Statische Methoden in Python

Definition :

Statische Methoden sind Methoden, die innerhalb einer Klasse definiert sind, aber keine Instanz spezifischen zugreifen oder ändern. Sie werden mit dem @staticmethod Dekorator definiert.

- Keine self oder cls Parameter: Statische Methoden nehmen weder self (Instanzreferenz) noch cls (Klassenreferenz) als Parameter.

- Klassen-Level-Aufruf: Sie können auf der Klasse selbst aufgerufen werden, ohne dass eine Instanz der Klasse erforderlich ist.

- Dienstprogrammfunktionalität: Typischerweise verwendet für Dienstprogrammfunktionen, die eine Aufgabe unabhängig von Klassen- oder Instanzzustand ausführen.

Statische Methoden in Python

Beispiel :

```
class MathOperations:
    @staticmethod
    def add(x, y):
        return x + y

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

# Statische Methoden direkt auf der Klasse aufrufen
result_add = MathOperations.add(5, 3)
result_subtract = MathOperations.subtract(10, 4)

print("Addition:", result_add)  # Ausgabe: Addition: 8
print("Subtraktion:", result_subtract)  # Ausgabe: Subtraktion: 6
```

Statische Methoden in Python

Eine **Klassenvariable** ist ein Attribut, das von allen Instanzen einer Klasse geteilt wird. Sie wird direkt in der Klasse (außerhalb von __init__) definiert.

- Wird nur einmal für die gesamte Klasse gespeichert.
- Alle Objekte greifen auf denselben Wert zu
- Änderungen an der Klasse wirken auf alle Instanzen

Eine **Klassenmethode** ist eine Methode, die auf die Klasse selbst zugreift – nicht auf eine Instanz. Sie wird mit dem Dekorator @classmethod gekennzeichnet und erhält cls als ersten Parameter.

- Zugriff auf Klassenvariablen (cls.variablenname)
- Wird typischerweise für alternative Konstruktoren oder klassenweite Logik verwendet

Statische Methoden in Python

Wenn eine Methode Zugriff auf Klassenvariablen benötigt, ist es besser, eine **Klassenmethode** zu verwenden. Klassenmethoden sind dafür konzipiert, auf Klassenebene zu arbeiten und können über den cls-Parameter auf Klassenvariablen zugreifen.

```
class MyClass:
    class_variable = 0

    @classmethod
    def class_method(cls):
        cls.class_variable += 1


MyClass.class_method()
print(MyClass.class_variable)  # Ausgabe: 1
```

Statische Methoden in Python

Obwohl statische Methoden keinen direkten Zugriff auf Klassenvariablen haben, können sie dennoch explizit auf sie zugreifen, indem sie die Klasse namentlich referenzieren. Dies wird jedoch als unkonventionell angesehen und untergräbt den Zweck von statischen Methoden.

Statische Methoden sind entworfen, um unabhängig vom Zustand der Klasse oder ihrer Instanzen zu arbeiten. Ihr Zweck ist es, Aufgaben auszuführen, die keinen Zugriff auf oder Änderungen an den Daten der Klasse oder ihrer Instanzen erfordern.

```
class MyClass:
    class_variable = 0

    @staticmethod
    def static_method():
        MyClass.class_variable += 1  # Expliziter Zugriff auf die Klassenvariable


MyClass.static_method()
print(MyClass.class_variable)  # Ausgabe: 1
```

Statische Methoden in Python

Unterschied zu Instanzmethoden in Python

Instanzmethoden sind Funktionen, die innerhalb einer Klasse definiert sind und auf Instanzen (Objekten) dieser Klasse operieren. Sie haben Zugriff auf die Attribute der Instanz (und auch die Klassenattribute) und können deren Zustand ändern.

- Erster Parameter self:
  - Instanzmethoden nehmen immer mindestens einen Parameter, typischerweise self, der sich auf die spezifische Instanz der Klasse bezieht, die die Methode aufruft.
  - Dadurch kann die Methode auf Instanzattribute und andere Methoden zugreifen.

- Zugriff auf Instanzattribute:
  - Instanzmethoden können Instanzattribute lesen und ändern, was dynamisches Verhalten basierend auf dem Zustand der Instanz ermöglicht.

Statische Methoden in Python

Beispiel Instanzmethode :

```
class MyClass:

    def __init__(self, value):
        self.value = value  # Instanzattribut

    def instance_method(self):
        return self.value


# Erstellen einer Instanz der Klasse
obj = MyClass(10)

# Aufrufen der Instanzmethode
print(obj.instance_method())  # Ausgabe: 10
```

Statische Methoden in Python

Mehr erkunden :

https://www.geeksforgeeks.org/python/class-method-vs-static-method-vs-instance-method-in-python/

https://www.programiz.com/python-programming/methods/built-in/staticmethod

https://www.programiz.com/python-programming/methods/built-in/classmethod