DS 3 – Weitere Übungen

Verzweigungen: Programme treffen Entscheidungen

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: Bestanden? ●○○

Lies das Programm, **ohne** es auszuführen. Gib an, was es bei den Eingaben `70`, `50` und `49` ausgibt. Überprüfe danach am Computer.

```
1 punkte = int(input("Punkte: "))
2 if punkte >= 50:
3     print("bestanden")
4 else:
5     print("nicht bestanden")
6 print("Ende")
```

```
Punkte: 65
bestanden
Ende
```

Deine Antwort:

…

Lösung

✅ **Lösung**

Lösung

Eingabe	Bedingung `punkte >= 50`	Ausgabe
`70`	`True`	`bestanden`, danach `Ende`
`50`	`True`	`bestanden`, danach `Ende`
`49`	`False`	`nicht bestanden`, danach `Ende`

Für die Eingabe `70` sieht das so aus:

Lösung

```
1 punkte = int(input("Punkte: "))
2 if punkte >= 50:
3     print("bestanden")
4 else:
5     print("nicht bestanden")
6 print("Ende")
```

```
Punkte: 65
bestanden
Ende
```

Lösung

Getestet mit den Eingaben: `70`

Lösung

Ausgabe:

```
Punkte: «70»
bestanden
Ende
```

Lösung

`Ende` wird **immer** ausgegeben, denn Zeile 6 ist nicht eingerückt und gehört zu keinem Block. Bei `50` ist die Bedingung wegen `>=` erfüllt: 50 ist ein Grenzfall und gehört noch zu „bestanden“.

Aufgabe 2: Einrückung entscheidet ●○○

Die beiden Programme unterscheiden sich nur in der Einrückung der letzten Zeile. Gib für beide an, was sie ausgeben. Erkläre den Unterschied in einem Satz.

Programm A:

```
1 x = 3
2 if x > 5:
3     print("groß")
4     print("fertig")
```

Programm B:

```
1 x = 3
2 if x > 5:
3     print("groß")
4 print("fertig")
```

```
fertig
```

Deine Antwort:

…

Lösung

✅ **Lösung**

Lösung

Programm A gibt **nichts** aus. Programm B gibt aus:

Lösung

Ausgabe:

```
fertig
```

Lösung

Die Bedingung `3 > 5` ist falsch. In Programm A sind beide `print`-Zeilen eingerückt und gehören zum dann-Teil, also werden beide übersprungen. In Programm B gehört `print("fertig")` nicht mehr zur Verzweigung und wird deshalb in jedem Fall ausgeführt.

Aufgabe 3: Gerade oder ungerade? ●○○

Das Programm soll ausgeben, ob eine eingegebene ganze Zahl gerade oder ungerade ist. Ergänze die beiden Lücken `___` und teste dein Programm mit `8`, `7` und `0`.

Lückencode

```
zahl = int(input("Zahl: "))
if ___:
    print(zahl, "ist gerade.")
___
    print(zahl, "ist ungerade.")
```

Lösung

✅ **Lösung**

Lösung

```
zahl = int(input("Zahl: "))
if zahl % 2 == 0:
    print(zahl, "ist gerade.")
else:
    print(zahl, "ist ungerade.")
```

```
Zahl: 15
15 ist ungerade.
```

Lösung

Getestet mit den Eingaben: `7`

Lösung

Ausgabe:

```
Zahl: «7»
7 ist ungerade.
```

Lösung

Eine Zahl ist gerade, wenn beim Teilen durch 2 kein Rest bleibt. In die zweite Lücke gehört `else:` mit Doppelpunkt. Hinter `else` steht **keine** Bedingung. Die `0` ist gerade, denn `0 % 2` ergibt `0`.

Aufgabe 4: Fieber messen ●○○

Das Programm ordnet eine gemessene Körpertemperatur ein. Teste es mit den Eingaben aus der Tabelle. Überlege die Ausgabe zuerst und trage sie ein. Prüfe danach am Computer.

```
temp = float(input("Temperatur in Grad Celsius: "))
if temp < 37.5:
    print("normal")
elif temp < 38.5:
    print("erhöht")
else:
    print("Fieber")
```

```
Temperatur in Grad Celsius: 23
normal
```

Eingabe	36.8	37.4	37.5	38.4	38.5	39.2
Ausgabe						

Lösung

✅ **Lösung**

Lösung

Eingabe	36.8	37.4	37.5	38.4	38.5	39.2
Ausgabe	normal	normal	erhöht	erhöht	Fieber	Fieber

Die Grenzwerte sind `37.5` und `38.5`. Weil dort `<` steht, gehört der Grenzwert selbst schon zur **nächsten** Stufe: Bei `37.5` ist `temp < 37.5` falsch, aber `temp < 38.5` wahr. Bei `38.4` ist die zweite Bedingung noch wahr, bei `38.5` nicht mehr.

