

Python LAP – Fangfragen (Übungsblatt)

Beantworte die folgenden Fragen schriftlich. Die Aufgaben prüfen dein Verständnis der Python-Grundlagen. Achte besonders auf Referenzen, Mutability und Gültigkeitsbereiche.

1) Warum ändert sich die Liste a, obwohl nur b verändert wird?

```
a = [1, 2, 3]
b = a
b.append(4)
```

2) Ist zahlen[0] eine Methode? Begründe.

3) Erkläre den Unterschied zwischen:

```
a = liste
a = liste.copy()
```

4) Warum ist x = (5) kein Tuple?

5) Warum gibt dieser Code keinen Fehler aus, obwohl Tupel unveränderlich sind?

```
t = ('A', [1, 2])
t[1].append(3)
```

6) Kann man über ein Tuple mit einer for-Schleife iterieren? Warum?

7) Ist s = {} ein Set? Wenn nein, was ist es?

8) Warum ist folgender Code nicht erlaubt?

```
s = {1, 2, 3}
print(s[0])
```

9) Was ist der Unterschied zwischen remove() und discard()?

10) Warum ist folgender Zugriff falsch?

```
noten = {'Anna': 5.5}
print(noten[0])
```

11) Was ist der Unterschied zwischen noten['Ben'] und noten.get('Ben')?

12) Was gibt folgender Code aus?

```
for x in dict:
    print(x)
```

13) Warum ist print() kein Rückgabewert?

14) Was gibt folgender Code aus?

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

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

15) Warum sollte man das Schlüsselwort global möglichst vermeiden?

16) Warum ist folgende Lambda-Funktion ungültig?

```
lambda x:  
    x + 1
```

17) Was gibt folgender Ausdruck zurück?

```
(lambda x: x * 2)(3)
```

18) Wann ist lambda ungeeignet und def besser?

19) Warum funktioniert dieser Code nicht?

```
def aussen():  
    def innen():  
        print('Hi')  
  
innen()
```

20) Was ist eine Closure? (1 Satz)

21) Was bewirkt das Schlüsselwort nonlocal?

22) Verändert .count() eine Liste oder ein Tuple?

23) Warum ist liste[0] kein Methodenaufruf?

24) Warum kann diese Schleife endlos laufen?

```
while True:  
    print('Hallo')
```

25) Was ist der Unterschied zwischen break und continue?

26) Welche Datentypen in Python sind mutable und welche immutable?

27) Welche Datenstruktur verwendest du für: - eindeutige Werte - Schlüssel-Wert-Paare - veränderbare Reihenfolge

28) Warum ist Python interpretiert, aber trotzdem performant?