

Das unendliche Gaming-Inventar

Thema: Listen-Grundlagen, Index-Zugriff & Listen-Operationen

Name: _____

Klasse: _____ Datum: _____

Grundlagen & Merkkasten

WISSEN

Bislang konnte jede Variable immer nur genau einen einzigen Wert speichern. In Programmen – wie bei einem Rucksack in einem Computerspiel – müssen wir jedoch oft ganze Sammlungen von Daten verwalten.

Hierfür nutzen wir Listen: Eine geordnete, veränderbare Kette von Elementen. Jedes Element erhält einen festen Platzhalter, den sogenannten Index, der in der Informatik immer bei 0 beginnt!

<code>inventar = ['Trank', 'Schild']</code>	Erstellt eine Liste mit eckigen Klammern. Elemente werden mit Kommas getrennt.
<code>print(inventar[0])</code>	Index-Zugriff: Liest das allererste Element der Liste aus (Zählung startet bei 0).
<code>inventar.append('Schwert')</code>	Fügt ein neues Element ganz am Ende der Liste hinzu.
<code>inventar[1] = 'Goldschild'</code>	Ersetzt/Überschreibt das Element an Index-Position 1 mit einem neuen Wert.
<code>inventar.remove('Trank')</code>	Sucht den genannten Wert in der Liste und entfernt dessen erstes Vorkommen.
<code>del inventar[0]</code>	Entfernt das Element an der konkreten Index-Position 0.

1. Code-Puzzle (Reihenfolge & Einrückung)

AUFGABE 1

Bringe die Code-Blöcke in die richtige Reihenfolge:

1. Erstelle die Liste `rucksack` mit den Items `'Heiltrank'` und `'Holzschild'`.
2. Füge mit `.append()` das Item `'Feuerschwert'` hinzu.
3. Gib das erste Item (Index 0) auf der Konsole aus.
4. Gib die gesamte Liste aus.

☐	<code>rucksack = ['Heiltrank', 'Holzschild']</code>
☐	<code>rucksack.append('Feuerschwert')</code>
☐	<code>print('Erstes Item:', rucksack[0])</code>
☐	<code>print(rucksack)</code>
☐	<code>print('Erstes Item:', rucksack[1])</code>
☐	<code>rucksack = rucksack + 'Feuerschwert'</code>

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

Repariere das Inventar-Skript:

1. An Index 0 soll der `'Heiltrank'` ausgelesen werden. Behebe den falschen Index!
2. Ersetze das Item an Index 1 durch `'Goldschild'`.
3. Entferne das verbrauchte Item `'Heiltrank'` mit `.remove()`.

Fehlerhafter Ausgangs-Code:

```

1 items = ["Heiltrank", "Holzschild"]
2
3 # 1. Erstes Element anzeigen (Achtung: Indexfehler!)
4 erstes = items[1]
5 print("Erstes Item:", erstes)
6
7 # 2. Zweites Element aufrüsten
8 items[1] = "???"
9
10 # 3. Altes Item löschen
11 items.delete("Heiltrank")
12
13 print("Aktualisiertes Inventar:", items)

```

Deine Korrektur / Notizen:

3. Programmier-Challenge

AUFGABE 3

Schreibe ein Inventar-Verwaltungssystem:

1. Erstelle eine Liste `beutel` mit den zwei Elementen `'Heiltrank'` und `'Holzschild'`.
2. Füge mit der Methode `.append()` das Element `'Feuerschwert'` ans Ende der Liste an.
3. Speichere das Element an Index-Position 0 in einer Variablen `haupt_item`.
4. Gib auf der Konsole exakt folgendes Muster aus:

Slot 1: <haupt_item>

Inventar: <beutel>

Dein Python-Code:

Transfer-Aufgabe: Der Off-by-One-Denkfehler

AUFGABE 4

Gegeben ist eine Liste mit 4 Elementen:

```

beute = ["Gold", "Rubin", "Smaragd", "Diamant"]
print(beute[4])

```

Warum stürzt das Programm mit einem `IndexError: list index out of range` ab, obwohl die Liste 4 Elemente enthält?
Welcher Index muss stattdessen in den eckigen Klammern stehen, um an den `"Diamant"` zu gelangen?