**Lernnachweis zu Kompetenz B4G: Funktionen Map, Filter und Reduce anwenden**

**Kompetenz:** Ich beherrsche die Anwendung der Funktionen Map, Filter und Reduce auf Listen.

In dieser Lernsequenz demonstrierte ich die individuelle Anwendung von Map, Filter und Reduce auf Listen in einer Flask-Anwendung.

**Funktionalität im Code (ca. 150-200 Zeichen):** Die Flask-Anwendung bietet drei Routen: /map\_example, /filter\_example und /reduce\_example. Jede zeigt die Anwendung einer Funktion auf einer Liste. Zum Beispiel wird mit Map und einem Lambda-Ausdruck die Verdopplung der Zahlen durchgeführt.

**Umsetzung im Code mit Flask:**

Ein Bild, das Text, Screenshot enthält.

Automatisch generierte Beschreibung

**Reflexion:** Die individuelle Anwendung von Map, Filter und Reduce auf Listen bietet nicht nur leistungsstarke Werkzeuge zur Datenmanipulation, sondern fördert auch einen funktionalen Programmierstil. Diese Konzepte erweitern meine Fähigkeiten zur eleganten Transformation, Filterung und Aggregation von Daten.

**Lernnachweis zu Kompetenz B4F: Kombinierte Anwendung von Map, Filter und Reduce**

**Kompetenz:**

Ich kann die Funktionen Map, Filter und Reduce kombiniert verwenden, um Daten zu verarbeiten und zu manipulieren, die komplexere Transformationen erfordern.

In dieser Lernsequenz habe ich in einer Flask-Anwendung die kombinierte Anwendung von Map, Filter und Reduce demonstriert, um komplexe Datenverarbeitungen durchzuführen.

**Funktionalität im Code:** Die Flask-Anwendung enthält eine Route /combined\_example, die eine Liste von Zahlen nach bestimmten Kriterien transformiert. Dies erfolgt durch die kombinierte Anwendung von Map, Filter und Reduce mit Hilfe von Lambda-Ausdrücken.

**Umsetzung im Code mit Flask:**

**Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

Automatisch generierte Beschreibung**

**Reflexion:**

Die kombinierte Anwendung von Map, Filter und Reduce eröffnet vielfältige Möglichkeiten für komplexe Datenmanipulationen. Dies ermöglicht eine elegante Verarbeitung von Daten durch Transformation, Filterung und Aggregation in einer effizienten und funktionalen Weise. Die praxisnahe Umsetzung in der Flask-Anwendung verdeutlicht die Anwendbarkeit dieser Konzepte in realen Szenarien, durch die Erstellung eines API-Endpoints.

**Lernnachweis zu Kompetenz B4E**

**Kompetenz:**

Ich kann Map, Filter und Reduce verwenden, um komplexe Datenverarbeitungsaufgaben zu lösen, wie z.B. die Aggregation von Daten oder die Transformation von Datenstrukturen.

In dieser Lernsequenz habe ich in einer Flask-Anwendung die Anwendung von Map, Filter und Reduce zur Lösung einer komplexen Datenverarbeitungsaufgabe demonstriert.

**Funktionalität im Code:**

Die Flask-Anwendung bietet eine Route /complex\_processing, die Datenstrukturen aggregiert und transformiert. Dabei werden nur Elemente berücksichtigt, deren Wert größer als 10 ist. Die Anwendung von Map, Filter und Reduce ermöglicht eine effiziente und klare Lösung.

**Umsetzung im Code mit Flask:**

**Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

Automatisch generierte Beschreibung**

**Reflexion:**

Die Lösung von Datenverarbeitungsaufgaben erfordert oft komplexe Transformationen. In diesem Beispiel habe ich Map, Filter und Reduce verwendet, um Datenstrukturen zu aggregieren. Durch die Anwendung dieser Funktionen konnte ich eine klare und effiziente Lösung für die gewünschte Aufgabe implementieren.