Funktionen und Parameter

Programmstrukturen

Eine **Funktion** ist ein **Programmblock** mit einem Namen. Funktionen werden genau einmal pro Programm *definiert* und können danach beliebig oft *aufgerufen* werden.

Funktionen definieren

Um eine Funktion zu definieren, verwenden wir den Befehl `def`, gefolgt vom Funktionsnamen, Klammern, einem Doppelpunkt und dem Programmblock:

```
def quadrat():
    repeat 4:
        forward(100)
        right(90)
```

Danach können wir an beliebigen Stellen in unserem Programm durch den Aufruf von `quadrat()` ein Quadrat gemäss dieser Definition zeichnen lassen.

Funktionen definieren wir immer am Anfang des Programms, im sogenannten **Funktionsblock**. Ein **Funktionsname** darf nur aus Grossbuchstaben, Kleinbuchstaben, Zahlen und dem Unterstrich (_) bestehen; er darf nicht mit einer Zahl anfangen.

Funktionen aufrufen

Eigene Funktionen, die einmal definiert sind, können danach auf die gleiche Art verwendet werden, wie die bereits bekannten Funktionen (Befehle) der `gturtle`:

```
from gturtle import *

def quadrat():
    repeat 4:
        forward(100)
        right(90)

makeTurtle()

repeat 3:
    quadrat()
    forward(100)
```

TigerJython macht dabei keinen Unterschied, ob eine Funktion von `gturtle` importiert oder in Ihrem Programm definiert wurde.

Achten Sie darauf, keine Funktionsnamen zu verwenden, welche `gturtle` bereits definiert hat (z.B. `forward`), ansonsten kann TigerJython nicht mehr auf die Funktionen der `gturtle` zugreifen und Ihre Schildkröte sitzt fest.

Funktionen mit Parametern

Eine Funktion kann über sogenannte **Parameter** gesteuert werden. Wir kennen das bereits von Funktionen wie `forward(...)`, wo wir in den Klammern angeben können, wie weit sich die Schildkröte bewegen soll.

Um eine Funktion mit Parametern zu definieren, schreiben wir den Namen des Parameters zwischen die Klammern. Wir können den Wert danach über diesen Namen ansprechen, d.h. wir können mit dem Wert Berechnungen anstellen, oder ihn an andere Funktionen weiterreichen. Wenn eine Funktion mehr als nur einen Parameter haben soll, dann werden diese mit Komma getrennt; für die Namen der Parameter gelten die gleichen Einschränkungen wie für die Namen von Funktionen.

```
def vieleck(ecken, laenge):
    repeat ecken:
        forward(laenge)
        right(360 / ecken)
```

Somit können wir nun allerlei Formen zeichnen:

```
vieleck(3, 20)
forward(50)
vieleck(4, 50)
forward(50)
vieleck(8, 30)
```

Wir können auch Funktionen schreiben, die andere eigene Funktionen aufrufen:

```
def quadrat(laenge, farbe):
    setPenColor(farbe)
    vieleck(4, laenge)
```

Programmaufbau

Damit ein Programm mit eigenen Funktionen übersichtlich bleibt, verwendet man folgende Struktur:

1. Im **Import-Block** werden fremde Programmteile importiert. In unseren Beispielen ist das jeweils die `gturtle`.
2. Der **Funktionsblock** ist dazu da, eigene Funktionen zu definieren, die im vorliegenden Programm genutzt werden.
3. Das **Hauptprogramm** schliesslich ist alles, was weder mit einem `import` noch mit einem `def` anfängt; es ist der letzte Teil des Programms.

Folgendes Beispiel illustriert die einzelnen Blöcke:

```
from gturtle import *

def vieleck(ecken, laenge):
    repeat ecken:
        forward(laenge)
        right(360 / ecken)

def quadrat(laenge):
    vieleck(4, laenge)

makeTurtle()

repeat 3:
    quadrat(50)
    forward(100)
```

Der Import-Block umfasst in diesem Beispiel lediglich die erste Zeile, darauf folgt die Definition von zwei Funktionen im Funktionsblock und schliesslich das Hauptprogramm, welches mit dem Aufruf der Funktion `makeTurtle()` anfängt.