DRAWING GEOMETRIC SHAPES

This document will teach you how to draw geometric shapes using Scratch. We'll start by reviewing some laws that relate the angles and sides of triangles, and then we'll use these laws to draw different shapes, including triangles, trapezoids, and kites.

The Laws of Sines and Cosines

The laws of sines and cosines express the relation between the side lengths and angles of any triangle. Consider the triangle shown in Figure 1.

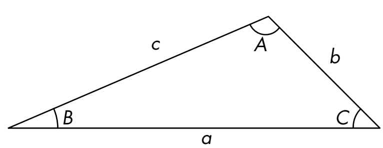

Figure 1: A triangle

The *law of sines* states:

$$\frac{a}{\sin A} = \frac{b}{\sin B} = \frac{c}{\sin C}$$

The *law of cosines* states:

$$c^2 = a^2 + b^2 - 2ab\cos C$$

When angle C is 90° (that is, a right triangle), $\cos C = 0$, and the law of cosines reduces to the well-known *Pythagorean theorem*:

$$c^2 = a^2 + b^2$$

The Pythagorean theorem states that in a right triangle, the square of the length of the hypotenuse equals the sum of the squares of the lengths of the legs. In the following section, we'll use these simple laws to draw different types of triangles.

Drawing Triangles

Using the relative motion commands in Scratch, we can draw any triangle. We'll start with some of the frequently used triangles, such as the isosceles right triangle, the 30-60-90 right triangle, and the 3:4:5 right triangle. Then we'll move on to draw "general triangles"—that is, those that are not specifically right triangles.

Isosceles Right Triangle

Consider the isosceles right triangle whose leg measures x units, as shown in Figure 2 (left).

Figure 2: Drawing an isosceles right triangle

Using the Pythagorean theorem, we find that the length of the hypotenuse is $\sqrt{2}x$. With this information, we can easily draw an isosceles right triangle using the script shown in Figure 2. Use the illustration on the right to trace its execution. The arrow in this figure represents the initial position and orientation of the sprite before the script executes. The last **turn** command in the script ensures that the sprite's final orientation is the same as its starting orientation.

30-60-90 Triangle

As its name indicates, the angles of the 30-60-90 triangle measure 30°, 60°, and 90°. Using trigonometry, we can determine that the ratios between the lengths of the three sides of this triangle are $1{:}\sqrt{3}{:}2$, shown in Figure 3 (left).

Figure 3: Drawing a 30-60-90 triangle

With this information, you can create a script to draw this triangle by using **move** and **turn** commands in the correct sequence, as shown in the figure. The illustration on the right shows the path a sprite would take to draw a 30-60-90 triangle.

3:4:5 Triangle

The 3:4:5 triangle is a right triangle whose side length ratios are 3:4:5. Using trigonometry, we can find the angles of the two vertices, shown in Figure 4.

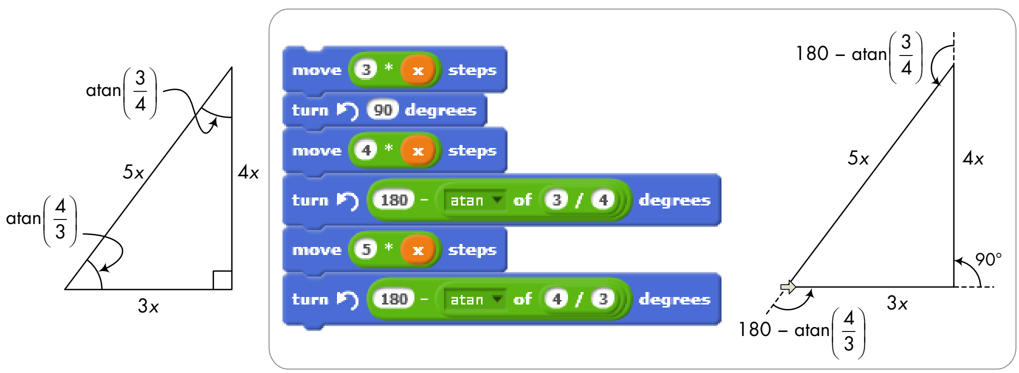

Figure 4: Drawing a 3:4:5 triangle

We know all the sides and angles of a 3:4:5 triangle, so we can create the script shown in the figure to draw it. Use the illustration on the right to trace through the drawing sequence.

