

Aufgabenblatt zu Funktionen, Listen und Mengen

Erstellen Sie die Funktionen dieses Aufgabenblattes alle in einem neuen Modul `aufgaben_listen.py`.

- E: Einstieg, Kennenlernen des Stoffes. Muss gelöst werden (können)
- Ü: Zusätzliches Übungsmaterial. Dient rein zur Übung und Repetition.
- V: Vertiefung und weitergehende Inhalte. Falls es als Aufgabe aufgegeben wird, muss es ebenfalls gelöst werden.
- H: Herausfordernde und weiterführende Aufgaben. Diese Aufgaben können Pflicht sein, sind es aber in der Regel nicht.

Erster Teil (Funktionen)

Aufgabe 1 (E)

- a) Schreiben Sie eine Funktion `ist_gerade`, die einen Parameter `zahl` hat und die in der Shell ausgibt, ob die übergebene Zahl gerade ist oder nicht.
- b) Verändern Sie obige Funktion so, dass keine Ausgabe in der Shell erfolgt, sondern so dass sie stattdessen ein Boolean (True oder False) zurückgibt, je nachdem ob die Zahl gerade ist oder nicht.

Aufgabe 2 (E)

Importieren Sie aus dem Modul `random` die Funktion `randint` und vom Modul `time` die Funktion `sleep`. Informieren Sie sich mittels `help` über die Funktionsweise dieser beiden Funktionen. Erstellen Sie eine Funktion `warte_zufaellig`, die eine zufällige Anzahl Sekunden lang wartet und dann «jetzt bin ich fertig» in der Shell ausgibt.

Zweiter Teil

Aufgabe 3 (E)

Schreiben Sie eine Funktion mit Namen `schrumpf`, die eine Liste als Argument erwartet und Schritt für Schritt von hinten her ein Element herauslöscht und die danach verbleibende Liste jeweils ausgibt.

Aufgabe 4 (Ü)

Schreiben Sie eine Funktion `halbiere`, die eine Liste von Elementen als Argument erwartet und die hintere Hälfte der Einträge löscht. Bei einer ungeraden Anzahl Einträgen, soll die Anzahl der gelöschten Einträge kleiner sein als diejenige der nicht gelöschten Einträge.

Aufgabe 5 (E)

Schreiben Sie eine Funktion `contains_threes`, die eine Liste von Zahlen als Argument erwartet und ausgibt, welche Zahlen der 3er Reihe darin vorkommen, der Grösse nach geordnet.

Aufgabe 6 (V)

Schreiben Sie eine Funktion mit Namen `listen`, die einen Integerwert `n` als Argument erwartet und dann eine Liste aus Listen bildet, so dass das erste Element `[1]` ist, das zweite `[1, 2]`, das dritte `[1, 2, 3]` usw. bis zum letzten Element `[1, 2, 3, ..., n]`. Wie viele Elemente hat die Liste `listen(9)`?

Aufgabe 7 (Ü)

Schreiben Sie eine Funktion mit Namen `mitte`, die eine Liste als Argument erwartet und eine neue Liste zurückgibt, die alle Elemente bis auf das erste und das letzte enthält. `mitte ([1, 2, 3, 4])` sollte also den Rückgabewert `[2, 3]` liefern.

Dritter Teil

Aufgabe 8 (E)

Schreiben Sie eine Funktion `ist_sortiert`, die eine Liste als Parameter erwartet und `True` zurückliefert, wenn die Liste aufsteigend sortiert ist, und ansonsten den Wert `False` zurückgibt. Sie können davon ausgehen (als Vorbedingung), dass die Elemente der Liste mit den relationalen Operatoren `<`, `>`, usw. vergleichbar sind.

Beispielsweise sollte `ist_sortiert([1, 2, 2])` den Wert `True` und `ist_sortiert(['b', 'a'])` den Wert `False` zurückliefern.

