

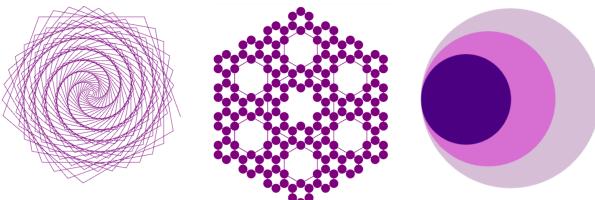
Einstieg in Python mit Turtlegrafik

Für den Einstieg ins Programmieren verwenden wir **Turtlegrafik**.

Das ist eine einfache Art, mit Python **Zeichnungen zu erstellen**. Dabei steuert man eine kleine Schildkröte (engl. turtle) auf dem Bildschirm mit Befehlen in Python.

Die Schildkröte:

- kann sich vorwärts und rückwärts bewegen,
- kann sich drehen
- und hinterlässt beim Bewegen eine Linie.



```
from gturtle import *
setFillColor("purple")
startPath()
repeat 5:
    forward(200)
    right(144)
fillPath()
```



Turtlegrafik ist **keine eigene Programmiersprache**, sondern ein sogenanntes **Modul** für Python.

Ein Modul ist eine für einen bestimmten Zweck gedachte **Erweiterung** für Python, d.h. eine Sammlung von **zusätzlichen Befehlen**.

Python selbst kann schon sehr viel und mit Modulen kann man Python noch weiter ausbauen.

1. Turtle bewegen

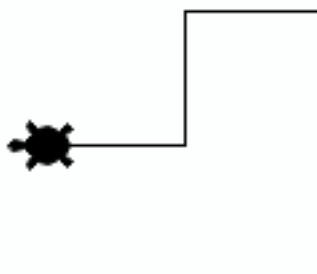
Ein **Programm** besteht aus einer Folge von **Programmzeilen**, die der Reihe nach (als Sequenz) abgearbeitet werden. Mit dem Turtle-Modul können wir eine kleine Schildkröte mit **Befehlen** steuern und sehen die dadurch erzeugten Bewegungen im **Grafikfenster**.

Turtlebefehle werden grundsätzlich Englisch geschrieben und enden immer mit einem **Klammerpaar**, das man **Parameterklammer** nennt. Dieses kann weitere Angaben für den Befehl enthalten. Selbst wenn keine Angaben nötig sind, muss in Python ein **leeres Klammerpaar** vorhanden sein. Es ist ebenfalls wichtig, die **Gross-/Kleinschreibung** exakt einzuhalten.

Bei der Bewegung hinterlässt die Schildkröte eine Spur. Es ist so, als ob sie einen auf der Zeichenfläche aufliegenden Zeichenstift (pen) mit sich tragen würde. Damit kann man beliebige Figuren zeichnen.

Bevor die Schildkröte loslegen kann, muss man den Computer anweisen, die Turtlebefehle aus dem Modul bereit zu stellen. Dazu schreibt man den Befehl `from gturtle import *` als erste Zeile ins Editorfenster.

Mit dem Befehl `forward(100)` bewegt sich die Schildkröte um 100 Schritte vorwärts, mit `left(90)` dreht sie sich um 90 Grad nach links und mit `right(90)` dreht sie um 90 Grad nach rechts. Mit diesen Befehlen können wir die Schildkröte steuern.



Erste Turtle-Zeichnung

```
from gturtle import *
forward(100)
left(90)
forward(50)
left(90)
forward(50)
right(90)
forward(50)
```

[Programm öffnen](#)

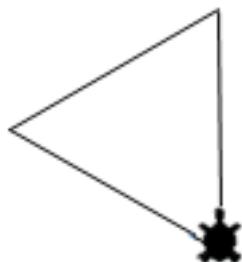
Beispiele ausprobieren

Um die Befehle zu lernen und das Programmieren zu üben ist es ratsam, wenn Sie die Beispiele jeweils in WebTigerPython ausprobieren. Klicken Sie dazu jeweils auf den bereitgestellten Link und experimentieren Sie.