General Triangle

For our last example, let's consider the scalene triangle shown in Figure 5 (left).

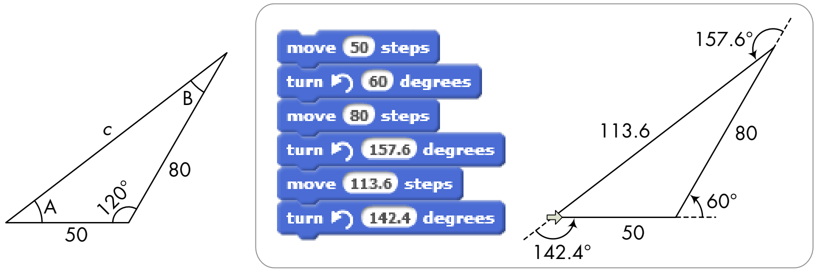

Figure 5: Drawing a scalene triangle

Once you know the sides and angles of a triangle, drawing it is a matter of executing three sequences of **move** and **turn** commands with the appropriate move steps and turn angles. In this case, we already know the lengths of two legs and the measure of one angle. Applying the law of cosines, we have

$$c^2 = 50^2 + 80^2 - 2(50)(80)\cos 120 = 12{,}900$$

which gives $c = 113.58$.

Applying the law of sines, we have

$$\frac{50}{\sin B} = \frac{80}{\sin A} = \frac{113.58}{\sin 120} = 131.15.$$

Solving these equations gives us $A = 37.6°$ and $B = 22.4°$.

Writing a script to draw this triangle just involves plugging in the values for each **turn** and **move** block used by the sprite. You can see the finished script in Figure 5. Use the illustration on the right as a visual guide to the sprite's movement.

Next, we'll look at four-sided geometric shapes, also known as quadrilaterals.

Drawing Quadrilaterals

In this section, you'll draw rectangles, parallelograms, rhombuses, trapezoids, and kites.

Drawing Rectangles

Let's start with the rectangle shown in Figure 6 (left), where `side1` and `side2` represent the lengths of two of its sides.

Figure 6: Drawing a rectangle

If you start at the lower-left corner, you need to move side1 steps, turn left 90°, move side2 steps, and turn left 90°, and then repeat these four instructions one more time. The figure shows a script that implements this sequence of instructions, and the illustration on the right will help you trace it.

Drawing Parallelograms and Rhombuses

Let's take on a slightly more difficult problem and draw a parallelogram, as shown in Figure 7 (left).

Figure 7: Drawing a parallelogram

The sprite starts at the lower-left corner, and the first instruction is to move forward side1 steps. Since the interior angle is *A*, the sprite has to turn left by 180° – *A* to orient itself to draw the second side. After moving side2 steps, the sprite needs to turn left again so that it points west. This time, the turn angle is *A*. To draw the other two sides, we just repeat this sequence.

The figure shows a script that implements these steps. You can draw different parallelograms by substituting different numbers for side1, side2, and the angle A. Note that setting A = 90° results in a rectangle.

Since a rhombus is an equilateral parallelogram, we can adapt the script in Figure 7 to draw any rhombus. All four sides of a rhombus are equal, so use the same number for side1 and side2. Try it out!

Drawing Trapezoids

Let's take what we learned so far a step further and explore the problem of drawing trapezoids. To keep things simple, we'll focus on one particular trapezoid, shown in Figure 8 (left); the long base is 200 units, the height is 90 units, and the two base angles are 45° and 60°.

