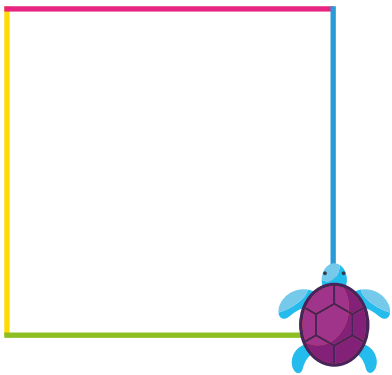
Programmiere ein Quadrat I

Lass die Turtle ein Quadrat mit einer Seitenlänge von 200 zeichnen.

Überlege, wie die Befehle dafür lauten und um wie viel Grad sich die Turtle an jeder Ecke drehen muss.

Ändere deinen Code so, dass alle Seiten eine andere Farbe haben.



Tipp:
Folgende Befehle benötigst du, um das Quadrat zu zeichnen:

♻ turtle → forward(200)
♻ turtle → right turn(90)
♻ turtle → setPenColor(blue)

Du kannst die Turtle natürlich auch links herum laufen lassen und eine andere Farbe verwenden.

Du brauchst einen kleinen Denkanstoß?

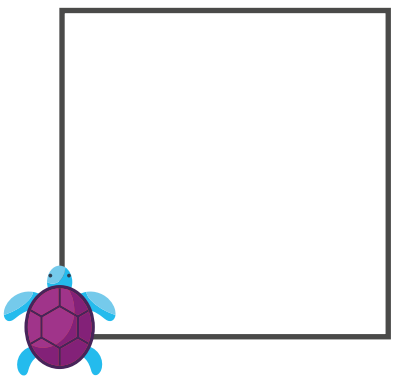
Lege das Quadrat doch einmal mit der Geheimschrift auf dem Boden aus oder laufe es im Raum ab.

Erklärung:
Mit einem Befehl sagst du der Turtle, was sie machen soll.

Für dein Portfolio 1 | A
Du kannst auch die richtigen Befehle erst einmal in dein Portfolio schreiben.

Programmiere ein Quadrat II

Lass die Turtle ein Quadrat mit einer Seitenlänge von 400 zeichnen. Aber Achtung: Die Turtle soll nicht aus dem Bild laufen.



Zusatzaufgabe: Lass die Turtle verschiedene Quadrate auf deinen Bildschirm malen. Nutze dazu verschiedene Seitenlängen und Farben.

Tipp:
Damit die Turtle nicht aus dem Bild läuft, kannst du den Stift anheben und die Turtle erst einmal laufen lassen ohne zu zeichnen:

Vergiss aber nicht, den Stift wieder abzusetzen!

Denke daran!
Du kannst ganz einfach die Länge der Seiten verändern, wenn du die Zahl in der Klammer hinter dem *forward*-Befehl änderst.

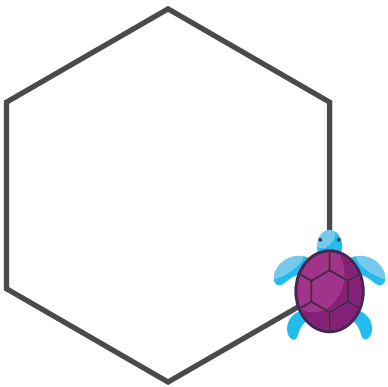
Suche mit anderen gemeinsam nach möglichen Lösungswegen. Zusammen schafft man noch viel mehr!

Für dein Portfolio 1 | B
Du hast es geschafft! Schreibe auf, wie du zu deiner Lösung gekommen bist, und klebe dein Quadrat in dein Portfolio.

Programmiere regelmäßige Vielecke I

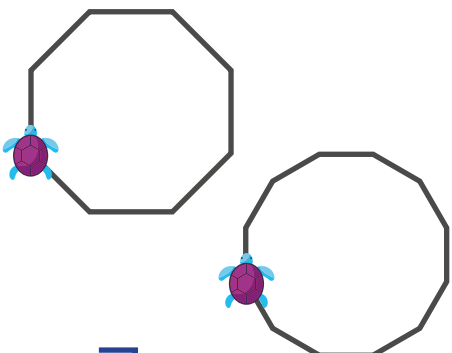
Lass die Turtle ein Quadrat mit einer Seitenlänge von 200 zeichnen.

