

Probeklausur

Musterlösung

Frage 1: Ausdrücke in Python

Geben Sie für jeden der folgenden Ausdrücke das Ergebnis in Python und dessen Datentyp (`str`, `int`, `float` oder `bool`) an.

- a) $10//4$ 2 Typ: int
- b) $2+2*3**2$ 20 Typ: int
- c) $2*\text{float}("2.5")$ 5.0 Typ: float
- d) $2*"2.5"$ '2.52.5' Typ: string
- e) $5*5<6*6$ True Typ: bool
- f) $\min(6,0\text{b}101)$ 5 Typ: int
- g) $1\text{e}-3$ 0.001 Typ: float
- h) $\text{bin}(0\text{xf})$ '0b1111' Typ: string
- i) `True and False` False Typ: bool
- j) `(a==b) or not (a==b)` True Typ: bool

Frage 2: Code beschreiben

Betrachten Sie die folgende Python-Funktion:

```
1 def f(liste1, liste2):
2     ret = []
3     for v in liste1+liste2:
4         if v in liste1 and v in liste2:
5             if v not in ret:
6                 ret.append(v)
7     return ret
```

a) Beschreiben Sie in einem Satz, was die Befehle in den folgenden Zeilen bewirken:

- Zeile 2: Weise der Variable `ret` die leere Liste zu.
- Zeile 3: Iteriere über jedes Element der beiden als Parameter übergebenen Listen (Die zwei Listen werden dafür aneinandergehängt).
- Zeile 4: Überprüfe ob das aktuelle Element `v` in beiden Listen `liste1` und `liste2` auftaucht.
- Zeile 5: Wenn die Bedingung aus Zeile 4 zutrifft: Überprüfe ob das aktuelle Element `v` nicht in der Liste `ret` enthalten ist.
- Zeile 6: Wenn die Bedingungen aus Zeile 4 und 5 zutreffen: Füge das aktuelle Element `v` der Liste `ret` hinzu.

Frage 2 (Fortsetzung)

b) Was ist das Ergebnis von `f([1,3,4,5], [1,2,5,9,12])?`

[1,5]

c) Beschreiben Sie die Funktionsweise der Funktion `f` in ihren eigenen Worten (1-2 Sätze).

Die Funktion findet den Mengenschnitt der beiden Listen `liste1` und `liste2`. Das bedeutet, sie gibt eine Liste ohne Duplikate zurück, die alle gemeinsamen Elemente der beiden Listen enthält.

Frage 3: Pythonbefehle

Schreiben sie jeweils eine oder zwei Zeilen in Python, die...

- a) 500 mal den String "Python" auf dem Bildschirm ausgeben

```
print('Python'*500)
```

- b) Die Variable i um eins erhöht falls q gerade ist.

```
if q%2==0:  
    i+=1
```

- c) Das Element 0 aus der liste zahlen entfernt

```
zahlen.remove(0)
```

- d) Die Zahlen von 1 bis 1000 auf dem Bildschirm ausgibt

```
for i in range(1,1001)  
    print(i)
```

Frage 4: Die Funktion beginnt_mit entwerfen

Schreiben Sie eine Funktion `beginnt_mit(s, anfang)`.

Diese Funktion gibt True zurück wenn `s` mit dem String `anfang` beginnt und sonst False. Implementieren Sie dabei die Funktionalität selbst und greifen Sie nicht auf die Methode `string.startswith` zurück

Beispiel: für `beginnt_mit("Hallo Welt", "Hallo")` gibt die Funktion True, für `beginnt_mit("Hallo Welt", "Welt")` aber False zurück.

```
def beginnt_mit(s, anfang):  
    index = 0  
    # die folgende Zeile ist optional:  
    if len(anfang)>len(s):  
        return False  
    for c in anfang:  
        if c != s[index]:  
            return False  
        index+=1  
    return True
```

Frage 5: Code ausführen

Sie definieren folgende Funktion in Python:

```
def d(liste):
    ret = []
    tmp = []
    for v in liste:
        if v in tmp:
            ret.append(v)
        else:
            tmp.append(v)
    return ret
```

a) Was ist das Ergebnis von `d(['a', 'b'])`?

[]

b) Was ist das Ergebnis von `d([1,2,3,2,1])`?

[2,1]

c) Beschreiben Sie die Funktionsweise der Funktion `d` in ihren eigenen Worten
(1-2 Sätze).

Die Funktion gibt eine neue Liste mit allen doppelten Elementen der Eingabeliste aus.

Frage 6: Multiple Choice

Kreuzen Sie die richtigen Antworten an. Mehrere Antworten können richtig sein.

a) Welche der folgenden for-Schleifen sind syntaktisch korrekt?

- `for i in 10:`
- `for c in "abcdefg":`
- `for i in range(1,11):`
- `for s in ["Eins","Zwei","Drei"]:`

b) Welche der folgenden booleschen Ausdrücke sind True?

- `"aAbBcC".upper() == "AAB"+"BCC"`
- `[1]*10 == [1+1]*5`
- `bool("")`
- `bool([1, 2, 3])`

c) Eine Python Sequenz `seq` unterstützt immer ...

- die Indizierung mit `seq[]`
- den Aufruf als Funktion mit `seq()`
- die Iteration mit for: `for v in seq`
- Die Umwandlung in eine Ganzzahl mit `int(seq)`

d) Was ist wahr über Methoden?

- Methoden nennt man Funktionen die Teil eines Python Moduls sind.
- Methoden sind stets mit einem Datentyp (also einer Klasse) verbunden.
- Die Methode `upper()` von Python strings ruft man mit `str.upper()` auf.
- `int("4")` ruft eine spezielle Methode, einen Konstruktor auf.