Aufgabe 1

Beschreibe die Programmiersprache Python. Wer hat sie entwickelt? Was ist ihr Einsatzgebiet? Was worin bestehen die Unterschiede zu anderen Programmiersprachen?

Aufgabe 2

```
print(1 + 2 * 3 - 4)
```

Aufgabe 3

```
print(17 // 3)
```

Aufgabe 4

```
print(17 % 3)
```

Aufgabe 5

```
print(2**3)
```

Aufgabe 6

```
print(4**0.5)
```

Aufgabe 7

```
print(-3**2)
```

Aufgabe 8

Welcher Datentyp hat das Resultat des Ausdrucks in Python? (ohne Berechnung)

```
print(type(4 * 9))
print(type(5.0 + 3))
print(type(8 / 2))
print(type(8 // 3))
print(type(7.0 // 4))
print(type(9**0.5))
```

Aufgabe 9

```
a = -1
b = 7
a = b
b = a
print(a)
print(b)
```

Aufgabe 10

```
a = 7
a += 3
a *= 9
print(a)
```

Aufgabe 11

```
a, b, c = 5, 3, 8
x, y, z = c, a, b
print(z)
```

Aufgabe 12

```
x = y = z = 3
print(x+y+z)
```

Aufgabe 13

```
print(5 != 5)
```

Aufgabe 14

```
print(True and False)
```

Aufgabe 15

```
print((5 < 7) or (3 > 8))
```

Aufgabe 16

```
print(not (2 < 1) and (7 > 6))
```

Aufgabe 17

```
print(True and True and True and True and False)
```

Aufgabe 18

```
x = 3.1
print(2.8 < x < 7.6)
```

Aufgabe 19

```
print(False or not True and False)
```

Aufgabe 20

```
print(not not not not not not not not True)
```

Aufgabe 21

```
x = 7
if x != 1:
    x = x + 3
x = x + 1
print(x)
```

Aufgabe 22

```
x = 7
if x != 5:
    x = x + 4
else:
    x = x + 3
x = x + 1
print(x)
```

Aufgabe 23

```
z = 9
if z < 2:
    z = z + 1
elif z < 4:
    z = z + 2
elif z < 6:
    z = z + 3
else:
    z = z + 4
print(z)
```

Aufgabe 24

```
b = 7
if b == 7:
    if b > 5:
        b = b + 1
    else:
        b = b + 2
else:
    if b >= 4:
        b = b + 3
    else:
        b = b + 4
print(b)
```

Aufgabe 25

```
for i in range(3, 5):
    print(2*i)
```

Aufgabe 26

```
for k in range(1, 10, 4):
    print(k)
```

Aufgabe 27

```
for j in range(10, 7, -1):
    print(j)
```

Aufgabe 28

```
for e in [3, -8, 7, -4, 0]:
    if e > 0:
        print(e)
```

Aufgabe 29

```
i = 0
while (i < 10):
    i = 2*i+1
    print(i)
```

Aufgabe 30

```
i = 0
while (i < 10):
    i = 2*i+1
print(i)
```

Aufgabe 31

```
i = 4
s = 0
while (s < 20):
    s = s + i
    i = i + 3
    print(s)
```

Aufgabe 32

Handelt es sich beim folgenden Python-Programmfragment um eine Endlosschleife?

```
i = 0
while (i < 10):
    print(i)
```

Aufgabe 33

Handelt es sich bei dem folgenden Python-Programmfragment um eine Endlosschleife?

```
i = 10
while i != 0:
    i = i - 2
```

Aufgabe 34

```
for e in [4, -7 , 6, 8]:
    if e > 5:
        break
    else:
        print(e)
```

Aufgabe 35

```
for i in range(1, 11):
    if i % 3 == 0:
        continue
    else:
        print(i)
```

Aufgabe 36

```
for i in range(2, 4):
    for j in range(3, 5):
        print(i+j)
```

