

## Ablauf der Übung

Deadline: : 15.10.2020 09:00 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>  
<Author, Mat-Nr>  
"""
```

---

### U1Bsp1

Schreiben Sie ein Python Programm, das einen String in folgendem Format mittels print() ausgibt (verwenden Sie die Strings-Sonderzeichen (Slide 62):

```
Twinkle, twinkle, little star,  
    "How I wonder what you are!"  
        Up above the world so high,  
        Like a diamond in the sky.  
Twinkle, twinkle, little star,  
    \How\ I wonder what you are.
```

### U1Bsp2

Gegeben ist die Variable text → `text = 'PROG VO bis'`

Schreiben Sie ein Python Programm, das

- die ersten 4 Zeichen des Strings ausgibt (PROG)
- den Teil VO ausgibt
- die letzten 2 Zeichen ausgibt
- die Länge von `text` ausgibt

Output:

```
PROG  
VO  
is  
11
```

### U1Bsp3

Lesen Sie via CLI-Input zwei Zahlen ein und speichern Sie diese in die Variable x und y. Multiplizieren Sie beide Zahlen (in das richtige Datenformat umwandeln – siehe Slide 65-67)

Abschließend geben Sie die gesamte Rechnung inkl. Ergebnis mittels print() aus.

Output:

```
Zahl1: 10
```

# Übungsblatt 1

Abgabe als ZIP-Datei

Deadline: 15.10.2020 09:00 Uhr

Zahl2: 5  
Rechnung  $10 * 5 = 50$

## U1Bsp4

Schreiben Sie ein Python Programm welches einen Rabatt aufgrund der Betragshöhe berechnet und ausgibt. Der Betrag wird als ganze Zahl durch den Benutzer als Input übergeben.

Der Rabatt wird anhand des Betrages berechnet:

- weniger als 1000: 5 %
- von 1000 bis 1999: 10 %
- von 2000 bis 4999: 15 % + danach noch 2% Skonto
- darüber: 20 % + danach noch 3% Skonto

Geben Sie abschließend den dementsprechenden Zahlungsbetrag aus.

Output:

Betrag (€): 2000  
Zahlungsbetrag: 1666.0€

## U1Bsp5

Schreiben Sie folgendes Python Programm. Es werden 3 Inputs übergeben:

- 2 Strings
- 1 int

Das Zeichen an der Stelle [int] der beiden Strings soll verglichen werden. Überprüfen Sie vor dem Vergleich ob sich der übergebene int Wert im Bereich der String Indizes befindet (Länge vergleichen mit int-Wert). Wenn die Strings gleich sind normalisieren Sie diese auf Kleinbuchstaben `<str variable>.lower()` und geben Sie das Zeichen aus. Bei Ungleichheit geben Sie nur ungleich aus.

Output gleich:

String1: test  
String2: test  
Index: 2  
s

Output ungleich:

ungleich

## U1Bsp6

Schreiben Sie ein Python Programm, das eine Passwortabfrage mittels einer While Schleife durchführt. Das Passwort wird mittels `input()` übergeben. Nur bei richtiger Eingabe soll eine Meldung erscheinen (Hinweis: `else`-Zweig der While-Schleife). Nach 5 falsch eingegebenen Passwörtern soll die While-Schleife beendet werden. Falls eine leere Eingabe erfolgt (nur Enter) soll der Counter nicht erhöht werden (*continue* innerhalb Schleife).

Abschließend soll beim richtigen Passwort Login erfolgreich und bei zu vielen Versuchen Login gesperrt ausgegeben werden.

Output bei richtigem Passwort:

Login erfolgreich

Output nach 5 falschen Versuchen:

Login gesperrt

## U1Bsp7

Schreiben Sie ein Programm, das die Zahlen von 1 bis 50 mit den folgenden Ausnahmen ausgibt:

Ist eine Zahl ohne Rest durch 3 teilbar, soll stattdessen der Text HIPP ausgegeben werden.

Ist eine Zahl ohne Rest durch 5 teilbar, soll stattdessen der Text HOPP ausgegeben werden.

Ist eine Zahl sowohl ohne Rest durch 3 als auch ohne Rest durch 5 teilbar, soll stattdessen der Text HIPP HOPP ausgegeben werden.

Brechen Sie ab der Zahl 19 die for-Schleife ab.

Tipp: Rest mit Modulo (Operator %) siehe Slide 70-71) berechnen

### Output:

```
HIPP
HOPP
HIPP
HIPP
HOPP
HIPP
HIPP HOPP
HIPP
```

## U1Bsp8

Schreiben sie ein Programm, das einen ganzzahligen Euro Betrag einliest und anschließend in die dafür notwendigen Scheine und Münzen ausgibt. Es soll die optimale Anzahl an Scheinen und Münzen ausgeben.

Input (Euro): 304

Resultat: 1 x 200-Euro-Scheine, 1 x 100-Euro-Schein, 2 x 2 Euro-Münzen

## U1Bsp9

Schreiben sie ein Programm, dass die Fibonacci-Reihe ausgibt. Beginnen sie mit 0 und 1. Im Anschluss ergibt jeweils die Summe zweier aufeinanderfolgender Zahlen die unmittelbar danach folgende Zahl.

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

Geben sie mindestens die ersten 50 Zahlen aus.