

Ablauf der Übung

Deadline: : 19.10.2020 23:55 Uhr

- Die finalen Python Dateien sollen am E-Campus in der Übungsabgabe als ZIP Datei hochgeladen werden.
- Jedes Python Programm enthält in den ersten Zeilen einen Header in dem der Name, Autor, eine kurze Beschreibung enthalten sind.
- Der Code soll sauber geschrieben und wichtige Code-Zeilen kommentiert werden. Der Output sollte genauso wie in der Angabe aussehen!

```
"""
<Name>
<Description>
<Mat-Nr>
"""
```

U1Bsp1

Gegeben ist das Tuple `t = ('P', 'R', 'O', 'G', '1', '_', 'V', 'O')`

Das Python Programm soll

- das 3. Element ausgeben (O)
- das zweitletzte Element ausgeben
- den selben Tupel hinten anhängen und t ausgeben
- Ausgeben wie oft der Buchstabe O im Tupel auftritt (tuple.count() Methode)

Output:

```
O
V
('P', 'R', 'O', 'G', '1', '_', 'V', 'O', 'P', 'R', 'O', 'G', '1', '_', 'V', 'O')
4
```

U1Bsp2

Schreiben Sie ein Python Programm, das überprüft ob ein übergebener String ein [Palindrom](#) ist (Tipp: list.reverse() Methode. Der String wird mittels input() übergeben. Achten Sie dabei, dass das eingelesene Wort sofort normalisiert wird (Kleinbuchstaben oder Großbuchstaben). Als Ergebnis soll nur True oder False ausgegeben werden. (Überprüfung kann direkt im print erfolgen, `word == tmp`)
Testwörter: Lagerregal Kajak Ebbe

Output gleich:

```
Wort eingeben: Kajak
True
```

U1Bsp3

Gegeben sind zwei Listen `a = [1, 1, 'hello', 'hello', 2, 5, 8, 13, 22, 2, 'world']`
`b = [55, 5, 1, 2, 2, 5]`

Das Python Programm soll:

- die Duplikate der Listen entfernen und die restlichen Elemente zusammenfügen
- die Unterschiede beider Listen ohne Duplikate ausgeben

Übungsblatt 2

Abgabe als ZIP-Datei

Deadline: 19.10.2020 23:55 Uhr

- Die Liste b sortiert ohne Duplikate

Tipps: `set.union()` Methode, `set.difference()` Methode, `sorted(set())` Methode

Output (kann auch als Liste ausgegeben werden):

```
{1, 2, 5, 8, 'world', 13, 'hello', 22, 55}
{8, 'world', 13, 'hello', 22}
[1, 2, 5, 55]
```

U1Bsp4

Gegeben ist das Dictionary `pets = {'cat': 4, 'dog': 15, 'bird': 5, 'velociraptor': 150}`

Das Python Programm soll:

- einen weiteren Eintrag mouse mit dem Wert 1 hinzufügen
- den Wert von Dog auf 20 ändern
- den Eintrag bird löschen
- mit einer for loop den gesamten Inhalt in Form von Key: Value ausgeben

Output z.B. `print(f'{key}: {value}')`:

```
cat: 4
dog: 20
velociraptor: 150
mouse: 1
```

U1Bsp5

Ratespiel: Umgekehrter Weg in `1_random_example.py` zu sehen.

Diesmal sollen nicht Sie die Zufallszahl erraten, sondern der Computer, d.h.

- Sie haben im Kopf ihre Nummer (1-100)
- das Programm ratet eine Zahl
 - Sie als User sagen niedriger (lower) oder höher (higher) oder richtig für die korrekte Zahl
- Abschließend soll ihr Programm die Anzahl der Versuche ausgeben

Tipp: Sie können ihr Programm vorerst immer um eins höher/niedriger raten lassen

Ziel der Übung ist es strategisch vorzugehen:

- Überlegen Sie sich oder diskutieren Sie mit ihren KollegInnen eine Strategie aus und setzen Sie diese um z.B.: Mitte des Bereichs wird immer geraten

Output genauso anpassen:

```
computer guess: 50
lower
computer guess: 25
higher
computer guess: 37
higher
computer guess: 44
higher
computer guess: 45
higher
computer guess: 46
correct - 6 attempts
```

U1Bsp6

Gegeben ist folgender Satz als String:

```
sentence = 'Jim quickly realized that the beautiful gowns are expensive'
```

Erstellen Sie ein Dictionary **count_letters** mit Keys für Buchstaben im Satz. Die Werte der Keys sollen die Häufigkeit des Vorkommens (int) darstellen (z.B. {'J': 1, 'i': 5, 'm': 1, ...})

Geben Sie abschließend das Dictionary aus.

Output:

```
{'J': 1, 'i': 5, 'm': 1, 'q': 1, 'u': 3, 'c': 1, 'k': 1, 'l': 3, 'y': 1, 'r': 2, 'e': 8, 'a': 4, 'z': 1, 'd': 1, 't': 4, 'h': 2, 'b': 1, 'f': 1, 'g': 1, 'o': 1, 'w': 1, 'n': 2, 's': 2, 'x': 1, 'p': 1, 'v': 1}
```

U1Bsp7

- Programmieren Sie in Python Lotto 6 aus 45 (ohne Zusatzzahl)
- Hinweis:
 - Versuchen Sie den Code in Funktionen aufzuteilen und diese aufzurufen

1) Eingabe: Das Programm soll vom Benutzer einen Lottotipp entgegennehmen. Hierfür sollen sie sechs Zahlen per Standardeingabe in ein Array einlesen. Stellen Sie sicher, dass nur Zahlen zwischen 1 und 45 erlaubt sind und keine Zahl doppelt verwendet werden darf. Lesen Sie danach die Anzahl der Ziehungen (max. 1). Überprüfen Sie die Eingaben auf Zahlen (z.B. isNumeric), bei einer falschen Eingabe (zu viele, zu wenige oder doppelte Zahlen) soll der User erneut nach den Zahlen gefragt werden.

Die Eingabe können Sie mit `.split()` nach Leerzeichen in eine Liste splitten somit haben Sie alle Eingaben wie folgt `[20, 17, 15, 10, 1, 45]`.

2) Ziehungen (ohne Zusatzzahl): Bei einer Ziehung werden insgesamt 6 Zahlen zufällig im Bereich 1-45 gezogen. Stellen sie sicher, dass keine Zahl doppelt gezogen werden kann. Überlegen Sie sich, welche Möglichkeiten es gibt, soll eine Ziehung zu realisieren.

3) Vergleichen sie die Ziehung (= zufällig gezogenen „Kugeln“) mit dem abgegebenen Tipp. Stellen sie fest, ob es sich um einen 0er (keine richtig), 1er, 2er, 3er, 4er, 5er oder Lotto-6er handelt

Output:

```
tip input: 20 17 15 10 1 45
draw count: 150
0er: 101
1er: 20
2er: 10
3er: 9
4er: 8
5er: 2
6er: 0
```