

# Serie 06

## Programmieren für Naturwissenschaften FS 2023

Gruppe: Sofia Kessler, Florian Mohaupt, Lukas Batschelet

## Aufgabe 1: Operatorenreihenfolge

### Aufgabe

Betrachten Sie den untenstehenden Python-Code und geben Sie an, welche Werte in den Variablen `a`, `b`, `c`, `d`, `e` und `f` gespeichert sind, wenn der Code vollständig durchlaufen wurde. Erwähnen Sie jeweils auch den Datentyp der jeweiligen Variable. Versuchen Sie die Aufgabe ohne einen Computer zu lösen.

```
a, b = 2, 2
c = 4

d = a + b * c
c *= c % a
e = d / a + c
f = e // b
```

## Lösung

```
d = a + b * c
d = 2 + 2 * 4
d = 10 # int

c *= c % a
c = c * (c % a)
c = 4 * (4 % 2)
c = 0 # int

e = d / a + c
e = 10 / 2 + 0
e = 5.0 # float

f = e // b
f = 5.0 // 2
f = 2.0 # float
```

## Aufgabe 2: Notenberechnung

### Aufgabe

Schreiben Sie ein Programm, welches vom Benutzer die maximal möglichen und die tatsächlich erreichten Punkte einer Prüfung einliest ( $m$  und  $e$ ). Danach berechnen Sie anhand der Formel

$$Note = \frac{e}{m} \cdot 5 + 1$$

die  $Note$  und geben diese auf zwei Nachkommastellen gerundet aus (hierzu können Sie die eingebaute Funktion `round` verwenden).

## Mögliche Lösung

```
points_max = int(input("Maximal erreichbare Punkte: "))
points = int(input("Erreichte Punkte: "))

# Notenberechnung
grade = (points / points_max) * 5 + 1
round(grade, 3)

print("Maximale Punkte:", points_max)
print("Erreichte Punkte:", points)
print("Note:", grade)
```

[Die abgelegte Datei](#), fügt zusätzlich eine Schleife ein um mehrere Noten berechnen zu können. Ebenfalls wird eine gewisse Wertprüfung durchgeführt.

## Aufgabe 3: Listenmanipulationen

### Aufgabe

Gegeben sei eine Liste `numbers`, welche ganze Zahlen enthält. Schreiben Sie je ein Code-Fragment, das Folgendes erledigt:

1. Fügen Sie an das Ende der Liste `numbers` den Wert `17` hinzu.
2. Fügen Sie der Liste `numbers` den Wert `99` an Position `0` hinzu.
3. Addieren Sie zum Element an Position `4` in `numbers` den Wert `18`.
4. Lesen Sie das Element an Position `3` aus `numbers` aus und weisen Sie dieses der Variable `value` zu.
5. Geben Sie die Grösse der Liste `numbers` aus.
6. Ersetzen Sie das Element an Position `1` mit dem Wert `17`

## Mögliche Lösung

```
# Erstellen der Liste
numbers = (list(range(0,5,1)))

# Manipulationen
numbers.append(17)
numbers.insert(0, 99)
numbers[4] += 18
value = numbers[3]
print("Länge der Liste:", len(numbers))
numbers[1] = 1

print(numbers)
```

## Aufgabe 4:

### Aufgabe

Betrachten Sie die Liste `values` in folgender Abbildung:

values	-1	99	7	7	4	3	4	5	8
	0	1	2	3	4	5	6	7	8

Was ist:

1. `values[1]`
2. `values[2] + values[5]`
3. `values[2+5]`
4. `len(values)`
5. `values[len(values)-1]`
6. `values[1:3]`
7. `values[:4]`
8. `values[2:]`
9. `values[::2]`
10. `values[-1:1:-1]`

## Lösung

1. 99
2. 10
3. 5
4. 9
5. 8
6. [99, 7]
7. [-1, 99, 7, 7]
8. [7, 7, 4, 3, 4, 5, 8]
9. [-1, 7, 4, 4, 8]
10. [8, 5, 4, 3, 4, 7, 7]