

1. (3 Punkte) Erstelle eine leere Liste `a`.
Füge in `a` nacheinander die Zahlen ein: 6, 2, 17
Gib die Liste aus.
Gib die Länge von `a` aus.

2. (4 Punkte) In dieser Aufgabe soll, ausgehend von einer Liste `a`, mittels slicing oder indexing der angegebene Wert erreicht werden. Beispiel: `a = ['abc', [4,7]]`. Der Wert `'ab'` kann mittels `a[0][:2]` erreicht werden, der Wert 7 kann mittels `a[1][1]` erreicht werden.

`a = [42, 'abc', ['de', 102], ['uvw']]`

Mit welchem Ausdruck kann man, von `a` ausgehend, folgende Werte erreichen:

- a. 42 b. 102 c. 'c' d. 'w'

3. (5 Punkte) Gegeben ist eine Liste `a = [4,6,10,2,7,18]`. Erstelle jeweils mittels slicing oder indexing einen Ausdruck, der sich zu folgenden Werten auswertet:

- a. `[4, 6]`
b. `[18]`
c. `7`
d. `[6, 10, 2]`
e. `[7, 18]`

4. (3 Punkte) Was erscheint auf der Konsole?

```
a = [24,6,7,20,5,11]
b = []
while a:
    x = a.pop()
    if x % 2 == 1:
        b.append(2*x)
print(a)
print(b)
```

5. (2 Punkte) Die Liste `b` enthält einige Zahlen. Gehe mit der Laufvariablen `z` durch die Elemente von `b` und gib das doppelte jedes Elements von `b` aus.

6. (2 Punkte) Die Liste `b` enthält einige Zahlen. Gehe mit der Laufvariablen `k` durch die Indizes von `b` und gib das dreifache jedes Elements von `b` aus.

7. (1 Punkt) Gegeben ist die Liste `a = [12, 5, 9]`. Weise mit unpacking den Variablen `x`, `y`, `z` die Werte 5, 9 und 12 zu.

8. (2 Punkte) Erstelle mit einer List-Comprehension eine Liste `a`, die die geraden Zahlen von 2 bis 40 enthält.

9. (2 Punkte) Erstelle mit einer List-Comprehension eine Liste `a`, die alle ungeraden Zahlen von 1 bis 30 enthält.

10. (3 Punkte) Erstelle mit einer List-Comprehension eine Liste `a`, die alle Zahlen von 1 bis 50 enthält, die durch 4 teilbar sind.

11. (3 Punkte) Gegeben ist die Liste `b = [3, 7, 2, 9, 12, 4]`. Erstelle mit einer List-Comprehension eine neue Liste `a`, die nur die Zahlen aus `b` enthält, die größer als 5 sind.