Codeklasse

Der Monster-Index & Profil-Speicher

Thema: Dictionaries & Key-Value-Paare

Name: _____

Klasse: _____ Datum: _____

Grundlagen & Merkkasten

WISSEN

Bisher haben wir Listen genutzt, um mehrere Werte zu sammeln. Doch bei komplexen Wesen oder Profilen wird es unübersichtlich: Was bedeutet `daten[1]`? Ist das der Typ, das Level oder die Lebenspunkte?
Ein Dictionary (Wörterbuch) löst dieses Problem: Statt numerischer Positionen verwendet es sprechende Schlüssel (Keys) wie `"name"`, `"typ"` oder `"hp"`, denen jeweils ein bestimmter Wert (Value) zugeordnet ist.

`monster = {'name': 'Flamadrag', 'hp': 90}`	Erstellt ein Dictionary mit geschweiften Klammern und Schlüssel-Wert-Paaren.
`print(monster['name'])`	Liest den Wert zum Schlüssel 'name' aus.
`monster['level'] = 12`	Fügt ein neues Schlüssel-Wert-Paar hinzu oder überschreibt einen bestehenden Wert.
`monster['hp'] += 10`	Erhöht den bestehenden Wert eines numerischen Schlüssels.

1. Code-Puzzle (Reihenfolge & Einrückung)

AUFGABE 1

Bringe die Code-Kacheln in die richtige Reihenfolge, um ein Profil-Dictionary für `Flamadrag` anzulegen, das Level zu aktualisieren und die Daten sauber auszugeben.

[ ]	`monster = {'name': 'Flamadrag', 'typ': 'Feuer', 'level': 12}`
[ ]	`monster['level'] = monster['level'] + 1`
[ ]	`print('Monster:', monster['name'])`
[ ]	`print('Typ:', monster['typ'])`
[ ]	`print('Level:', monster['level'])`
[ ]	`print('Monster:', monster[0])`
[ ]	`monster = ['Flamadrag', 'Feuer', 12]`

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

Im folgenden Code haben sich zwei typische Dictionary-Fehler eingeschlichen: Ein Syntax-Fehler bei der Definition und ein ungültiger Schlüsselzugriff.

Korrigiere den Code, sodass das Wesen `Flamadrag` mit `90` HP initialisiert wird, `15` HP Heilung erhält und beide Werte korrekt ausgegeben werden.

Fehlerhafter Ausgangs-Code:

```
# Finde und repariere die Fehler:
kreatur = {
    "name": "Flamadrag"
    "hp": 90
}

kreatur["hp"] = kreatur["hp"] + 15

print("Name:", kreatur["name"])
print("Aktuelle HP:", kreatur["lebenspunkte"])
```

Deine Korrektur / Notizen:

3. Programmier-Challenge

AUFGABE 3

Erstelle einen vollständigen Monster-Index-Eintrag:

1. Erstelle ein Dictionary namens `profil` mit folgenden Schlüsseln:
- `"name"` mit dem Text `'Flamadrag'`
- `"typ"` mit dem Text `'Feuer'`
- `"level"` mit der Ganzzahl `12`
- `"hp"` mit der Ganzzahl `90`
2. Erhöhe das Level um `1` (also `profil["level"] += 1`).
3. Füge dem Dictionary ein neues Schlüssel-Wert-Paar hinzu:
- Schlüssel `"status"` mit dem Text `'bereit'`
4. Gib die Werte formatiert auf der Konsole aus:
- `Monster: <Name>`
- `Level: <Level>`
- `Status: <Status>`

Dein Python-Code:

Transfer-Aufgabe: Liste vs. Dictionary

AUFGABE 4

Erkläre anhand eines Beispiels, wann der Einsatz einer Liste sinnvoll ist und wann ein Dictionary die bessere Wahl darstellt.
Was passiert, wenn man versucht, auf einen Schlüssel zuzugreifen, der im Dictionary überhaupt nicht existiert?