Methoden

Online Python-Interpreter:

- https://www.programiz.com/python-programming/online-compiler/
- https://www.online-python.com

Oft benötigt man Funktionen in einem Programm öfter. Du hast schon mit den Schleifen ein Verfahren kennengelernt, um Wiederholungen im Programmcode zu vermeiden. Dabei ging es meist um einige wenige Anweisungen.

Die erste Methode

Das Schlüsselwort für eine Methode in Python ist **def**. Alles, was an Code zu dieser Methode gehören soll, wird wie bei den Schleifen eingerückt. Methoden müssen (vorerst) oben im Programmcode definiert werden.

Eine erste Methode:

method01.py

```
# Das ist die Methode
def Oerks():
   print("Örks!")
```

Wenn du die in Python-Online ablaufen lässt, passiert nichts. Dazu musst du die Methode im Hauptprogramm explizit aufrufen. Um für den Pythoninterpreter die Methode kenntlich zu machen, musst du nach dem Namen der Methode runde Klammern () setzen.

method02.py

```
# Das ist die Methode
def Oerks():
   print("Örks!")

# Hauptprogramm
# Hier kommt der Methodenaufruf
Oerks()
```

Glückwunsch. Jetzt kannst du sehr leicht „Örks!“ in deinen Programm ausgeben, aber so der richtige Gewinn ist das nicht.

Werte an Methoden übergeben

Oft möchte man mit Hilfe eine Methode Berechnungen durchführen, wobei man z.B. einen Wert an die

Methode aus dem Hauptprogramm übergibt. Jetzt kommen die Klammern nochmal zu neuen Ehren.

method03.py

```
# Das ist die Methode
def Halbiere(wert):
   print(f'Die Hälfte von {wert} ist {wert/2}.')

# Hauptprogramm
# Hier kommt der Methodenaufruf
Halbiere(4)
```



Quasi nebenbei hast du eine neue Art kennengelernt, mit print() Text mit Variablen auszugeben. Die Ausgabe beginnt mit einem „f“ für „formattet“, steht in einfachen Anführungszeichen und enthält die Variablen im Text in geschweiften Klammern.

Du kannst natürlich auch Variablen übergeben:

method03.py

```
# Das ist die Methode
def Halbiere(wert):
   print(f'Die Hälfte von {wert} ist {wert/2}.')

# Hauptprogramm
# Hier kommt der Methodenaufruf
zahl = 4
Halbiere(zahl)
```

Häh? In der Methode steht doch „wert“ und nicht „zahl“. Warum klappt das? „wert“ ist ein Parameter unserer Methode. Der Parameter wird beim Methodenaufruf auf den Wert der Variablen Zahl gesetzt. Wie der Parameter heißt, ist nicht wichtig. Den Parameter „wert“ gibt es wiederum nur innerhalb der Methode. Ich kann ihn im Hauptprogramm nicht ausgeben.

Das hier ...

method03.py

```
# Das ist die Methode
def Halbiere(wert):
   wert = wert/2

# Hauptprogramm
```

```
# Hier kommt der Methodenaufruf
zahl = 4
Halbiere(zahl)
print(wert)
```

... geht schief! Die Variable „wert“ gibt es nur innerhalb der Methode! Es kann sogar eine Variable mit dem gleichen Namen in unterschiedlichen Methoden UND dem Hauptprogramm geben, die außer dem Namen nichts miteinander zu tun haben.

Werte von Methoden zurückgeben

Bisher haben wir die Ausgabe der Rechnung durch die Methode gemacht. Da die Parameter nur innerhalb der Methode gelten, können wir nicht im Hauptprogramm auf sie zugreifen. Das ist blöd. Aber natürlich gibt es eine Lösung mit dem Schlüsselwort **return**.

method04.py

```
# Das ist die Methode
def Halbiere(wert):
    return wert/2

# Hauptprogramm
zahl = 4

# setze "ergebnis" auf den Wert, den die Methode "Halbiere()"
zurückgibt.
# die Variable ergebnis enthält jetzt den Wert, den die Methode
Halbiere() zurückgibt.

ergebnis = Halbiere(zahl)
print(ergebnis)
```

Zusammenfassung

Alles zusammengenommen kennst du das vom Beispielprogramm in einer unserer Online-Programmierumgebungen. Neu ist hier nur das Vorgehen, wenn du mehrere Parameter an die Methode übergibst.

method05.py

```
def sum(a, b):
    return (a + b)

a = int(input('Enter 1st number: '))
b = int(input('Enter 2nd number: '))
```

```
print(f'Sum of {a} and {b} is {sum(a, b)}')
```

From:
https://schule.riecken.de/ - **Unterrichtswiki**

Permanent link:
https://schule.riecken.de/doku.php?id=informatik:algorithmisch:python:methoden

Last update: **2024/07/16 12:01**