Robot shape-up

How did the coder get the robot into shape? By programming it to drive in squares!

In this activity, you will be the Edison robot's 'personal trainer' by getting the robot to move in different shapes. To do this, there are a few things we need to learn:

- Part 1: What is a variable?
- Part 2: What is a 'for' loop?
- Part 3: How do you drive a different shape?

Is this your first time using Edison robots or EdPy? Start with the activity *Edison and EdPy: Get started* first! Ask your teacher for a copy.

Part 1: What is a variable?

In computer programming, we often use the same bit of information multiple times in a single program. Variables make it a lot easier to do this.

Did you know….

A **variable** is a bit of memory that is used to store a value in a program. You can think of a variable like a container that you can use to store some bit of information in a way that will make sense to a computer.

When you use a variable, you tell the computer to store a specific bit of information inside that variable. You can then use that variable in different places in the program. Any time the computer sees the variable, it will recall whatever information is stored inside the variable.

Look at the following program:

```
1
2  #-------------Setup----------------
3
4  import Ed
5
6  Ed.EdisonVersion = Ed.V2
7
8  Ed.DistanceUnits = Ed.CM
9  Ed.Tempo = Ed.TEMPO_MEDIUM
10
11 #--------Your code below-----------
12 degreesToTurn = 90
13 Ed.Drive(Ed.SPIN_RIGHT,Ed.SPEED_6,degreesToTurn)
14
```

Line 13 of this program includes the Ed.Drive function, which has three input parameters: direction, speed, and distance. In this program, the distance parameter is not a number but is 'degreesToTurn'.

This is a variable.

Remember, a variable represents a value that is set somewhere in your program. Look at line 12 of the program. This line is assigning a value to the variable. In Python, the equal sign (=) is used to assign values to variables. Line 12 is assigning the variable 'degreesToTurn' the value of 90.

Try it!

Write the program above in EdPy. Download it and run it in your robot to see what it does.

1. What did the robot do when you ran this program?

_____

_____

_____

Okay, but why use variables?

One big benefit of variables is that you can use variables to store values that are used in several places throughout a program. This can be very helpful, especially if the value of a variable changes. By using variables, you only need to make the change in one line of the code to adjust the value everywhere that variable is being used in your program.

Let's try writing a program that uses the variable 'degreesToTurn' in multiple places. This time, write a program for Edison using EdPy so that your robot can drive in a square.

Hint: Try adding additional Ed.Drive() function commands to your program. And don't forget to use the 'degreesToTurn' variable in more than one command!

Download your program and test it using activity sheet 1, placing your Edison at the 'start' point and following the lines. Be sure your program ends with your Edison in the same spot it started.

2. Do you have any duplicate lines of code in your program? If so, what are they and how many times did you use each line?

_____

_____

_____

_____

Part 2: What is a 'for' loop?

You probably noticed that there's a pattern to the code you need to get Edison to drive in a square. You need the Ed.Drive() command with a direction parameter of Ed.FORWARD four times, once for each side of the square. You also needed to use Ed.Drive() command with a direction parameter of Ed.SPIN_RIGHT four times to turn each corner.

Did you find writing the same commands many times a bit boring?

Repeating the same commands over and over is no problem for a computer, but writing out a program this way isn't very efficient. Instead, it is better to use a loop structure.

Did you know….

A **loop** is a special piece of code that tells a computer to repeat something multiple times. Loops are a type of **control structure** because loops control other bits of code in a program.

When you have a program that repeats a set of commands multiple times, using a loop helps make writing the program more efficient because you don't need to write every command individually. Instead, you can use a loop to tell the computer to repeat the commands multiple times. That means you need to use fewer lines of code, which also helps reduce the likelihood of mistyping and having a syntax error in the program.

We can write a program that uses less code but still gets Edison to drive in a square by using a type of loop called a 'for' loop. In Python, a 'for' loop is a control structure which can be used to repeat sets of commands or statements any number of times. Using a 'for' loop allows you to repeat (also called 'iterate') a block of statements as many times as you like.

