

Aufgabenblatt 1

Allgemeine Informationen zum Aufgabenblatt:

- Die Abgabe erfolgt in TUWEL. Bitte laden Sie Ihre Python-Datei bis spätestens **Freitag, 27.03.2020 20:00 Uhr** in TUWEL hoch.
- Beachten Sie bitte folgende Punkte
 - Ihre Programme sollten ausführbar sein, d.h. einzelne Programme sollten z.B. keine Syntaxfehler oder Typfehler enthalten.
 - Bei kleinen logischen Fehlern (falsche Ergebnisse) werden wir keine Punkte abziehen. Daher sollten Sie immer versuchen, alle Aufgaben vollständig abzugeben.
 - Ihre Programme sollten nur Konstrukte verwenden, die bisher in der Vorlesung vorgekommen sind. Sie sollten daher keine speziellen Aufrufe verwenden, die möglicherweise einzelne Aufgaben abkürzen.

In diesem Aufgabenblatt werden folgende Themen behandelt:

- Variablen
- Datentypen
- Verzweigungen

Aufgabe 1 (1 Punkt)

In dieser Aufgabe müssen Sie unterschiedliche Ausdrücke überprüfen und zu jedem Ausdruck einen kurzen Kommentar in die Abgabedatei schreiben. Ein Kommentar sollte enthalten:

- Kurze Beschreibung der Operation
- Ergebnis
- Typ des Ergebnisses

Dazu zwei Beispiele:

```
3 + 4.0    # Addition zweier Zahlen, 7.0, float
```

```
"Hello"*2 # String vervielfältigen, "HelloHello", str (String)
```

Sie dürfen zusätzlichen Python-Code schreiben, um die Ergebnisse und Typen zu überprüfen. Die Antworten sollten aber so strukturiert sein, dass eine Kommentarzeile immer neben dem jeweiligen Beispiel steht.

Folgende Ausdrücke sind zu überprüfen:

- `2 ** 3`
- `5.0 * 2.5`
- `"One" + "Two"`
- `2 * (5 / 2)`
- `22 // 5 % 2`
- `int("10") + float(5 // 2)`
- `"Hello World "[6:] + "Hello World"[:5]`
- `"step on no pets"[:-1]`
- `2 < 3 and 4 < 5`
- `bool("") or False`

Aufgabe 2 (1 Punkt)

In dieser Aufgabe müssen Sie die folgenden zwei Programme schreiben:

1. Erweitern Sie das Programm für die Schaltjahresberechnung aus den Folien um folgende Funktionalität:

- Lesen Sie zuerst das aktuelle Kalenderjahr (`current_year`) ein.
- Lesen Sie danach das Jahr (`year`) ein, für das die Berechnung durchgeführt werden soll.
- Unterscheiden Sie dann folgende Fälle:
 - `year < current_year`: Ausgabe mit Jahr gefolgt von `: Long time ago -` und danach die Unterscheidung (`"not a leapyear"` oder `"a leapyear"`)
 - `year > current_year`: Ausgabe mit Jahr gefolgt von `: The future is always beginning now -` und danach die Unterscheidung (`"not a leapyear"` oder `"a leapyear"`)
 - `year == current_year`: Ausgabe mit Jahr gefolgt von `:"` und danach die Unterscheidung (`"Not a leapyear"` oder `"A leapyear"`)

- Beispiele für den Ablauf des Programms:

```
Current year: 2020
Year to analyze: 1997
1997 : Long time ago - not a leapyear
```

```
Current year: 2020
Year to analyze: 2020
2020 : A leapyear
```

```
Current year: 2020
Year to analyze: 2024
2024 : The future is always beginning now - a leapyear
```

2. Sie haben in der Angabedatei schon eine Implementierung gegeben. Hier müssen Sie zuerst das gegebene Programm verstehen und dann verbessern.

- Ergänzen Sie diese Implementierung um ein paar Kommentarzeilen (vor dem Programmcode), in denen Sie beschreiben, was das Programm macht.
- Implementieren Sie dann eine alternative Variante, die bei jedem Vergleich weniger Zuweisungen und daher weniger Zeilen benötigt.

Aufgabe 3 (1 Punkt)

Implementieren Sie ein Programm, das ein eingelesenes Gewicht (**weight**) einer Kategorie zuordnet. Dabei wird davon ausgegangen, dass nur ganze Zahlen eingelesen werden. Die Kategorisierung ist in Tabelle 1 aufgelistet. In der dritten Spalte finden Sie den String, der bei einem bestimmten Gewicht ausgegeben werden soll.

Kategorie	Gewichtsbereich	Ausgabestring
Extra Leicht	1kg – 4kg	"Gewicht --> Extra Leicht"
Leicht	5kg – 9kg	"Gewicht --> Leicht"
Mittelschwer	10kg – 24kg	"Gewicht --> Mittelschwer"
Schwer	25kg – 49kg	"Gewicht --> Schwer"
Extra Schwer	≥ 50 kg	"Gewicht --> Extra Schwer"

Tabelle 1: Gewichtskategorisierung mit den dazugehörigen Ausgabestrings.

Falls eine negative Zahl oder die Zahl 0 eingegeben wird, dann sollte die Ausgabe lauten: "Falsche Eingabe! Gewicht muss > 0 sein!" Realisieren Sie diese Gewichtskategorisierung mittels if-Anweisungen auf zwei verschiedene Arten:

1. Verschachtelung einfacher if-Anweisungen

- Kein elif erlaubt
- Kein else erlaubt
- Keine logischen Operatoren (and, or) erlaubt, Bedingungen dürfen nur eine einfache Form (z. B. $x < 0$) haben, also z. B.

```
if ...:
    if ...:
        ...
```

2. Mehrfachverzweigung (mit elif und else), also z. B.

```
if ...:
    ...
elif ...:
    ...
elif ...:
    ...
```