

Der Schiri-Check

Thema: Mehrfachverzweigungen & Logische Operatoren

Name: _____

Klasse: _____ Datum: _____

Grundlagen & Merkkasten

WISSEN

Oft reicht eine einfache Entweder-Oder-Entscheidung nicht aus. Wenn ein Notenspiegel mehrere Stufen umfasst oder Bedingungen kombiniert werden müssen, kommen Mehrfachverzweigungen und logische Operatoren ins Spiel.

Mit `elif` (kurz für "else if") können beliebig viele Zwischenbedingungen geprüft werden. Operatoren wie `and` verlangen, dass alle Teilaussagen wahr sind, während `or` bereits wahr ist, wenn mindestens eine Teilbedingung zutrifft. Mit `not` lässt sich ein Wahrheitswert gezielt umkehren.

<code>if punkte >= 90: ... elif punkte >= 75: ... else: ...</code>	Mehrfachverzweigung: Bewertet Bedingungen streng der Reihe nach von oben nach unten.
<code>if alter >= 16 and hat_pass:</code>	Logisches UND: Trifft nur zu, wenn BEIDE Bedingungen True sind.
<code>if foul or meckern:</code>	Logisches ODER: Trifft zu, wenn MINDESTENS EINE Bedingung True ist.
<code>if not erledigt:</code>	Logisches NICHT: Kehrt den Wahrheitswert um (True wird False, False wird True).

1. Code-Puzzle (Reihenfolge & Einrückung)

AUFGABE 1

Bringe die Kacheln in die richtige logische Reihenfolge, um ein Schiedsrichter-Prüfprogramm zu erstellen.

Der Kandidat erhält die Lizenz nur, wenn er mindestens 16 Jahre alt ist **und** die Regelprüfung bestanden hat.

```
[ ] alter = 16
[ ] test_bestanden = True
[ ] if alter >= 16 and test_bestanden:
[ ]     print('Lizenz erteilt')
[ ] else:
[ ]     print('Kriterien nicht erfuehlt')
[ ] if alter >= 16 or test_bestanden:
[ ] elif test_bestanden == False:
```

Hinweis: Trage in die eckigen Klammern die korrekte Reihenfolge (1, 2, 3...) ein. Streiche überflüssige oder fehlerhafte Zeilen (Distraktoren) durch!

2. Fehlersuche & Code-Reparatur

AUFGABE 2

Für die Schiedsrichter-Lizenz gelten strenge Regeln: Man erhält die Lizenz nur, wenn man mindestens 80 Punkte erzielt hat **und** die Theorieprüfung bestanden ist.

Hat man zwar mindestens 50 Punkte erreicht, aber die Kriterien für die Lizenz nicht erfüllt, muss man in die **Nachprüfung**. Ansonsten ist man **Durchgefallen**.

Im vorgegebenen Code haben sich zwei typische Fehler eingeschlichen:

- Logikfehler:** Es wurde der Operator `or` statt `and` verwendet. Dadurch würde ein Kandidat fälschlicherweise die Lizenz erhalten, obwohl die Theorie nicht bestanden wurde (`theorie_bestanden = False`)!
- Syntaxfehler:** In der zweiten Bedingung fehlt das Schlüsselwort für die alternative Mehrfachverzweigung (`elif`).

Repariere beide Fehler, sodass bei den vorgegebenen Werten korrekterweise **Urteil: Nachprüfung** ausgegeben wird.

Fehlerhafter Ausgangs-Code:

```
1 punkte = 85
2 theorie_bestanden = False
3
4 if punkte >= 80 or theorie_bestanden:
5     print("Urteil: Lizenz erteilt")
6 punkte >= 50:
7     print("Urteil: Nachprüfung")
8 else:
9     print("Urteil: Durchgefallen")
```

Deine Korrektur / Notizen:

3. Programmier-Challenge

AUFGABE 3

Programmiere den digitalen Schiri-Assistenten für ein hitziges Match in der Sportart `Fussball`. Ein Foulspiel soll anhand der verbleibenden Spielzeit und der Art des Vergehens bewertet werden.

1. Erstelle eine Variable `restzeit` mit dem Wert `15` (in Minuten).
2. Erstelle eine Variable `taktisches_foul` mit dem Wert `True`.
3. Entscheide mit einer `if - elif - else`-Kette über die Strafe:
 - Wenn `restzeit <= 10` **oder** `taktisches_foul == True` ist:
Gib Strafe: Zeitstrafe aus.
 - Wenn `restzeit <= 30` ist (und keine Zeitstrafe ausgesprochen wurde):
Gib Strafe: Verwarnung aus.
 - In allen anderen Fällen (`else`):
Gib Strafe: Ermahnung aus.

Dein Python-Code:

Transfer-Aufgabe: Auswertungs-Reihenfolge bei Verzweigungen

AUFGABE 4

Betrachte folgenden Code-Ausschnitt:

```
punkte = 95
if punkte >= 50:
    print("Bestanden")
elif punkte >= 90:
    print("Hervorragend")
```

Erkläre, warum bei einem Punktestand von 95 niemals "Hervorragend" ausgegeben wird. Wie muss die Struktur korrigiert werden?
