

Prozeduren in Python

Eine **Prozedur** fasst mehrere Schritte zu einem kleinen „Mini-Programm“ zusammen.

Wenn man die Prozedur **aufruft**, werden genau diese Schritte der Reihe nach ausgeführt.

Sie erledigt also eine Aufgabe, gibt aber keinen Wert zurück.

In Python schreibt man eine Prozedur mit **def**.

Beispiel 1

```
1 def begruessung(name):  
2     print("Hallo " + name + "!")  
3  
4 # Aufruf der Prozedur  
5 begruessung("Anna")
```

In diesem Beispiel fragt die Prozedur kein Ergebnis ab. Sie führt lediglich die Ausgabe aus.

Aufgaben:

1. Ändere im ersten Beispiel den Namen auf „Paul“ und führe das Programm aus.
2. Schreibe eine eigene Prozedur mit dem Namen `verabschiedung(name)`, die den Benutzer mit „Tschüss, [Name]!“ verabschiedet.
Tipp: Verwende `print()`.

Beispiel-Aufruf:

`verabschiedung("Lena")`

Erwartete Ausgabe:

Tschüss, Lena!

Platz für deinen Code:

Funktionen in Python

Eine Funktion, die beispielsweise das Quadrat einer Zahl berechnet, nimmt eine Zahl entgegen und multipliziert sie mit sich selbst.

Möchte man zum Beispiel das Quadrat der Zahl 2 berechnen, ruft man die Funktion mit `print(quadrat(2))` auf – und erhält das Ergebnis 4.

Beispiel 2:

Wir berechnen das Quadrat einer Zahl.

```
1 def quadrat(x):  
2     return x * x  
3  
4 print(quadrat(2))
```

Erklärung:

- `def quadrat(x) :` definiert die Funktion `quadrat` mit einem Parameter `x`.
- `return x * x` gibt das Ergebnis zurück.
- `print(quadrat(2))` ruft die Funktion auf und zeigt das Ergebnis an.

Mit einer Prozedur könnten wir das Ergebnis nicht direkt weiterverwenden, weil sie keinen Rückgabewert hat.

Aufgaben:

1. Teste das Beispiel und ändere die Zahl 2 auf einen anderen Wert.
2. Schreibe eine Funktion **wuerfel**, die das Volumen eines Würfels berechnet. Der Funktionsaufruf könnte z. B. so aussehen:

```
print(wuerfel(3))
```

und sollte 27 ausgeben, wenn die Kantenlänge 3 ist.