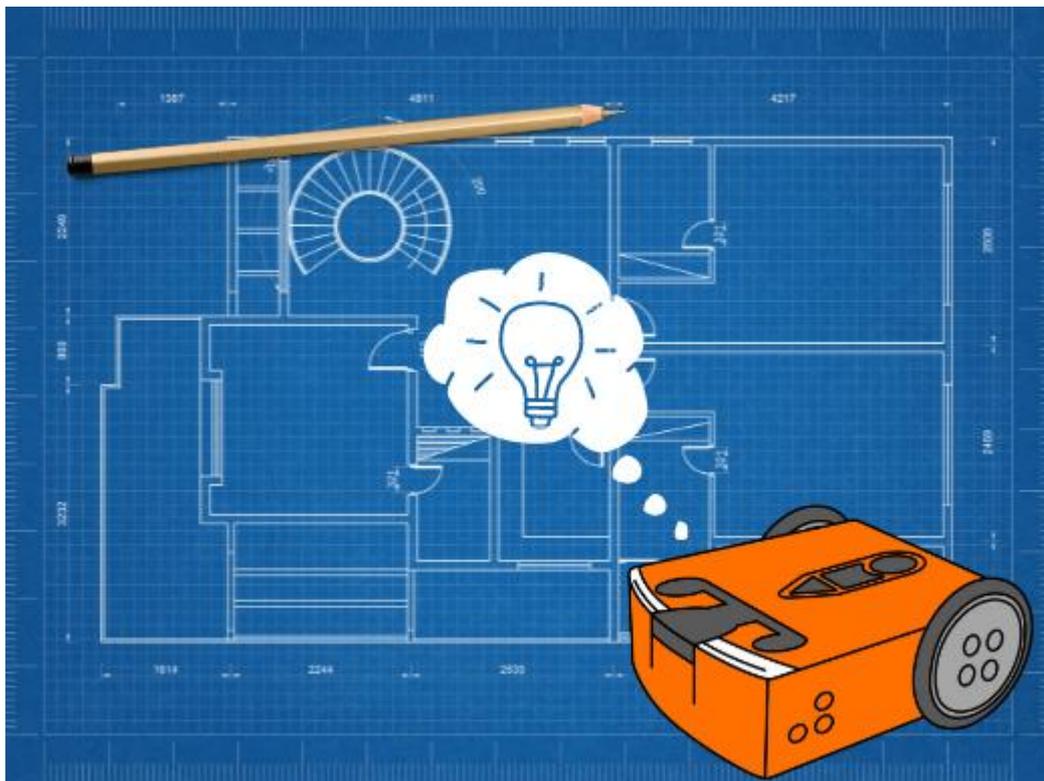




Edison the blueprint designer

Teacher's notes



The *Edison the blueprint designer* lesson set by [Robert Spargo](#) and [Jin Peng](#) is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](#).

Contents

About this lesson and guide	3
Lesson overview	4
Before you start.....	6
Part 1: What is an Edison robot?.....	8
Part 2: How do you use EdScratch with Edison?	8
Part 3: Let's try to make Edison draw a square!.....	9
Part 4: What are variables and iteration?	10
Part 5: Your turn! Designing blueprints!.....	11

About this lesson and guide

This guide offers teachers and instructors overview information, facilitation recommendations and other supporting information for the *Edison the blueprint designer* lesson available at [Edison the blueprint designer](#).

Do you need to read this whole guide to run the lesson? **Absolutely not!**

Once the robots and programming devices are set up, you can start learning along with your students! The student sheets for this lesson have been designed to allow students to work through the different parts of the lesson independently, learning about the Edison robot, how to use the EdScratch programming environment, and the key computer science learning objectives of the lesson. This guide simply offers further information for teachers and instructors to help make using this lesson easy and fun.

Each part of the lesson is included in this guide along with any relevant supporting information for that part. Supporting information is divided into the following sections:

Overview

General information about the section and key learning objectives covered.

Delivery recommendations

Suggestions for how you can cover the lesson section if you want to run the lesson in a more facilitator-led capacity.

Tips and tricks

Helpful hints and ways to overcome common issues students may encounter.

Creative Commons licence attribution details

The *Edison the blueprint designer* lesson set is comprised of the student sheets and this guide. This set is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](#).

Developed and written by: Robert Spargo

Illustrations by: Jin Peng

The *Edison the blueprint designer* lesson set was developed using resources from the [EdScratch Lesson Plans Set](#).

LESSON OVERVIEW

Introduce the key computational concepts of variables and iteration using Edison robots and the [Scratch-based programming language EdScratch](#). Variables, which can be created and utilised through the special 'Data' category in EdScratch, are introduced and then explored in this lesson. The concept of using maths in programming, and the practice of tracing code are also explored and practised.