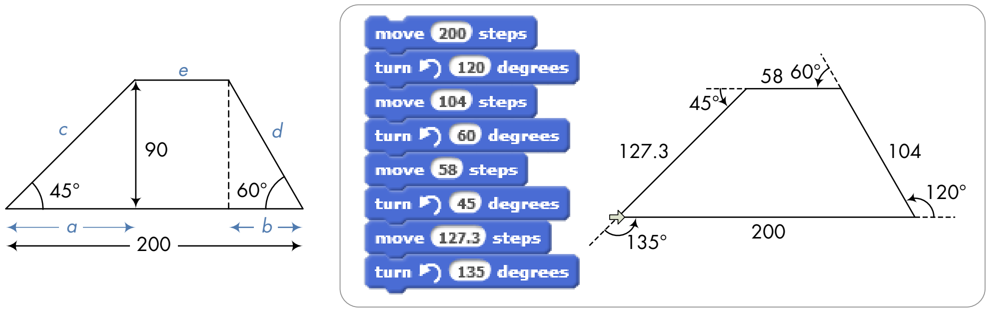

Figure 8: Drawing a trapezoid

The first step is to figure out the values of the three sides (c, d, and e). Using trigonometry, we find that

$$a = \frac{90}{\tan 45} = 90$$

and, using the Pythagorean theorem,

$$c = \sqrt{90^2 + 90^2} = 127.3.$$

Similarly,

$$b = \frac{90}{\tan 60} = 52$$

and

$$d = \sqrt{90^2 + 52^2} = 104.$$

Thus, the length of the short base is

$$e = 200 - (a + b) = 58.$$

With these numbers in hand, we can write the script in Figure 8.

Drawing Kites

The last shape in our quadrilateral-drawing journey is the kite. As an example, we'll focus on the kite shown in Figure 9 (left).

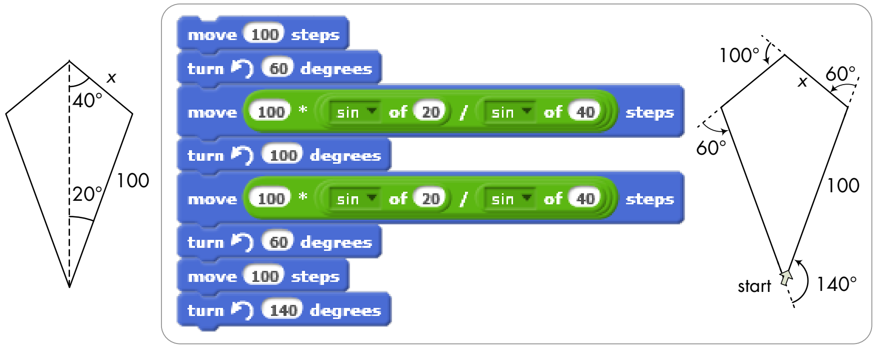

Figure 9: Drawing a kite

You can divide a kite into two triangles to help you find the missing side. Using the law of sines, we have

$$\frac{x}{\sin 20} = \frac{100}{\sin 40}$$

which gives

$$x = 100\frac{\sin 20}{\sin 40}.$$

We almost have all the numbers we need to draw our kite. Use the numbers in the figure and the information you've learned about triangles to figure out how much the sprite should turn after drawing each side. Our final script for drawing the kite is shown in Figure 9, along with an illustration of how it works.

Now that you've seen how to draw triangles and quadrilaterals, the next section will cover polygons with more sides.

Drawing Polygons

If you're excited about how easy it is to draw geometric shapes in Scratch, you might be wondering whether there are some rules for drawing more complex shapes. Look back at the shapes we've drawn already, and you'll see that the sum of the exterior angles has been 360° for each figure. The same is true for polygons as well. In this section, you'll learn about a simple, yet important, rule that allows you draw regular polygons of any order.

The Rule of 360

The previous examples revealed a consistent pattern regarding the sum of the exterior angles of the shapes we considered. If you examine these shapes carefully, you'll find that the sum of the exterior angles the sprite turned as it drew each one of these figures is 360°. The question now is, does this pattern hold for other polygons? The answer is yes, and we could prove it mathematically. However, I'll not bother you with that. Instead, I hope Figure 10 will convince you.