Verändere deinen Code so, dass die Turtle nun ein regelmäßiges 6-Eck mit einer Seitenlänge von 200 zeichnet.



Zusatzaufgabe:
Lass die Turtle ein regelmäßiges 8-Eck, 12-Eck, 36-Eck und 360-Eck zeichnen.

Tipp:
Vom Quadrat zum 6-Eck oder zum 8-Eck gibt es einige Veränderungen. Du wirst mehr Befehlszeilen brauchen und auch die Zahlen werden anders sein.



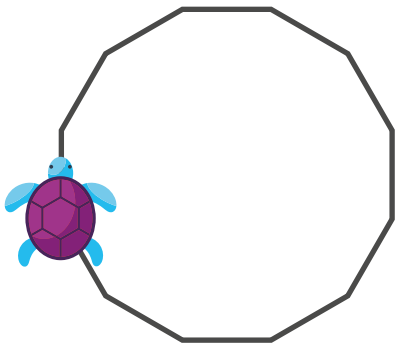
Du brauchst einen kleinen Denkanstoß?
Den passenden Winkel für verschiedene Vielecke kannst du einfach berechnen:

Die Turtle dreht sich insgesamt immer um 360 Grad. Teile dies durch die Anzahl der Ecken und du weißt, wie groß der Winkel an der Ecke ist.

Für dein Portfolio 2 | A
Das Portfolio kann dir dabei helfen, die Winkel zu berechnen.

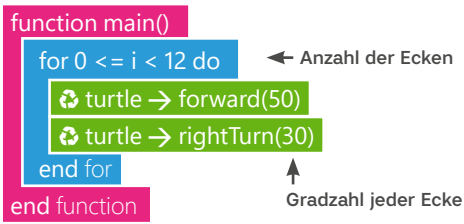
Programmiere regelmäßige Vielecke II

Lass die Turtle mit Hilfe einer Schleife ein 6-Eck mit einer Seitenlänge von 200 zeichnen.



Zusatzaufgabe: Zeichne mit Hilfe einer Schleife verschiedene regelmäßige Vielecke, z. B. ein 8-Eck, 12-Eck, 36-Eck oder 360-Eck.

Tipp:
So sieht z. B. ein 12-Eck mit einer Seitenlänge von 50 aus, das mit einer Schleife gezeichnet wurde:



Du brauchst einen kleinen Denkanstoß?
Dein 12-Eck passt nicht auf den Bildschirm? Verkleinere einfach die Zahl beim *forward*-Befehl.

Erklärung:
Mit einer Schleife kannst du eine Befehlsabfolge beliebig oft wiederholen.

Für dein Portfolio 2 | B & C
B: Die Schleife wirst du noch ganz oft benötigen. Dokumentiere deshalb für dich im Portfolio, was eine Schleife ist.

C: Klebe ein Vieleck in dein Portfolio.

Programmiere einen Mäander I

Lass die Turtle den abgebildeten Mäander mit einer Seitenlänge von 100 zeichnen.

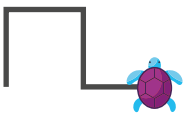
Denke auch daran, dass du mit der Schleife wiederkehrende Befehle wiederholen kannst.

Definiere eine Variable *laenge* mit der Länge 50. Ersetze dann alle Längenangaben deines Mäanders durch die Variable *laenge*.

Tipp:
Bevor du eine neue Variable verwenden kannst, musst du sie zuerst benennen und einen Wert für sie festlegen.

```
function main()
    var laenge :=50
    for 0 <= i < 4 do
        turtle → forward(laenge)
        turtle → rightTurn(90)
        turtle → forward(laenge)
```

Denke daran!
Da du eine Schleife benutzt, musst du nur ein Segment des Mäanders programmieren. Vergiss die letzte Drehung nicht!

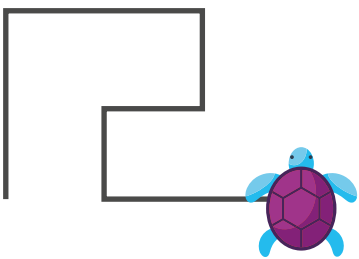


Erklärung:
Eine Variable ist ein Merkkasten, der für eine beliebige Zahl steht. Den Wert der Zahl kannst du verändern.

