



### Funktionen

Funktionen werden in der Programmierung genutzt, um Code wiederverwendbar zu machen und in sinnvolle Einheiten zu strukturieren. In Python werden sie mit dem Keyword `def` definiert, gefolgt vom Namen der Funktion und runden Klammern.

Innerhalb der runden Klammern können kommagetrennt sogenannte **Parameter** an die Funktion übergeben werden.

Das errechnete Ergebnis kann mit dem Keyword `return` zurückgegeben werden.

Die folgende Funktion berechnet beispielsweise das Quadrat einer Zahl und gibt den errechneten Wert zurück:

```
1 def quadriere(n):  
2     return n*n  
3 vier = quadriere(2) #Der berechnete Wert wird in eine Variable  
    gespeichert
```



### Aufgabe 1: Volumenrechner

Schreiben Sie ein Programm, das das Volumen eines Quaders berechnet und ausgibt.

Nutzen Sie eine Funktion, die die Eingabe des Nutzers einliest und anschließend verifiziert, dass die eingegebenen Werte für die Kantenlängen a, b und c gültig sind:

- Es muss sich um eine Zahl handeln (Tipp: `variable.isnumeric()`)
- Die Zahl muss größer als 0 sein
- Bei ungültiger Eingabe soll das Programm eine Fehlermeldung ausgeben und beendet werden (mit `exit()`)

Lesen Sie die Zahlen dann mithilfe der Funktion ein und berechnen Sie das Volumen.



### Standardwerte für Parameter

Man kann die Werte der übergebenen Parameter auch direkt auf einen Standardwert festlegen: `def potenziere(basis, exponent=2)` legt beispielsweise den Exponenten standardmäßig auf 2 fest.

Die Funktion kann nun mit `potenziere(10)` aufgerufen werden (was 100 ergeben würde), oder weiterhin mit `potenziere(10, 3)` (was 1000 wäre).



### Aufgabe 2: Listenelemente filtern

Schreiben Sie eine Funktion `elements_bigger_than(l, n)`, die eine Liste von Zahlen `l` und eine Zahl `n` als Parameter annimmt.

Die Funktion soll eine Liste zurückgeben, welche alle Zahlen der Liste `l` beinhaltet, welche größer als `n` sind.

Rufen Sie die Funktion anschließend mit einer Beispielliste auf und geben Sie den Rückgabewert aus.



### Aufgabe 3: Pythagoräische Tripel

Drei Zahlen, für die der Satz des Pythagoras gilt, nennt man *pythagoräisches Tripel*. Ein Beispiel für ein solches Tripel sind die Zahlen 3, 4 und 5, denn  $3^2 + 4^2 = 5^2$ .

Schreiben Sie eine Funktion `is_pythagorean(a, b, c)`, welche für drei Zahlen überprüft, ob es sich um ein pythagoräisches Tripel handelt.