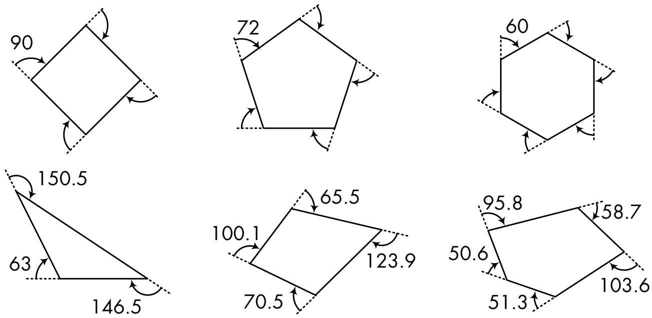

Figure 10: The sum of the exterior angles of any polygon is 360°.

The top row shows three regular polygons (that is, all sides have the same length). The figure clearly shows that the sum of all exterior angles for each polygon is 360°. The bottom row shows three irregular polygons. Again, if you add up the exterior angles, you'll find that the sum is 360° in each case. We conclude from these examples that the sum of a sprite's turns when it draws any polygon is 360°. This will be the basis for the polygon-drawing script presented in the next section.

Drawing Regular Polygons

Equipped with the 360° rule, you should now be able to draw any regular polygon. Your script will be similar to that shown in Figure 11.

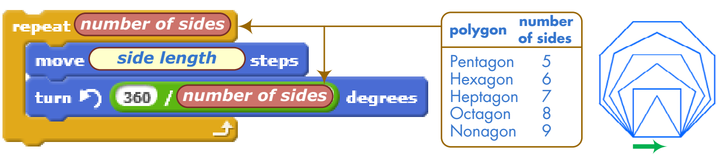

Figure 11: Drawing regular polygons

You can substitute any integer value (that is, whole number) for the number of sides and any value for the side length to control the size of the polygon. The figure also shows six polygons of the same side length that were drawn using this script. The sprite started at the position and direction indicated by the green arrow. The location of the output polygon on the Stage depends on the initial direction of the sprite that executes this script.

Create the script shown in Figure 11 and run it using different values for the number of sides. What happens when this number becomes larger? Does this give you an idea of how to draw circles?

Rotated Polygon Art

In Chapter 2, we created a script for drawing rotated squares (see Figure 2-18), and with some minor tweaks to that script, you can also draw rotated polygons. Figure 12 shows a generic version of the script, where we can use any integer number for **repeat count**.

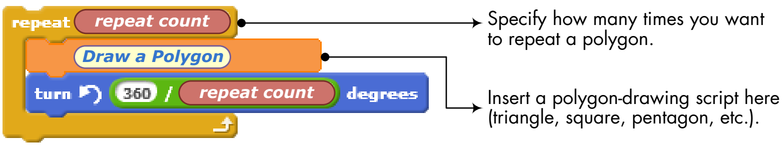

Figure 12: Template script for creating patterns of rotated polygons

Figures 13 through 19 show some of the patterns that can be obtained by rotating triangles, squares, pentagons, hexagons, heptagons, octagons, and decagons.