Aufgabe 37

```
L = [3, -7, 1, 5, 8]
print(L[2])
```

Aufgabe 38

```
L = [[5, 3, 2], [1, 7], [9, 4, 8]]
print(L[2][0])
```

Aufgabe 39

```
L = [9, 3, 4, 2, 7, 5]
print(L[-2])
```

Aufgabe 40

```
L = [9, 3, 4, 2, 7, 5]
print(len(L))
```

Aufgabe 41

```
L = [[5, 3, 2], [1, 7], [9, 4, 8]]
print(len(L))
```

Aufgabe 42

```
L = [8,1,4,9]
L[2] = 7
print(L)
```

Aufgabe 43

```
L = [5, 2, 3]
L.append(7)
print(L)
```

Aufgabe 44

```
L = [5, 9, 8]
L.insert(1, 2)
print(L)
```

Aufgabe 45

```
L = [7, 9, 5, 8, 2]
L.pop()
print(L)
```

Aufgabe 46

```
L = [7, 9, 5, 8, 2]
x = L.pop()
print(x)
```

Aufgabe 47

```
L = [7, 9, 5, 8, 2]
x = L.pop(2)
print(x)
```

Aufgabe 48

```
A = [9, 3, 2]
B = [4, 1]
print(A + B)
```

Aufgabe 49

```
print(3 * [7,2])
```

Aufgabe 50

```
L = [7, 9, 5, 8, 2, 1, 4, 3]
print(L[2:5])
```

Aufgabe 51

```
L = [7, 9, 5, 8, 2, 1, 4, 3]
print(L[:3])
```

Aufgabe 52

```
L = [7, 9, 5, 8, 2, 1, 4, 3]
print(L[6:])
```

Aufgabe 53

```
A = [6, 7, 2]
B = A
B[1] = 9
print(A)
```

Aufgabe 54

```
A = [6, 7, 2]
B = A[:]
B[1] = 9
print(A)
```

Aufgabe 55

```
L = [6, 1, 2, 7, 9]
L.reverse()
print(L)
```

Aufgabe 56

```
L = [6, 1, 2, 7, 9]
L.sort()
print(L)
```

Aufgabe 57

```
T = (9, 3, 4, 2, 7, 5)
print(T[3])
```

Aufgabe 58

```
L = (9, 3, 4, 2, 7, 5)
L[3] = 8
print(L)
```

Aufgabe 59

```
L = (9, 3, 4, 2, 7, 5)
print(L[1:4])
```

Aufgabe 60

```
L = [a for a in range(2, 5)]
print(L)
```

Aufgabe 61

```
A = [1, 2, 3]
B = [2*x for x in A]
print(B)
```

Aufgabe 62

```
Z = [[0 for i in range(2)] for j in range(3)]
print(Z)
```

Aufgabe 63

```
L = [0 if i % 2 == 0 else 1 for i in range(7)]
print(L)
```

Aufgabe 64

```
def f(x):
    return 2*x + 4

print(f(9))
```

Aufgabe 65

```
def f(x):
    return 19

print(f(3))
```

Aufgabe 66

```
def f(x):
    x + 1

print(f(6))
```

Aufgabe 67

```
def f(a, b):
    return 2*a + b

print(f(5, 7))
```

Aufgabe 68

```
def f(a, b):
    return 2*a + b

print(f(b=5, a=7))
```

Aufgabe 69

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

print(f(f(f(1))))
```

Aufgabe 70

```
def f(x):
    return 2*x - 1
def g(x):
    return x*x

print(f(5) + g(3))
```

Aufgabe 71

```
def f(x):
    a = 4
    print(a+x)

a = 3
f(5)
print(a)
```

Aufgabe 72

```
def f(x=2, y=1):
    return 2*x + 3*y

print(f(3))
```

Aufgabe 73

```
def f(L, x):
    L.append(x)

L = [1, 2, 3]
f(L, 4)
print(L)
```

Aufgabe 74

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

print(f(4))
```

Aufgabe 75

```
def f(n):
    if n < 2:
        return n
    else:
        return 2*f(n-2) + 1

print(f(5))
```

