

# Kontrollstrukturen

Python-Übungen – Tim Poerschke (timkodierte.de)

---

## Aufgabe 1: Ping Pong! (if-Verzweigungen)

Schreiben Sie ein Programm, dass auf eine Nutzereingabe reagiert. Das Programm soll „Pong“ ausgeben, wenn der Nutzer „Ping“ eingegeben hat. Bei allen anderen Eingaben soll „Falsche Eingabe“ angezeigt werden.

*Hinweis: Die Nutzereingabe können Sie wie folgt abfragen.*

```
1 user_input = input("Eingabe bitte: ")
```

Die Lösung finden Sie in der Datei `1_PingPong.py`.

## Aufgabe 2: Eingaben vergleichen (if-Verzweigungen)

Fordern Sie zwei Nutzereingaben an und konvertieren Sie diese jeweils in einen Integer. Überprüfen Sie, welche der Eingaben die größere Zahl war. Mögliche Ausgaben des Programms:

```
Eingabe 1 war größer
Eingabe 2 war größer
Die Eingaben waren identisch
```

Die Lösung finden Sie in der Datei `2_EingabenVergleichen.py`.

## Aufgabe 3: Was ist die Ausgabe? (if-Verzweigungen)

```
1 a = True
  b = False
  numero = 0

5 if a == b:
    print("Aha! True ist gleich False!")
  elif not a or b:
    print("nicht a oder b")
  elif (a and not b) and not numero:
10    print("Komplizierte Bedingung hat True ergeben")
  else:
    print("Keine Bedingung hat True ergeben...")
```

Was wird dieses Programm ausgeben? Versuchen Sie die Ausgabe herauszufinden, ohne das Skript auszuführen. Um Ihr Ergebnis zu prüfen, können Sie das Programm `3_AusgabeIf.py` ausführen.

## Aufgabe 4: Summieren wie die Sumerer (Schleifen)

Implementieren Sie ein Python-Programm, welches die folgende Summenformel berechnet:

$$\sum_{x=3}^{13} (x-2)^2$$

Passen Sie das Programm anschließend so an, dass der Nutzer die Unter- und Obergrenze (jeweils inklusiv) angeben kann. Dazu können Sie die `input`-Funktion oder die Kommandozeilenparameter verwenden.

*Optional:* Wandeln Sie die `while`-Schleife in eine äquivalente `for`-Schleife um.

Die Lösung finden Sie in der Datei `4_Summieren.py`.

## Aufgabe 5: Pattern Printing #1 (Schleifen)

Entwerfen Sie ein Python-Programm, welches das nachfolgende Muster ausgibt:

```
1 22 333 4444 5555 ...
```

Die Lösung finden Sie in der Datei `5_PatternPrinting1.py`.

## Aufgabe 6: Pattern Printing #2 (Schleifen)

Entwerfen Sie ein Python-Programm, welches das nachfolgende Muster ausgibt:

```
1          *
      *  *
    *    *
  *      *
*        *
5 *          *
```

Die Lösung finden Sie in der Datei `6_PatternPrinting2.py`.