

## Programmieren mit Jython – Teil 2

### Lernziele

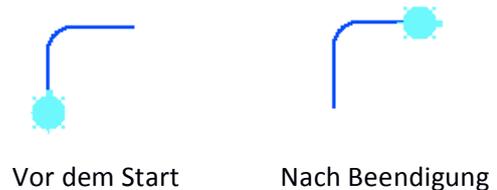
- Unterprogramme: Wie bringt man der Turtle neue Befehle bei?
- Flexibler Einsatz von Unterprogrammen durch Übergabe von Zahlenwerten (Argumenten).

### Beispiel

Das folgende Programm **definiert** ein **Unterprogramm**, das einen Haken nach rechts zeichnet. In der Abbildung sieht man Start- und Endposition der Turtle:

```
from turtle import *
makeTurtle()

def HakenR():
    forward(35)
    rightArc(15, 90)
    forward(35)
```



**Beachte:** Wie schon bei der Repeat-Schleife gehört alles was nach dem Doppelpunkt eingerückt ist (d.h. nach rechts verschoben) zum Unterprogramm. Das Einrücken des Programmtextes ist also Teil der Programmiersprache!

Der Befehl `rightArc(r, w)` zeichnet einen Kreisbogen mit Radius  $r$  und Winkel  $w$ .

Startet man das obige Programm, geschieht jedoch gar nichts, da es ausser der **Definition** von `HakenR()` keine Anweisungen enthält! Das neu definierte Unterprogramm muss zuerst **aufgerufen** werden, damit es ausgeführt wird. Das geschieht zum Beispiel in folgendem **Hauptprogramm**:

```
repeat 2:
    HakenR()
```

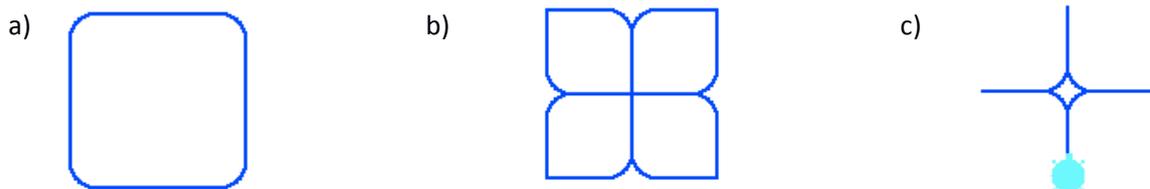


Startet man nun das Programm, wird eine Heftklammer gezeichnet:

**Beachte:** Die erste Zeile des Hauptprogramms beginnt wieder ganz links (also nicht eingerückt). Der Aufruf von `HakenR()` ist aber wieder eingerückt, da er zur Repeat-Schleife gehört. Dadurch wird das Unterprogramm `HakenR()` insgesamt 2 mal aufgerufen.

### Aufgaben

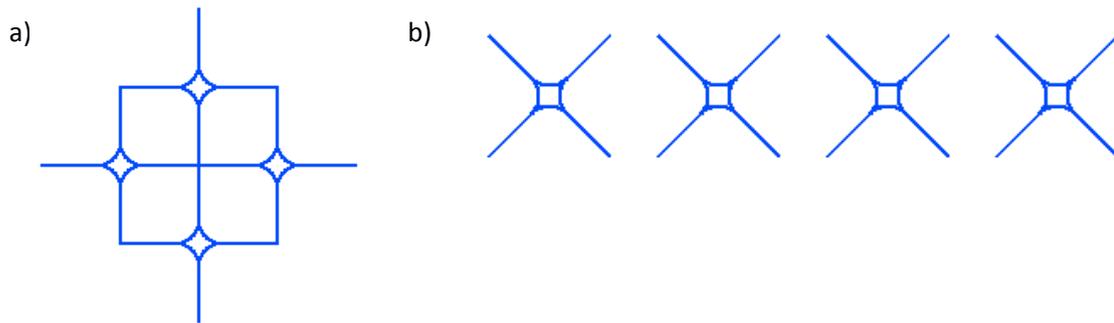
2.1) Verwende das Unterprogramm `HakenR()` und zeichne damit folgende Figuren.



2.2) Definiere nun ein neues Unterprogramm `Stern()`, das die obige Figur 2.1 c) zeichnet.

Zu Beginn und am Ende soll die Turtle unten sitzen und nach oben schauen.

2.3) Verwende nun das unter 2.2 geschriebene Unterprogramm Stern() und zeichne damit folgende Figuren.



