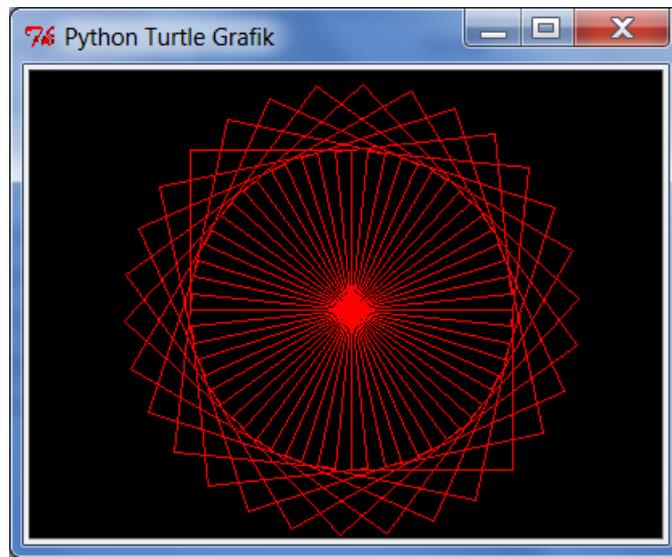


2. Aufgabe: Zeichne mit der Turtle folgendes Bild:



Was vermutest du, wie entsteht das Muster?
Es sieht komplizierter aus, als es ist ... ;)

Der Algorithmus (die Lösungsidee):

Das Quadrat wird gezeichnet
die Ausrichtung der Turtle wird etwas gedreht
das Quadrat wird gezeichnet
die Ausrichtung der Turtle wird etwas gedreht
das Quadrat wird gezeichnet

...usw.

Die Programmierung

Damit wir die Turtle benutzen können, brauchen wir auf jeden Fall am Anfang:

```
from turtle import*           # importiere die Turtle-Funktionen
shape („turtle“)             # nimm die Turtle (=Schildkröten) -Form als Ausgabe
```

Das Problem: Wenn wir viele Quadrate zeichnen wollen, wird das Skript sehr lang.

Die Lösung: Funktion definieren

Deshalb schnüren wir das Quadrat zu einer Funktion zusammen.

Wir brauchen dann nur noch „quadrat()“ schreiben, damit alle Befehle, die eingerückt unter „def quadrat():“ stehen, ausgeführt werden.

```
def quadrat():                # "quadrat()" soll die neue Abkürzung für die eingerückten Anweisungen darunter
                               sein
    fd(100)
    lt(90)
    fd(100)
    lt(90)
    fd(100)
    lt(90)
    fd(100)
    lt(90)
quadrat()                     # zeichne ein Quadrat, wie es oben definiert ist
```

Vergleiche dein Ergebnis mit:

→ **02a_spitzendeckchen_Funktion_definieren.py**

Jetzt können wir bequem viele Quadrate zeichnen. Für das „Spitzendeckchen“ muss nach jedem Quadrat die Turtle etwas gedreht werden. Schreibe ein Programm dafür und probiere es aus!

Vergleiche dein Ergebnis mit:

→ **02a_spitzendeckchen_drehen.py**

Wie viele Drehungen hast du geschafft?

Das Problem: Wenn ich 30 Drehungen einzeln aufschreiben soll, wird das Skript immer noch sehr lang. Kann ich nicht einfach sagen, dass 30mal ein Quadrat gezeichnet und gedreht werden soll?

Die Lösung: Schleife programmieren

Dieser Befehl kann helfen:

```
for i in range (30):           # wiederhole die nachfolgenden, eingerückten Befehle 30mal
```

Probiert es aus! Benutzt wieder die Tabulator-Taste zum Einrücken der Befehle.

Vergleiche dein Ergebnis mit

→ **02c_spitzendeckchen_Schleife.py**

Die 30 Quadrate sind in dieser Variante noch nicht schön gleichmäßig auf den ganzen Kreis verteilt. Wie bekomme ich heraus, um welchen Winkel sich die Turtle drehen muss, damit wir 30 Quadrate im Kreis unterbekommen?

Ich lasse das einfach von Python ausrechnen!

Ein Operator zum Ausrechnen ist das Zeichen /

```
100 / 10                       #teile 100 durch 10
```

Weißt du, wie groß der Innenwinkel eines Kreises ist? Dann ist es nicht mehr schwer, alle 30 Quadrate gleichmäßig darin zu verteilen.

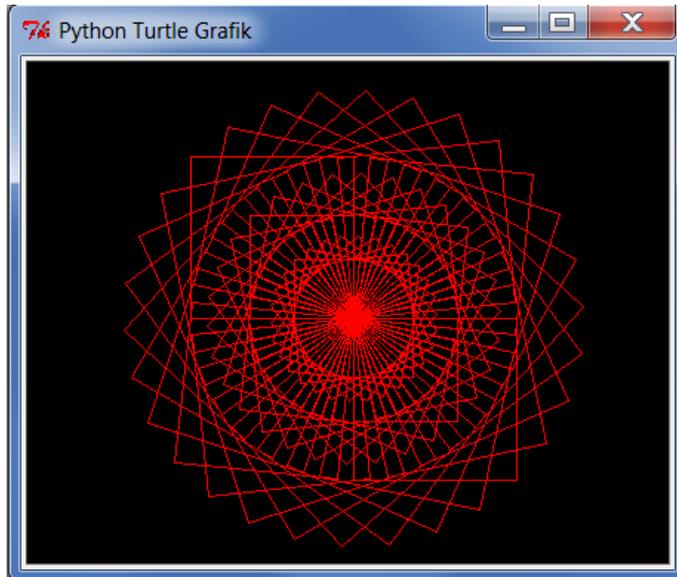
Mit „speed“ können wir die Turtle noch schneller laufen lassen, damit das Zeichnen nicht ewig dauert:

```
speed (10)                     # setze die Geschwindigkeit auf 10 (oder einen anderen Wert zwischen 0 und 10)
```

Bist du auf die Lösung gekommen? Dann vergleiche dein Ergebnis mit

→ **02d_spitzendeckchen_Schleife_drehwinkel.py**

Darf's noch ein bisschen komplizierter sein? Dann probiert doch mal Folgendes aus:



Was vermutest du: Wie entsteht das Muster?

Richtig, wir brauchen dafür Quadrate mit verschiedenen Seitenlängen. Lass deine Turtle verschieden lange Strecken laufen!

Und vergleiche dein Ergebnis mit

→ **02e_spitzendeckchen_drehwinkel_mehrfach.py**

Vielleicht möchtest du das Ganze noch etwas bunter gestalten? Du erinnerst dich noch, wie das geht – oder?

Zum Beispiel so:

→ **02f_spitzendeckchen_bunter.py**