Aufgabe 5: Fehler finden: Medaillen beim Sportfest ●●○

Beim Sportfest gibt es ab 60 Punkten Gold, ab 45 Punkten Silber und ab 30 Punkten Bronze. Wer weniger hat, bekommt keine Medaille. Das Programm soll die Medaille ausgeben. Es enthält **drei Fehler**. Einer davon erzeugt keine Fehlermeldung, sondern eine falsche Ausgabe.

```
1 punkte = int(input("Punkte: "))
2 if punkte >= 30:
3     print("Bronze")
4 elif punkte >= 45:
5     print("Silber")
6 elif punkte >= 60
7     print("Gold")
8 else:
9 print("keine Medaille")
```

```
  File "<Zelle 8>", line 6
    elif punkte >= 60
                     ^
SyntaxError: expected ':'
```

1. Finde die drei Fehler. Notiere jeweils die Zeile und was falsch ist.
2. Schreibe das Programm richtig auf. Teste es mit den Grenzwerten 29, 30, 44, 45, 59 und 60.

Deine Antwort:

…

Lösung

✅ Lösung

Lösung

1. **Zeile 6:** Am Ende der `elif`-Zeile fehlt der Doppelpunkt (`SyntaxError: expected ':'`).
2. **Zeile 9:** Die Anweisung im `else`-Block ist nicht eingerückt (`IndentationError`).
3. **Zeilen 2–7:** Die Reihenfolge ist falsch. Schon bei 30 Punkten ist die erste Bedingung wahr. Dann wird „Bronze“ ausgegeben, auch bei 70 Punkten, und die anderen Fälle werden nie erreicht. Bei `>=` muss die **größte** Grenze zuerst geprüft werden.

Lösung

```
1 punkte = int(input("Punkte: "))
2 if punkte >= 60:
3     print("Gold")
4 elif punkte >= 45:
5     print("Silber")
6 elif punkte >= 30:
7     print("Bronze")
8 else:
9     print("keine Medaille")
```

```
Punkte: 65
Gold
```

Lösung

Getestet mit den Eingaben: `60`

Lösung

Ausgabe:

```
Punkte: «60»
Gold
```

Lösung

Eingabe	29	30	44	45	59	60
Ausgabe	keine Medaille	Bronze	Bronze	Silber	Silber	Gold

Aufgabe 6: Was ziehe ich an? ●●○

Schreibe ein Programm, das die Temperatur (ganze Zahl in Grad Celsius) einliest und eine Kleidungsempfehlung ausgibt:

- unter 10 Grad: „Zieh eine Jacke an."
- 10 bis unter 20 Grad: „Ein Pulli reicht."
- ab 20 Grad: „T-Shirt-Wetter!"

Danach soll das Programm fragen, ob es regnet (`ja`/`nein`). Nur wenn die Antwort `ja` ist, gibt es zusätzlich „Nimm einen Regenschirm mit." aus. Teste mit allen Zweigen und den Grenzwerten.

Lösung

✅ **Lösung**

Lösung

```python
temperatur = int(input("Temperatur in Grad Celsius: "))
if temperatur < 10:
    print("Zieh eine Jacke an.")
elif temperatur < 20:
    print("Ein Pulli reicht.")
else:
    print("T-Shirt-Wetter!")

regen = input("Regnet es? (ja/nein): ")
if regen == "ja":
    print("Nimm einen Regenschirm mit.")
```

```
Temperatur in Grad Celsius: 23
T-Shirt-Wetter!
Regnet es? (ja/nein): ja
Nimm einen Regenschirm mit.
```

Lösung

Getestet mit den Eingaben: `15`, `ja`

Lösung

Ausgabe:

```
Temperatur in Grad Celsius: «15»
Ein Pulli reicht.
Regnet es? (ja/nein): «ja»
Nimm einen Regenschirm mit.
```

Lösung

Für die Temperatur braucht man eine **mehrseitige** Verzweigung, für den Regen eine **einseitige**: Bei `nein` passiert einfach nichts. Die zweite Verzweigung beginnt mit einem neuen `if`, weil die Regenfrage unabhängig von der Temperatur ist. Mit `elif` würde sie nie gestellt, wenn schon eine Temperatur-Bedingung zugetroffen hat. Sinnvolle Testfälle: `9`, `10`, `19`, `20`, jeweils einmal mit `ja` und einmal mit `nein`.

