LU02.A06 - Einkaufsliste

Entwickle eine Funktion, die in einer Einkaufsliste mit Produktdetails die Produktpreise anpasst. Die Aufgabe ist, die Preise der Produkte zu verändern, ohne die ursprüngliche Liste zu verändern. Die Preise werden in einem separaten Wörterbuch (Dictionary) gespeichert. Erstellen Sie ebenfalls eine Liste die die Gesamtkosten der Einkaufsliste mit den neuen und den alten Preisen berechnet und ausgibt.

Anleitung in Schritten

1. **Definieren der Einkaufsliste:** Definiere die ursprüngliche Einkaufsliste als Liste von Dictionaries, wobei jedes Dictionary ein Produkt enthält (z.B. {'Produkt': 'Apfel', 'Menge': 5, 'Preis': 2.3}).
2. **Preisliste erstellen:** Erstelle ein separates Dictionary für die Preise der Produkte.
3. **Funktion zur Preisanpassung:** Schreibe eine Funktion, die die Einkaufsliste und die Preisliste akzeptiert. Diese Funktion sollte eine neue, aktualisierte Liste erstellen, ohne die ursprünglichen Listen zu ändern.
4. **Gesamtpreis berechnen:** Entwickle eine Funktion, die die Gesamtkosten der Einkaufsliste basierend auf den Preisen und den Mengen berechnet.
5. **Testen Sie Ihre Funktionen:** Verwende den `if __name__ == '__main__':` Block, um verschiedene Produkte und Preise zu testen.

Vorlage

```
products = [
    {'product': 'Apple', 'quantity': 5, 'price': 1.2},
    {'product': 'Banana', 'quantity': 2, 'price': 0.8},
    # Add more products as needed
]

new_prices = {
    'Apple': 0.5,
    'Banana': 0.3,
    # Add more prices as needed
}

def update_prices(products, new_prices):
    # Update the price in the new products dictionary
    pass

def calculate_total(products):
    # Calculate the total cost based on products and prices
    pass
```

```python
if __name__ == '__main__':
    # Test your functions here
    pass
```

Zusammenfassung

Diese Aufgabe hilft, das Konzept der Unveränderlichkeit in Python zu verstehen, indem sie eine realistischere Situation nachahmt, in der Listen und Dictionaries zusammen verwendet werden. Die Verwendung von unveränderlichen Werten kann dazu beitragen, Fehler zu vermeiden und den Code klarer und verständlicher zu machen.

M323-LU02, M323-A1I, M323-C1I

© Kevin Maurizi

From:
https://wiki.bzz.ch/ - **BZZ - Modulwiki**

Permanent link:
https://wiki.bzz.ch/modul/m323/learningunits/lu02/aufgaben/immutable3

Last update: **2026/08/18 10:20**