Für dein Portfolio 3 | A
Du kannst erst einmal mit der Geheimschrift überlegen, wie die einzelnen Befehle lauten könnten.

Programmiere einen Mäander II

Zeichne den abgebildeten Mäander. Die kurzen Seiten haben eine Länge von 50, die langen Seiten eine Länge von 100.



Zusatzaufgabe:
Denke dir deinen eigenen Mäander aus und programmiere ihn.

Tipp:
Verwende diesmal 2 Variablen. Nenne sie am besten so, dass du sie leicht erkennst, z. B. *lang* und *kurz*.

```
var lang :=100
var kurz :=50
```

Du brauchst einen kleinen Denkanstoß?
Falls du das Muster anfangs zu kompliziert findest, lauf es doch wieder erst einmal im Raum ab. So verstehst du besser, wie sich die Turtle drehen muss.

Für dein Portfolio 3 | B & C
B: Um die einzelnen Befehle besser zu verstehen, kannst du wieder mit der Geheimschrift arbeiten.

C: Du kannst auch einen Mäander in dein Portfolio kleben.

Programmiere einen komplizierten Mäander I

Lass die Turtle den abgebildeten Mäander mit 4 Wiederholungen zeichnen. Die kurzen Seiten haben eine Länge von 100, die langen Seiten eine Länge von 200.

Zusatzaufgabe: Programmiere den Mäander so, dass die lange Seite 150 beträgt, und finde heraus, wie lang die kurze Seite sein muss.

Tipp:
Der Code für den Mäander könnte so anfangen:

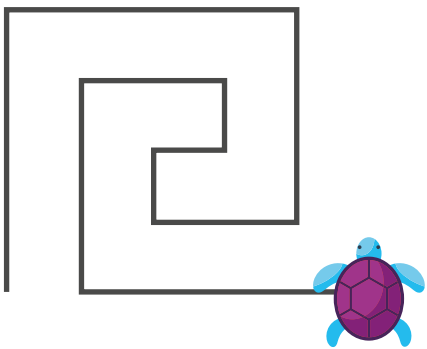
Denke daran!
Dein erster Schritt ist auch hier, zuerst die Variablen zu benennen und den Wert festzulegen. Erst dann kommt die Schleife.

Für dein Portfolio 4 | A
Wenn du dir unsicher bist, welche Befehle du in die Schleife schreiben musst, kannst du dir einen Tipp geben lassen.

Programmiere einen komplizierten Mäander II

Lass die Turtle den abgebildeten Mäander zeichnen. Die kürzeste Seite hat eine Länge von 100.

Verändere die Seitenlängen so, dass die kürzeste Seite eine Länge von 50 hat.

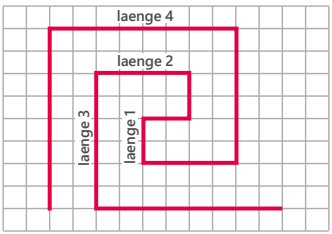


Zusatzaufgabe: Verändere die Seitenlängen des Mäanders so, dass die längste Seite eine Länge von 300 hat.

Tipp:
Du brauchst vier Variablen, die du z. B. so nennen kannst:

```
var laenge1 := 100
var laenge2 := 200
var laenge3 :=
var laenge4 :=
```

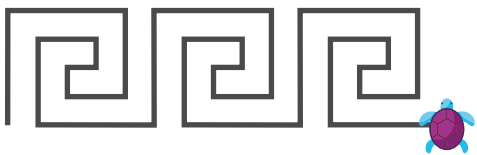
Denke daran!
Ab jetzt musst du ein bisschen rechnen. Dein Mäander soll etwas kleiner oder größer werden, also musst du die Variablen entsprechend anpassen. Die Kästchen helfen dir bei der Berechnung der Längen.



Für dein Portfolio 4 | B
Im Portfolio kannst du die unterschiedlichen Längen deines Mäanders berechnen.

Programmiere einen komplizierten Mäander III

Lass die Turtle den abgebildeten Mäander zeichnen. Verwende dafür nur eine Variable.



