Python-Programmieraufgaben Lösungen+ Prüfungsvorbereitung

Aufgabe 1

```
def list_max(L):
    '''Gibt das Maximimum der Elemente von L zurück.'''
    m = float('-inf')
    for x in L:
        if x > m:
            m = x
    return m

LoL = [ [], [14], [3,1,-4, 19, 12], [3, 3, 3, 3, 3]]
for L in LoL:
    print(L, list_max(L))
```

Aufgabe 2

```
def list_find(item, L):
    '''Gibt True zurück, wenn "item" Element der Liste "L" ist.'''
    for x in L:
        if x == item:
            return True
    return False

# Testcode
print(list_find(5, [3, 5]))
print(list_find('x', ['ab', 'xz', 'pq']))
print(list_find(False, [True, True, True]))
```

Aufgabe 3

```
def mean(L): # pythonic
    '''Gibt arithmetisches Mittel der Elemente in L zurück'''
    return sum(L)/len(L)

def mean_classic(L):
    '''Gibt arithmetisches Mittel der Elemente in L zurück'''
    summe = 0
    for i in range(len(L)):
        summe = summe + L[i]
    return summe/len(L)

# Was fehlt: Vorher überprüfen, ob die Liste leer ist
# und dann "False" oder "None" zurückgeben.
```

Aufgabe 4

```python
def is_sorted(L):
    '''Gibt True zurück, wenn die Elemente in L aufsteigend sortiert sind'''
    n = len(L)
    if n < 2:
        return True
    for i in range(0, n-1):
        if L[i] > L[i+1]:
            return False
    return True

LoL = [[0], [31], [-4, 3, 5, 7, 12, 29], [2, 6, 9, 15, 11]]
for L in LoL:
    print(L, is_sorted(L))
```

Aufgabe 5

```python
from random import randint

def random_list(n, a, b): # pythonic
    '''Gibt eine Liste mit n zufälligen Zahlen a <= x <= b zurück'''
    return [randint(a,b) for in range(n)]

def random_list_classic(n, a, b):
    '''Gibt eine Liste mit n zufälligen Zahlen a <= x <= b zurück'''
    L = []
    for x in range(0, n):
        L.append(x)
    return L

print(random_list(5, 1, 9))
print(random_list_classic(1, 9))
```

Aufgabe 6

```python
def gcd(a, b):
    '''Gibt den grössten gemeinsamen Teiler von a und b zurück'''
    while  b != 0:
        a, b = b, a % b
    return a

print(gcd(24,15))
print(gcd(17, 0))
```

Aufgabe 7

```
def reverse_string(string): # pythonic
    '''Gibt die Zeichen von "string" in umgekehrter Reihenfolge zurück'''
    return string[::-1]

def reverse_string_classic(string):
    new_string = ''
    n = len(string)
    for i in range(0, n):
        new_string += string[n-i-1]
    return new_string

string = 'ABCDEFG'
print(reverse_string(string))
print(reverse_string_classic(string))
```

Aufgabe 8

```
def factorial(n):
    '''Gibt n! = n*(n-1)*...*2*1 zurück'''
    fact = 1
    for k in range(n):
        fact *= k
    return fact
```

Aufgabe 9

```
def count_letters(string):
    '''Gibt Dictionary mit Zeichenhäufigkeiten von 'string' zurück'''
    D = dict()
    for char in string:
        if char not in D:
            D[char] = 1
        else:
            D[char] += 1
    return D

print(count_letters('MISSISSIPPI'))
```

Aufgabe 10

```
def reverse_string(string): # pythonic
    '''Gibt die Zeichen von "string" in umgekehrter Reihenfolge zurück'''
    return string[::-1]

def reverse_string_classic(string):
    new_string = ''
    n = len(string)
    for i in range(0, n):
        new_string += string[n-i-1]
    return new_string

string = 'ABCDEFG'
print(reverse_string(string))
print(reverse_string_classic(string))
```

Aufgabe 11

```
def count_words(string):
    '''Gibt Dictionary mit der Häufigkeit aller Wörter in 'string' zurück'''
    string = string.lower()
    array = string.split() # trennt an Leerzeichen und Zeilenschaltungen
    D = dict()
    for word in array:
        if word not in D:
            D[word] = 1
        else:
            D[word] += 1
    return D

print(count_words('Das ist wie es ist'))
```

Aufgabe 12

```
def sum_of_n(n):
    '''Gibt 1 + 2 + ... + n in O(n) zurück'''
    return sum([i for i in range(1,n+1)])

def sum_of_n_fast(n):
    '''Gibt 1 + 2 + ... + n in O(1) zurück'''
    return n*(n+1)//2
```

Aufgabe 13

```
def is_palindrome(string):
    '''Gibt True zurück, wenn string ein Palindrom ist und False sonst.'''
    return string == string[::-1] # pythonic!

def is_palindrome_classic(string):
    '''Gibt True zurück, wenn string ein Palindrom ist und False sonst.'''
    n = len(string)
    for i in range(n):
        if string[i] != string[n-i-1]:
            return False
    return True

s1 = 'ANNA'
s2 = 'BEN'
print(s1, is_palindrome(s1)) # => True
print(s2, is_palindrome(s2)) # => False

print(s1, is_palindrome_classic(s1)) # => True
print(s2, is_palindrome_classic(s2)) # => False
```

Aufgabe 14

```
def zero_matrix(m, n):
    '''Gibt die Nullmatrix mit m Zeilen und n Spalten zurück'''
    return [[0 for i in range(n)] for j in range(m)]
```

Aufgabe 15

```
def matrix_add(A, B):
    '''Gibt die Summe der Matrizen A und B zurück'''
    m, n = len(A), len(A[0])
    return [[A[i][j]+B[i][j] for j in range(n)] for i in range(m)]

A = [[1,2], [3,4], [5,6]]
B = [[3,1], [0,2], [4,1]]
C = matrix_sum(A, B)
print(C)
```

Aufgabe 16

```
def transpose_matrix(A):
    '''Gibt die Transponierte A^T von A zurück'''
    m, n = len(A), len(A[0])
    return [[A[i][j] for i in range(m)] for j in range(n)]

A = [[1,2], [3,4], [5,6]]
B = transpose_matrix(A)
print(B)
```

Aufgabe 17

```
def write_integers(n, filename):
    '''Schreibt die ersten n natürlichen Zahlen in eine Datei'''
    fd = open(filename, mode='w')
    for i in range(1, n+1):
        fd.write(f'{i}\n')
    fd.close()

write_integers(50, 'integers.txt')
```

Aufgabe 18

```
def add_numbers_of_file(filename):
    '''Addiert alle Gleitkommazahlen in der Datei "filename".'''
    fd = open(filename, mode='r')
    s = 0
    for line in fd:
        s += float(line)
    fd.close()
    return s

s = add_numbers_of_file('messwerte.txt')
print(s)
```

Aufgabe 19

```
def fibonacci(n):
    '''Gibt die Liste der ersten n Fibonacci-Zahlen zurück'''
    F = []
    if n > 0:
        F.append(1)
    if n > 1:
        F.append(1)
    for i in range(2,n+1):
        F.append(F[-2] + F[-1])
    return F


for k in range(1, 7):
    print(k, fibonacci(k))
```