

Übungsblatt 2

Aufgabe 1

Schreiben Sie ein Programm, das das folgende Muster ausgibt:

```
1 ( 1 0 )
2 ( 2 0 ) ( 2 1 )
3 ( 3 0 ) ( 3 1 ) ( 3 2 )
4 ( 4 0 ) ( 4 1 ) ( 4 2 ) ( 4 3 )
5 ( 5 0 ) ( 5 1 ) ( 5 2 ) ( 5 3 ) ( 5 4 )
6 ( 6 0 ) ( 6 1 ) ( 6 2 ) ( 6 3 ) ( 6 4 ) ( 6 5 )
7 ( 7 0 ) ( 7 1 ) ( 7 2 ) ( 7 3 ) ( 7 4 ) ( 7 5 ) ( 7 6 )
8 ( 8 0 ) ( 8 1 ) ( 8 2 ) ( 8 3 ) ( 8 4 ) ( 8 5 ) ( 8 6 ) ( 8 7 )
9 ( 9 0 ) ( 9 1 ) ( 9 2 ) ( 9 3 ) ( 9 4 ) ( 9 5 ) ( 9 6 ) ( 9 7 ) ( 9 8 )
```

Aufgabe 2

Was ist die Ausgabe des folgenden Programms?

```
1 a = 5
2 b = 10
3 print(a == b)
4 print(a != b)
5 print(True == (a == b))
6 print(True == (a != b))
7 print((True == (a != b)) or (True == (a == b)))
8 print(True and False and True)
9 print(True and False and False or True)
10 print((True and False) and (False or True))
11 print(True and (False and False) or True)
```

Aufgabe 3

Setzen Sie in der folgenden Berechnung alle Klammern, um die Präzedenzen der verwendeten Operatoren zu verdeutlichen.

Tipp: Sie können die Präzedenzen der Operatoren zum Beispiel auf <https://www.programiz.com/python-programming/precedence-associativity> finden.

```
1 e = True
2 f = False
3 g = - 2 ** 8 / 9 + 20 <= 250 and not e or f
```

Aufgabe 4

Schreiben Sie den Funktionskörper einer Funktion, die eine Zahl n als Eingabe erhält und alle Zahlen zwischen 0 und n ausgibt, welche durch 7 teilbar sind. Verwenden Sie dazu eine `while`-Schleife.

```
1 def meine_funktion(n):
2     # Ihr Code
```

Beispiel: Wird `meine_funktion(49)` aufgerufen, soll Folgendes ausgegeben werden:

```
1 0 7 14 21 28 35 42 49
```

Aufgabe 5

Schreiben Sie den Funktionskörper einer Funktion, die zwei Zahlen x und y als Eingabe erhält und die kleinste Zahl findet, die sowohl durch x als auch durch y teilbar ist. Die Funktion soll diese Zahl zurückgeben.

Lösen Sie diese Aufgabe, ohne eine Multiplikation zu verwenden.

```
1 def meine_zweite_funktion(x, y):
2     # Ihr Code
```

Aufgabe 6

In der folgenden Funktion ist ein Logikfehler passiert. Finden Sie den Fehler und korrigieren Sie die Funktion.

```
1 def falsche_funktion(a, b):
2     if a and b != 1:
3         print("Weder a noch b sind gleich 1.")
4     else:
5         print("Entweder a oder b oder sogar beide sind gleich 1.")
```

Aufgabe 7

Erklären Sie, weshalb das folgende Programm die gezeigte Ausgabe produziert.

```
1 a = 100
2 b = 200
3
4 def fun1():
5     print(a, b)
6
7 def fun2():
8     a = 2
9     print(a, b)
10
11 def fun3():
12     a = 6
13     b = 10
14     print(a, b)
15
16 fun1()
17 fun2()
18 fun3()
```

Ausgabe:

```
1 100 200
2 2 200
3 6 10
```

Aufgabe 8

Erklären Sie, weshalb `fun4()` nicht ausführbar ist, aber `fun5()` funktioniert.

```
1 a = 100
2 b = 200
3
4 def fun4():
5     a += 1
6     b += 1
7     print(a, b)
8
9 def fun5():
10    a = 0
11    b = 10
12    a += 1
13    b += 1
14    print(a, b)
```