Zusatzaufgabe: Lass die Turtle verschiedene Formen, wie z. B. eine quadratische Schnecke zeichnen.

Tipp:
Der Code für den Mäander könnte so aussehen:

```
turtle → forward(laenge1*4)
turtle → rightTurn(90)
turtle → forward(laenge1*4)
turtle → rightTurn(90)
turtle → forward(laenge1*3)
turtle → rightTurn(90)
turtle → forward(laenge1*2)
turtle → rightTurn(90)
turtle → forward(laenge1*1)
turtle → rightTurn(90)
```

Denke daran!
Wenn ein Muster für dich zu kompliziert ist, mal es zuerst auf. Dann kannst du die einzelnen Schritte einzeichnen. Du kannst die Muster aber auch im Raum ablaufen.

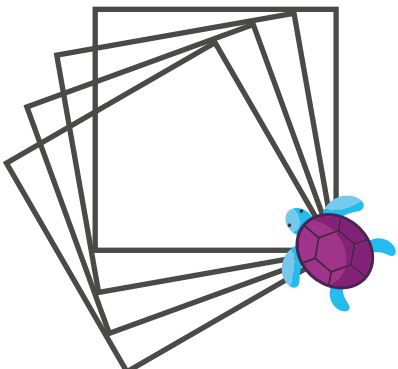
Wenn du mit Variablen arbeitest, musst du nicht alles selbst rechnen.

Für dein Portfolio 4 | C & D
C: Dein Portfolio kann dir einen Tipp geben, wie du aus vier Variablen eine machst. Dazu braucht es einen Rechenvorgang.

D: Du kannst dann auch ein Muster deiner Wahl aufkleben.

Programmiere eine Spirale I

Lass die Turtle das abgebildete Muster aus Quadraten nachzeichnen. Verwende für die Seitenlänge eine Variable.



Zusatzaufgabe: Verändere deinen Code so, dass die Turtle sich einmal ganz herumdreht.

Experimentiere mit den Zahlen, ändere die Farben oder nimm eine andere Grundform. So entstehen wunderschöne Blüten.

Tipp:
Du kannst statt einer Farbe auch die Variable i aus der Schleife eintragen. Dann erhält jede neue Form eine andere Farbe.

```
for 0 <= i < 20 do
    turtle → setPenColor(i)
    for 0 <= j < 12 do
        turtle → forward(30)
        turtle → leftTurn(30)
    end for
    turtle → leftTurn(18)
end for
```

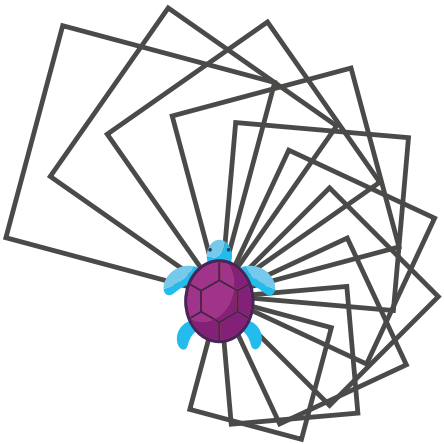
Denke daran!
Verwende für das Muster zwei Schleifen: eine für das Quadrat und eine für die Wiederholungen der Grundform. Verwende zusätzlich einen Merkkasten (Variable) für die Seitenlänge des Quadrats, die 100 betragen soll.

Für dein Portfolio 5 | A
Im Portfolio kannst du dir eine kleine Gedankenstütze geben lassen, die dir dabei hilft, das Muster zu formen.

Programmiere eine Spirale II

Lass die Turtle das abgebildete Muster zeichnen.

Die Grundform ist ein Quadrat. Es wird nach jedem Schleifendurchlauf größer, so dass eine Spirale entsteht.



Zusatzaufgabe: Verändere die Werte so, dass du unterschiedliche Spiralen erhältst. Du kannst die Spiralen auch bunt machen.

Tipp:
Für die Spirale musst du am Ende der Schleife, die das Quadrat malt, die Variable größer machen. Dies machst du mit einer Addition.

```
var laenge := 100
for 0 <= i < 10 do
    for 0 <= j < 4 do
        turtle → forward(laenge)
        turtle → leftTurn(90)
    end for
    var laenge := laenge+10 ←
    turtle → leftTurn(15)
end for
```

Denke daran!
Wie groß die unterschiedlichen Formen werden, kannst du durch den Wert am Ende der Schleife bestimmen. *laenge+10* ist dabei natürlich kleiner als *laenge+20*.

