

# LU05b - Innere und äußere Funktionen in Python

In Python können Funktionen innerhalb anderer Funktionen definiert werden. Diese inneren Funktionen sind auch als verschachtelte oder lokale Funktionen bekannt. Sie sind ein wichtiges Konzept, das in Closures, Decorators und anderen höherstufigen Funktionen in Python verwendet wird.

## Grundkonzepte

- **Äußere Funktion:** Dies ist die umschließende oder übergeordnete Funktion, die die innere Funktion enthält.
- **Innere Funktion:** Dies ist die Funktion, die innerhalb der äußeren Funktion definiert ist. Sie hat Zugriff auf die Variablen und Parameter der äußeren Funktion, auch wenn die äußere Funktion ihren Ausführungskontext bereits verlassen hat.

## Beispiel für innere Funktion

Ein einfaches Beispiel, um das Konzept zu veranschaulichen:

```
def outer_function(x):
    def inner_function(y):
        return x + y
    return inner_function

add_five = outer_function(5)
print(add_five(3)) # Output: 8
```

In diesem Beispiel ist `inner_function` die innere Funktion und `outer_function` die äußere Funktion. `add_five` ist ein Beispiel für ein Closure, das die innere Funktion zusammen mit dem Umgebungskontext (der Wert von `x`) speichert.

## Zugriff auf den Umgebungskontext

Innere Funktionen haben Zugriff auf die Variablen und Parameter der äußeren Funktion, selbst wenn die äußere Funktion bereits beendet wurde. Dies wird als Closure bezeichnet.

```
def outer_function(x):
    def inner_function(y):
        return x + y
    return inner_function

add_five = outer_function(5)
```

```
print(add_five(3)) # Output: 8
```

In diesem Beispiel behält `add_five` den Wert von `x` (5) bei, auch nachdem `outer_function` beendet wurde.

## Verwendungszwecke

- **Closures:** Zum Speichern von Zuständen zwischen verschiedenen Aufrufen.
- **Decorators:** Innere Funktionen werden oft in Decorators verwendet, um das Verhalten von Funktionen zu modifizieren.
- **Datenkapselung:** Zum Verbergen der Implementierungsdetails und zum Exponieren nur der erforderlichen Schnittstellen.
- **Temporäre Verwendung:** Für Funktionen, die nur in einem begrenzten Kontext verwendet werden und nicht im gesamten Programm erforderlich sind.

Durch das Verständnis der Funktionsweise innerer und äußerer Funktionen kann man komplexere und modularere Code-Strukturen in Python erstellen.

---

M323-LU05



© Kevin Maurizi

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

Permanent link:  
<https://wiki.bzz.ch/modul/m323/learningunits/lu05/innerfunctions>

Last update: **2024/03/28 14:07**

