

**Variablen**

In der Programmierung werden Variablen wie in der Mathematik als Platzhalter für bestimmte Werte verwendet. Beispielsweise setzt man mit `x = 5` den Wert der Variablen `x` auf 5. Der Wert der Variablen kann dann später im Code wiederverwendet werden: Beispielsweise gibt `print(x)` nun den Wert der Variablen `x` aus, also 5.

**Aufgabe 1: Variablen in Python**

- Schreiben Sie ein Programm, in dem Sie vier Variablen mit Ihrem Vornamen, Nachnamen, Alter und Körpergröße in Metern anlegen. Geben Sie anschließend einen Text aus, der Sie – unter Verwendung der eben angelegten Variablen – vorstellt.
- Schreiben Sie ein Programm, in dem zwei (Zahl-)Werte in Variablen gespeichert werden. Geben Sie die Namen der Variablen mit den zugehörigen Werten anschließend aus. Berechnen Sie zusätzlich die Summe, Differenz, Produkt und Quotient der beiden Variablen und geben Sie auch diese aus.
- Schreiben Sie ein Programm, das eine Variable für die Kantenlänge eines Würfels anlegt und anschließend die Oberfläche, das Volumen und die Länge der Raumdiagonale berechnet.

**Datentypen**

Da Computer nur mit Zahlen umgehen können, müssen wir als Programmierer dem Programm mitteilen, um welche Art von Daten es sich handelt. Die Art der Daten, die sich hinter einer Variable verbergen, nennt man auch *Datentyp*. Die am häufigsten verwendeten Datentypen, die Python (ebenso auch die meisten anderen Programmiersprachen) unterstützt, sind:

- `int`: steht für Integer, also Ganzzahlen wie 0, 3, -15 usw.
- `float`: Gleitkommazahlen, also z.B. 3.14, 15.2, -0.0014
- `bool`: Bool'sche bzw. Wahrheitswerte, die entweder `True` oder `False` sind
- `str`: Strings, das sind Zeichenketten wie "Hallo", "Informatik", "25 + 4 = 29".

Python nimmt uns hier viel Arbeit ab: Wir können einer Variable einfach einen Wert zuweisen und der Datentyp wird automatisch passend gesetzt. In anderen Programmiersprachen wie Java oder C++ muss man immer angeben, welchen Datentyp eine Variable haben soll.

Aufgabe 2: Arithmetische Operationen bei verschiedenen Datentypen

Schreiben Sie ein Programm, jeweils eine Variable der oben genannten Datentypen anlegt.

- a) Testen Sie ausgiebig, was passiert, wenn Sie zwei Variablen mit verschiedenen Datentypen mit einer arithmetischen Operation (wie z.B. +) verknüpfen.
- b) Speichern Sie die jeweiligen Ergebnisse, falls möglich, in einer neuen Variablen ab.
- c) Schreiben Sie zu jeder der neuen Variablen einen Kommentar, in dem Sie erklären, wie ihr Wert zustande kommt.
- d) Geben Sie die Werte und Datentyp der erstellten Variablen aus. Sie können den Datentyp einer Variablen mit `type()` abfragen.

Aufgabe 3: Typecasts

Mit den Python-Funktionen `int()`, `str()`, `float()` und `bool()` können Datentypen ineinander umgewandelt werden: Beispielsweise ist `float(23)` vom Typ `float` (und kein Integer mehr). Diese Konvertierung nennt man *Typecast*.

- a) Testen Sie die Typecasts zwischen den verschiedenen Datentypen in einem Programm. Welche Typecasts sind erfolgreich, welche schlagen fehl? Warum?

- b) In welchen Fällen könnte die Verwendung eines Typecasts nützlich oder sogar notwendig sein?

- c) Was ist der Unterschied zwischen dem *String* "23" und dem *Integer* 23?



Aufgabe 4: Verschiedene Umrechnungen

- a) Recherchieren Sie die Formel zur Umwandlung einer Temperatur von Fahrenheit in Celsius. Setzen Sie die Umrechnung in einem Python-Programm um und lassen Sie sich den umgerechneten Wert ausgeben.
- b) Legen Sie eine Variable an, die eine Anzahl an Sekunden enthält. Berechnen Sie, wie viele Minuten und Stunden diese Anzahl entspricht, speichern Sie die Werte in neue Variablen und geben Sie sie aus.
- c) Legen Sie die für ein Quadrat, ein Dreieck und einen Kreis benötigten Variablen an. Berechnen Sie jeweils Flächeninhalt und Umfang und geben Sie die Werte aus.