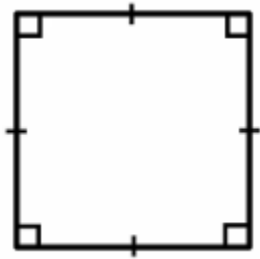


# Lab 1.3: Squares and Triangles and Stars, Oh, My!

In this lab, you will write your first SNAP programs to draw some simple shapes on the stage.

## 1. Drawing a square

**1.1)** Write a SNAP script that draws a square when the number 1 is pressed on the keyboard. Remember that each corner of a square is a 90° angle, as shown in the figure below.



**1.2)** Add code so that the sprite says the word "square" while it is drawing. The sprite should stop saying "square" once it has finished drawing the square.

**1.3)** Add code so that pressing the space bar clears the pen marks from the stage.

### Content

#### Ways to transform requirements into algorithms

An algorithm is a set of precise instructions that solve a problem or perform a given task. When you wrote a SNAP program to draw a square, you essentially developed an algorithm to create a square. Your algorithm probably involved moving a certain amount of spaces in a straight line, then turning 90 degrees.

In the tasks below, you will be asked to create programs, and therefore develop algorithms, for other shapes. As you do this, think carefully about how the requirements of the shape influence the algorithm you create.

For example, what characteristics of a pentagon influence the algorithm. The number of sides? The number of vertices? How are these characteristics apparent in your algorithm?

## 2. Adding more shapes

**2.1)** Now that you've drawn a square, add code to draw the shapes in the following table. Each shape should be drawn when the number next to it is pressed on the keyboard. (For example, pressing 3 on the keyboard should draw a diamond.)

When this key is pressed...	Draw a ...
1	Square
2	Equilateral triangle
3	Diamond
4	Pentagon
5	Parallelogram ("leaning rectangle")
6 (optional)	5-pointed star

You may want to draw a diagram similar to the one above of a square to figure out the angles in each shape.

**2.2)** Add code so that as each shape is being drawn, the sprite is saying the name of shape. The sprite should stop saying the name of the sprite when it is finished drawing.

**2.3)** Modify your code so that each different shape is drawn in a *different* color and with a *different line thickness*. So, for example, if the square is drawn in blue with a line thickness of 3, each other shape must be drawn in a color that is not blue and with a line thickness that is not 3.

**2.4)** Add code so that the sprite is hidden when it is not drawing. This will make sure that the sprite is not obstructing the view of your beautiful artwork!

<b>Lab 1.3 Criteria</b>	
1 - Square	0.25 points
2 - Equilateral Triangle	0.25 points
3 - Diamond	0.5 points
4 - Pentagon	0.5 points
5 - Parallelogram	0.5 points
Star (bonus!)	0.5 points
<b>PROJECT TOTAL</b>	<b>2.5 points</b>