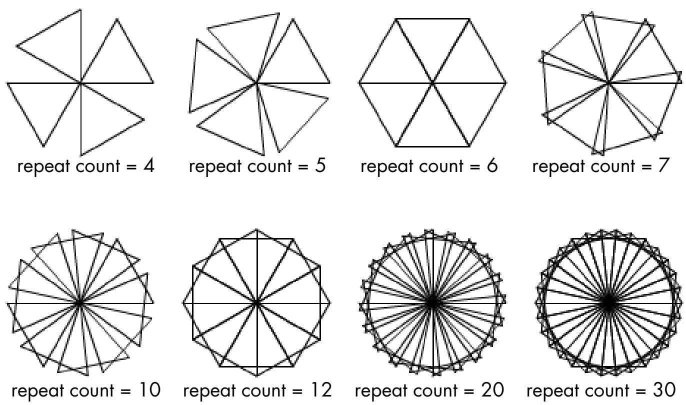

Figure 13: Patterns obtained by rotating an equilateral triangle

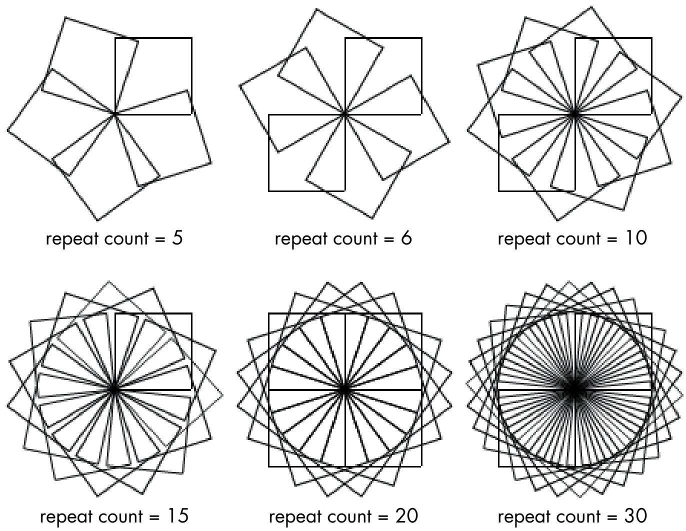

Figure 14: Patterns obtained by rotating a square

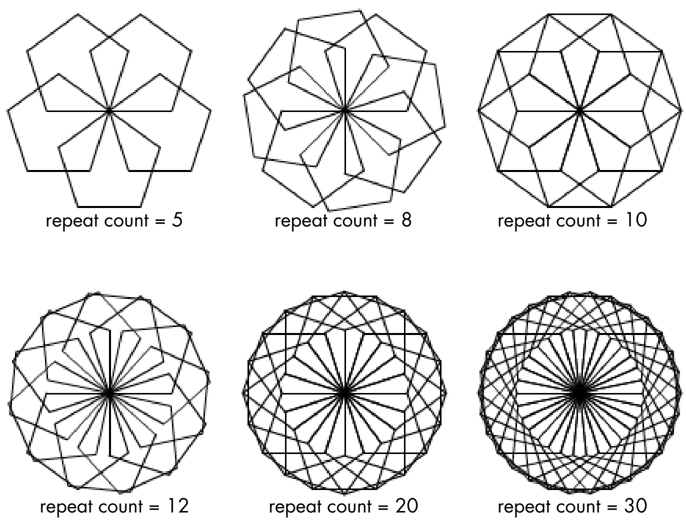

Figure 15: Patterns obtained by rotating a pentagon

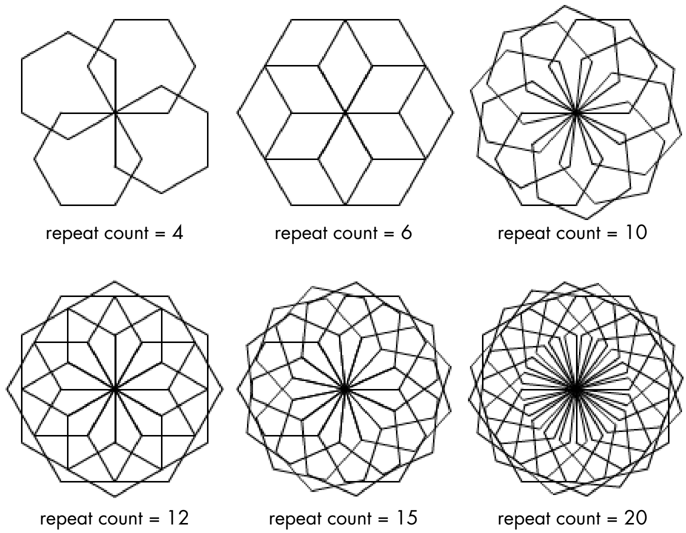

Figure 16: Patterns obtained by rotating a hexagon