Grade levels	Difficulty	Duration
Year 5+	Advanced	60 minutes

Prerequisite knowledge

To be successful with this activity, it is recommended that students:

- Have used block-based coding applications
- Understand sequence and sequential programming
- Understand definite loops ('for' loops)
- Have a sound understanding of shapes and degrees as a measurement for angles

Computer science and computational thinking topics covered

- Variables and iteration
- Code tracing
- Computations in programs
- Pseudocode

Tie-ins to other subjects

- Mathematics: Geometry

Supplies you need

- Full set of Edison robots¹ and EdComm programming cables
- Full set of prepared programming devices (computers or tablets) (Please see [‘Setting up your programming devices for EdScratch’](#) in this guide.)
- 4x AAA batteries per robot, regular disposable-alkaline or NiMH rechargeable batteries are recommended (For more information on batteries, please see [‘Get Edison ready’](#) in this guide).
- Print-outs or digital copies of the student sheets
- In order to make Edison 'Draw' the shapes and blueprints, you will need a way to attach a marker or pen, this could be achieved by using our EdSketch attachment, or you could make your own with EdCreate parts or even 'maker-

¹This activity assumes Version 2.0 Edison robots. If you have Version 1 Edison robots, you will need to adjust the activity to have the robot turn using seconds as the input parameter rather than degrees. Learn more at <https://meet Edison.com/meet-edscratch-edison-robots-scratch-language/#V1-EdScratch>

space' craft materials. There is also a penholder available for 3d printing at: <https://www.thingiverse.com/thing:2949946>

Aren't variables hard? A special note about this lesson

The primary focus of this lesson is variables and iteration, two of the most fundamentally important parts of general-purpose programming languages. It's not uncommon for students (and instructors!) to feel like these are 'hard' areas of programming, especially if this is your first time working with variables. Learning about variables and maths in programming is an important step in meaningful computer science education. Just like anything that is brand new, becoming comfortable with maths and variables in programs can take some time. Encourage students to be patient with themselves as they work through the activity. They will soon see the creative potential these new skills unlock inside programming!

Some great advice from the Hour of Code team

It's okay not to know! Respond to student questions and struggles with phrases like:

- "I don't know. Let's figure this out together."
- "Technology doesn't always work out the way we want."
- "Learning to program is like learning a new language; you won't be fluent right away."

And don't forget to have fun! (^_^)

Before you start

Before using Edison with your students, you will need to set up the programming devices, (i.e. the computers or tablets) you will be using with the EdScratch app and get the Edison robots ready.

Additional details can be found in the free *Getting started with Edison guide* available at <https://meetiedison.com/content/Get-started-with-Edison-guide-English.pdf>

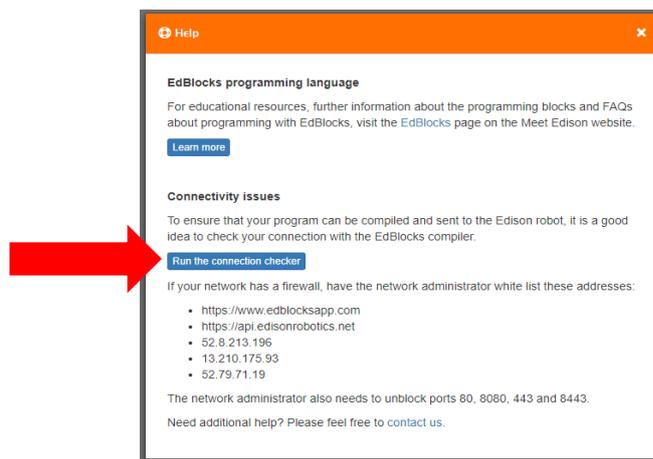
Setting up your programming devices for EdScratch

Depending on the type of programming devices you are using, there are a few things you will need to do to prepare them to be able to work with the EdScratch app.

If you are using computers running Windows operating systems, you will need to disable sound enhancements. Please go to <https://meetiedison.com/edison-robot-support/trouble-shooting/#soundenhancements> to find step-by-step video guides showing you how to disable sound enhancements for standard Windows sound enhancements as well as the most common third-party software programs.

To be able to program Edison, most devices will need the volume turned up to maximum or 100%. As many devices have built-in safety settings that reduce the volume when an audio device is connected using the headphone jack, it is also important to double check that your volume is turned all the way up after plugging in the EdComm programming cable to your device.

You also need to check that the EdScratch app can access the compiler (which is what converts the EdScratch programs into a format that can be sent to the Edison robot) on the devices you will be using. Inside the EdScratch app at www.edscratchapp.com, open 'Menu' in the upper left-hand corner and select 'Help'. This will open a pop-up which includes a button to 'run the connection checker':



Click this button to check your connection and follow the steps provided.

Additional details specific to different types of devices (Windows, Macs and tablets) can be found in the free *Getting started with Edison guide* available at <https://meetiedison.com/content/Get-started-with-Edison-guide-English.pdf>

Get Edison ready

To get Edison ready for use, you need to:

1. Open the battery compartment at the back of Edison and remove the EdComm programming cable.
2. Insert 4 'AAA' batteries. Please refer to the picture to ensure that the batteries are inserted correctly. Be sure to reclose the battery case by clipping the battery cover back on.

Please note: Low or flat batteries can cause a range of issues with Edison. For this reason, please be sure always to use fresh, fully charged batteries in your robots.

Choosing batteries: If using disposable batteries with Edison, only ever use alkaline batteries.

(These are the most common standard AAA batteries you will find in just about any shop.) If you are using rechargeable batteries with Edison, only use nickel metal hydride (NiMH) rechargeable batteries. