

# Lernnachweis zu Kompetenz A1G

## Eigenschaften von Funktionen (z.Bsp. pure function) und Unterschied zu anderen Programmierstrukturen (z.Bsp. zu Prozedur).

### Das Konzept von Pure Functions

Pure Functions sind ein Grundkonzept der funktionalen Programmierung. Sie zeichnen sich durch zwei Hauptmerkmale aus:

1. **Deterministisch:** Das Ergebnis einer pure function hängt ausschließlich von den übergebenen Eingabewerten ab und hat keine Seiteneffekte. Das bedeutet, dass sie bei gleichen Eingabewerten immer das gleiche Ergebnis liefern.
2. **Keine Seiteneffekte:** Eine pure function verändert nichts in ihrem Umfeld. Sie wirkt sich nicht auf den globalen Zustand aus und verändert keine Eingabeparameter.

### Warum Pure Functions?

Pure Functions bieten mehrere Vorteile:

- **Testbarkeit:** Da das Ergebnis nur von den Eingaben abhängt, sind pure functions leicht zu testen.
- **Wartbarkeit:** Sie sind leichter zu verstehen und zu warten, da sie keine externen Zustände verändern.
- **Parallelisierung:** Da sie keine Seiteneffekte haben, können sie sicher parallel ausgeführt werden.
- **Codequalität:** Die Verwendung von Pure Functions trägt zur Verbesserung der Codequalität bei.
- **Modularität:** Pure Functions tragen zur Bildung von unabhängigen Modulen im Code bei.

### Vergleich mit Unpure Functions

Im Gegensatz zu Pure Functions können unpure Funktionen Seiteneffekte haben oder von externen Zuständen abhängig sein, was sie schwerer zu verstehen und zu warten macht.

### Beispiel einer Unpure Function

```
total = 0

def add_to_total(x):
    global total
    total += x
```

### Beispiel einer Pure Function

```
def add(x, y):
    return x + y
```