Aufgabe 7: Schaltjahr mit Verzweigung ●●○

Im Abschnitt über Operatoren hast du im Auftrag *Schaltjahr-Check* einen Wahrheitswert berechnet. Schreibe das Programm jetzt so, dass es einen vollständigen Satz ausgibt, zum Beispiel `2024 ist ein Schaltjahr.` oder `2023 ist kein Schaltjahr.` Teste mit 2024, 2023, 1900 und 2000.

Zur Erinnerung: Ein Jahr ist ein Schaltjahr, wenn es durch 4 teilbar ist, aber nicht durch 100 – **außer** es ist auch durch 400 teilbar.

Lösung

✅ **Lösung**

Lösung

```
1 jahr = int(input("Jahreszahl: "))
2 if (jahr % 4 == 0 and jahr % 100 != 0) or jahr % 400 == 0:
3     print(jahr, "ist ein Schaltjahr.")
4 else:
5     print(jahr, "ist kein Schaltjahr.")
```

```
Jahreszahl: 2024
2024 ist ein Schaltjahr.
```

Lösung

Getestet mit den Eingaben: `1900`

Lösung

Ausgabe:

```
Jahreszahl: «1900»
1900 ist kein Schaltjahr.
```

Lösung

Eingabe	2024	2023	1900	2000
Ausgabe	Schaltjahr	kein Schaltjahr	kein Schaltjahr	Schaltjahr

Die Bedingung aus dem *Schaltjahr-Check* steht jetzt direkt hinter `if`. Du kannst sie auch zuerst in einer Variablen speichern (`schaltjahr = ...`) und dann `if schaltjahr:` schreiben. Das ist ebenfalls richtig, denn hinter `if` darf alles stehen, was `True` oder `False` ergibt.

Aufgabe 8: Welches Dreieck? ●●●

Aus drei Seitenlängen `a`, `b` und `c` kann man nur dann ein Dreieck bauen, wenn je zwei Seiten zusammen länger sind als die dritte: `a + b > c`, `a + c > b` und `b + c > a`.

Schreibe ein Programm, das drei Seitenlängen einliest. Wenn sich kein Dreieck bauen lässt, soll es das ausgeben. Sonst soll es mit einer **verschachtelten** Verzweigung ausgeben, ob das Dreieck

- **gleichseitig** ist (alle drei Seiten gleich lang),
- **gleichschenklig** ist (genau zwei Seiten gleich lang) oder
- **allgemein** ist (alle Seiten verschieden).

Teste mit 3, 4, 5 und mit 5, 5, 8 und mit 5, 5, 5 und mit 1, 2, 5.

Lösung

✅ Lösung

Lösung

```
a = float(input("Seite a: "))
b = float(input("Seite b: "))
c = float(input("Seite c: "))
if a + b > c and a + c > b and b + c > a:
    if a == b and b == c:
        print("Das Dreieck ist gleichseitig.")
    elif a == b or b == c or a == c:
        print("Das Dreieck ist gleichschenklig.")
    else:
        print("Das Dreieck ist allgemein.")
else:
    print("Aus diesen Seiten kann man kein Dreieck bauen.")
```

```
Seite a: 3
Seite b: 4
Seite c: 5
Das Dreieck ist allgemein.
```

Lösung

Getestet mit den Eingaben: `5`, `5`, `8`

Lösung

Ausgabe:

```
Seite a: «5»
Seite b: «5»
Seite c: «8»
Das Dreieck ist gleichschenklig.
```

Lösung

Eingaben	Ausgabe
`3`, `4`, `5`	`Das Dreieck ist allgemein.`
`5`, `5`, `8`	`Das Dreieck ist gleichschenklig.`
`5`, `5`, `5`	`Das Dreieck ist gleichseitig.`
`1`, `2`, `5`	`Aus diesen Seiten kann man kein Dreieck bauen.`
`2`, `3`, `5`	`Aus diesen Seiten kann man kein Dreieck bauen.`

- **Zeile 4:** Die äußere Verzweigung prüft, ob es überhaupt ein Dreieck ist. Alle drei Teilbedingungen müssen erfüllt sein, deshalb `and`.
- **Zeilen 5–10:** Die innere Verzweigung wird nur für echte Dreiecke ausgeführt. Die Reihenfolge ist wichtig: Ein gleichseitiges Dreieck erfüllt auch `a == b`. Deshalb muss „gleichseitig“ **zuerst** geprüft werden.

Der letzte Testfall ist ein Grenzfall: `2 + 3` ist genau `5`, aber nicht **größer** als 5. Die drei Seiten würden flach aufeinanderliegen. Deshalb steht in der Bedingung `>` und nicht `>=`.