

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.

Lösung:

```
a = []
a.append(6)
a.append(2)
a.append(17)
print(a)
print(len(a))
```

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'

Lösung: a. `a[0]` b. `a[2][1]` c. `a[1][2]` d. `a[3][0][2]`

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]`

Lösung:

a. `a[:2]` , b. `a[-1:]` , c. `a[4]` , d. `a[1:4]` , e. `a[-2:]`

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)
```

Lösung:

```
[]
[22, 10, 14]
```

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.

Lösung:

```
for z in b:
    print(2*z)
```

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.

Lösung:

```
for k in range(len(b)):
    print(3*b[k])
```

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.

Lösung:

```
z, x, y = a
```

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

Lösung:

```
a = [2*x for x in range(1,21)]
```

9. (2 Punkte) Erstelle mit einer List-Comprehension eine Liste `a`, die die Zahlen von 1 bis 10 enthält.

Lösung:

```
a = [x for x in range(1, 11)]
```

10. (2 Punkte) Erstelle mit einer List-Comprehension eine Liste `a`, die die Quadrate der Zahlen von 1 bis 10 enthält.

Lösung:

```
a = [x**2 for x in range(1, 11)]
```

11. (2 Punkte) Erstelle mit einer List-Comprehension eine Liste `a`, die alle ungeraden Zahlen von 1 bis 30 enthält.

Lösung:

```
a = [x for x in range(1, 31) if x % 2 == 1]
```

12. (3 Punkte) Erstelle mit einer List-Comprehension eine Liste a, die alle Zahlen von 1 bis 50 enthält, die durch 4 teilbar sind.

Lösung:

```
a = [x for x in range(1, 51) if x % 4 == 0]
```

13. (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.

Lösung:

```
a = [x for x in b if x > 5]
```

14. (3 Punkte) Erstelle mit einer List-Comprehension eine Liste a, die die geraden Quadratzahlen von 1 bis 20 enthält.

Lösung:

```
a = [x**2 for x in range(1, 21) if x % 2 == 0]
```