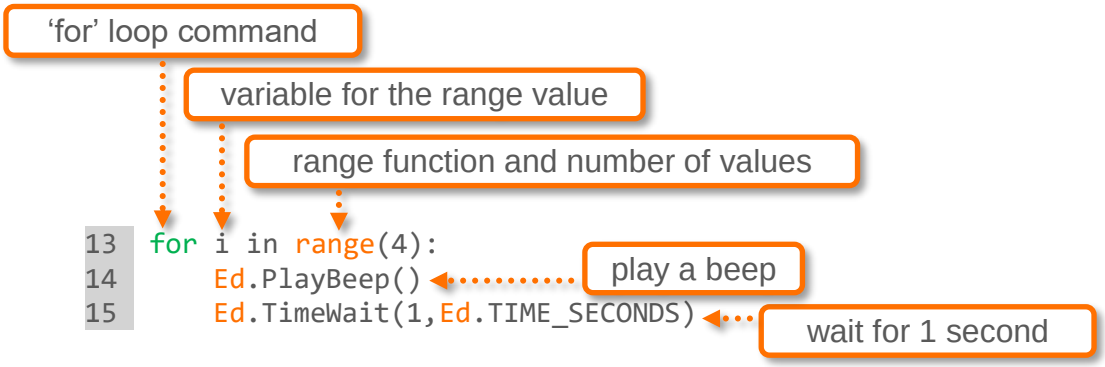
The 'for' loop often goes together with the 'range()' function in Python. The range() function returns a set of values within a certain range. The range() function helps tell the 'for' loop how many times to repeat a set of commands.

Let's look at how the 'for' loop and range() function work together. Look at this EdPy program:

```
13  for i in range(4):
14      Ed.PlayBeep()
15      Ed.TimeWait(1,Ed.TIME_SECONDS)
```

If you run this program in Edison, the robot will play a beep four times with a one-second delay in between each beep.

Let's look at what each part of this program is doing:



- **Line 13:** This line sets up the 'for' loop using the range() function.
- **Line 14 and Line 15:** These two lines are both inside the loop. Each contains a command the program should execute each time the loop iterates.

In Python there are a few syntax rules for using loops. First, you need to use a colon (:) at the end of the code line that sets up the loop. Second, all the commands you want to include in the loop need to be indented. Every indented line will be included in the loop and together they make up the loop's statement block.

This loop is being set up using the range() function with the input parameter of 4. In EdPy, the range() function only has one input parameter. That input parameter determines the upper limit of the set. The lower limit is always 0. That means that the range() function returns values from 0 to (input parameter – 1). In this example, range(4), the upper limit is set to 4. That means that there are 4 values in the set: 0, 1, 2, 3.

Did you notice the little 'i' in line 13? That's a variable.

In this program, the 'for' loop iterates four times. The variable 'i' is being used to store the current iteration of the 'for' loop. In other words, this variable is storing the count for where the loop is up to, starting with '0'. Each time the code iterates, the loop executes all the code inside the loop, the value of variable 'i' increments (goes up by 1) and then the loop starts again. When the value of that variable is at the upper limit, the program knows to stop iterating, and moves on with the program.

Try it!

Write a program using a 'for' loop and the range() function so that when your Edison robot drives, it makes a square. You should be able to complete the program using just two Ed.Drive() functions: one for forward and one for spin.

Hint: Look out for syntax errors! Remember to include a colon at the end of the code line that sets up the loop. Also, remember that all of the commands you want inside your loop need to be indented.

3. What value do you need to have as the input parameter in the range() function to get Edison to drive a square?

   _____

4. Why do you need to have that be the value?

_____

_____

Part 3: How do you drive a different shape?

Now that you know how to get Edison to drive a square using a 'for' loop with the range() function and variables, you have all the tools you need to drive any other regular polygon.

Try it!

Write a new program using a 'for' loop and the range() function that gets your Edison to drive in a different shape, either a triangle or a hexagon. Activity sheet 2 has a triangle and activity sheet 3 has a hexagon. Choose which shape you want to try, then write a program for Edison to drive that polygon. Download and test your program in your robot.

Hint!

A regular polygon means a shape where all the sides are equal. What does that mean for the angles in that shape?

Challenge time!

