

Namensräume (Namespaces) in Python

Einführung

Namensräume in Python sind Container, die Namen zu Objekten (wie Variablen und Funktionen) mappen. Sie sind ein Schlüsselkonzept in Python und spielen eine zentrale Rolle bei der Organisation und Isolierung von Code.

Was ist ein Namensraum?

Ein Namensraum ist eine Abstraktion, die es Python erlaubt, alle Namen, die in einem Programm definiert sind, zu verwalten und sicherzustellen, dass es keine Namenskonflikte gibt. Namen innerhalb eines Namensraums sind eindeutig und unabhängig von Namen in anderen Namensräumen.

Arten von Namensräumen

Lokale Namensräume

Lokale Namensräume enthalten Namen, die innerhalb einer Funktion definiert sind. Diese Namen sind nur innerhalb der Funktion sichtbar und existieren nur für die Dauer der Funktionsausführung.

Globale Namensräume

Globale Namensräume enthalten Namen, die auf der obersten Ebene eines Skripts oder Moduls definiert sind. Diese Namen sind im gesamten Modul sichtbar.

Eingebaute Namensräume

Eingebaute Namensräume enthalten Namen, die in Python vordefiniert sind, wie `print`, `id` und `type`. Diese Namen sind überall im Programm verfügbar.

Namensraum-Hierarchie

Python sucht Namen in einer hierarchischen Reihenfolge: zuerst im lokalen Namensraum, dann im umschließenden Namensraum (falls vorhanden), dann im globalen Namensraum und schließlich im eingebauten Namensraum.

Beispiele

Zugriff auf verschiedene Namensräume

```
x = "global" # Globaler Namensraum

def funktion():
    x = "lokal" # Lokaler Namensraum
    print(x)
```

```
funktion() # Ausgabe: lokal  
print(x)   # Ausgabe: global
```

Modifikation von globalen Namen innerhalb einer Funktion

```
x = "global"  
  
def funktion():  
    global x  
    x = "modifiziert"  
  
funktion()  
print(x) # Ausgabe: modifiziert
```