

Hinweise:

- **Bearbeitungszeit:** 120 Minuten
 - **Hilfsmittel:** keine.
 - Schreiben Sie bitte auf jedes Blatt Ihren **Name**.
 - Sind die Ausgaben von Python-Programmen anzugeben, so sind die **Zeilenumbrüche** zu beachten. Ausnahme: Listen und Matrizen müssen nur sinngemäß angegeben werden.
 - Sind Sie der Meinung, dass ein dargestelltes Python-Programm mit einer Fehlermeldung abbricht, so geben Sie bitte die Art des Fehlers an (Division durch Null, Syntaxfehler usw.).
-

1. Geben Sie den Wertebereich eines vorzeichenlosen Integer-Werts der Länge 8 Bit an! (1 Punkt)
2. Stellen Sie die Binärzahl 1111001001111011 als Hexadezimalzahl dar! (2 Punkte)
3. Für ein Data-Science-Projekt sollen 600 000 Bilder der Größe 400×240 Pixel aus dem Internet herunter geladen werden. Die Farbinformation ist mit 3 Byte pro Pixel kodiert. Die Daten sind komprimiert mit einer Kompressionsrate von 0.25 Prozent (Beispiel: 1000 Byte Bilddaten benötigen nur 250 Byte Speicherplatz). Der Datentransfer erfolgt mit 7.2 MB pro Sekunde. Wie viele Minuten dauert das Herunterladen? (3 Punkte)
4. Nennen Sie je ein Beispiel für eine interpretierte und eine kompilierte Programmiersprache. (2 Punkte)
5. Erläutern Sie den Unterschied zwischen den Vergleichsoperatoren == und **is**! (2 Punkte)
6. Geben Sie die Nummern von genau drei Zeilen an, die einen Syntaxfehler enthalten:

```
1 a = 2
2 b = 1. 5
3 print('Jetzt', "wird", 'gerechnet...')
4 for k in range[10]:
5     a = a * b
6     print(a)
7 print('Das war's!')
```

(3 Punkte)

7. Welche Ausgaben liefert das folgende Programm?

```
1 a = [4, 2, 6, 7]
2 b = a
3 a[3] = 23
4 c = b[1:]
5
6 print(a[-2])
7 print(b[1:2])
8 print(c)
```

(3 Punkte)

8. Welche Ausgaben liefert das folgende Programm?

```
1 def f(a):
2     a[0] = a[1]
3     a[2][0] = 10
4     return a
5
6 c = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
7 d = f(c)
8
9 print(d[1])
10 print(d[2])
11 print(d[0] is d[1])
```

(3 Punkte)

9. Welche Ausgaben liefert das folgende Programm?

```
1 def f(a):
2     print(a)
3     if a >= 1:
4         return f(a - 1)
5     else:
6         return a
7
8 f(3.5)
```

(3 Punkte)

10. Worin unterscheiden sich die Datentypen `tuple` und `list` in Python? (1 Punkt)

11. Erläutern Sie kurz den Unterschied zwischen normalen String-Literalen wie "Hallo!" und f-Strings wie f"Hallo!" in Bezug auf das Einbetten von Daten (z.B. Zahlen) in die Zeichenkette. (2 Punkte)

12. In der Python-Dokumentation wird die String-Methode `startswith` wie folgt beschrieben:

```
str.startswith(prefix[, start[, end]])
```

Return `True` if string starts with the *prefix*, otherwise return `False`. *prefix* can also be a tuple of prefixes to look for. With optional *start*, test string beginning at that position. With optional *end*, stop comparing string at that position.

Welche der folgenden Aufrufe sind demnach zulässig? Geben Sie auch die Rückgabewerte der zulässigen Aufrufe an.