Aufgabe 76

```
wort = 'HALLO'
print(wort[1])
```

Aufgabe 77

```
wort = "HALLO"
print(wort[-1])
```

Aufgabe 78

```
wort = 'hundert'
print(wort[1:4])
```

Aufgabe 79

```
satz = "Was soll das?"
print(len(satz))
```

Aufgabe 80

```
wort = 'Feeler'
wort[2] = 'h'
print(wort)
```

Aufgabe 81

```
a = 'abc'
b = 'xyz'
print(a + b)
```

Aufgabe 82

```
print('a' + 2 * 'n' + 'a')
```

Aufgabe 83

```
satz = 'Das\nist\nso.'
print(satz)
```

Aufgabe 84

```
satz = '''Woher
weisst du das?'''
print(satz)
```

Aufgabe 85

```
satz = '''Das \
ist vielleicht \
sinnlos!'''
print(satz)
```

Aufgabe 86

```
text = '15'
print(int(text)+5)
```

Aufgabe 87

```
text = '7'
print(int(text, 2))
```

Aufgabe 88

```
satz = "Das ist das Zeichen \\."
print(satz)
```

Aufgabe 89

```
text = "Sag \"Hallo\""
print(text)
```

Aufgabe 90

```
a = 'Das'
b = 'ist'
c = 'schlecht'
print(a, b, c, sep='.')
```

Aufgabe 91

```
wort = 'Ragusa'
print('usa' in wort)
```

Aufgabe 92

```
text = '{0} + {1} = {2}'.format(3, 4, 7)
print(text)
```

Aufgabe 93

```
text = '{1} Meter kosten {0} Fr.'.format(20, 30)
print(text)
```

Aufgabe 94

```
text = 'gut gemacht!'
print(text.capitalize())
```

Aufgabe 95

```
wort = 'abracadabra'
print(wort.count('ab'))
```

Aufgabe 96

```
wort = 'Mississippi'
print(wort.find('is'))
```

Aufgabe 97

```
liste = ['25', '2', '2013']
print('.'.join(liste))
```

Aufgabe 98

```
wort = "HAMMER"
wort.lower()
print(wort)
```

Aufgabe 99

```
wort = "Hammer"
wort = wort.replace('m', 'c', 1)
wort = wort.replace('m', 'k', 1)
print(wort)
```

Aufgabe 100

```
satz = 'Das ist gut.'
satz = satz.strip('.')
print(satz)
```

Aufgabe 101

```
satz = 'Das ist gut'
abc = satz.split(' ')
print(abc)
```

Aufgabe 102

```
wort = "ANANAS"
wert = wort.split('N')
print(wert)
```

Aufgabe 103

```
text = '7'
text = text.zfill(3)
print(text)
```

Aufgabe 104

```
a = 'mit'
b = list(a)
b.reverse()
c = ''.join(b)
print(c)
```

Aufgabe 105

```
D = {3: 6, -3: -9, 6: -6}
print(D[1+2])
```

Aufgabe 106

```
D = {'a':'d', 'e':'g', 'd':'e', 'g':'a'}
print(D[D[D['e']]])
```

Aufgabe 107

```
D = {'a': 3, 'e': 2, 'g': -3, 'f': 4}
D.pop('a')
print(len(D))
```

Aufgabe 108

```
D = {'b': -2, 'e': 8, 'd': 2, 'g': -6}
del D['b']
print(len(D))
```

Aufgabe 109

```
D = {'c': -6, 'd': 4, 'g': -9}
D.pop('b')
print(D)
```

Aufgabe 110

```
D = {'b': 6, 'g': 0, 'f': 8}
E = {'b': -2, 'g': 0, 'f': 8}
D.update(E)
print(D)
```

Aufgabe 111

```
D = {'e': -4, 'g': 7, 'f': 3}
for x in D.keys():
    print(D[x])
```

Aufgabe 112

```
D = {'a': 3, 'b': -2, 'e': -3}
for y in sorted(D.values()):
    print(y)
```

Aufgabe 113

```
D = {3:2, 2:0, 4:5}
for v, w in D.items():
    print(v-w)
```

Aufgabe 114

```
A = [3, 7, 9]
B = [8, 5, 2]
D = dict(zip(A, B))
for x, y in D.items():
    print(x+y)
```

Aufgabe 115

```
D = {5:4, 7:8, 3:2}
E = {y:x for x, y in D.items()}
print(sorted(E))
```

Aufgabe 116

Gegeben ist das folgendes Python-Modul:

```
class Rechteck:

    anzahl = 0

    def __init__(self, laenge, breite):
        self.a = laenge
        self.b = breite
        Rechteck.anzahl += 1

    def umfang(self):
        return 2*(self.a + self.b)

    def inhalt(self):
        return  self.a * self.b

    def add(self, other):
        return Rechteck(self.a+other.a, self.b+other.b)

    def get_anzahl():
        return Rechteck.anzahl

r1 = Rechteck(2,5)
r2 = Rechteck(4,3)
r3 = r1.add(r2)

print(r1.umfang())
print(r2.inhalt())
print(r3.b)
print(Rechteck.get_anzahl())
```

(a) Welche Ausgabe(n) macht das Programm?

(b) Erkläre die folgenden Begriffe anhand des obigen Quellcodes.

