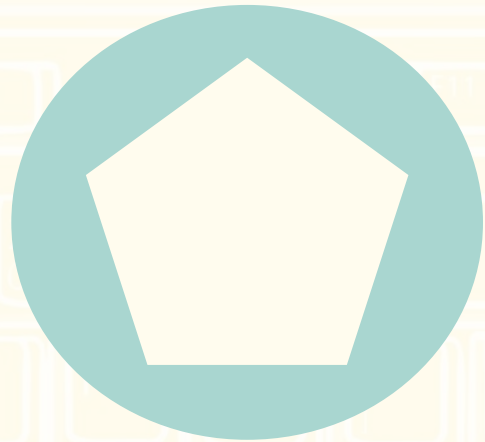
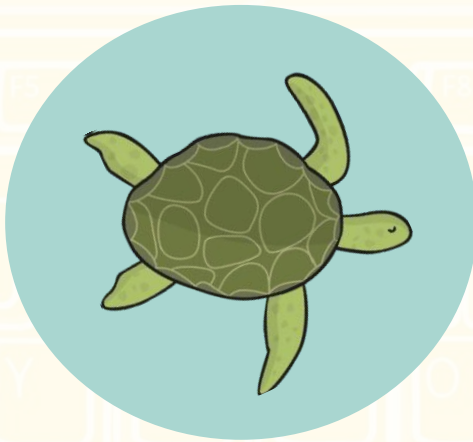




# Computing

Programming Turtle Logo and Scratch

# Regular Polygons



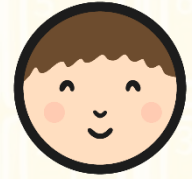
# Aim

- I can create and debug algorithms that draw regular polygons.

# Success Criteria

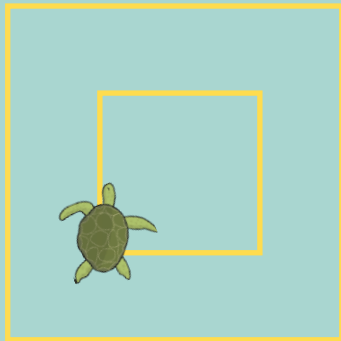
- I can write commands in the correct order.
- I can write a variable value where required.
- I can correct any mistakes.
- I can rotate the turtle angles other than  $90^\circ$ .
- I can use calculations as a variable.

# Write the Algorithm

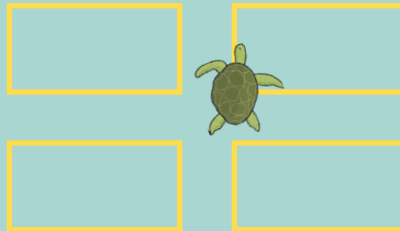


Write algorithms for these different shapes and test them:

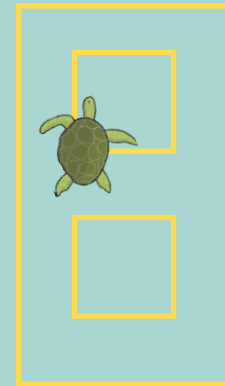
A square of 120 sides with a square of 60 sides inside.



4 rectangles of 30 and 50 sides with a space of 20 between each.



A rectilinear number 8.

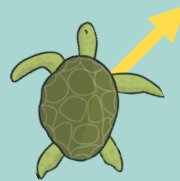


# Turning Angles Other Than 90°

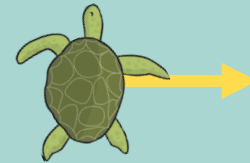
The turtle can be commanded to turn any angle. The angle of the turn is measured in degrees.



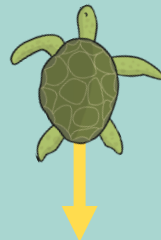
rt will turn the turtle an angle of  $45^\circ$  to the right.



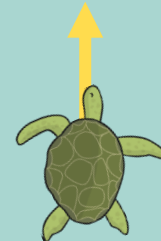
$90^\circ$  is a quarter turn.



$180^\circ$  is a half turn.



$360^\circ$  is a full turn.



# What Angle To Turn?

To work out the angle to turn for any polygon you need to divide 360 by the number of sides.

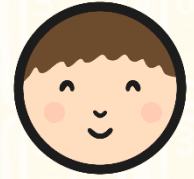
In a square there are 4 turns, so the turn is  
 $360 \div 4 = 90$ .



So, for a hexagon, it is  
 $360 \div 6 = 60$ .



# Drawing Different Polygons

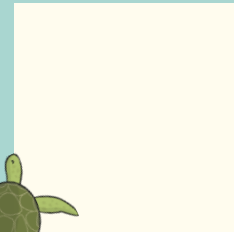


Using a Turtle Logo programme on a computer or tablet, draw different regular polygons.

How will you work out what angle you need to turn?

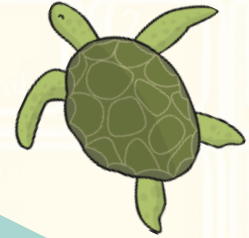


Remember a whole turn is  $360^\circ$ , and with a square you turned a quarter turn, which is  $90^\circ$ .



# Let Turtle Logo Work it Out!

You can write an algorithm that includes a calculation.