Aufgabe 9 (E)

Schreiben Sie eine Funktion mit dem Namen `schnipp`, die eine Liste erwartet, die das erste und das letzte Element entfernt und den Rückgabewert `None` liefert.

Aufgabe 10 (E)

Zwei Wörter werden als Anagramm bezeichnet, wenn sie die Buchstaben des einen Wortes so umstellen können, dass Sie das andere Wort ergeben. Schreiben Sie eine Funktion `ist_anagramm`, die zwei Strings erwartet und `True` liefert, wenn es sich um Anagramme handelt.

Aufgabe 11 (E)

Schreiben Sie eine Funktion mit dem Namen `hat_duplikate`, die eine Liste erwartet und `True` zurückliefert, wenn eines der Elemente mehr als einmal darin enthalten ist. Die ursprüngliche Liste soll dabei nicht verändert werden.

Hinweis: Gehen Sie geschickt vor und nutzen Sie die vorgestellten Datentypen. Eine Zeile genügt!

Aufgabe 12 (E)

Schreiben Sie eine Funktion mit dem Namen `entferne_duplikate`, die eine Liste als Parameter erwartet und eine Liste zurückliefert, die jedes Element der Originalliste nur einmal enthält.

Tipp: Sie brauchen nicht in der gleichen Reihenfolge vorzukommen. Auch hier genügt eine Zeile!

Aufgabe 13* (H)

Schreiben Sie eine Funktion mit Namen `every_third`, die eine Liste von Zeichen (Buchstaben, Zahlen) entgegen nimmt und jedes dritte Element daraus löscht.

Tipp: Testen Sie ihre Funktion eingehend! (Die Aufgabe ist weniger einfach als sie erscheint)

Vierter Teil

Aufgabe 14 (E)

- a) Schreiben Sie eine Funktion `starts_with_d(text)`, die einen Text einliest und all jene Worte in einer alphabetisch sortierten Liste ausgibt, die mit einem 'd' oder einem 'D' anfangen.
- b) Schreiben Sie eine Funktion `ends_with_e(text)`, die einen Text einliest und all jene Worte in einer nach Wortlänge sortierten Liste ausgibt, die mit einem 'e' enden. Um eine Liste von Strings nicht alphabetisch sondern nach der Länge zu sortieren, verwenden Sie den Parameter `key` der Funktion `sort`.
`worte.sort(key=len)` sortiert Ihnen eine Liste `worte` nach Länge der Worte.

Aufgabe 15 (V)

Schreiben Sie eine Funktion `ohne_umlaute(text)`, die einen Text einliest, alle Worte entfernt, die einen Umlaut enthalten und die schliesslich den Satz ohne diese Worte wieder als String zurückgibt.

Aufgabe 16 (V)

Schreiben Sie eine Funktion `primes(n)`, die Ihnen alle Primzahlen bis zum Wert `n` in einer Liste ausgibt. Nutzen Sie dazu eine selbstgeschriebene Funktion `is_prime(x)`, die `True` zurückgibt, falls `x` eine Primzahl ist und `False` sonst. Mit dieser Hilfsfunktion können Sie in `primes(n)` eine Listen-Abstraktion nutzen, so dass eine Zeile genügt.

Aufgabe 17 (E)

Nutzen Sie Listen-Abstraktion um folgende Listen zu erzeugen:

- a) Eine Liste aller Zahlen in der Siebenerreihe bis 1000
- b) Eine Liste der ersten 10 Kubikzahlen beginnend bei der Kubikzahl von 1
- c) Eine Liste aller Primzahltriple der Zahlen bis 1000. Ein Primzahltriple ist ein Tripel (a, b, c) , so dass entweder $a+2=b$ und $b+4=c$ oder $a+4=b$ und $b+2=c$ gilt. Nutzen Sie ruhig bereits erstellte Funktionen.
- d) Eine Liste, die aus einer vorgegebenen Liste von Worten all jene Worte auswählt, in denen ein 'a' aber kein 'o' drin vorkommt.