

Aufgabe 1 Zusammengesetzte Datentypen

In der Vorlesung haben wir die zusammengesetzten Datentypen `list` und `tuple` angeschaut. Recherchieren Sie in Gruppen in der Pythondokumentation folgende Datentypen:

- `set` (Mengen)
- `dict` (Wörterbuch)

Erstellen Sie eine Übersicht zu der Syntax, nützlichen Methoden und Eigenschaften (Reihenfolge, veränderbar ect.). Die Gruppen stellen sich die Ergebnisse anschließend gegenseitig vor. Eine Vorlage mit den Informationen zu Listen und Tupeln finden Sie in der Datei `ZusammengesetzteDT-Vorlage.pdf` (auch `.tex` zum Ausfüllen).

Aufgabe 2 Zauberbücher (Ergänzung zu Aufgabe 3 Tutoriumsblatt 2)

Wir können in den Beutel auch noch ein Zauberbuch hinzufügen, welches wir mit einem Wörterbuch implementieren. In dem Zauberbuch steht zu jedem bekannten Zauber eine Beschreibung, es können beliebig viele Zaubersprüche eingetragen werden. Dann kann zu dem Code der letzten Woche zum Beispiel noch folgendes hinzugefügt werden:

```
print("Ich habe sogar ein Zauberbuch dabei, das hat tolle  
Zaubersprüche: ")

for zauber, beschreibung in my_character[4].items():
    print("Da gibt es den Zauber ", zauber)
    print("und der kann folgendes: ")
    print(beschreibung)
```

Aufgabe 3 Zwei-Faktor-Codematrix auslesen mit Python (auf Wunsch)

In dieser Aufgabe geht es darum, ein kleines Programm in Python zu bauen, das einem die Lösung der FU-Codematrix ausgibt.

Die Benutzung soll so aussehen, dass die Bezeichnungen der abgefragten Felder im Programm eingegeben werden sollen und die Lösung ausgegeben wird.

Im Folgenden wird eine mögliche Vorgehensweise beschrieben, wie man das Programm implementieren kann. Es ist Ihnen überlassen, ob Sie dem folgen möchten, oder nicht.

- Ein kurzer Text erklärt die Verwendung des Programms.
- Die von der Code-Matrix abgefragten Felder sollen eingelesen werden.
- Es muss verifiziert werden, dass die Felder alle gültig sind (erstes Zeichen liegt zwischen A und I, zweites Zeichen zwischen 1 und 9) und dass es sich um genau sechs Felder handelt.
- Die entsprechenden Werte werden ausgelesen. Hier können Sie wählen, ob Sie mit einem Wörterbuch oder einer Liste von Listen arbeiten möchten. (Testinstanzen können mit Funktionen `generateTestFUMatrix()` oder `generateTestFUDict()` aus `matrixGen.py` generiert werden.)
- Die Lösung wird auf der Konsole ausgegeben, die Werte getrennt von Leerzeichen.

optionaler Programmaufbau:

- (a) Matrix erzeugen:

ACHTUNG: Schreiben Sie ihre richtige Codematrix NICHT in den Quellcode des Programms!

Generieren Sie eine Testmatrix mit den bereit gestellten Funktionen oder legen Sie im selben Ordner, in dem ihr Programm liegt, eine Datei "secret_matrix.py" an.

Speichern Sie die Matrix in beiden Fällen in einer Variablen "MY_MATRIX" als eine Liste von neun neunelementigen Listen bzw. als Wörterbuch. Damit Sie bei der zweiten Variante Zugriff auf die Matrix haben, fügen Sie in ihr Programm `from secret_matrix import MY_MATRIX*` ein.

