

Praktische Beispiele

1 Ein einfaches Spiel

1. Programmieren Sie das folgende kleine Spiel: Der Computer „denkt“ sich eine geheime Zahl aus. Der Benutzer muss versuchen, diese Zahl in mehreren Versuchen zu erraten. Nach jedem Versuch gibt das Spiel aus, ob der Benutzer zu niedrig, zu hoch, oder richtig geraten hat. Die maximale Anzahl der Versuche und der Wertebereich der Zufallszahlen können vom Benutzer gewählt werden. Das Spiel ist beendet, wenn die maximale Anzahl an Versuchen erreicht wurde, oder aber der Spieler die geheime Zahl errät. *Tipp*: Sie können mit `userInput = int(input())` eine Zahl von der Kommandozeile einlesen. Verwenden Sie das Modul `random`, um Zufallszahlen zu erzeugen (<http://docs.python.org/3/library/random.html>).
2. Erweitern Sie das Programm um einen Modus für zwei Spieler. Diese spielen gegeneinander, und dürfen abwechselnd raten.

2 Einkaufsliste

Schreiben Sie ein Programm, das eine Einkaufsliste verwaltet. Es sollte zumindest möglich sein, neue Punkte hinzuzufügen, erledigte Einkäufe zu streichen, und die aktuelle Liste auszugeben. Verwenden Sie dazu ein Menü auf der Kommandozeile, das dem Benutzer die Optionen erklärt, etwa `l` für `list`, `a` zum Hinzufügen (*add*), und `r` zum Entfernen (*remove*).

3 Filmempfehlungen

Schreiben Sie ein Programm, welches Filme empfehlen kann. Als Datengrundlage können Sie das bekannte MovieLens¹-Datenset verwenden.

1. Lesen Sie eine (kleine) Variante des MovieLens-Datensets ein. Python bringt bereits einen CSV-Reader (und -Writer) mit².
2. Vergleichen Sie Filme anhand ihrer Tags. Sie können zum Beispiel die Anzahl identischer Tags verwenden. In einem Blog³ sind auch andere Ähnlichkeitsmaße gut erklärt. Überlegen Sie sich, welche Ähnlichkeitsmaße geeignet sind und warum.
3. Geben Sie die n ähnlichsten Filme zu einem gegebenen Film aus. Sie können mit `userInput = input()` eine Film-ID oder Namen von der Kommandozeile einlesen und dazu die passenden Filme ausgeben.

¹<https://grouplens.org/datasets/movielens/>

²<https://docs.python.org/3/library/csv.html>

³<http://dataaspirant.com/2015/04/11/five-most-popular-similarity-measures-implementation-in-python/>