Leere Zeilen im Code (wie die zweite Zeile im Beispiel) werden in Python **ignoriert** und dienen nur der besseren Lesbarkeit des Programms.

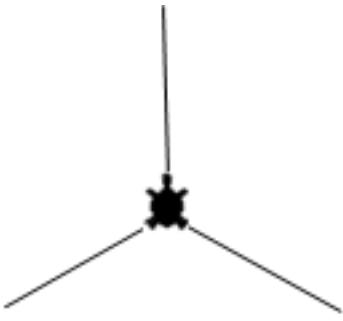
Aufgabe 1

Zeichnen Sie ein gleichseitiges Dreieck mit der Seitenlänge 100. Finden Sie den richtigen Drehwinkel?



Aufgabe 2

Die Schildkröte kann sich auch rückwärts bewegen. Soll sie beispielsweise 100 Schritte rückwärtsgehen, so verwendet man den Befehl `back(100)`. Zeichnen Sie die dargestellte Figur.



Tipp für die Aufgaben

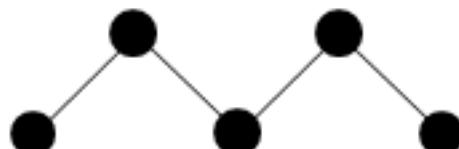
Speichern Sie Ihre Lösungen jeweils als **Python-Dateien mit sinnvollen Namen** auf Ihrem Computer ab, damit Sie sie wieder finden.

Alternativ können Sie den Code von Ihren Lösungen ins **OneNote kopieren**.

Beginnen Sie jede Aufgabe mit einem **leeren Editorfenster**.

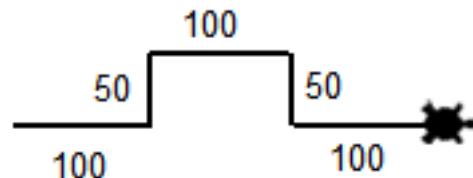
Aufgabe 3

Mit dem Befehl `dot(20)` zeichnet die Schildkröte dort, wo sie sich gerade befindet, einen gefüllten Kreis mit dem Durchmesser 20. Zeichnen Sie die folgende Figur.



Aufgabe 4

Zeichnen Sie die abgebildete Figur mit den angegebenen Längen.



2. Farben verwenden

Die Turtlegrafik verwendet die sogenannten **X11-Colors**. Dabei handelt es sich um eine **vordefinierte Liste von Farbnamen** die ursprünglich aus dem X Window System (X11) stammen. Jede Farbe hat einen fest zugeordneten RGB-Wert. Sie können die Farben im Internet unter w3schools.com/colors/colors_x11.asp nachschlagen. Einige Beispiele sind: *yellow, gold, orange, red, maroon, violet, magenta, purple, navy, blue, skyblue, cyan, turquoise, blueviolet, lime, green, darkgreen, chocolate, brown, black, gray, white*, und noch viele mehr.



Kerze zeichnen

```
1 from gturtle import *
2 setPenWidth(60)
3 setPenColor("red")
4 forward(100)
5 penUp()
6 forward(50)
7 penDown()
8 setPenColor("yellow")
9 dot(40)
10 setPenWidth(5)
11 setPenColor("black")
12 back(15)
13 hideTurtle()
```

[Programm öffnen](#)

Wir wollen eine brennende rote Kerze zeichnen. Zuerst zeichnet man mit einem dicken roten Stift die Kerze selbst. Dann fährt man mit abgehobenem Stift ein wenig weiter und zeichnet mit dem Befehl `dot(40)` einen gelbgefüllten Kreis. Um den Docht zu malen, fährt man mit einem schwarzen, dünnen Stift leicht zurück.

Dazu brauchen wir neue Befehle: `setPenColor()` für die Stiftfarbe, `setPenWidth()` für die Stiftdicke, `penUp()` und `penDown()` um den Stift hochzuheben und ihn wieder abzusenken.

Zuletzt verstecken wir noch mit `hideTurtle()` das Bild der kleinen Schildkröte.

