $5 + 9i$ . Geben Sie `liste2` aus.
- Löschen Sie in `liste1` den Eintrag 'Track'.
- Geben Sie `liste2` noch einmal aus. Was fällt Ihnen auf?
- Was hätten Sie in h) anders machen müssen, damit sich `liste2` durch Teil i) nicht ändert?

### Aufgabe 6: (Funktion definieren)

*Befehle:* `def`, `return`, `print`

Erstellen Sie ein PYTHON-Skript *meine\_erste\_funktion.py* mit folgendem Inhalt:

- Schreiben Sie eine Funktion  $(v, o) = \text{zylinder}(r, h)$ , die als Eingabeparameter den Radius  $r$  und die Höhe  $h$  eines Zylinders übergeben bekommt, das Volumen und die Oberfläche des Zylinders berechnet und diese Werte zurückgibt.
- Das Skript soll `zylinder(r, h)` mit  $r = 12,3$  und  $h = 4,56$  aufrufen und das Ergebnis wie folgt ausgeben (ohne Zeilenumbruch und mit den richtigen Werten):  
Ein Zylinder mit dem Radius  $r$  und der Höhe  $h$  hat ein Volumen von  $v$  und eine Oberfläche von  $o$ .
- Kommentieren Sie Ihren Quelltext sinnvoll.

*Hinweis:* Achten Sie auf die korrekte Einrückung der Funktion. Beachten Sie auch, dass bei der Ausgabe in b) vor dem '.' kein Leerzeichen ist. Schauen Sie sich daher ggf. die Hilfe zu `print` an.

**Aufgabe 7:** (Erster Plot)

*Befehle:* `lambda`, `*`, `*`, `plot`, `linspace`, `matplotlib.pyplot`

Schreiben Sie das PYTHON-Skript `mein_erster_plot.py` mit folgendem Inhalt:

- (a) Definieren Sie die *lambda-Funktion*  $f(x) = 0,5 + 5 \cdot 10^{-2}x(x + 2,5)(x - 3)$
- (b) Plotten Sie  $f(x)$  auf dem Intervall  $[-3, 3, 5]$  mit 1000 äquidistanten Punkten in einer roten Linie (in der Hilfe zu `plot` gibt es Hinweise zur Farbe).

*Hinweis:* Sie müssen vorher geeignete Module importieren.

**Aufgabe 8:** (while Schleife)

*Befehle:* `while`, `np.random.rand`

Erstellen Sie das PYTHON-Skript `meine_erste_schleife.py`. Weisen Sie der Variable `zahl` den Wert 0 zu. Addieren Sie nun solange eine zufällige *float-Zahl* aus dem Intervall  $[-1, 2)$ , bis `zahl` einen Wert größer als 5 erreicht. Geben Sie nach jeder Addition folgende Zeile aus:

```
*zahl* hat jetzt den Wert 'Wert von Zahl'
```

Ihre Ausgabe könnte also so aussehen:

```
*zahl* hat jetzt den Wert 0.17839840258771433
*zahl* hat jetzt den Wert -0.370048343385214
*zahl* hat jetzt den Wert 1.4336435557605018
*zahl* hat jetzt den Wert 2.116586247392288
*zahl* hat jetzt den Wert 1.875671647436247
*zahl* hat jetzt den Wert 3.848956487506835
*zahl* hat jetzt den Wert 3.8039196454976496
*zahl* hat jetzt den Wert 4.846861447239473
*zahl* hat jetzt den Wert 5.371616625870708
```

*Hinweis:* Achten Sie auf die korrekte Einrückung der Schleife. Überlegen Sie sich ggf. wie Sie an Zufallszahlen in dem gewünschten Intervall kommen.