DS 10 – Weitere Übungen

Zeichenketten: Texte Zeichen für Zeichen

Hier kannst du das Gelernte weiter üben. Die Punkte im Titel zeigen, wie schwer eine Aufgabe ist: ●○○ leicht, ●●○ mittel, ●●● knifflig. Fang mit den leichten an.

Die Aufgaben stehen auch im Skript im Abschnitt „Weitere Übungen“; die Lösungen im Lösungsheft.

Aufgabe 1: Index und Zeichen ●○○

Gegeben ist `wort = "Freibad"`. Bestimme **ohne Computer** die Werte der Ausdrücke. Prüfe danach mit `print(...)` am Computer.

`wort[0]`	`wort[3]`	`wort[-1]`	`len(wort)`	`wort[len(wort) - 2]`

Gib außerdem an, an welchem Index das Zeichen `b` steht.

Deine Antwort:

…

Aufgabe 2: Ausgabe vorhersagen ●○○

Gib **ohne Computer** an, was das Programm ausgibt. Überprüfe danach am Computer.

```
1 text = "Klasse 10a"
2 print(len(text))
3 print(text[7] + text[8])
4 print("s" in text)
5 print("S" in text)
6 print(text.lower())
```

```
10
10
True
False
klasse 10a
```

Deine Antwort:

…

Aufgabe 3: Initialen ●○○

Implementiere eine Funktion `initialen(vorname, nachname)`, die die Initialen mit Punkten zurückgibt, zum Beispiel für „Mia“ und „Becker“ die Zeichenkette `M. B.`. Teste die Funktion mit zwei verschiedenen Namen.

Aufgabe 4: Buchstaben zählen ●○○

Implementiere eine Funktion `zaehle(text, zeichen)`, die zurückgibt, wie oft `zeichen` in `text` vorkommt. Teste sie mit diesen Aufrufen:

Aufruf	erwartet	tatsächlich
`zaehle("Banane", "a")`	`2`	
`zaehle("Mississippi", "s")`	`4`	
`zaehle("Hallo", "x")`	`0`	

Aufgabe 5: Zwei Fehler finden ●●○

Das Programm soll zählen, wie oft der Buchstabe e im Satz vorkommt, egal ob groß oder klein. Erwartet ist 7. Stattdessen bricht es mit einem Fehler ab.

```
satz = "Eine Elefantenherde"
satz.lower()
anzahl = 0
for i in range(1, len(satz) + 1):
    if satz[i] == "e":
        anzahl = anzahl + 1
print(anzahl)
```

```
Traceback (most recent call last):
  File "<Zelle 2>", line 5, in <module>
IndexError: string index out of range
```

1. Gib an, welche Fehlermeldung erscheint, und erkläre sie.
2. Das Programm enthält noch einen zweiten Fehler. Er würde dafür sorgen, dass die großen E nicht mitgezählt werden. Erkläre ihn.
3. Schreibe das Programm richtig auf.

Deine Antwort:

…

Aufgabe 6: Ohne Leerzeichen ●●○

Implementiere eine Funktion `ohne_leerzeichen(text)`, die eine neue Zeichenkette zurückgibt, in der alle Leerzeichen fehlen. Teste mit `"Hallo Welt"` (erwartet: `HalloWelt`) und mit `"0151 234 567"` (erwartet: `0151234567`).

> **Tipp:** Baue wie im Abschnitt *Einen neuen Text aufbauen* einen neuen Text auf. Beginne mit `""` und hänge nur die Zeichen an, die **kein** Leerzeichen sind. Ein Leerzeichen schreibst du als `" "`.

Aufgabe 7: Vokale tauschen ●●○

Ein bekanntes Sprachspiel: Man ersetzt in einem Satz alle Vokale durch denselben Vokal. Aus „Wir sehen uns morgen“ wird dann „Wir sihin ins mirgin“. Implementiere eine Funktion `nur_i(text)`, die eine neue Zeichenkette zurückgibt, in der jeder Vokal (a, e, i, o, u, auch großgeschrieben) durch ein kleines `i` ersetzt ist. Alle anderen Zeichen bleiben unverändert. Teste mit `"Wir sehen uns morgen"` und mit `"Otto"`.

Aufgabe 8: Handynummer verbergen ●●●

Auf Rechnungen wird eine Handynummer oft nur teilweise gezeigt, zum Beispiel `*********678`. Implementiere eine Funktion `verberge(nummer)`, die eine neue Zeichenkette zurückgibt. Darin ist jedes Zeichen durch `*` ersetzt – außer den **letzten drei** Zeichen. Teste mit `"015112345678"` (erwartet: `*********678`) und `"12345"` (erwartet: `**345`).

Tipp: Hier brauchst du die **Position** jedes Zeichens. Lass die Schleife deshalb über die Indizes laufen: `for i in range(len(nummer)):`. Überlege: Ab welchem Index beginnen die letzten drei Zeichen? Bei 12 Zeichen sind das die Indizes 9, 10 und 11.