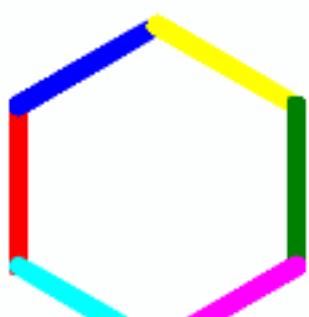
Bei den **Farbnamen** spielt die Gross-/Kleinschreibung keine Rolle, sie müssen aber in Anführungszeichen stehen.

Aufgabe 5

Ergänzen Sie das untenstehende Programm so, dass die Schildkröte ein regelmässiges Sechseck zeichnet und wählen Sie für jede Seite eine andere Farbe.

```
1 from gturtle import *
2 setPenWidth(5)
3 setPenColor("red")
4 forward(80)
5 right(60)
6 ...
```

[Programm öffnen](#)



Aufgabe 6

Zeichnen Sie gefüllte Kreise mit den Farben "red", "yellow", "blue" und "magenta". Zeichnen Sie weitere drei Kreise mit Farben Ihrer Wahl, die Sie unter w3schools.com/colors/colors_x11.asp finden.



Aufgabe 7

Zeichnen Sie eine Verkehrsampel. Das schwarze Rechteck können Sie mit der Stiftbreite 80 zeichnen.



3. Wiederholungen

Bei den letzten Aufgaben haben wir gemerkt, dass wir häufig dieselben Befehle nacheinander wiederholen. Wir können nun eine oder mehrere Programmzeilen zu einem Programmblöck zusammenfassen und ihn dann **eine bestimmte Anzahl mal wiederholt durchlaufen** lassen. Dadurch erspart man sich viel Schreibarbeit und das Programm wird übersichtlicher.

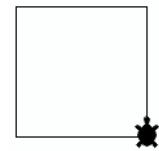
Um ein Quadrat zu zeichnen, muss die Schildkröte vier Mal die Befehle `forward(100)` und `left(90)` ausführen. Dies kann man in WebTigerPython elegant mit dem Befehle `repeat` programmieren.

Die Wiederholung wird mit `repeat 4:` eingeleitet. Dabei ist der **Doppelpunkt** sehr wichtig. Vergisst man ihn, so ergibt sich bei der Programmausführung eine Fehlermeldung.

Die Befehle im Programmblöck nach `repeat 4:` muss man alle **gleichweit einrücken**. Dazu verwendet man immer 4 Leerschläge, man kann aber auch die Tabulator-Taste brauchen, um sie zu erzeugen. Man spricht bei der Wiederholstruktrur auch vom **Durchlaufen einer Schleife**.

 **Das Einrücken ist sehr wichtig**

Alles was wiederholt werden muss, kommt direkt nach `repeat 4:` und muss **gleichweit eingerückt** sein. Alles was danach **nicht mehr eingerückt** ist, wird **nicht wiederholt**.



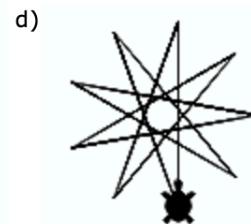
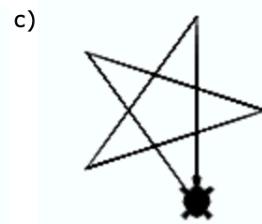
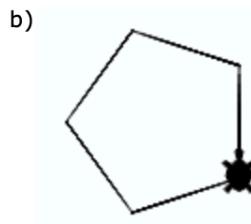
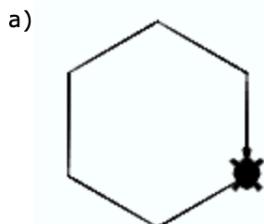
Viereck mit repeat

```
1 from gturtle import *
2 repeat 4:
3     forward(100)
4     left(90)
```

[Programm öffnen](#)

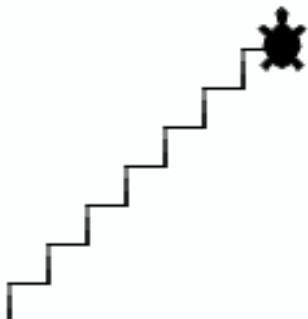
Aufgabe 8

Experimentieren Sie mit dem Programm aus dem Viereck-Beispiel. Ändern Sie die Anzahl Wiederholungen und den Drehwinkel so, dass die Turtle die folgenden Figuren zeichnet.



