

# Serie 06B

## 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, 1
c = 3

d = c + a - b
e = a * b + a
e += a ** c
f = e // a / b
c *= c & b
a = e + f - c
```

## Lösungen

```
d = c + a - b
d = 3 + 2 - 1
d = 4 # int

e = a * b + a
e = 2 * 1 + 2
e = 4 # int

e += a ** c
e = e + (a ** c)
e = 4 + (2 ** 3)
e = 12 # int

f = e // a / b
f = 12 // 2 / 1
f = 6.0 # float

c *= c % b
c = c * (c % b)
c = 3 * (3 % 1)
c = 0 # int

a = e + f - c
a = 12 + 6.0 - 0
a = 18.0 # float
```

## Aufgabe 2: Listenmanipulationen

### Aufgabe

Betrachten Sie die Liste `values` in folgender Abbildung:

|               |           |          |          |          |          |          |           |          |          |
|---------------|-----------|----------|----------|----------|----------|----------|-----------|----------|----------|
| <b>values</b> | <b>-1</b> | <b>2</b> | <b>7</b> | <b>7</b> | <b>4</b> | <b>3</b> | <b>99</b> | <b>5</b> | <b>8</b> |
|               | 0         | 1        | 2        | 3        | 4        | 5        | 6         | 7        | 8        |

Was ist:

1. `values[2]`
2. `values[1] + values[3]`
3. `values[1 + 3]`
4. `values.index(7)`
5. `values.pop()`
6. `values[3:5] = [5, 7]`
7. `values.sort()`
8. `values[2:]`
9. `values[-1:1:-1]`

Welche Code-Fragmente werden benötigt um folgende Aufgaben zu erledigen?

10. Addieren Sie zum Element an Position `4` den Wert `18`.
11. Lesen Sie den Wert an Position `3` aus und speichern Sie ihn in der Variable `number`
12. Ersetzen Sie den Wert an Position `2` mit dem Wert an Position `7` und umgekehrt

## Lösungen

1. `values[2] = 7`
2. `values[1] + values[3] = 9`
3. `values[1 + 3] = 4`
4. `values.index(7) = 2`
5. `values.pop() = 8`  
- Neue Liste: `[-1, 2, 7, 7, 4, 3, 99, 5]`
6. `values[3:5] = [5, 7]`  
- Neue Liste: `[-1, 2, 7, 5, 7, 3, 99, 5]`
7. `values.sort()`  
- Neue Liste: `[-1, 2, 3, 5, 5, 7, 7, 99]`
8. `values[2:] = [3, 5, 5, 7, 7, 99]`
9. `values[-1:1:-1] = [99, 7, 7, 5, 5, 3]`
10. `values[4] += 18`  
- Neue Liste: `[-1, 2, 7, 5, 25, 3, 99, 5]`
11. `number = values[3]`
12. `values[2], values[7] = values[7], values[2]`  
- Neue Liste: `[-1, 2, 99, 5, 25, 3, 7, 5]`

### Aufgabe 3:

#### Aufgabe

Schreiben Sie ein Programm, welches einen Radius  $r$  vom Benutzer abfragt und anschliessend die Fläche  $f$  und den Umfang  $u$  eines Kreises mit Radius  $r$  berechnet. Die Resultate sollen auf **zwei** Nachkommastellen gerunden ausgegeben werden (hierzu können Sie die eingebaute Funktion `round` verwenden).

### Mögliche Lösung

```
import math

radius = float(input("Geben Sie den Radius ein: "))

area = radius ** 2 * math.pi
scope = radius * 2 * math.pi

print("Radius =", radius)
print("Fläche =", round(area, 2))
print("Umfang =", round(scope, 2))
```