

Wenn nichts anderes steht, ist die Ausgabe des Python-Programmfragments anzugeben.

Aufgabe 1

```
def f(x):  
    return 3*x + 10  
  
print(f(7))
```

Aufgabe 2

```
def volumen(a, b, c):  
    return a * b * c  
  
print(volumen(7, 2, 5))
```

Aufgabe 3

```
def minSecToSec(m, s):  
    return 60*m + s  
  
print(minSecToSec(2, 45))
```

Aufgabe 4

```
def secToMinSec(sec):  
    return [sec // 60, sec % 60]  
  
print(secToMinSec(200))
```

Aufgabe 5

```
def kreisflaeche(radius):  
    pi = 3.14  
    return pi * radius**2  
  
print(kreisflaeche(10))
```

Aufgabe 6

```
def nullstelle(a, b):  
    if a == 0:  
        return 'keine Nullstelle'  
    else:  
        return -b/a  
  
print(nullstelle(2, 3))
```

Aufgabe 7

```
def f(x):  
    return 73  
  
print(f(7))
```

Aufgabe 8

```
def f():  
    return 2  
  
print(f())
```

Aufgabe 9

```
def sign(x):  
    if x < 0:  
        return -1  
    elif x > 0:  
        return 1  
    else:  
        return 0  
  
print(sign(-7))
```

Aufgabe 10

```
def f(x):  
    5*x + 3  
  
print(f(2))
```

Aufgabe 11

```
def f(x):  
    return 3*x - 1  
  
print(f(f(f(1))))
```

Aufgabe 12

```
def f(x):  
    return 3*x + 1  
  
def g(x):  
    return 2*x  
  
print(f(g(1)))
```

Aufgabe 13

```
def f(a, b):  
    return 2*a + b  
  
print(f(b=4, a=5))
```

Aufgabe 14

```
def quadratwurzel(x):  
    if x < 0:  
        return 'geht nicht'  
    else:  
        return x**0.5  
  
print(quadratwurzel(-4))
```

Aufgabe 15

```
def diskriminante(a, b, c):  
    return b*b - 4*a*c  
  
print(diskriminante(5, 10, 1))
```

Aufgabe 16

```
def determinante(M):  
    return M[0][0]*M[1][1] - M[0][1]*M[1][0]  
  
print(determinante([[2, 1], [3, 10]]))
```

Aufgabe 17

Welchen Wert gibt das folgende Programm aus und warum?

```
def change(x):  
    a = x  
  
a = 1  
change(2)  
print(a)
```

Aufgabe 18

```
def f(n):
    if n == 0:
        return 1
    else:
        return f(n-1)+n

print(f(4))
```

Aufgabe 19

Erstelle ein Modul mit dem Namen `rechteck-umfang.py` und schreibe dort eine Python-Funktion `rechteckUmfang(a,b)`, die aus den Seitelängen `a` und `b` den Umfang des Rechtecks berechnet und als Wert zurückgibt. Teste die Funktion mit den folgenden Werten:

a	b	umfang
3	4	14
15	15	60
0	6	12
3.2	5.6	17.6

Aufgabe 20

Schreibe ein Modul mit dem Namen `imperial.py`, das die folgenden Umrechnungsfunktionen vom angloamerikanischen ins metrische Masssystem enthält:

- `inchToCm(x)`: Rechnet x *inch* (Zoll) in Zentimeter um und liefert das Resultat als Wert zurück. Verwende die Beziehung $1 \text{ inch} = 2.54 \text{ cm}$.
- `poundToKg(x)`: Rechnet x *pound* (Pfund) in Kilogramm um und liefert das Resultat als Wert zurück. Verwende die Beziehung $1 \text{ pound} = 0.45359237 \text{ kg}$.
- `momentToSec(x)`: Rechnet x *moment* in Sekunden um und liefert das Resultat als Wert zurück. Verwende die Beziehung $1 \text{ moment} = 1.5 \text{ s}$.

Aufgabe 21

Schreibe ein Modul mit dem Namen `kreis.py`, das die folgenden Funktionen definiert.

- `umfang(r)`: Berechnet aus dem Radius r eines Kreises seinen Umfang u und gibt ihn zurück.
- `inhalt(r)`: Berechnet aus dem Radius r eines Kreises seinen Flächeninhalt A und gibt ihn zurück.