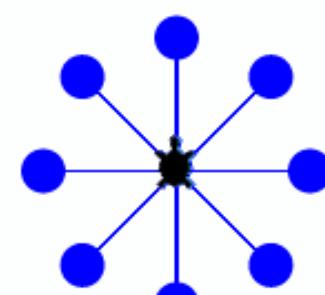
Aufgabe 9

Zeichnen Sie eine Treppe mit 7 Stufen.



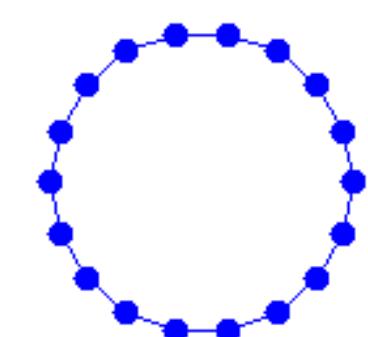
Aufgabe 10

Zeichnen Sie die angegebene Figur. Dazu brauchen Sie auch die Befehle `back()` und `dot()` und die Stiftfarbe "blue".



Aufgabe 11

Zeichnen Sie eine Perlenkette, die aus 18 Perlen (dots) besteht. Zwischen den Perlen muss die Schildkröte jeweils einige Schritte vorwärts gehen und um einen kleinen Winkel (z.B. 20°) nach links drehen.



Verschachtelte Schleifen

Richtig spannend und anspruchsvoll wird es, wenn man zwei **repeat-Strukturen ineinander verschachtelt**. Dabei muss man immer denken, dass zuerst die «innere», weiter eingerückte Wiederholstruktur durchlaufen wird, bevor sich die «äussere», weniger eingerückte Struktur wiederholt.

Verschachtelte Schleifen  Py

```

1 from gturtle import *
2 repeat 7:
3     repeat 4:
4         forward(30)
5         right(90)
6         forward(30)

```

[Programm öffnen](#)

In diesem Beispiel zeichnet die innere Schleife ein einzelnes Quadrat und die Schildkröte befindet sich nachher wieder in der linken unteren Ecke des Quadrats. Sie wird dann in der äusseren Schleife vorgeschoben und das Quadrat wird erneut gezeichnet.

 **Wichtig**

Versuchen Sie diesem Code Schritt für Schritt zu folgen und zu verstehen, was genau passiert!



Aufgabe 12

Versuchen Sie zuerst auf einem Blatt Papier herauszufinden, was das folgende Programm zeichnet. Lassen Sie es dann laufen, um Ihre Vermutung zu bestätigen.

 Py

```

1 from gturtle import *
2
3 repeat 5:
4     repeat 4:
5         forward(100)
6         right(90)
7         left(36)

```

[Programm öffnen](#)

 **Auftrag**

Überlegen Sie sich eine **kleine Zeichnung**, die Sie mit der Schildkröte programmieren möchten. Dies kann ein Buchstabe, ein Symbol, eine Flagge oder irgendeine kleine kreative Form sein.

- **Programmieren** Sie die Zeichnung in WebTigerPython.
- Erstellen Sie einen **Screenshot** von der fertigen Zeichnung.
- **Senden** Sie mir den **Screenshot** und Ihren **Code** via **Teams**.

★ Figuren füllen (freiwilliges Zusatzkapitel)

Die Schildkröte kann sich an ihre Bewegung erinnern, um so eine von ihr gezeichnete **geschlossene Figur mit einer Farbe zu füllen**.