- (a) `"Hallo!".startswith("H")`
- (b) `"Hallo!".startswith("H", 0)`
- (c) `"Hallo!".startswith("H", end=3)`
- (d) `"Hallo!".startswith("lo", 2)`
- (e) `"Hallo!".startswith("Ha", 0, 10)`

(5 Punkte)

13. Ergänzen Sie den folgenden Python-Code so, dass die Funktion `get_mean` den Mittelwert einer Liste aus Zahlen zurück gibt.

```
1 def get_mean(l):
2
3     if _____:
4         print('List is empty. Returning 0 as mean!')
5         return 0
6
7     list_sum = _____
8     for value in _____:
9         list_sum _____ value
10
11     return _____
12
13 print(get_mean([2, 3, 1, -4]))
14 print(get_mean([0, 3, 1]))
```

(5 Punkte)

14. Welche Ausgaben liefert das folgende Programm?

```
1 class Person:
2
3     def __init__(self, name):
4         self.name = name
5
6     def print_name(self):
7         print(self.name)
8
9 class Child(Person):
10
11     def print_name(self):
12         print('Little', self.name)
13
14 john = Person('John')
15 tom = Child('Tom')
16
17 print('A: ', end='')
18 john.print_name()
19 print('B: ', end='')
20 tom.print_name()
21 print('C: ', end='')
22 Child.print_name(john)
23 print('D: ', end='')
24 Person.print_name(tom)
25 print('E: ', end='')
26 print(john.name)
27 print('F: ', end='')
28 print(tom.name)
```

(6 Punkte)

15. Warum sollten NumPy-Arrays und Pandas-DataFrames nicht Zeile für Zeile aufgebaut oder erweitert werden? Warum ist dies bei Python-Listen kein Problem? Erläutern Sie dazu kurz die Unterschiede in der Speicherung von NumPy-Arrays und Python-Listen. (4 Punkte)

16. Welche Ausgaben liefert das folgende Programm?

```
1 import numpy as np
2
3 A = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
4
5 B = A[0:1, 0:2]
6 C = A[:, ::2]
7
8 print(A.shape)
9 print(A)
10
11 print(B.shape)
12 print(B)
13
14 print(C.shape)
15 print(C)
```

(3 Punkte)

17. Welche Ausgaben liefert das folgende Programm?

```
1 import numpy as np
2
3 a = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9])
4 B = np.array([[1, 2], [3, 4]])
5 C = np.array([[True, False], [False, True]])
6
7 print(a[B])
8 print(B[C])
```

(2 Punkte)

18. Welche Ausgaben liefert das folgende Programm?

```
1 import numpy as np
2 import pandas as pd
3
4 s = pd.Series({2: 5, 7: 3, 5: 2, 4:1})
5
6 print(s.loc[2])
7 print(s.iloc[2])
```

(2 Punkte)

19. Welche Ausgaben liefert das folgende Programm?

```
1 import numpy as np
2 import pandas as pd
3
4 a = np.array([[1, 2, 3], [2, 4, 5], [3, 5, 6]])
5 df = pd.DataFrame(a, index=['top', 'middle', 'bottom'],
6                   columns=['left', 'middle', 'right'])
7
8 print(df.loc['top', 'middle'])
9 print(df.iloc[2, 1])
10 print(df.loc['bottom', df.columns[2]])
11 print(df['middle'].iloc[1])
```

(4 Punkte)

20. Wie unterscheiden sich Ordinaldaten von numerischen Daten? (1 Punkt)

21. Welche Art Daten werden durch die Pandas-Typen Timedelta und Timestamp repräsentiert? (2 Punkte)

22. Bringen Sie die Zeilen des folgende Python-Programms in eine funktionsfähige Reihenfolge (es gibt mehrere Möglichkeiten). Geben Sie die Ausgaben Ihres Programms an.

```
A: df['is_even'] = [False, True, True]
B: print(df.loc['B':'D', 'number'].sum())
C: print(df.at['A', 'is_even'])
D: df = pd.DataFrame([2, 1, 4], index=['A', 'B', 'C'])
E: import pandas as pd
F: df.loc['D', :] = (3, False)
G: df = df.drop(index=['A'])
H: df.columns=['number']
```

(6 Punkte)