

Serie 05

Programmieren für Naturwissenschaften FS 2023

Gruppe: Sofia Kessler, Florian Mohaupt, Lukas Batschelet

Aufgabe 1: Hello World

Aufgabe

Schreiben Sie ein Programm, welches folgende Ausgabe generiert (ersetzen Sie dabei [Ihr Name] mit Ihrem Namen):

```
Hallo, mein Name ist [Ihr Name].  
Ich freue mich, Python zu lernen!
```

Zusatzaufgabe: Schaffen Sie es, die zweizeilige Ausgabe mit einer einzigen print Anweisung zu erzeugen?

```
print("Hallo unsere Namen sind Sofia, Florian und Lukas", "Wir freuen uns Python zu lernen!", sep="\n")
```

Aufgabe 2: Kommentare und Skript-Modus

Aufgabe a

Begründen Sie, weshalb die folgenden zwei Kommentare nicht optimal sind:

```
print("Hallo") # gibt Hallo aus  
print("test") # muss später geändert werden
```

- # gibt Hallo aus gibt keine neue Information welche nicht sofort aus dem Programm klar wird.
- # muss später geändert werden erklärt nicht, was, wann, wo und weshalb etwas geändert werden muss.

Aufgabe b

Welche Ausgabe erzeugen die folgenden zwei Anweisungen, wenn diese im Skript-Modus ausgeführt werden (Hinweis: `"\n"` repräsentiert das Zeilenumbruchzeichen)?

```
print("Hallo", "Python!", "Alles klar?", sep="\n", end=" ")  
print("Ja!")
```

```
Hallo  
Python!  
Alles klar? Ja!
```

Aufgabe 3: Rechnung

Aufgabe

Schreiben Sie ein Programm, das den arithmetischen Ausdruck $3 \cdot (2^5 + 17)^2$ ausrechnet und das Resultat der Berechnung folgendermassen ausgibt:

```
3 * (2^5 + 17)^2 = 7203
```

```
print("3 * (2^5 + 17)^2 =", end=" ")  
print(3 * (2 ** 5 + 17) ** 2)
```

Aufgabe 4: Operationenhierarchie

Aufgabe

Geben Sie für die folgenden Ausdrücke jeweils an, in welcher Reihenfolge die einzelnen Operationen durchgeführt werden.

- $a + b + c + d + e$
- $a + b * c - d / e$
- $a / (b + c) - d \% e$
- $a / (b * (c + (d - e)))$
- $a += b * c + d / e$
- $a = b + c + d * e$

- Zeile 1: $a + b + c + d + e$

```
a + b
(a + b) + c
((a + b) + c) + d
(((a + b) + c) + d) + e
```

- Zeile 2: $a + b * c - d / e$

```
b * c
d / e
a + (b * c)
(a + (b * c)) - (d / e)
```

- Zeile 3: $a / (b + c) - d \% e$

```
b + c
a / (b + c)
d \% e
(a / (b + c)) - (d \% e)
```

- Zeile 4: $a / (b * (c + (d - e)))$

```
d - e
c + (d - e)
b * (c + (d - e))
a / (b * (c + (d - e)))
```

- Zeile 5: $a += b * c + d / e$

```
b * c
d / e
(b * c) + (d / e)
a + ((b * c) + (d / e))
a = (a + ((b * c) + (d / e)))
```

- Zeile 6: $a = b + c + d * e$

```
d * e
b + c
(b + c) + (d * e)
a = ((b + c) + (d * e))
```

Aufgabe 5: Eingebaute Funktionen

Aufgabe

Die Funktion `print` ist ein Beispiel einer sogenannt *eingebauten Funktion*. Geben Sie ein weiteres Beispiel einer eingebauten Funktionen an und beschreiben Sie kurz, was diese tut. Machen Sie sich hierzu mit der Dokumentation von Python vertraut:

<https://docs.python.org/3/library/functions.html>

Die `range()` Funktion in Python

Die `range()` Funktion in Python erzeugt eine Sequenz von Zahlen. Diese Sequenz kann zum Beispiel in `for`-Schleifen verwendet werden. Die Funktion kann bis zu drei Parameter annehmen, die das Verhalten der erzeugten Sequenz steuern.

Grundlegende Verwendung

Mit `range(5)` erzeugen Sie eine Sequenz von Zahlen von 0 bis 4:

```
for x in range(5):  
    print(x)
```

Ausgabe:

```
0  
1  
2  
3  
4
```

Anfang und Ende festlegen

Durch die Verwendung von zwei Parametern können Sie einen Start- und einen Endwert für die Sequenz festlegen:

```
for x in range(5, 10):  
    print(x)
```

Ausgabe:

```
5  
6  
7  
8  
9
```

Schrittgröße angeben

Ein dritter Parameter legt die Schrittgröße zwischen den Zahlen in der Sequenz fest:

```
for x in range(10, 21, 2):  
    print(x)
```

Ausgabe:

```
10  
12  
14  
16  
18  
20
```