

1. Quadrate

Aufgabe:

1. Zeichne mit Hilfe der Turtle ein Quadrat.
2. Verändere das Aussehen des Quadrates, z.B. durch Anpassen von Größe, Stifffarbe und -größe.

Zusatzaufgabe: Lass die Turtle nacheinander drei unterschiedlich aussehende Quadrate zeichnen.

Tip:

Die Turtle läuft immer in die Richtung, in die ihr Kopf zeigt.



Im Glossar

Quadrat zeichnen

Die Befehle zum Zeichnen des Quadrats findest du in der Kategorie „Bewegung“

Aussehen des Quadrats

Die Befehle zum Verändern der Stifffarbe und -größe findest du in der Kategorie „Aussehen“

Zum Starten:

1. Öffne „IDLE (Python GUI)“, wähle „File“ > „New File“. Es öffnet sich ein neues Fenster.
2. Wähle hier „File“ > „Save as“ und speichere dein Projekt unter einem aussagekräftigen Namen ab.

Beispiellösungen

```
from turtle import*  
shape("turtle")
```

Wichtig

Beginne jedes Skript mit den folgenden beiden Befehlen

```
 pencolor("green") #setze Stifffarbe auf grün  
 pensize(2) #setze Stiftgröße auf 2
```

```
 fd(100) #gehe 100 Pixel vorwärts  
 lt(90) #drehe dich um 90° nach links  
 fd(100)  
 lt(90)  
 fd(100)  
 lt(90)  
 fd(100)  
 lt(90)
```

Quadrat

Um ein Quadrat zu zeichnen, muss die Turtle jeweils vier mal vorwärts gehen und sich um 90° (nach links oder rechts) drehen

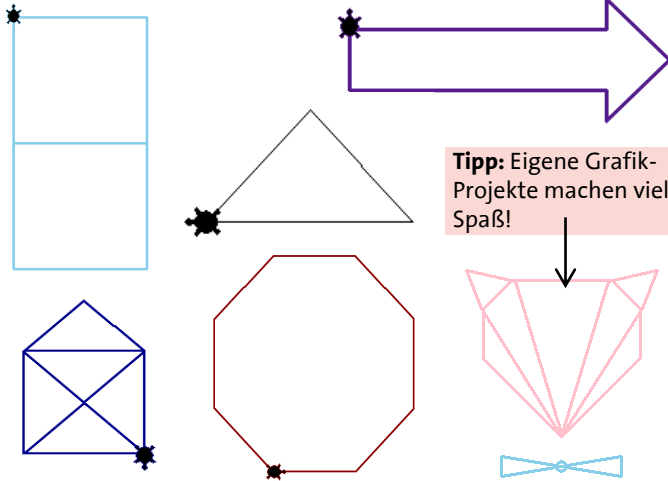
Dein Programm testen

- Wähle „Run“ > „Run module“
alternativ: Drücke Taste f5
- Bestätige das Speichern der vorgenommenen Änderungen
- Das Grafik- Fenster öffnet sich und zeigt dein Programm

2. Geometrische Formen

Aufgabe:

1. Zeichne mit Hilfe der Turtle die abgebildeten Formen.
2. Überlege dir ein eigenes Motiv, was du mit der Turtle zeichnest (z.B. dein eigener Name, ein Tier o.Ä.).



Tipp: Eigene Grafik-Projekte machen viel Spaß!

Im Glossar

Formen zeichnen

Die Befehle zum Zeichnen der Formen findest du in der Kategorie „Bewegung“

Aussehen der Formen

Die Befehle zum Verändern des Aussehens findest du in der Kategorie „Aussehen“

Ein neues Projekt anlegen:

Lege für jede Form ein neues Projekt an. Dazu gehst du im Shell Fenster auf „File“ > „New File“. Es öffnet sich ein neues Fenster. Wähle hier „File“ > „Save as“ und speichere dein Projekt unter einem aussagekräftigen Namen ab.