So for a square you can make  
the turn  $360/4$   
(representing  $360 \div 4$ )

The command will be repeated  
`4[fd 100 rt 360/4]`

So, for a heptagon,  
it is  
repeat `7[fd 80 rt 360/7]`

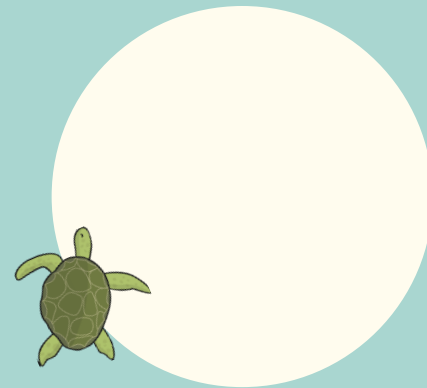
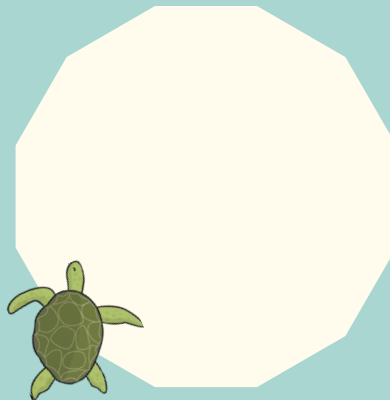
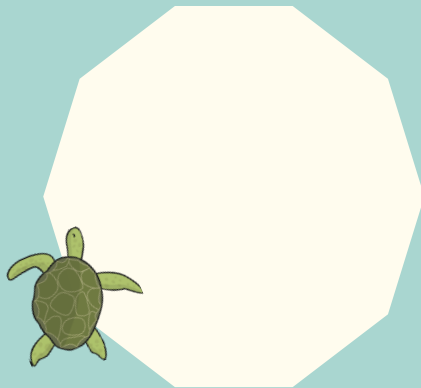
What happens when you draw a polygon with many sides?  
Try writing an algorithm for a shape with 360 side. What happens?



# Write Your Own



Write your own algorithms that include calculations to draw regular polygons that have a different numbers of sides.

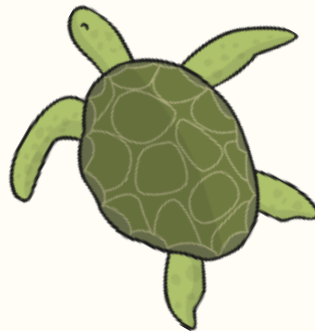


# 17 Sides?



Can you write  
the algorithm for a  
regular 17 sided shape?

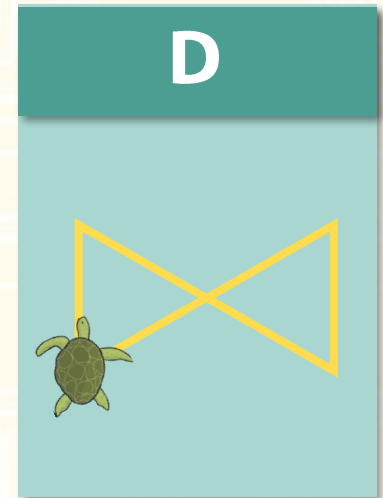
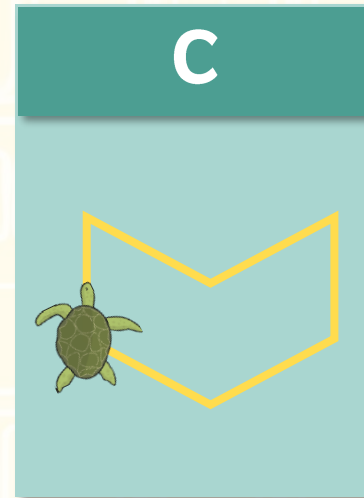
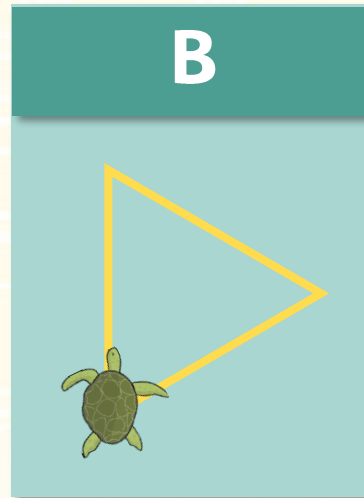
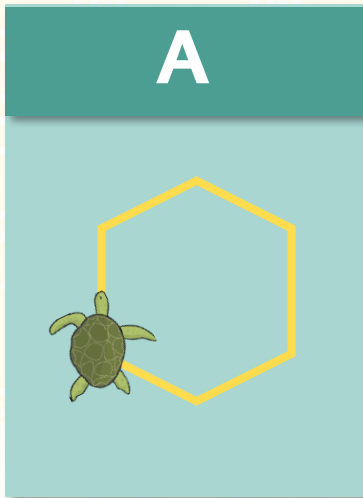
**repeat 17[fd 40 rt 360/17]**



# Which Shape Will Be Drawn?



fd 120 rt 120 fd 120 lt 60 fd 120 rt 120  
repeat 3[fd 120 rt 60]



**Incorrect:**

All turns would be rt 60 or lt 60.

**Incorrect:**

Click on the shape that you think is correct

All turns would be rt 60 or lt 60.



**Incorrect:**

Second line would repeat the first line.

# Aim



- I can create and debug algorithms that draw regular polygons.

# Success Criteria

- I can write commands in the correct order.
- I can write a variable value where required.
- I can correct any mistakes.
- I can rotate the turtle angles other than  $90^\circ$ .
- I can use calculations as a variable.

