

Aufgabenblatt 3

Abgabe: Die Lösungen sollten zeitnah als IPYNB-Datei fertiggestellt werden. Der Code muss ausreichend kommentiert sein und die Variablen müssen sinnvoll benannt werden. Sie müssen die Aufgabe selbst programmiert haben. Sie können Fragen in Form von Kommentaren im Code stellen, falls etwas nicht funktioniert hat. Die Antwort erfolgt dann im Praktikum mündlich. Sie dürfen nie mehr als drei Aufgabenblätter im Rückstand sein.

Hilfsmittel: Kein Copy-Paste aus dem Internet, alles muss selbstständig programmiert sein. Sie dürfen die IPython-Notebook-Skripte aus der Vorlesung (liegen nach der jeweiligen Vorlesung auf Ilias) und die Python-Einführung auf Ilias verwenden. Außerdem ist die Hilfe-Funktion `help(...)` und die Methode `dir(...)` zur Auflistung der verfügbaren Funktionen zu empfehlen.

Anwesenheit: Grundsätzlich herrscht Anwesenheitspflicht. Ein Attest ist notwendig, wenn jemand nicht kommen kann. Bei unentschuldigtem Fehlen ist das Praktikum nicht bestanden. Von der Teilnahme an der Klausur wird in diesem Fall dringend abgeraten.

Aufgabe 3.0

Schreiben Sie die Funktion `scheresteinpapier()`, welche das Spiel Schere, Stein, Papier spielt. Der Computer soll eine zufällige Auswahl treffen und der Mensch soll eine zufällige Eingabe machen. Bei Gleichheit oder wenn der Mensch verloren hat, wird wiederholt. Wenn der Mensch gewinnt, ist das Spiel zu Ende.

```
>>> scheresteinpapier ()
Eingabe (SCHERE, STEIN, PAPIER)? schere
schere schere
unentschieden
Eingabe (SCHERE, STEIN, PAPIER)? papier
papier schere
Verloren. Nochmal.
Eingabe (SCHERE, STEIN, PAPIER)? papier
papier stein
gewonnen
```

Aufgabe 3.1

Für die Implementierung des folgenden Spiels wird eine Zufallszahl benötigt. Diese können Sie wie folgt bekommen:

```
from random import randint
r = randint ( 0, 100 )  # Zufallszahl zwischen 0 und 100
```

einbinden können.

- (a) Schreiben sie eine Funktion `ratespiel(n)`, die eine zufällige Zahl zwischen 0 und `n` erzeugt und den Benutzer so oft interaktiv raten lässt, bis er die Zahl erraten hat. Nach jedem Rateversuch, den der Benutzer über die Tastatur eingeben soll, muss die Funktion als Rückmeldung entweder 'zu gross', 'zu klein' oder 'richtig' auf dem Bildschirm ausgeben. Sobald der Benutzer die Zahl richtig erraten hat, soll die Funktion beendet werden und die Anzahl der Rateversuche als Integer-Zahl zurückliefern.

Aufgabe 3.2

Schreiben Sie eine Python-Funktion, die eine Liste aller Worte aus Kleinbuchstaben (keine Umlaute) erzeugt, die aus genau 4 Buchstaben bestehen und die in der Mitte **nicht** zwei Konsonanten enthält.

Beispiel der Liste: ['aaaa', 'aaab', 'aaac', ..., 'zzzx', 'zzzy', 'zzzz']

Länge der Liste: $456976 = 26^4$

Mit der Konsonaten-Bedingung:

['aaaa', 'aaab', 'aaac', ..., 'zzux', 'zzuy', 'zzuz']

Länge der Liste: 158860

Aufgabe 3.3

Gegeben sei ein in der Variablen `d` befindliches `dict`-Objekt. Geben Sie Python-Code an, um ...

- (a) zu berechnen, wie viele Einträge `d` enthält.
- (b) zu berechnen, ob es einen Schlüssel 'Hallo' gibt.
- (c) zu entscheiden, ob alle Schlüssel den Typ `int` haben.
- (d) einen neuen Eintrag zu erzeugen, der dem Schlüssel 1 den Wert 2 zuordnet.