Beispiellösung

```
from turtle import*  
shape("turtle")
```

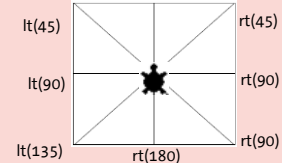
```
fd(100)  
lt(90)  
fd(10)  
rt(135)  
fd(50)  
rt(90)  
fd(50)  
rt(135)  
fd(10)  
lt(90)  
fd(100)  
rt(90)  
fd(50)
```

```
#gehe 100 Pixel vorwärts  
#drehe dich um 90° nach links  
#gehe 10 Pixel vorwärts  
#drehe dich um 135° nach rechts
```



Winkel

Winkel bemessen sich immer aus der Perspektive des Schildkrötenkopfes



Fehler finden

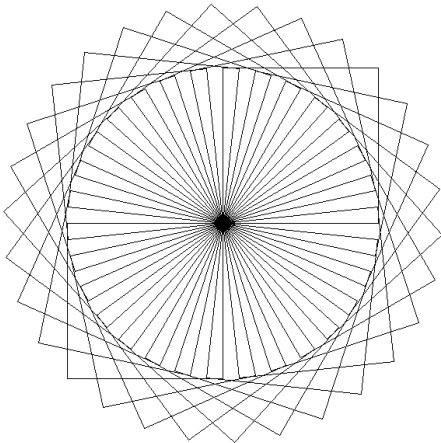
- Beginnt dein Skript mit dem Befehl `from turtle import*
shape ("turtle")`
- Hast du alles richtig geschrieben?
- Hast du alle Zeichen und Klammern richtig gesetzt?
- Hast du alle Leerzeichen richtig gesetzt?

3. Mandalas

Aufgabe:

1. Zeichne mit Hilfe der Turtle dieses Mandala.
2. Vereinfache den Code mit Hilfe einer Funktion und einer Schleife auf max. 14 Zeilen.

Zusatzaufgabe: Schaffst du es auch mit 9 Zeilen Code?



Tip:

Dieses Mandala setzt sich aus 30 Quadraten zusammen, die um einen bestimmten Winkel gedreht werden, so dass ein Kreis entsteht. Wie groß ist der Dreh- Winkel?

Im Glossar

Funktion und Schleife

Die notwendigen Befehle findest du in der Kategorie „Funktionen und Steuerung“.

Funktionen und Schleifen:

Mehrere Befehle können in einer Funktion zusammengefasst werden. Sollen Befehle häufiger wiederholt werden, dann werden sie in eine Schleife geschrieben

Beispiel-Lösungen

```
from turtle import*  
shape("turtle")
```

```
def quadrat(): # definiere eine Funktion namens quadrat, bestehend  
              # aus den 8 nachfolgenden Befehlen
```

```
    fd(100)  
    lt(90)  
    fd(100)  
    lt(90)  
    fd(100)  
    lt(90)  
    fd(100)  
    lt(90)
```

```
for i in range(30): # Schleife: wiederhole folgende Befehle 30mal  
    quadrat()      # führe Funktion quadrat aus  
    lt(12)         # drehe dich um 12° nach links
```

Lösung der Zusatzaufgabe

Ein Mandala in 9 Zeilen Code

Um den Code noch weiter zu vereinfachen, kannst du in die Funktion des Quadrats ebenfalls eine Schleife integrieren

```
def quadrat():  
    for i in range(4):  
        fd(100)  
        lt(90)
```

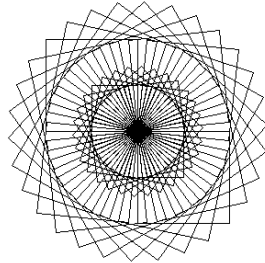
4. Komplexe Mandalas

Aufgabe:

1. Zeichne 10 Quadrate mit unterschiedlichen Seitenlängen. Definiere dazu einen Parameter.
2. Nutze den Parameter, um ein komplexes Mandala zu zeichnen.

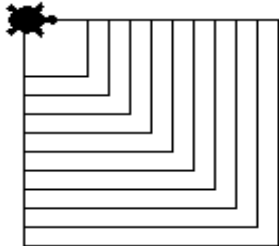
Tipp:

Lege für den Parameter „seitenlänge“ einen beliebigen Wert fest. In einer Schleife soll der Wert für die Seitenlänge immer mehr zunehmen.



