

✓ Test 14.01.2025

WICHTIG: Schreibe Deinen Namen in das folgende Feld

1 Beginnen Sie mit dem Programmieren oder generieren Sie Code mit KI.

Aufgabe 1 - Einfache Syntax: Jede Codezeile sollte den angegebenen Text ausgeben. Leider sind die Zeilen fehlerhaft. Schreibe in den Kommentar am Ende jeder Zeile, was Du ändern musstest, damit der Text ausgegeben wird. Es kann mehrere Fehler oder keinen geben.

(3 Punkte: 0.5 Punkt je Beispiel)

```
1 print("Zen of Python:") # 1:
2 print("Beautiful is better than ugly") # 2:
3 print("Simple ist better than complex") # 3:
4 print("Complex ist better than complicated") # 4:
5 print("Readability counts") # 5:
6 print("Errors should never pass silently") # 6:
```

⇒ Zen of Python:
Beautiful is better than ugly
Simple ist better than complex
Complex ist better than complicated
Readability counts
Errors should never pass silently

Aufgabe 2 - Input: Schreibe ein Programm, das den Namen des Benutzers erfragt und ihn freundlich begrüßt. So könnte die Ausgabe aussehen:

```
Wie heißt Du? Benjamin
Hallo Benjamin! Schön, dass Du da bist.
```

Schaffst Du es, dass vor dem Ausrufezeichen kein Leerzeichen ist?

(6 Punkte: 1 = Variable, 1 = Zuweisung, 1 = input(), 1 = print(), 1 = Stringaufbau, 1 = plus für Konkatination)

```
1 benutzername = input("Wie heißt Du?")
2 print("Hallo " + benutzername + "! Schön, dass Du da bist.")
```

⇒ Hallo Ben! Schön, dass Du da bist

Aufgabe 3 - Konvertierung: Eine amerikanische Firma hat Dich als Entwickler eingestellt. Dein Lohn wird in Dollar bezahlt. Doch wie viele Euro sind das?

Schreibe ein Programm, das eine Eingabe entgegennimmt, aus der Eingabe eine Zahl macht und den Betrag in Euro (Kurs 0.97) ausgibt. Es sollen jeweils passende Texte dabei sein, die dem Benutzer helfen.

Beispielausgabe:

Bitte Betrag in Dollar eingeben: 100

Betrag in Euro: 97.0

(6 Punkte: 1 = Variable, 1 = Zuweisung, 1 = input(), 1 = Konvertierung, 1 = print(), 1 = Stringaufbau)

```
1 dollar = int(input("Bitte Betrag in Dollar eingeben: "))
2 print("Betrag in Euro: ", dollar * 0.97)
```

↔ Betrag in Euro: 97.0

Aufgabe 4 - Bedingungen: Es soll das Spiel "Zahlen raten" programmiert werden. Gegeben sind das Erzeugen der Zufallszahl und die Schleife mit der Benutzereingabe. Die Schleife wird verlassen, wenn die richtige Zahl geraten worden ist.

Das Programm soll passende Ausgaben machen, wenn die geratene Zahl zu groß oder zu klein ist. Ergänze das Programm um "if"-Abfragen, die diese Ausgaben machen.

Die Ausgabe des Programms könnte so aussehen:

```
Die gesuchte Zahl ist größer
Die gesuchte Zahl ist kleiner
Die gesuchte Zahl ist größer
Die gesuchte Zahl ist größer
Treffer! Diese Zahl war gesucht: 74
```

(5 Punkte: 0.5+0.5 = if, 1+1 = Bedingungen, 0.5+0.5 = Einrückungen, 0.5+0.5 = print())

```
1 import random
2
3 zufallszahl = random.randint(1, 100)
4
5 while True:
6     geratene_zahl = int(input("Bitte geratene Zahl eingeben: "))
7     if (geratene_zahl == zufallszahl):
8         print("Treffer! Diese Zahl war gesucht:", zufallszahl )
9         break
10    # hier Code ergänzen
11
```

↔ Die gesuchte Zahl ist größer
Die gesuchte Zahl ist kleiner
Die gesuchte Zahl ist größer
Treffer! Diese Zahl war gesucht: 74

Aufgabe 5 - Listen: Wir haben eine List von Noten von Schüler:innen und wollen den Mittelwert berechnen. Also die Summe aller Noten geteilt durch die Anzahl an Noten. Schreibe in das folgende Codefeld ein Programm, das den Mittelwert berechnet und ausgibt. Du darfst davon ausgehen, dass die Liste immer die gleiche Anzahl an Elementen hat.

(5 Punkte: 1 = Listenelemente ansprechen, 1 = Addition aller Elemente, 1 = Teilen, 1 = print, 1 = Stringaufbau)

```
1 noten = [12, 9, 14, 8, 4, 7]
2
3 print("Der Durchschnitt ist ", (noten[0] + noten[1] + noten[2] + noten[3] + noten[4] +
4
```

→ Der Durchschnitt ist 9.0

Bonus: Falls die Anzahl der Noten in der Liste vorher nicht bekannt ist, muss das Programm flexibler werden. Kannst Du das Programm so anpassen, dass alle Elemente in der Liste addiert werden - unabhängig von der Anzahl? Zum Berechnen des Durchschnitts muss dann die Anzahl der Elemente benutzt werden.

Tip 1: Wenn die Elemente der Liste in einer Schleife addiert werden sollen, muss die Variable, die die Summe enthält, vor der Schleife mit dem Wert 0 initialisiert werden.

```
summe = 0
```

Tip 2: Mit einer "for"-Schleifen kann man über alle Elemente einer Liste laufen. Beispiel:

```
for note in noten
```

Tip 3: Die Anzahl der Elemente in einer Liste liefert die Funktion len()

```
len(noten)
```

(8 Bonuspunkte = 1 = Variable, 1 = Initialisierung, 1 = For-Schleife, 1 = Summieren, 1 = print, 1 Stringaufbau, 1 = Teilen, 1 = Länge der Liste)

```
1 noten = [12, 9, 14, 8, 4, 7]
2
3 summe = 0
4 for note in noten:
5     summe = summe + note
6
7 print("Der Durchschnitt ist ", summe / len(noten) )
8
```

→ Der Durchschnitt ist 9.0

