

Alle Programme, die wir bisher gesehen oder selbst geschrieben haben, waren relativ simpel gestrickt: Wir haben mithilfe von Variablen irgendwelche Formeln zur Berechnung neuer Werte angewendet. In der Regel sind Programme aber deutlich komplexer und müssen auf die Daten reagieren.

💡 Verzweigungen in Python

Der Ablauf eines Programms kann in Python mit den Keywords `if`, `elif` und `else` beeinflusst werden. Abhängig von einer *Bedingung* (englisch: condition) werden so bestimmte Teile des Codes ausgeführt oder auch nicht.

- Der Code innerhalb des `if`-Blocks wird nur ausgeführt, wenn die zugehörige Bedingung zutrifft.
- Der Code im `elif`-Block wird ausgeführt, wenn die Bedingung beim `if` nicht zutrifft, aber die Bedingung beim `elif` schon.
- Der Code im `else`-Block wird immer dann ausgeführt, wenn keine der vorherigen Bedingungen zugetroffen ist. Daher muss hier auch keine zusätzliche Bedingung angegeben werden.

Nach dem jeweiligen Keyword folgt die Bedingung (außer beim `else`) und anschließend ein Doppelpunkt. Innerhalb eines `if`-, `elif`- oder `else`-Blocks muss der Code eingerückt werden. Sowohl der `elif`-Block als auch der `else`-Block sind optional und können komplett weggelassen werden, wenn sie nicht benötigt werden.

Umgesetzt sähe das ganze beispielsweise folgendermaßen aus:

```
1 alter = 0.5
2 if (alter > 18):
3     # Wird ausgeführt, wenn alter > 18
4     print("Nutzer ist volljährig!")
5 elif (alter < 1):
6     # Wird ausgeführt, wenn alter <= 18 und alter < 1
7     print("Nutzer ist ein Baby!")
8 else:
9     # Wird ausgeführt, wenn alter <= 18 und alter >= 1
10    print("Nutzer ist minderjährig!")
```

Dieser Code gibt beispielsweise "Nutzer ist ein Baby!" aus, denn die Bedingung in Zeile 2 trifft nicht zu, die Bedingung in Zeile 5 allerdings schon.

Aufgabe 1: Teilbarkeit durch 5

Schreiben Sie ein Programm, das vom Nutzer eine Zahl einliest. Überprüfen Sie, ob es sich um eine durch 5 teilbare Zahl handelt und geben Sie abhängig davon einen entsprechenden Text aus.

Logische Operatoren

Wenn man das Ergebnis einer solchen Bedingung mal testweise in einer Variablen speichert, sieht man, dass es sich dabei um einen Boole'schen Wert handelt. `bedingung1 = 17 < 25` wäre beispielsweise `True`, während `bedingung2 = 25 < 17` den Wert `False` hätte.

Man kann Bedingungen auch noch komplizierter gestalten, indem man sie mit Klammern und den Keywords `and`, `or` und `not` verbindet. Beispielsweise wäre `bedingung3 = (17 < 25) and (25 < 50)` zutreffend, da sowohl die erste als auch die zweite Klammer beide `True` sind.

So sähe beispielsweise eine Verzweigung mit etwas komplizierteren Bedingungen aus:

```
1 alter = 13
2 groesse = 155
3 if (alter >= 14) and (groesse >= 150):
4     print("Du darfst Achterbahn fahren")
5 elif not (alter >= 14) and (groesse >= 150):
6     print("Du bist zwar groß genug, aber leider zu jung")
7 elif (alter >= 14) and not (groesse >= 150):
8     print("Du bist zwar alt genug, aber leider zu klein")
9 else:
10    print("Du bist leider zu jung und zu klein")
```

Die obige Person ist zwar groß genug, aber leider zu jung zum Achterbahn fahren.

Aufgabe 2: Schaltjahrrechner

Schreiben Sie ein Programm, das prüft, ob ein vom Nutzer eingegebenes Jahr ein Schaltjahr ist:

- Ein Jahr ist Schaltjahr, wenn es durch 4 teilbar ist,
- aber nicht durch 100,
- außer es ist auch durch 400 teilbar.