ETH zürich
Departement Informatik

Programmieren und Problemlösen

Dr. Dennis Komm
https://lec.inf.ethz.ch/ppl/
22. März 2021

Übungsblatt 1
Lösungsvorschläge

Aufgabe 1

Schreiben Sie ein Programm, das nach Ihrem Namen und einer Zahl fragt. Das Programm soll anschliessend „Herzlich Willkommen“ gefolgt von Ihrem Namen ausgeben. Danach soll angegeben werden, ob die Zahl durch 3 teilbar ist, oder nicht.
Hier ist ein Beispiel einer Ausgabe:

```
1 Bitte geben Sie Ihren Namen ein: Name
2 Bitte geben Sie eine Zahl ein: 14
3 Herzlich Willkommen Name!
4 Die Zahl 14 ist nicht durch 3 teilbar.
```

Lösung

```
1 name = input("Bitte geben Sie Ihren Namen ein: ")
2 zahl = int(input("Bitte geben Sie eine Zahl ein: "))
3 print("Herzlich Willkommen", name, "!")
4 if zahl % 3 == 0:
5     print("Die Zahl", zahl, "ist durch 3 teilbar.")
6 else:
7     print("Die Zahl", zahl, "ist nicht durch 3 teilbar.")
```

Die `input()`-Funktion gibt einen String zurück. Da wir in Zeile 2 eine Zahl definieren, muss der String zusätzlich in einen Integer umgewandelt werden. Dazu verwenden wir `int()`.
In Zeilen 4–7 prüft ein `if-else`-Statement, ob die Zahl durch 3 teilbar ist und gibt jeweils eine entsprechende Meldung aus.

Aufgabe 2

Schreiben Sie ein Programm, welches alle Elemente in der Liste `data`, deren Wert gleichzeitig auch ihre Position beschreibt, ausgibt.

```
data = [13, 4, 23, 5, 6, 714, 8, 200, 18, 9, 20, 17]
```

Lösung

```
data = [13, 4, 23, 5, 6, 714, 8, 200, 18, 9, 20, 17]
for i in range(0, len(data)):
    if data[i] == i:
        print(i)
```

Für jedes `i` in `range(0, len(data))` wird geprüft, ob es mit dem Element in `data` an der Position `i` übereinstimmt.

Aufgabe 3

Bestimmen Sie die Ausgabe des folgenden Programms.

```
for i in range(0, 5):
    for i in range(10, 17):
        print(i, end=" ")
    print()
```

Lösung

```
10 11 12 13 14 15 16
10 11 12 13 14 15 16
10 11 12 13 14 15 16
10 11 12 13 14 15 16
10 11 12 13 14 15 16
```

In der äusseren `for`-Schleife nimmt `i` die Werte 0 bis 4 an, dementsprechend wird die äussere sie insgesamt 5 Mal ausgeführt. In der inneren `for`-Schleife wird `i` dann jeweils mit den Werten 10 bis mit 16 überschrieben und ausgegeben, wobei `end= " "` die Ausgabe mit einem Leerzeichen beendet. Nach der inneren `for`-Schleife wird noch ein Zeilenumbruch mit dem Befehl `print()` ausgegeben.

Aufgabe 4

Schreiben Sie ein Programm, das Folgendes ausgibt. Versuchen Sie dazu nur eine `for`-Schleife und ein `if`-Statement zu verwenden.

```
 0     0     0     0     0
    0     0     0     0
 0     0     0     0     0
    0     0     0     0
 0     0     0     0     0
    0     0     0     0
 0     0     0     0     0
    0     0     0     0
 0     0     0     0     0
```

Lösung

```
for i in range(0,9):
    s1 = " 0    0    0    0    0 "
    s2 = "   0    0    0    0    "
    if i % 2 == 0:
        print(s1)
    else:
        print(s2)
```

Aufgabe 5

Schreiben Sie ein Programm, das das Muster aus Aufgabe 4 ausgibt, hierbei aber nur die Strings `" 0 "` und `"   "` verwendet.

Lösung

```
for i in range(0, 9):
    for j in range(0, 9):
        if i % 2 == 0:
            if j % 2 == 1:
                print("  ", end="")
            else:
                print(" 0 ", end="")
        else:
            if j % 2 == 1:
                print(" 0 ", end="")
            else:
                print("  ", end="")
    print()
```

In der ersten Zeile der Ausgabe (`i = 0`) enthalten alle geraden Positionen den String `" 0 "` und alle ungeraden den String `"   "`. In der zweiten Zeile (`i = 1`) ist es umgekehrt. Anschliessend wiederholt sich das Muster. Deswegen überprüfen wir im Code, ob es sich um

eine gerade oder einer ungerade Zeilennummer `i` handelt. Falls die Zeilennummer gerade ist (also `i % 2 == 0` gilt), gibt das Programm bei einer geraden Spaltennummer `j` den String `" O "` und bei einem ungeraden `j` den String `"   "` aus. Bei einer ungeraden Zeilennummer werden die Ausgaben der geraden und ungeraden Spaltennummern vertauscht.

Aufgabe 6

Schreiben Sie ein Programm, das Folgendes ausgibt.

```
1     O
2    OOO
3   OOOOO
4  OOOOOOO
```

Tipp 1: Um nur die Elemente an den Positionen $1, 2, 3, \ldots$ einer Liste oder eines Strings zu kopieren, kann das sogenannte „Slicing“ verwendet werden, welches wir noch später in der Vorlesung vorstellen werden: `neue_liste = meine_liste[1:]`.

Tipp 2: Wenn man eine Liste oder einen String umdrehen möchte, kann man Folgendes verwenden: `inverse_liste = meine_liste[::-1]`.

Lösung

```
1  for i in range(1, 5):
2      s = ""        # dies ist der String fuer die aktuelle Zeile
3      for j in range(0, 5):
4          if j < i:
5              s+="O"
6          else:
7              s+= " "
8      reverse_s = s[::-1]
9      s = s[1:]
10     print(reverse_s, end = "")
11     print(s)
```

Die Variable `i` nimmt die Werte 1 bis 4 an und sagt uns somit, in welcher Zeile wir uns gerade befinden. Für jede Zeile der Ausgabe wird ein String `s` erstellt. Die ersten `i` Positionen von `s` werden mit `"O"` gefüllt und der Rest mit `" "`.

Beispiel: Für `i = 2` ist `s = "O   "`.

In Zeile 6 wird Tipp 2 angewendet und damit `reverse_s` definiert, was dem invertierten String `s` entspricht.

Beispiel: Für `i = 2` ist `s = "O   "` und `reverse_s = "   O"`.

In Zeile 7 wird Tipp 1 angewendet.

Beispiel: Für `i = 2` ist `s = "O   "` und `reverse_s = "   O"`.

In Zeilen 8 und 9 werden schliesslich erst `reverse_s` und dann `s` ausgegeben.

Beispiel: Für `i = 2` ist die Ausgabe `"   OOO   "`.

Zusammengefasst wird für jede Zeile in der Ausgabe also ein String mit einer anderen Anzahl `"O"` und `" "` gebaut. Dieser String wird einmal umgedreht und einmal etwas gekürzt ausgegeben.