Tipp:

Dieses komplexe Mandala setzt sich aus zwei Mandalas mit unterschiedlichen Seitenlängen zusammen.



Im Glossar

Parameter

Die Befehle zur Definition eines Parameters findest du im Glossar unter „Steuerung und Funktionen“

Parameter definieren

Man teilt einem beliebigen Wort oder Buchstaben einen beliebigen Zahlenwert zu. Der Name kann beim Zeichnen verwendet werden. Der Wert des Parameters kann im Laufe des Programms geändert werden

Beispiel-Lösungen

```
from turtle import*  
shape("turtle")
```

```
seitenlänge=30
```

#Parameter: dem Wort seitenlänge wird der Wert 30 zugeordnet

```
def quadrat():  
    for i in range(4):  
        fd(seitenlänge)  
        lt(90)
```

gehe vorwärts um den Wert von Parameter

```
for i in range(10):  
    quadrat()  
    seitenlänge=seitenlänge+10 # addiere 10px zum Parameterwert von seitenlänge hinzu
```

Lösung der Zusatzaufgabe

```
def quadrat():  
    for i in range(4):  
        fd(seitenlänge)  
        lt(90)
```

Wiederhole die innere Schleife zweimal mit unterschiedlichen Werten für Parameter Seitenlänge

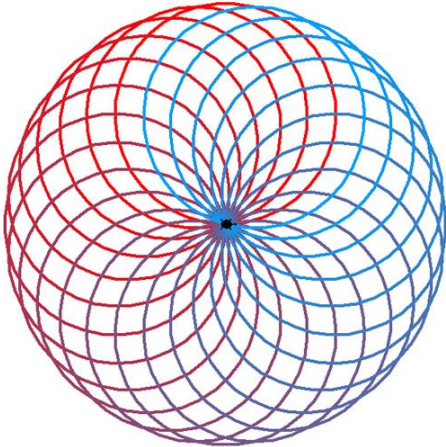
```
for i in range(2):  
    for i in range(30):  
        quadrat()  
        lt(12)  
    seitenlänge=seitenlänge+10
```



5. Kreismandalas mit Farbverläufen

Aufgabe:

1. Zeichne ein Kreismandala mit einem Farbverlauf.
Nutze hierfür die RGB-Farben



Tipp:

Arbeite mit den RGB Farben!

Nutze dazu die Befehle
`colormode(255)`
`pencolor(r,g,b)`

Definiere dann die Startwerte für die Parameter rot, grün und blau.

Im Glossar

Kreise zeichnen

Die Befehle zum Zeichnen von Kreisen findest du in der Kategorie „Bewegung“

Farbverläufe erstellen

Die Befehle findest du in der Kategorie „Aussehen“ und „Steuerung & Funktionen“

RGB Farbsystem:

RGB ist ein additives Farbsystem. Die Stiftfarbe wird auf einen Wert für rot grün und blau zwischen 0 und 255 festgesetzt. Über Parameter lassen sich die Farbwerte über die Zeit verändern.

Beispiel-Lösungen

```
from turtle import*  
shape("turtle")
```

```
colormode(255) #setze den colormode auf 255
```

```
r=255 #setze den Startwert für rot auf 255  
g=0 #setze den Startwert für grün auf 0  
b=0 #setze den Startwert für blau auf 0
```

```
for i in range(24):  
    pencolor(r,g,b) # nutze r,g,b Werte als Stiftfarben  
    circle(150) #zeichne einen Kreis mit dem Durchmesser 150px  
    lt(15)  
    r=r-11 #subtrahiere 11 zum Wert von rot  
    g=g+7 #addiere 7 zum Wert von grün  
    b=b+11 #addiere 11 zum Wert von blau
```

Tipp: Maximalwerte beachten

Durch den `colormode(255)` ist festgelegt, dass die Farb-Parameter einen max. Wert von 255 annehmen können. Beachtet dies bei deinen Rechenoperationen in der Schleife.