

LU05.A08 - Erstellung eines Multiplikators mit Closures

Erstelle eine äußere Funktion, die einen Multiplikator als Parameter akzeptiert und eine innere Funktion zurückgibt, die diesen Multiplikator verwendet, um Zahlen zu multiplizieren.

Aufgabenstellung

Definiere eine Funktion namens `create_multiplier`. Diese Funktion sollte einen Parameter `multiplier` akzeptieren und eine innere Funktion zurückgeben, die eine Zahl akzeptiert und sie mit dem `multiplier` multipliziert.

Code-Vorlage

`main.py`

```
"""Erstellung eines Multiplikators mit Closures.

Aufgabenstellung:
https://wiki.bzz.ch/modul/m323/learningunits/lu05/aufgaben/closures
"""

def create_multiplier(multiplier):
    """
    Erstellt eine innere Funktion, die Zahlen mit dem angegebenen
    Multiplikator multipliziert.

    Args:
        multiplier (int or float): Der Multiplikator, mit dem die
        Zahlen multipliziert werden sollen.

    Returns:
        function: Eine innere Funktion, die eine Zahl akzeptiert und
        sie mit dem Multiplikator multipliziert.
    """
    # TODO: Implementiere die Multiplikation
    ...

if __name__ == '__main__':
    multiply_by_three = create_multiplier(3)
```

```
print(multiply_by_three(4)) # Erwarteter Output: 12
```

Vorgehen

1. Akzeptiere das GitHub Classroom Assignment
2. Kclone dein persönliches Repository in die Entwicklungsumgebung
3. Implementiere die Funktion `create_multiplifier` gemäß der Aufgabenstellung
4. Teste die Funktion mit verschiedenen Multiplikatoren und Zahlen

Abgabe

Die Abgabe der Lösung erfolgt als Push in das persönliche GitHub-Repository.

⇒ *GitHub Repo für externe Besucher*

GitHub Repository <https://github.com/templates-python/m323-lu05-a08-closures>

Lernende am BZZ müssen den Link zum GitHub Classroom Assignment verwenden

© Kevin Maurizi

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

Permanent link:
<https://wiki.bzz.ch/modul/m323/learningunits/lu05/aufgaben/closures?rev=1711631267>

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

