

Lernnachweis zu Kompetenz B4G

Anwendung von Map, Filter und Reduce auf Listen

Die Kompetenz B4G umfasst die individuelle Anwendung der Funktionen Map, Filter und Reduce auf Listen. Diese Funktionen bieten eine leistungsstarke Möglichkeit, Listenoperationen auf kompakte und effiziente Weise durchzuführen. Im folgenden Beispiel werden diese Funktionen einzeln demonstriert.

Anwendung von Map, Filter und Reduce

1. Map - Anwendung einer Funktion auf jedes Element der Liste

```
# Ursprüngliche Liste von Zahlen
numbers = [1, 2, 3, 4, 5]

# Quadrat jeder Zahl mithilfe von Map
squared_numbers = list(map(lambda x: x**2, numbers))

# Ausgabe
print("Original:", numbers)
print("Quadrate:", squared_numbers)
```

2. Filter - Auswahl von Elementen basierend auf einer Bedingung

```
# Ursprüngliche Liste von Zahlen
numbers = [1, 2, 3, 4, 5]

# Filtern von geraden Zahlen mithilfe von Filter
even_numbers = list(filter(lambda x: x % 2 == 0, numbers))

# Ausgabe
print("Original:", numbers)
print("Gerade Zahlen:", even_numbers)
```

3. Reduce - Reduzierung der Liste auf ein einzelnes Ergebnis

```
from functools import reduce

# Ursprüngliche Liste von Zahlen
numbers = [1, 2, 3, 4, 5]

# Summierung der Zahlen mithilfe von Reduce
sum_of_numbers = reduce(lambda x, y: x + y, numbers)

# Ausgabe
print("Original:", numbers)
print("Summe:", sum_of_numbers)
```

Erklärung:

- *Map*: Die «map»-Funktion wendet die Quadratfunktion auf jedes Element der Liste an, um eine Liste der Quadrate zu erstellen.
- *Filter*: Die «filter»-Funktion selektiert nur die geraden Zahlen aus der Liste und erstellt eine neue Liste.
- *Reduce*: Die «reduce»-Funktion summiert alle Zahlen in der Liste auf, um ein einzelnes Ergebnis zu erhalten.

Die individuelle Anwendung von Map, Filter und Reduce ermöglicht eine präzise Steuerung der Liste, je nach den Anforderungen der Operation.

Dieses Beispiel demonstriert die praktische Anwendung der Kompetenz B4G durch die separate Verwendung der Funktionen Map, Filter und Reduce auf Listen in Python.