## Unterprogramme mit Argumenten

Das folgende Programm zeichnet eine Anzahl von aneinander liegenden Quadraten einer gegebenen Groesse. Die Ausdrücke Anzahl und Groesse sind **Argumente** des Unterprogramms QuadratZeile(). Sie werden im Unterprogramm wie Variablen verwendet:

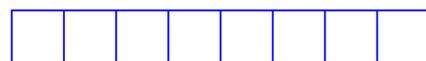
```

from turtle import *
makeTurtle()

def QuadratZeile(Groesse,Anzahl):
    repeat Anzahl:
        repeat 4:
            forward(Groesse)
            right(90)
            right(90)
            forward(Groesse)
            left(90)
        right(90)
    back(Groesse*Anzahl)
    left(90)
    forward(Groesse)

hideTurtle()
    QuadratZeile(30,8)

```



QuadratZeile(30,8)



QuadratZeile(80,3)



QuadratZeile(6,45)

Das Unterprogramm QuadratZeile() wird in der letzten Zeile mit den Werten Groesse=30 und Anzahl=8 aufgerufen. Beachte auch, dass zu Beginn die Turtle in der linken unteren Ecke sitzt und nach oben schaut, während sie am Ende in der linken oberen Ecke sitzt und ebenfalls nach oben schaut.

2.4) Benutze das Programm QuadratZeile() und definiere ein Unterprogramm QuadratNetz(), das mehrere solche Zeilen übereinander zeichnet. Dabei soll sowohl die Groesse der einzelnen Quadrate, als auch die AnzahlSpalten und die AnzahlReihen angegeben werden können.

Die Zeile, in der die Definition von QuadratNetz() beginnt, sollte also wie folgt lauten:

```

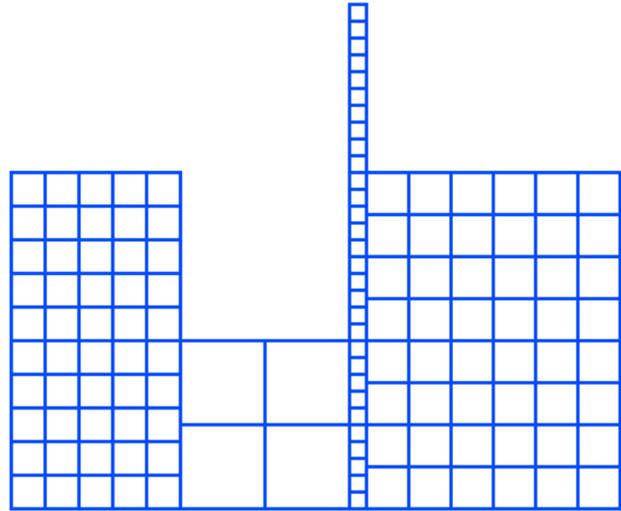
def QuadratNetz(Groesse,AnzahlSpalten,AnzahlReihen):

```

2.5) Zeichne nun folgendes Bild.

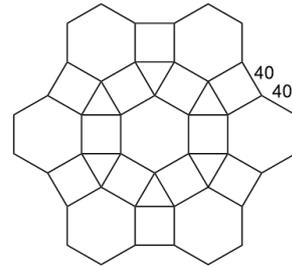
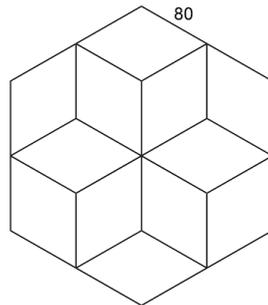
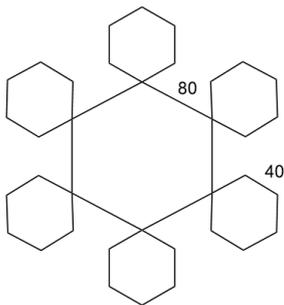
Es besteht aus 4 Teilen:

- 1) 5x10 Quadrate der Grösse 20
- 2) 2x2 Quadrate der Grösse 50
- 3) 1x30 Quadrate der Grösse 10
- 4) 6x8 Quadrate der Grösse 25



## Zusatzaufgaben

2.6) Analysiere die Figuren auf wiederkehrende gleiche Elemente. Schreibe ein Unterprogramm für ein Element. Rufe dann das Unterprogramm auf geeignete Weise wiederholt auf.



2.7) Zeichne eine Bienenwabe. Überlege gut, welches Element sich einfach wiederholen lässt.

## Hausaufgaben

Löse folgende Aufgaben mit **Papier und Bleistift** (ohne Computer):

H2.1) Schreibe ein Programm mit Unterprogramm `zack()`, welches diese Figur zeichnet:



H2.2) Was zeichnet folgendes Programm mit den Unterprogrammen `fr()`, `fl()`, `st()`?

```
def fr():  
    forward(20)  
    right(90)
```

```
def fl():  
    forward(20)  
    left(90)
```

```
def st():  
    fr()  
    fl()  
    fr()
```

```
repeat 4:  
    st()
```

Zeichne die Figur in den karierten Ausschnitt. Jedes Häuschen misst 10 Einheiten. Zu Beginn befindet sich die Turtle im eingezeichneten Punkt und schaut in Richtung des Pfeiles.

