
Übungsblatt 4 zur Vorlesung Grundlagen der Programmierung

Übung 1

Schleifen und Strings

Strings können in Schleifen ganz einfach traversiert werden. Probieren Sie folgenden Code aus:

```
iteration = 0
zaehler = 0
while iteration < 5:
    for zeichen in "hallo, welt":
        zaehler += 1
    print("Iteration " + str(iteration) + "; Zaehler ist: " + str(zaehler))
    iteration += 1
```

Welche der alternativen Formulierungen ergeben den gleichen Output? Wenn Sie sich unsicher sind, probieren Sie aus!

Variante 1:

```
for iteration in range(5):
    zaehler = 0
    while True:
        for zeichen in "hallo, welt":
            zaehler += 1
        print("Iteration " + str(iteration) + "; Zaehler ist: " + str(zaehler))
```

Variante 2:

```
for iteration in range(5):
    zaehler = 0
    while True:
        for zeichen in "hallo, welt":
            zaehler += 1
        print("Iteration " + str(iteration) + "; Zaehler ist: " + str(zaehler))
        break
```

Variante 3:

```
zaehler = 0
text = "hallo, welt"
for iteration in range(5):
    index = 0
    while index < len(text):
        zaehler += 1
        index += 1
    print("Iteration " + str(iteration) + "; Zaehler ist: " + str(zaehler))
```

Variante 4:

```
zaehler = 0
text = "hallo, welt"
for iteration in range(5):
    while True:
        zaehler += len(text)
        break
    print("Iteration " + str(iteration) + "; Zaehler ist: " + str(zaehler))
```

Variante 5:

```
zaehler = 0
text = "hallo, welt"
for iteration in range(5):
    zaehler += len(text)
    print("Iteration " + str(iteration) + "; Zaehler ist: " + str(zaehler))
```

Übung 2

Fingerübungen

Schreiben Sie folgende kleinere Programme.

- 1) Schreiben Sie ein Programm, welches bei einem String s die Anzahl Vokale bestimmt.

Beispielausgabe:

```
s = 'hstmeanstuasmeanshst'
Anzahl Vokale: 6
```

- 2) Schreiben Sie ein Programm, welches die Zeichen eines Strings in alphabetischer Reihenfolge ausgibt

Beispielausgabe:

```
s = 'hstmeanstuasmeanshst'
a e h m n s t u
```

- 3) Schreiben Sie ein Programm, welches ausgibt, wie häufig der String hst in einem beliebigen String s auftritt

Beispielausgabe:

```
s = 'hstmeanstuasmeanshst'
Anzahl von hst in String: 2
```

- 4) Schreiben Sie ein Programm, welches den längsten Substring in s findet, bei dem die Buchstaben in alphabetischer Reihenfolge vorkommen.

Beispielausgabe:

```
s = 'hstmeanstuasmeanshst'
Längster Substring in alphabetischer Reihenfolge: anstu
```

Hinweise zur Lösung:

- Bitte beachten Sie, dass Sie Zeichen miteinander vergleichen können. Probieren Sie folgende Ausdrücke in Python aus:

```
# Vergleiche funktionieren gemäß Alphabet
print("a" > "b")
print("b" > "a")

# Achtung jedoch bei Groß-/Kleinschreibung
print("A" > "a")

# Tipp: Strings können komplett zu Kleinbuchstaben konvertiert werden
print(str.lower("Hochschule Trier – Trier University of Applied Sciences"))

# Zur Erinnerung: mit "in" können Sie prüfen, ob ein Zeichen oder String
# in einem String vorkommt:
print("tuas" in "hstmeanstuasmeanshst")
```

Übung 3

Häuser auf der Kommandozeile

Sie möchten auf der Kommandozeile gerne vereinfachte Häuser malen. Hierfür darf der Nutzer die maximale Breite des Hauses angeben und die Höhe des Baukörpers. Die minimale Breite beträgt 5 und die minimale Höhe 3. Das Haus hat im linken Teil immer eine Tür der Höhe 2. Zur Vereinfachung soll die Breite immer eine ungerade Zahl sein.

Beispiel-Haus 1: Breite 5, Höhe Körper: 3

```

      *
    ***
  *****
 *   **
 *   **

```

Beispiel-Haus 2: Breite 9, Höhe Körper: 5

```

      *
    ***
  *****
 *******
 *******
 *******
 *   *****
 *   *****

```

Beispiel-Haus 4: Breite 17, Höhe Körper:7

```

      *
    * * *
  * * * * *
* * * * * * *
* * * * * * * *
* * * * * * * * *
* * * * * * * * * *
* * * * * * * * * * *
*   * * * * * * * *
*   * * * * * * * *

```

```
*  
 * * *  
  * * * * *  
   * * * * * * *  
    * * * * * * * *  
     * * * * * * * * *  
      * * * * * * * * * *  
       * * * * * * * * * * *  
        * * * * * * * * * * * *  
         * * * * * * * * * * * * *  
          * * * * * * * * * * * * * *  
           * * * * * * * * * * * * * * *  
            * * * * * * * * * * * * * * * *
```

Schreiben Sie ein Programm, welches auf der Kommandozeile unter Angabe der Breite (zur Vereinfachung wieder nur ungerade Werte zulässig) ein Dreieck und ein Kreuz bestehend aus Punkten [.] und Sternen [*] ausgibt:

Beispiel-Kreuz: Breite 7

*	*	*	*	*	*	*
.	*	*	*	*	*	*
.	.	*	*	*	*	*
.	.	.	*	*	*	*
.	.	.	.	*	*	*
.	*	*
.	*

*	*			
.	*	.	.	.	*	.			
.	.	*	.	*	.	.			
.	.	.	*	.	.	.			
.	.	*	.	*	.	.			
*	*			
.	*	.	.	.	*	.			
.			
.			
*	*			

Wurzelziehen durch Guess-and-Check

- 1) Wir möchten nun analog die *Quadratwurzel* einer beliebigen Zahl mithilfe der linearen Suche bestimmen. Bitte implementieren Sie die Lösung als Guess-and-Check-Verfahren.
- 2) Bitte adaptieren Sie den Code für die *Quadratwurzel*- und *Kubikwurzel*-Suche derart, dass die Lösung nicht auf ganzzahlige Werte als Lösung beschränkt ist. Bitte stellen Sie sicher, dass die Kubikwurzel auch weiterhin für negative Werte bestimmt werden kann!