- Klasse
- Instanz (oder Objekt)
- Objekteigenschaft (oder Objektattribut)
- Objektmethode
- Klasseneigenschaft (oder Klassenattribut)
- Konstruktor

Aufgabe 117

Welche Ausgaben macht das folgende Python-Modul?

```
class Car:

    max_speed = 120

    def __init__(self):
        self.speed = 0

    def accelerate(self, delta_v):
        self.speed = min(self.speed+delta_v, Car.max_speed)

    def brake(self, delta_v):
        self.speed = max(self.speed-delta_v, 0)

    def get_speed(self):
        return self.speed

c1 = Car()
c2 = Car()

c1.accelerate(50)
c1.accelerate(30)
c1.brake(40)

c2.accelerate(40)
c2.brake(50)

print(c1.get_speed())
print(c2.get_speed())
```

Aufgabe 118

Gegeben ist das folgende Python-Modul.

```
class Image:

    def __init__(self, width, height):
        self.w = width
        self.h = height
        self.img = [[0 for i in range(width)] for j in range(height)]

    def set_pixel(self, x, y, color=1):
        self.img[y][x] = color

    def write(self, filename):
        fd = open(filename, mode='w')
        fd.write('P1\n')
        fd.write('{0} {1}\n'.format(self.w, self.h))
        for i in range(0, self.h):
            for j in range(self.w):
                fd.write('{0} '.format(self.img[i][j]))
        fd.close()

I = Image(3, 3)
I.set_pixel(0, 0)
I.set_pixel(1, 1)
I.set_pixel(2, 2)
I.write('test.pbm')
```

(a) Beschreibe, was dieses Modul macht.

(b) Gibt es Daten aus? Wenn ja, welche Daten und in welcher Form?

Aufgabe 119

Gegeben ist das folgende Python-Modul zum Rechnen mit Brüchen.

```
from math import gcd # greatest common divisor

class Bruch:

    def __init__(self, z, n):
        t = gcd(z, n)
        self.z = z // t
        self.n = n // t
        if self.n < 0:
            self.z = -self.z
            self.n = -self.n

    def __str__(self):
        if self.n == 1:
            return '{0.z}'.format(self)
        else:
            return '{0.z}/{0.n}'.format(self)

    def __add__(self, other):
        z = self.z * other.n + self.n + other.z
        n = self.n * other.n
        return Bruch(z, n)

a = Bruch(3,9)
b = Bruch(1,6)
c = a + b

print(a)
print(c)
```

Hinweise: Die Spezialmethode `__str__` überschreibt die `str()`-Methode und wird implizit bei der Ausgabe mit `print(...)` ausgeführt. Die Spezialmethode `__add__` überschreibt den Python-Operator „`+`“, wobei hier `self` für den linken und `other` für den rechten Operator steht.

(a) Beschreibe die einzelnen Teile des Moduls so genau wie möglich.

(b) Welche Ausgaben macht das Modul?

(c) Ergänze das Modul mit einer Methode zum Multiplizieren von Brüchen, indem du den Operator `__mul__` überschreibst.

Aufgabe 120

Schreibe eine Python-Funktion `list_max(L)`, die das grösste Element der Liste `L` zurückgibt.

Hinweis: Weise dem Maximum `m` den provisorischen Wert `float('-inf')`; d. h. $-\infty$ zu und überprüfe in einer Schleife über alle Listenelemente, ob sich das jeweils aktuelle Maximum durch ein noch grösseres Element ersetzen lässt. Falls ja, ersetze es.

Aufgabe 121

Schreibe eine Python-Funktion `list_find(item, L)`, die `True` zurückgibt, wenn `item` in der Liste `L` ist und `False` sonst.

Aufgabe 122

Schreibe eine Python-Funktion `mean(L)`, welche das arithmetische Mittel der Elemente in der Liste `L` berechnet und zurückgibt.

Welcher Spezialfall muss berücksichtigt werden und wie soll man ihn verarbeiten?

Aufgabe 123

Schreibe eine Python-Funktion `is_sorted(L)`, die `True` zurückgibt, wenn die Liste `L` aufsteigend sortiert ist; d. h. wenn mit Ausnahme des letzten Elements kein Element grösser als sein Nachfolger ist. Andernfalls soll die Funktion `False` zurückgeben.

Überlege auch, welcher Rückgabewert bei Listen der Länge 0 oder 1 sinnvoll ist.

Aufgabe 124

Schreibe eine Python-Funktion `random_list(n, a, b)`, die eine Liste mit `n` Elementen zurückgibt, wobei jedes Element `r` eine ganze Zufallszahl mit $\texttt{a} \leq \texttt{r} \leq \texttt{b}$ ist.

Hinweis: Importiere dazu die Funktion `randint` aus dem `random`-Modul, die mit dem Aufruf `randint(a,b)` gleichverteilte ganze Zufallszahlen `r` mit $\texttt{a} \leq \texttt{r} \leq \texttt{b}$ zurückgibt.