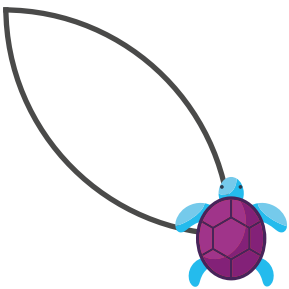
Erklärung:
Der Zuweisungsoperator := kann einer Variable einen neuen Wert zuweisen.

Für dein Portfolio 5 | B
Du kannst eines deiner Muster in dein Portfolio aufkleben.

Programmiere eine Blume I

Lass deine Turtle das abgebildete Blatt einer Blüte zeichnen. Beginne damit, einen Kreis zu zeichnen.

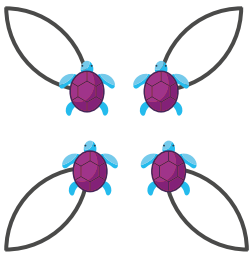
Lass die Turtle dann nur ein Viertel des Kreises zeichnen und forme so daraus das Blatt.



Tipp:
Die Turtle muss sich nach dem ersten Viertelkreis drehen. Schau dir den Winkel der Drehung genauer an.

```
for 0 <= i < 2 do
    for 0 <= j < 90 do
        turtle → forward(2)
        turtle → leftTurn(1)
    end for
    turtle → leftTurn(90)
end for
```

Denke daran!
Die Turtle kann das Blatt auch in eine andere Richtung zeichnen. Dann musst du nur die Drehung ändern.

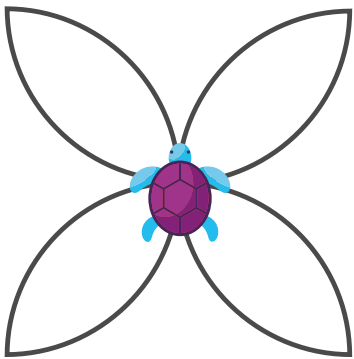


Für dein Portfolio 6 | A
Öffne das Portfolio und überlege noch einmal, wie man einen Kreis programmiert.

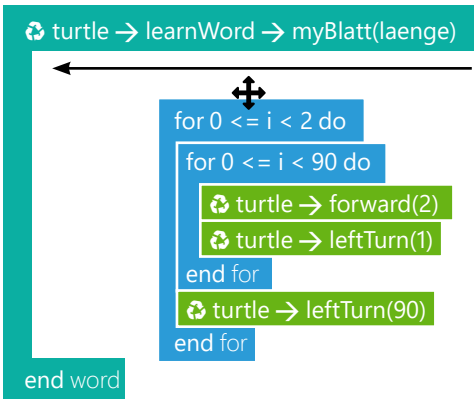
Programmiere eine Blume II

Lass die Turtle die abgebildete Blüte zeichnen. Beginne damit, der Turtle das Wort *Blatt* beizubringen.

Erstelle dann die Blüte mit Hilfe des neuen Unterprogramms *Blatt*.



Tipp:
Nachdem du ein neues Wort erstellt hast, kannst du deinen Code einfach hineinziehen.



Denke daran!
Ein neues Wort lernt die Turtle, indem du hinter deiner Funktion auf das + drückst und dann ein neues *„Word“* auswählst.

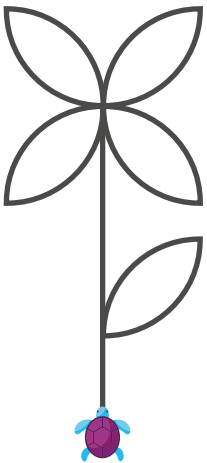
Du kannst das neue Wort z. B. *myBlatt* nennen.

Erklärung:
Mit einem Unterprogramm kannst du der Turtle neue Worte beibringen oder neue Funktionen erstellen.

Für dein Portfolio 6 | B
Schau dir die Aufgabe an, um dir zu überlegen, wie die Turtle die ganze Blüte malen könnte.

