

Funktionen in Python

Einführung

Funktionen in Python sind wiederverwendbare Codeblöcke, die für eine spezifische Aufgabe konzipiert sind. Sie verbessern die Modularität und Wiederverwendbarkeit des Codes und helfen bei der Strukturierung von Programmen.

Definition und Aufruf einer Funktion

Syntax

```
def funktionsname(parameter):  
    # Anweisungen  
    return wert
```

Beispiel

```
def gruß(name):  
    return f"Hello {name}!"  
  
print(gruß("Welt"))
```

Parameter und Argumente

Funktionen können Parameter enthalten, die beim Aufruf der Funktion Argumente aufnehmen.

Beispiel mit mehreren Parametern

```
def addiere(a, b):  
    return a + b  
  
print(addiere(5, 3))
```

Standardwerte für Parameter

Parameter können Standardwerte haben, die verwendet werden, falls beim Aufruf kein Argument übergeben wird.

Beispiel

```
def gruß(name="Welt"):
    return f"Hello {name}!"

print(gruß()) # Verwendet Standardwert "Welt"
```

Schlüsselwortargumente

Beim Aufruf einer Funktion können Argumente als Schlüsselwortargumente übergeben werden, wodurch die Reihenfolge der Argumente irrelevant wird.

Beispiel

```
def benutzer_info(name, alter):
    return f"Name: {name}, Alter: {alter}"

print(benutzer_info(alter=30, name="Anna"))
```

Variable Argumentlisten

Funktionen können eine variable Anzahl von Argumenten mittels `*args` (für nicht benannte Argumente) und `**kwargs` (für benannte Argumente) empfangen.

Beispiel mit `*args`

```
def summe(*zahlen):
    return sum(zahlen)

print(summe(1, 2, 3, 4))
```

Beispiel mit `**kwargs`

```
def benutzer_info(**daten):
    return ", ".join([f"{k}: {v}" for k, v in daten.items()])

print(benutzer_info(Name="Anna", Alter=30))
```