Hinweis: Weise der Konstanten `PI` den Wert `3.14159` zu.

Aufgabe 22

Erstelle ein Modul mit dem Namen `coords.py`, das die folgenden Funktionen für Listen der Form `[x,y]` definiert:

- `getX(P)` gibt die x -Koordinate von P zurück.
Beispiel: `print(getX([-2,7]))` \Rightarrow `-2`
- `getY(P)` gibt die y -Koordinate von P zurück.
Beispiel: `print(getY([-2,7]))` \Rightarrow `7`
- `add(P,Q)` bestimmt die Summe der jeweiligen Koordinaten und gibt sie wieder in Form einer Liste mit zwei Elementen als Wert zurück.
Beispiel: `print(add([1,2],[3,4]))` \Rightarrow `[4,6]`
- `sub(P,Q)` bestimmt die Differenz der jeweiligen Koordinaten und gibt sie wieder in Form einer Liste mit zwei Elementen als Wert zurück.
Beispiel: `print(sub([8,1],[3,-2]))` \Rightarrow `[5,3]`

Teste die Funktionen mit den Punkten $A=[3,7]$ und $B=[-1,1]$.

Aufgabe 23

Schreibe ein Modul mit dem Namen `quadglg.py`, das die Funktion `solve(a, b, c)` definiert, um die Menge der reellen Lösungen der quadratischen Gleichung $ax^2+bx+c=0$ als Wert zurückzugeben. *Hinweise:*

- Berechne zuerst die Diskriminante $D = b^2 - 4ac$
- Falls $D < 0$, gibt es keine reellen Lösungen.
- Falls $D = 0$, gibt es nur die Lösung $x = \frac{-b}{2a}$.
- Falls $D > 0$, gibt es die Lösungen $x_1 = \frac{-b + \sqrt{D}}{2a}$ und $x_2 = \frac{-b - \sqrt{D}}{2a}$
(Die Quadratwurzel \sqrt{D} kann mit der Anweisung `D**0.5` berechnet werden.)
- Gib die jeweilige Lösungsmenge als (möglicherweise leere) Liste `[...]` zurück.

Teste dein Programm mit folgenden Gleichungen:

Gleichung	Ausgabe
$2x^2 + 7x - 4 = 0$	<code>[0.5, -4]</code>
$x^2 + 3x + 7 = 0$	<code>[]</code>
$x^2 + 6x + 9 = 0$	<code>[-3]</code>
$16x^2 - 9 = 0$	<code>[0.75, -0.75]</code>

Aufgabe 24

Schreibe ein Python-Modul mit dem Namen `ggt.py`, das eine gleichnamige Funktion `ggT(a,b)` definiert, die den grössten gemeinsamen Teiler von zwei nichtnegativen ganzen Zahlen `a` und `b` berechnet und als Wert zurückgibt.

Bemerkung: Anstatt der aus dem Mathematikunterricht bekannten ggT-Berechnung mittels Primfaktorzerlegung verwendet man den viel effizienteren *Algorithmus von Euklid*, der wie folgt funktioniert:

- (a) Ist $b = 0$, dann ist der grösste gemeinsame Teiler in jedem Fall a , denn a teilt sowohl a als auch 0 und ist somit der $\text{ggT}(a, 0)$.
- (b) Für $b \neq 0$ berechnen wir fortlaufend den Divisionsrest $r = a \bmod b$ mit $0 \leq r < b$ und ersetzen danach a durch b und b durch r . Sobald $b = 0$ gilt, beenden wir die Berechnungen und geben a als Wert zurück. Hier ein Beispiel:

$$\begin{aligned}\text{ggT}(21, 15) &= \text{ggT}(15, 21 \bmod 15) = \text{ggT}(15, 6) \\ &= \text{ggT}(6, 15 \bmod 6) = \text{ggT}(6, 3) \\ &= \text{ggT}(3, 6 \bmod 3) = \text{ggT}(3, 0) \\ &= 3 \quad \text{wie in (a)}\end{aligned}$$

Teste die Funktion mit folgenden Werten:

Ausdruck	Wert
<code>print(ggT(5168,2413))</code>	19
<code>print(ggT(71,0))</code>	71
<code>print(ggT(0,222))</code>	222