

Unit 12

Dr. Günter Kolousek

21. Juli 2015

Lege wiederum ein Verzeichnis an. Nennes es `18_unit12`! In diesem Verzeichnis sollen alle Dateien der jeweiligen Einheit abgelegt werden.

1 Schulübungen

In diesen Beispielen werden verschiedene Aspekte geübt, wiederholt und vertieft.

1. Schreibe eine Funktion `union(a, b)` in einer Datei `sets.py`, die 2 Mengen als Parameter bekommt und die Vereinigungsmenge als **neue** Menge zurückliefert. Verwende die Methode `add`.
2. Schreibe eine Funktion `minus(a, b)` (wieder im Modul `sets`), die 2 unterschiedliche (aber u.U. durchaus gleiche) Mengen als Parameter bekommt und alle Elemente von `b` aus der Menge `a` entfernt und die Ergebnismenge zurückliefert. Verwende die Methode `discard`.

Beachte, dass sich damit die Menge `a` verändert hat. Das nennt man einen Nebeneffekt!
3. Schreibe eine Funktion `concat(a, b)` (von engl. concatenate) in einem Modul `lists`, die alle Elemente von `b` an die Elemente von `a` anhängt (in der selben Reihenfolge) und die Liste `a` zurückliefert. Verwende die Methode `append`.
4. Schreibe eine Funktion `insert(a, b, i)` (wieder im Modul `lists`), die jedes Element der Liste `b` in die Liste `a` vor dem Index `i` einfügt und die Liste `a` zurückliefert. Verwende die Methode `insert`. Die Liste `b` soll **nicht** verändert werden!

Beispiel:

```
>>> insert([1,4,5], [2, 3], 1)
[1, 2, 3, 4, 5]
```

Tipp: Verwende die Funktion `reversed`. Hilfe?

- a) Hmm, in der eingebauten Hilfe nachsehen (`help(reversed)`) ist immer ein guter Anfang. Diese ist in diesem Fall noch nicht so richtig verständlich, wenn man erst ein halbes Programmierjahr am Buckel hat.
 - b) Zweiter Versuch, wäre die Python-HTML-Hilfe zu Rate zu ziehen. Zu finden ist diese unter <http://www.python.org/doc> oder auch auf der [www2](http://www2.python.org/doc)! Aber die ist auch für den Anfang nicht so richtig aussagekräftig.
 - c) Dritter Versuch: einfach im Python-Interpreter ausprobieren: `reversed([1,2,3])` liefert so etwas wie `<list_reverseiterator object at 0x89b442c>`. Na ja, so viel über "Iterator" und "reverse" ist auch in der Doku gestanden. Aber probiere einmal `list(reversed([1,2,3]))`!
5. Schreibe eine Funktion `sort(a)` (wieder im Modul `lists`), die eine Liste `a` erhält und eine sortierte Liste zurückliefert. Verwende die Methode `sort`! D.h. die Liste `a` wird verändert.
 6. Schreibe eine Funktion `sort_descending(a)` (wieder im Modul `lists`), die eine Liste `a` erhält und eine absteigend sortierte Liste zurückliefert. Die Liste `a` soll **nicht** verändert werden. Verwende sowohl die Funktion `sorted` als auch die Methode `reverse`.

Beachte: Funktion `reversed` vs. Methode `reverse` und Funktion `sorted` vs. Methode `sort`.

Beachte: Der Type `tuple` kennt in Python keine Methoden `sort` und `reverse`, aber es funktionieren die Funktionen `sorted` und `reversed` sehr wohl. Warum?

7. Schreibe eine Funktion `minimum(seq)` (wieder im Modul `lists`), die das Minimum einer Sequenz von Zahlen findet und dieses zurückliefert.

Bei einer leeren Liste soll `None` zurückgeliefert werden.

8. Schreibe eine Funktion `maximum(seq)` (wieder im Modul `lists`), die eine Sequenz von Punkten der x/y Ebene (Tupel mit je 2 Zahlen) erhält und dasjenige Tupel zurückliefert, das den größten y-Wert enthält.

Beispiel::

```
>>> maximum([(2, 3), (4, 1), (1, 5), (3, 2)])
(1, 5)
```

Bei einer leeren Liste soll wieder `None` zurückgeliefert werden.

9. Schreibe das Modul `lists` so um, dass es jetzt **auch** als Programm verwendet werden kann und in diesem Fall der Benutzer nach beliebigen Anzahl von Zahlen gefragt wird. Es dürfen nur Zahlen eingegeben werden dürfen. Die Eingabe wird mit `CTRL-D` abgebrochen. Diese Zahlen sollen in einer Liste abgespeichert werden.

Nach erfolgter Eingabe ist die Funktion `minimum` aufzurufen und das Ergebnis auszugeben. Wurden keine Zahlen eingegeben, dann ist die Meldung `Es wurden keine Zahlen eingegeben!` auszugeben.

10. Schreibe eine Funktion `find(seq, num)` (wieder im Modul `lists`), die eine Sequenz von Zahlen `seq` und eine Zahl `num` als Parameter bekommt.

Die Funktion soll den Index der ersten gefundenen Position zurückliefern, wenn die Zahl enthalten ist und anderenfalls `-1`.