Time to choose your own shape to make with Edison. Choose a shape which has sides and angles to drive using your Edison robot. Make a workspace to test your program by either drawing your shape on paper or marking it out on the floor or a desk with coloured tape. Write a program for Edison so the robot will drive in your shape. Think about how your program can use variables and loops. Then download and test the program with your robot.

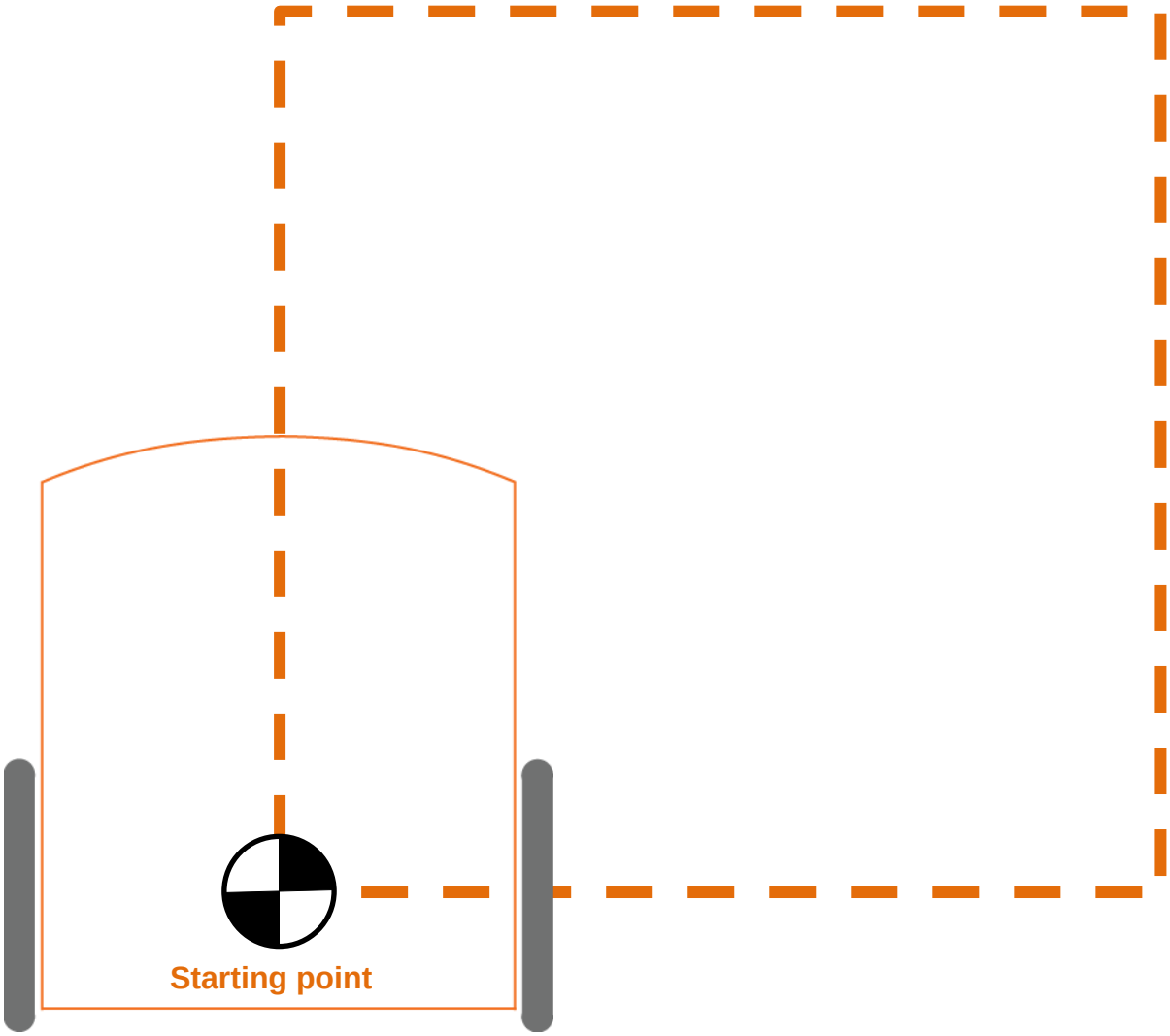
Not sure what shape to make? Here are some ideas:

- Try a different regular polygon: Choose a different regular polygon, like an octagon, a decagon or a dodecagon.
- Try an irregular quadrilateral: Choose an irregular quadrilateral, like a parallelogram, a rhombus or a trapezoid.