Aufgabe 125

Schreibe eine Python-Funktion `gcd(a, b)`, welche den grössten gemeinsamen Teiler (*greatest common divisor*) der beiden ganzen Zahlen `a` und `b` berechnet und zurückgibt.

Aufgabe 126

Schreibe eine Python-Funktion `zero_matrix(m, n)`, welche die Nullmatrix mit `m` Zeilen und `n` Spalten zurückgibt.

Beispiel: `zero_matrix(2, 3)` gibt als Wert `[[0, 0, 0], [0, 0, 0]]` zurück.

Aufgabe 127

Schreibe eine Python-Funktion `reverse_string(string)`, welche den String `string` in umgekehrter Zeichefolge zurückgibt.

Beispiel: `reverse_string('ABC')` $\Rightarrow$ `'CBA'`

Aufgabe 128

Schreibe eine Python-Funktion `factorial(n)`, welche die Fakultät von n berechnet:

$$n! = n \cdot (n-1) \cdot \ldots \cdot 2 \cdot 1$$

Es muss nicht geprüft werden, ob `n` eine gültige Eingabe ist.

Beachte, dass $0! = 1$ gilt. Vielleicht hilft die folgende Grafik um einzusehen, dass dies sinnvoll ist.

$$\begin{aligned} 3! &= 3 \cdot 2 \cdot 1 = 6 \\ 2! &= 2 \cdot 1 = 2 \\ 1! &= 1 = 1 \\ 0! &= 1 \end{aligned} \quad \begin{matrix} :3 \\ :2 \\ :1 \end{matrix}$$

Aufgabe 129

Schreibe eine Python-Funktion `add_matrices(A,B)`, welche die Summe der Matrizen `A` und `B` mit gleicher Dimension berechnet und zurückgibt.

Aufgabe 130

Schreibe eine Python-Funktion `count_letters(string)`, welche die Häufigkeit aller Zeichen in der Zeichenkette `string` bestimmt und als Python-Dictionary zurückgibt.

Aufgabe 131

Schreibe eine Python-Funktion `count_vowels(string)`, welche die Anzahl der Vokale (A,E,I,O,U) in der Zeichenkette `string` zählt und zurückgibt.

Beispiel: `count_vowels('Eremit')` $\Rightarrow$ 3

Aufgabe 132

Schreibe eine Python-Funktion `count_words(string)`, welche die Zeichenkette `string` mit der Methode `string.split()` in seine Wörter zerlegt und danach die Häufigkeit der Wörter als Python-Dictionary zurückgibt.

Aufgabe 133

Schreibe eine Python-Funktion `is_palindrome(string)`, die `True` zurückgibt, wenn `string` ein Palindrom ist; d. h. wenn `string` vorwärts und rückwärts gelesen das gleiche Wort ergibt.

Aufgabe 134

(a) Schreibe eine Python-Funktion `sum_of_n(n)`, welche die Summe der ersten natürlichen Zahlen von 1 bis `n` in $O(n)$ berechnet und zurückgibt.

(b) Schreibe eine Python-Funktion `sum_of_n_fast(n)`, welche die Summe der ersten natürlichen Zahlen von 1 bis bis `n` in $O(1)$ berechnet und zurückgibt.

Aufgabe 135

Schreibe eine Python-Funktion `transpose_matrix(A)`, welche die Transponierte $\mathtt{A}^\mathsf{T}$ der Matrix `A` zurückgibt.

Hinweis: Die Transponierte A^T einer Matrix A ist die Matrix, bei der die i-te Zeilen von A zur j-ten Spalte von A^T wird.

Beispiel: $A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \quad \Rightarrow \quad A^\mathsf{T} = \begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{pmatrix}$

Aufgabe 136

Schreibe eine Python-Funktion `write_integers(n, filename)`, welche die ganzen Zahlen $1, 2, \ldots, n$ zeilenweise in die Datei mit dem Namen `filename` schreibt.

Aufgabe 137

Schreibe eine Python-Funktion `add_floats_in_file(filename)`, welche die Summe der Zahlen in dieser Datei zurückgibt, wobei in jeder Zeile der Datei eine Gleitkommazahl steht.

Aufgabe 138

Schreibe eine Python-Funktion `fibonacchi(n)`, welche die Liste mit den ersten `n` Fibonacchi-Zahlen (1, 1, 2, 3, 5, 8, 13, ...) erzeugt und zurückgibt.