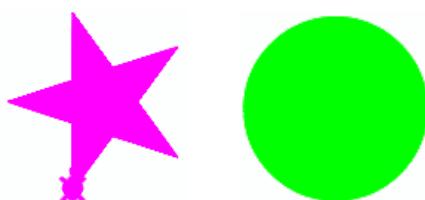
Um eine Figur auszufüllen, sagen wir zuerst der Schildkröte mit `startPath()`, dass sie sich ausgehend vom aktuellen Ort die nachfolgend gezeichnete Figur merken soll. Mit dem Befehl `fillPath()` wird der jetzige Ort mit dem Startort verbunden und die eingeschlossene Fläche ausgefärbt. Mit `setFillColor()` kann man die Füllfarbe angeben (sagt man nichts, so ist sie standardmäßig schwarz).



Ausgefüllter Stern Py

```
1 from gturtle import *
2 setFillColor("magenta")
3 startPath()
4 repeat 5:
5     forward(160)
6     right(144)
7 fillPath()
```

[Programm öffnen](#)



Wir können mit der Schildkröte einen gefüllten Kreis als Vieleck mit sehr vielen Seiten zeichnen.

Kreis als Vieleck Py

```
1 from gturtle import *
2 setFillColor("lime")
3 startPath()
4 repeat 120:
5     forward(3)
6     right(3)
7 fillPath()
8 hideTurtle()
```

[Programm öffnen](#)

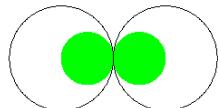
Aufgabe 13

Um den dargestellten 6er-Stern zu zeichnen, dreht die Schildkröte abwechselungsweise 140 und 80 Grad. Zeichnen Sie diese Figur.



Aufgabe 15

Diese lustige Figur besteht aus gefüllten und nicht gefüllten Kreisen. Zeichnen Sie sie wie im Beispiel mit Vielecken.



Aufgabe 17

Zeichnen Sie das rote Kreuz mit zwei repeat-Schleifen.



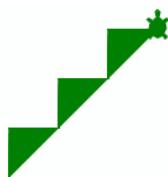
Aufgabe 14

Zeichnen Sie zwei gefüllte Halbkreise. Wenn Sie die Schildkröte vor dem Zeichnen mit `hideTurtle()` verstecken, wird das Bild viel schneller gezeichnet.



Aufgabe 16

Zeichnen Sie das angegebene Bild der gefüllten Treppe.



Lösungen

Aufgabe 1

```
1 from gturtle import * [2]
2 forward(100)
3 left(120)
4 forward(100)
5 left(120)
6 forward(100)
7 left(120)
```

[Programm öffnen](#)

Aufgabe 2

```
1 from gturtle import * [2]
2 forward(100)
3 back(100)
4 left(120)
5 forward(100)
6 back(100)
7 left(120)
8 forward(100)
9 back(100)
10 left(120)
```

[Programm öffnen](#)

Aufgabe 3

```
1 from gturtle import * [2]
2 dot(20)
3 right(45)
4 forward(100)
5 dot(20)
6 right(90)
7 forward(100)
8 dot(20)
9 left(90)
10 forward(100)
11 dot(20)
12 right(90)
13 forward(100)
14 dot(20)
15 left(90)
```

[Programm öffnen](#)

Aufgabe 4

```
1 from gturtle import * [2]
2 right(90)
3 forward(100)
4 left(90)
5 forward(50)
6 right(90)
7 forward(100)
8 right(90)
9 forward(50)
10 left(90)
11 forward(100)
```

[Programm öffnen](#)

Aufgabe 5

```
1 from gturtle import * [2]
2 setPenWidth(5)
3 setPenColor("red")
4 forward(80)
5 right(60)
6 setPenColor("blue")
7 forward(80)
8 right(60)
```

```
9 setPenColor("yellow")
10 forward(80)
11 right(60)
12 setPenColor("green")
13 forward(80)
14 right(60)
15 setPenColor("magenta")
16 forward(80)
17 right(60)
18 setPenColor("cyan")
19 forward(80)
20 right(60)
21 hideTurtle()
```

[Programm öffnen](#)