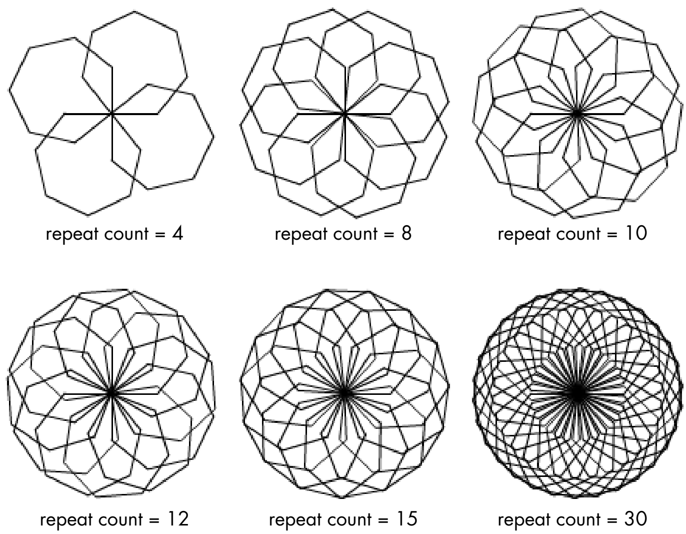

Figure 17: Patterns obtained by rotating a heptagon

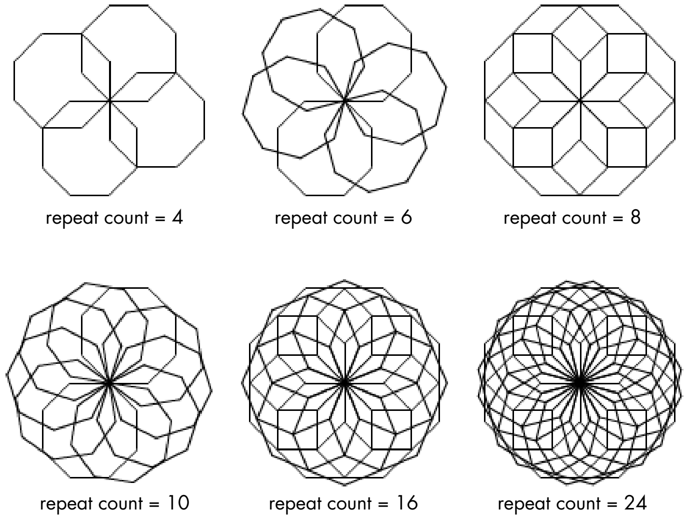

Figure 18: Patterns obtained by rotating an octagon

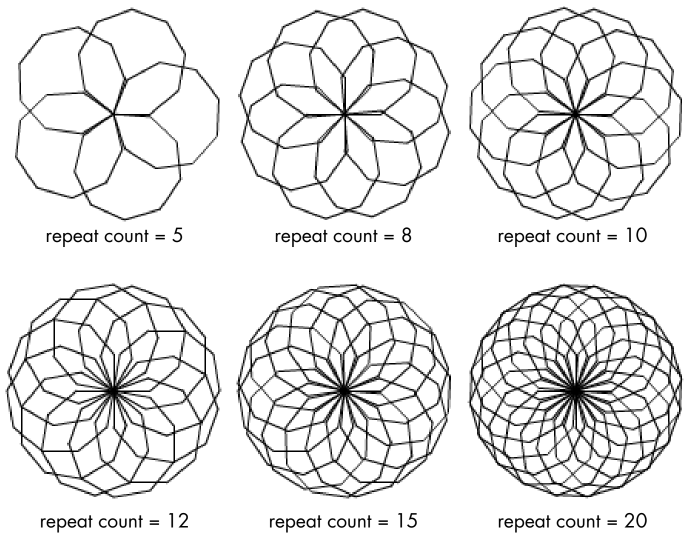

Figure 19: Patterns obtained by rotating a decagon

You now know how to draw a wide variety of geometric shapes. How will you use them in your Scratch projects?