

LU03d - First-Class Functions

In der funktionalen Programmierung sind Funktionen First-Class Citizens auf Deutsch **Bürger erster Klasse**. Dies bedeutet, dass Funktionen genauso behandelt werden wie andere Werte wie Zahlen, Strings oder Listen. Sie können als Argumente an andere Funktionen übergeben, von Funktionen zurückgegeben und in Variablen gespeichert werden.

Definition von First-Class Functions

First-Class Functions haben die folgenden Eigenschaften:

1. **Sie können in Variablen gespeichert werden:** Eine Funktion kann einer Variablen zugewiesen und später über diese Variable aufgerufen werden.
2. **Sie können als Argumente übergeben werden:** Funktionen können als Parameter an andere Funktionen übergeben werden.
3. **Sie können von anderen Funktionen zurückgegeben werden:** Funktionen können andere Funktionen erstellen und zurückgeben.
4. **Sie können in Datenstrukturen gespeichert werden:** Funktionen können in Listen, Dictionaries und anderen Datenstrukturen gespeichert werden.

Beispiele für First-Class Functions

Hier sind einige Beispiele, die die Konzepte von First-Class Functions in Python demonstrieren.

Funktionen in Variablen Speichern

```
def say_hello(name):  
    return f'Hello, {name}!'  
  
if __name__ == '__main__':  
    greeting = say_hello  
    print(greeting('Alice')) # Ausgabe: Hello, Alice!
```

Beachten Sie die fehlenden Klammern bei `greeting = say_hello`. Mit Klammern (`say_hello()`) würde die Funktion *aufgerufen* und ihr Ergebnis gespeichert. Ohne Klammern wird die Funktion *selbst* zugewiesen – `greeting` und `say_hello` sind danach zwei Namen für dasselbe Funktionsobjekt.

Funktionen als Argumente Übergeben

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

```
    return x + y

def subtract(x, y):
    return x - y

def operate(func, x, y):
    return func(x, y)

if __name__ == '__main__':
    print(operate(add, 3, 5)) # Ausgabe: 8
    print(operate(subtract, 10, 3)) # Ausgabe: 7
```

Derselbe Aufruf von `operate` liefert unterschiedliche Ergebnisse – je nachdem, welche Funktion übergeben wurde.

Funktionen von anderen Funktionen zurückgeben lassen

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

def subtract(x, y):
    return x - y

def get_operation(operator):
    if operator == '+':
        return add
    else:
        return subtract

if __name__ == '__main__':
    operation = get_operation('+')
    print(operation(5, 3)) # Ausgabe: 8
```

Bedeutung in der Funktionalen Programmierung

First-Class Functions sind ein zentraler Bestandteil der funktionalen Programmierung und ermöglichen Konzepte wie Higher-Order Functions, Closures und die allgemeine Nutzung von Funktionen als Daten. Sie fördern einen deklarativen Programmierstil, der den Code oft leichter verständlich, testbar und wartbar macht.

First-Class Functions erweitern das Konzept von Funktionen

über einfache Aktionen hinaus und machen sie zu vollwertigen Objekten innerhalb der Programmiersprache. Dieses Konzept ist eine Säule der funktionalen Programmierung und findet Anwendung in vielen modernen Programmiersprachen.

[M323-LU03](#), [M323-C2B](#), [M323-C2I](#)

 © Kevin Maurizi

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

Permanent link:
<https://wiki.bzz.ch/modul/m323/learningunits/lu03/firstclassfunctions>

Last update: **2026/08/25 11:56**