Programmiere eine Blume III

Erstelle mit Hilfe verschiedener Unterprogramme die vollständige Blume.



Zusatzaufgabe: Lass die Turtle einen ganzen Blumengarten zeichnen.

Tipp:
Der Code für deine Blume könnte so aussehen:

```
function main()
    turtle → myBlume(2)
end function
```

```
turtle → learnWord → myBlume(laenge)
    for 0 <= j < 4 do
        turtle → myBlatt(laenge)
        turtle → rightTurn(90)
    end for
end word
```

```
turtle → learnWord → myBlume(laenge)
    for 0 <= i < 90 do
```

Denke daran!
Du kannst dein Wort *myBlatt* nicht nur als Blütenblatt, sondern auch als Blatt am Stängel benutzen.

Du kannst der Turtle die unterschiedlichsten Wörter beibringen, damit dein Code nicht zu lang wird.

Für dein Portfolio 6 | C
Im Portfolio kannst du deine Blume und deinen Garten einkleben.

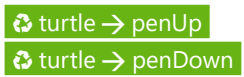
Programmiere ein Schachbrett I

Lass die Turtle eine Strichlinie zeichnen, bei der sich immer zwei Farben abwechseln.

Lass die Turtle aneinandergereihte Quadrate zeichnen. Zwischen ihnen soll etwas Platz liegen. Nutze auch hier zwei sich abwechselnde Farben.

Zusatzaufgabe: Versuche doch einmal, mehr als zwei Farben zu verwenden, die sich abwechseln.

Tipp:
Denke wieder an die *penUp*- und *penDown*-Befehle. Nur so kann die gestrichelte Linie entstehen.



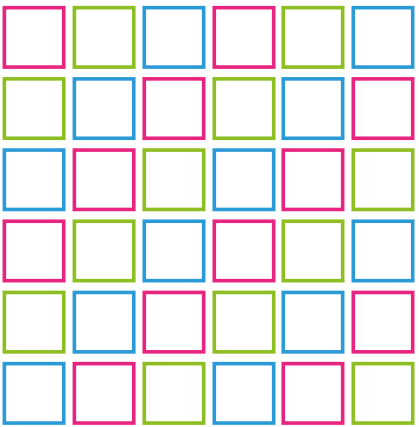
Denke daran!
Die Turtle kann Worte lernen. Wenn du ihr neue Worte beibringst, wird der Code viel leichter und übersichtlicher.

Für dein Portfolio 7 | A
Lass dir im Portfolio mit der Anordnung des Codes helfen.

Programmiere ein Schachbrett II

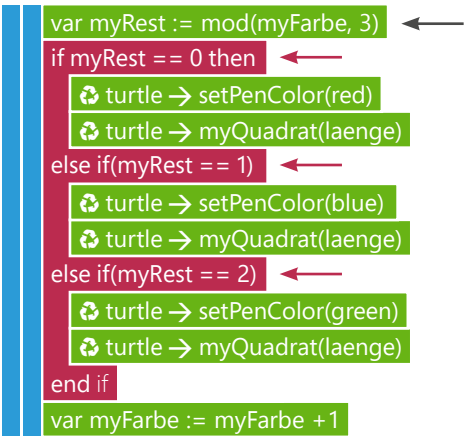
Lass die Turtle dein buntes Schachbrett zeichnen.

Du kennst nun alle Befehle und Programmierprinzipien.



Zusatzaufgabe: Lass deiner Kreativität freien Lauf! Programmiere zum Beispiel einen bunten Garten mit Blumen, Haus und Zaun.

Tipp:
Der Wert, durch den der *mod* teilt, entspricht immer der Anzahl der Entscheidungen (hier 3).



Erklärung:
Mit der *if-else*-Bedingung kannst du in deinem Code Entscheidungen festlegen. Wenn ein Wert z. B. 0 ist, dann malt die Turtle blau (siehe unser Beispiel links).

Erklärung:
Mit mod kannst du eine Division mit Rest darstellen. Die zugehörige Variable kannst du z. B. *myRest* nennen.

Für dein Portfolio 7 | B & C

B: Ergänze den Code in deinem Portfolio, um dir die *if-else*-Bedingung genauer anzuschauen.

C: Du kannst dein fertiges Schachbrett dann in dein Portfolio kleben.