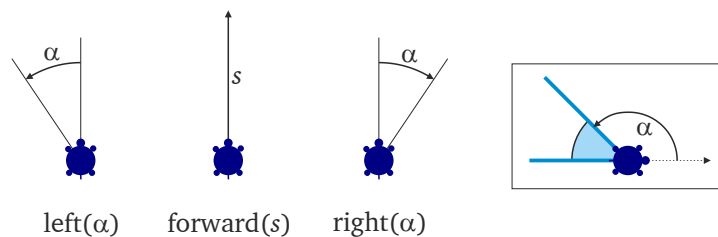


# Die Turtle steuern

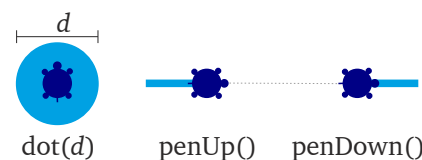
**Die Turtle laden steuern** Die Turtle kann geradeaus gehen oder sich an der aktuellen Position drehen. Zuerst muss das Turtle-Modul jeweils importiert und ein Turtle-Fenster erzeugt werden:

```
from gturtle import *      # <- Modul laden
makeTurtle()               # <- Fenster erzeugen

forward(100)               # <- Linien zeichnen ...
left(45)
```



**Abbildung 1:** Mit `forward(s)` geht die Turtle  $s$  Pixel vorwärts, mit `left(a)` bzw. `right(a)` drehst du die Turtle um den angegebenen Winkel  $a$  (in Grad). Achtung: Bei einer Ecke gibst du den *Aussenwinkel* an, um den sich die Turtle dreht (ganz rechts).



**Abbildung 2:** `dot(d)` zeichnet einen Punkt mit dem Durchmesser  $d$ . Mit `penUp()` und `penDown()` steuerst du, ob die Turtle eine Linie zeichnen soll.

**Farbe und Linienstärke** Die Farbe und Breite der Linien lässt sich mit `setPenColor("farbe")` (Gänsefüsschen nicht vergessen) bzw. mit `setPenWidth(breite)` angeben.

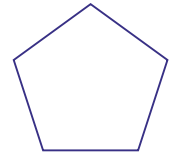
yellow	cyan	green
gold	blue	lime green
orange	navy	dark green
red	purple	white
dark red	magenta	gray
brown	sienna	black

**Abbildung 3:** Das ist eine Auswahl der Farben, die du mit `setPenColor("farbe")` auswählen kannst.

## Neue Befehle und Wiederholungen

**Wiederholungen** Codezeilen, die eingerückt unter einem `repeat n:` stehen werden nacheinander  $n$  Mal wiederholt.

**Beispiel: Ein Fünfeck zeichnen** Um ein Fünfeck zu zeichnen muss die Turtle 5 Mal das gleiche tun: Eine Strecke zeichnen und sich um  $72^\circ$  drehen. Das `hideTurtle()` wird nicht wiederholt.



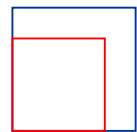
```
1 from gturtle import *
2 makeTurtle()
3
4 right(90)
5 repeat 5:
6     forward(100)
7     left(72)
8 hideTurtle()
```

**Die  $360^\circ$ -Regel** Woher weißt du, dass sich die Turtle beim Fünfeck jeweils um  $72^\circ$  drehen muss? Am Ende der Figur hat sich die Turtle um insgesamt  $360^\circ$  gedreht und in diesem Fall ist  $360^\circ : 5 = 72^\circ$ . Python kann die Division « $360/5$ » auch selber ausführen:

```
repeat 5:
    forward(100)
    left(360 / 5)
```

**Befehle definieren** Mit `def name():` wird ein neuer Befehl definiert, der danach mit `name()` aufgerufen und ausgeführt wird. In den Klammern dürfen auch Parameter stehen.

**Beispiel: Quadrate** In diesem Programm definieren wir den Befehl `quadrat` mit einer noch unbekannten Zahl (Parameter) `seite`. Danach zeichnen wir damit zwei Quadrate.



```
1 from gturtle import *
2 makeTurtle()
3
4 def quadrat(seite):
5     repeat 4:
6         forward(seite)
7         right(90)
8
9 quadrat(80)           # seite = 80
10 setPenColor("red")
11 quadrat(60)          # seite = 60
```

## AUFGABEN

1. Programmiere die drei Figuren aus der Abbildung 4 mit der Turtle.

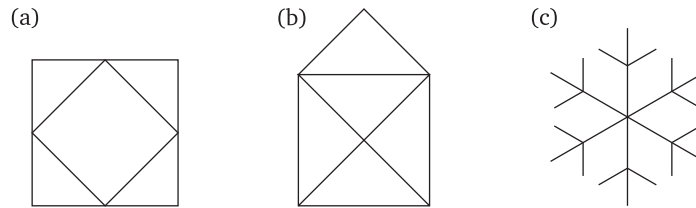


Abbildung 4: In der Mitte das «Haus von Nikolaus» und rechts eine einfache Schneeflocke.

2. Zeichne mit der Turtle einen fünfzackigen Stern oder ein Pentagramm wie in der Abbildung 5.

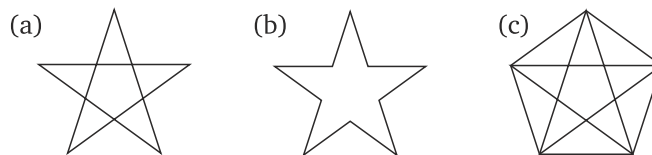


Abbildung 5: Fünfzackige Sterne mit dem Pentagramm.

3. Zeichne ein Schiff wie in der Abbildung 6 ganz links, oder eine Kirche, Lokomotive.

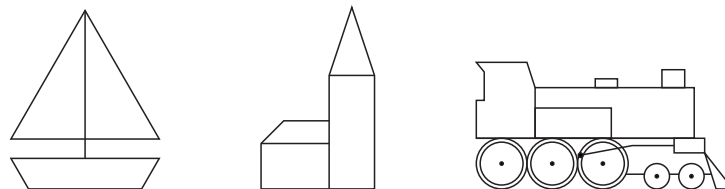


Abbildung 6: Figuren, die sich mit der Turtle zeichnen lassen.

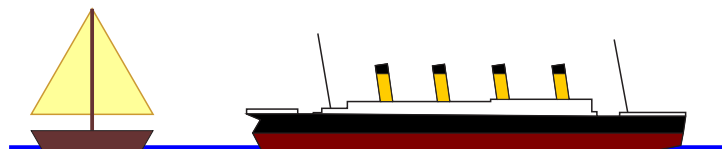


Abbildung 7: Fortgeschrittenere Varianten.