Aufgabe 6

```
1 from gturtle import * [2]
2 right(90)
3 setPenColor("red")
4 dot(60)
5 forward(60)
6 setPenColor("yellow")
7 dot(60)
8 forward(60)
9 setPenColor("blue")
10 dot(60)
11 forward(60)
12 setPenColor("magenta")
13 dot(60)
14 forward(60)
15 setPenColor("cyan4")
16 dot(60)
17 forward(60)
18 setPenColor("chocolate")
19 dot(60)
20 forward(60)
21 setPenColor("blueviolet")
22 dot(60)
23 hideTurtle()
```

[Programm öffnen](#)

Aufgabe 7

```
1 from gturtle import * [2]
2 setPenWidth(80)
3 forward(180)
4 right(180)
5 penUp()
6 forward(10)
7 penDown()
8 setPenColor("red")
9 dot(60)
10 penUp()
11 forward(80)
12 penDown()
13 setPenColor("yellow")
14 dot(60)
15 penUp()
16 forward(80)
17 penDown()
18 setPenColor("lime")
19 dot(60)
20 hideTurtle()
```

[Programm öffnen](#)

Aufgabe 8

Aufgabe a

```
1 from gturtle import *
2 repeat 6:
3   forward(100)
4   left(60)
```

[Programm öffnen](#)

Aufgabe b

```
1 from gturtle import *
2 repeat 5:
3   forward(100)
4   left(72)
```

[Programm öffnen](#)

Tipp für c) und d): Die Schildkröte «überspringt» Ecken → damit addieren sich die Winkel von allen überprungenen Ecken.

Aufgabe c

```
1 from gturtle import *
2 repeat 5:
3   forward(160)
4   left(144)
```

[Programm öffnen](#)

Aufgabe d

```
1 from gturtle import *
2 repeat 9:
3   forward(160)
4   left(160)
```

[Programm öffnen](#)

Aufgabe 9

```
1 from gturtle import * [2]
2 repeat 7:
3   forward(20)
4   right(90)
5   forward(20)
6   left(90)
```

[Programm öffnen](#)

Aufgabe 10

```
1 from gturtle import * [2]
2 setPenColor("blue")
3 repeat 8:
4   forward(100)
5   dot(20)
6   back(100)
7   right(45)
```

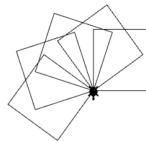
[Programm öffnen](#)

Aufgabe 11

```
1 from gturtle import * [2]
2 setPenColor("blue")
3 right(90)
4 repeat 18:
5   dot(15)
6   right(20)
7   forward(30)
8 hideTurtle()
```

[Programm öffnen](#)

Aufgabe 12



Aufgabe 13

```
1 from gturtle import * [2]
2 hideTurtle()
3 setFillColor("yellow")
4 startPath()
5 repeat 6:
```

```
6   forward(80)
7   right(140)
8   forward(80)
9   left(80)
10 fillPath()
```

[Programm öffnen](#)

Aufgabe 14

```
1 from gturtle import * [2]
2 setFillColor("red")
3 hideTurtle()
4 startPath()
5 repeat 180:
6   forward(1)
7   right(1)
8 fillPath()
9 startPath()
10 repeat 180:
11   forward(1)
12   left(1)
13 fillPath()
```

[Programm öffnen](#)

Aufgabe 15

```
1 from gturtle import * [2]
2 setFillColor("lime")
3 hideTurtle()
4 startPath()
5 repeat 360:
6   forward(1)
7   left(1)
8 repeat 360:
9   forward(1)
10  right(1)
11 fillPath()
12 repeat 360:
13  forward(2)
14  left(1)
15 repeat 360:
16  forward(2)
17  right(1)
```

[Programm öffnen](#)

Aufgabe 16

```
1 from gturtle import * [2]
2 setFillColor("green")
3 hideTurtle()
4 startPath()
5 repeat 3:
6   forward(50)
7   right(90)
8   forward(50)
9   left(90)
10 fillPath()
```

[Programm öffnen](#)

Aufgabe 17

```
1 from gturtle import * [2]
2 setFillColor("red")
3 hideTurtle()
4 startPath()
5 repeat 4:
6   repeat 3:
7     forward(80)
8     right(90)
9   left(180)
10 fillPath()
```

[Programm öffnen](#)