If you want, you can even attach a pen to your robot to turn Edison into an artist! Engineer a way to attach a pen to your Edison robot. You can use EdCreate parts or any other materials you like. Once you have built your pen attachment, download one of your polygon programs into your robot and put Edison onto some paper. Try running the program to see your robot make a geometric masterpiece!

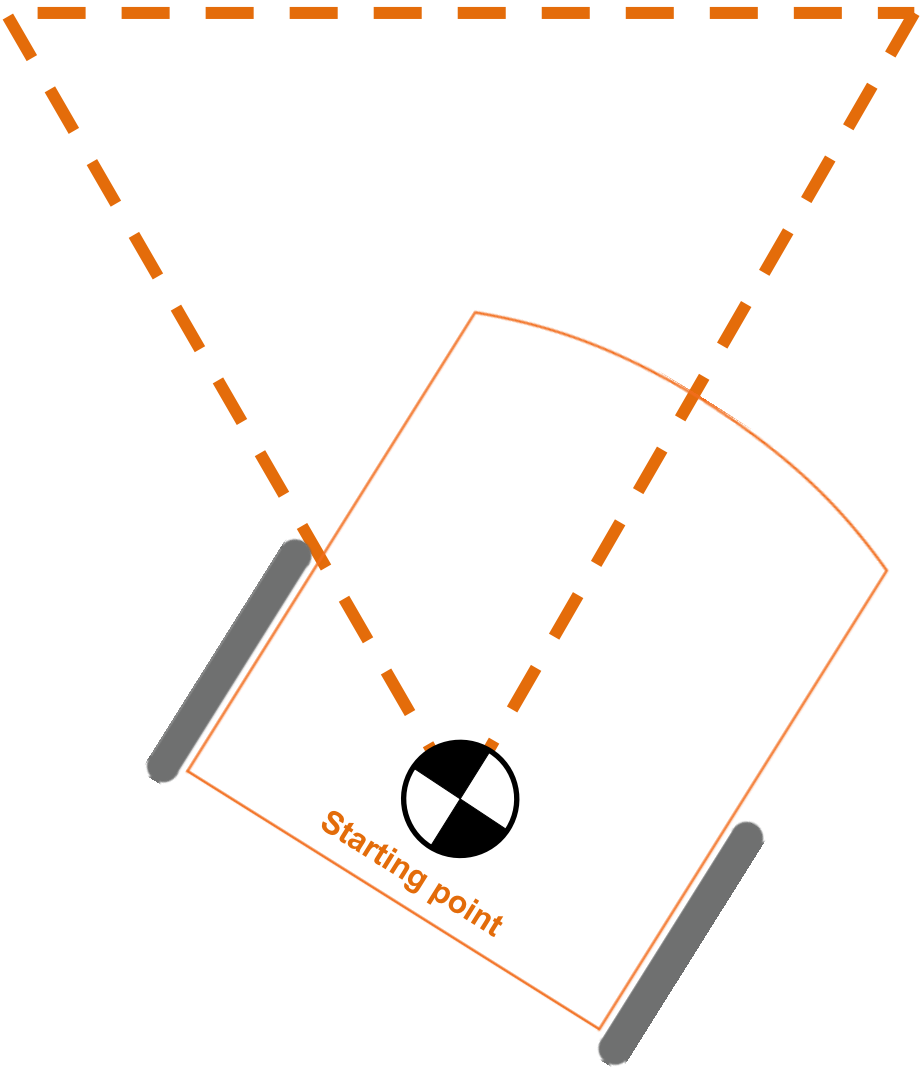
Activity sheet 1 : Drive a square

Print this page!



Activity sheet 2: Drive a triangle

Print this page!



Activity sheet 3: Drive a hexagon

Print this page!