- (b) Funktion zur Abfrage Felder (`get_valid_coordinates()`):

- Bitten Sie die Nutzer:in, die benötigten Felder der Matrix nacheinander einzugeben.
- Damit bei einem Tippfehler nicht die gesamte Sequenz wieder eingegeben werden muss, überprüfen Sie die Gültigkeit der Eingabe, gleich an dieser Stelle.
- Speichern Sie die Position entweder als ein Tupel von Ints (bei Liste von Listen) (`<Nummer der Zeile>1, <Nummer Spalte>`) oder String der Länge 2 (Wörterbuch) ab.
- Falls eine Eingabe ungültig war, soll erneut um die Eingabe dieses Feldes gebeten werden.
- Denken Sie besonders an diese Fehlerquellen:
 - Zu lange Eingabe
 - Kleinbuchstaben / Großbuchstaben
 - ungültige Ziffer an zweiter Stelle
 - Keine Ziffer an zweiter Stelle
- Geben Sie die Liste von Koordinaten oder Strings zurück.

¹Wie bekomme ich aus einem Buchstaben eine Zahl? Hier bietet sich z.B. die `ord()` Funktion an oder Sie verwenden ein Wörterbuch.

(c) Funktion zu Ermittlung der Werte der Code-Matrix:

- Schreiben Sie eine Funktion `get_values([<koordinaten>])` mit folgender Spezifikation:

Voraussetzung: $0 \leq \text{Koordinate} \leq 8$ für alle (Koordinate, Koordinate) $\in \text{coordinates}$

Effekt: Keiner

Ergebnis: Eine Liste der Elemente aus `MY_MATRIX` an den Stellen der Koordinaten ist geliefert.

(d) Alles zusammenfügen:

Rufen Sie `get_values()` auf der Ausgabe von `get_valid_coordinates()` auf und geben Sie das Ergebnis aus. Geben Sie am Anfang des Programms einen kurzen Text aus, der die Verwendung des Programms erklärt.

Aufgabe 4 Zwei-Faktor-Bonus Copy-Paste

Das Folgende können Sie noch umsetzen, falls Sie Lust haben, das Programm aus Aufgabe 3 etwas zu verfeinern (ob Sie das tun wollen ist jedoch vollkommen Ihnen überlassen). Dazu wird das Einlesen der Felder und die Ausgabe der Werte über die Zwischenablage gehandhabt, sodass die Felder im Browser mit `strg + c` kopiert werden können und die Werte mit `strg + v` eingefügt werden.

(a) Dafür wird das Programm ein Modul, `pyperclip`², verwenden, um unabhängig vom Betriebssystem auf die Zwischenablage zuzugreifen. Dieses Modul ist nicht in der Standardbibliothek enthalten und muss heruntergeladen werden. Python bietet dafür ein eigenes Programm an, nämlich `pip`. Geben Sie in einer Kommandozeile `pip install pyperclip` ein³.

Achtung: für die Verwendung unter Linux ist die Installation von `xclip` über den Paketmanager erforderlich.⁴

(b) Um das neue Modul verwenden zu können, müssen wir es mit `import pyperclip as pc` importieren. Jetzt kann auf die Funktionen des Moduls mit `pc.copy()` und `pc.paste()` zugegriffen werden.

(c) Schreiben Sie Ihre Funktion `get_valid_coordinates()` so um, dass der String, der von `pc.paste()` zurückgegeben wird, in eine Liste der Feldbezeichnungen zerlegt wird und jedes Feld auf seine Gültigkeit geprüft wird (paste umgeht iterative Eingabe). Zum Zerlegen können Sie die `split`-Funktion nutzen (Beispiel: `"a,1b,1c".split(",1")` \Rightarrow `['a', 'b', 'c']`). Denken Sie an sinnvolle Meldungen für die Nutzer:in, wenn keine gültige Folge von sechs Feldern gefunden werden konnte.

²<https://pypi.org/project/pyperclip/>

³Eine Anleitung, wie `pip` auf verschiedenen Betriebssystemen verwendet wird gibt es hier: <https://automatetheboringstuff.com/3e/appendixa.html>.

Dabei handelt es sich um ein kostenfrei lesbares Buch zum Einstieg in die Pythonprogrammierung, das auch wunderbar als Nachschlagewerk dienen kann.

⁴Unter Ubuntu/Debian verwenden Sie einfach `sudo apt install xclip`, unter Fedora `sudo dnf install xclip`

- (d) Ersetzen Sie die Ausgabe der Werte mit einer Meldung, dass das Programm erfolgreich war und kopieren Sie die Folge von Werten mit `pc.paste(<String von Werten>)` in die Zwischenablage.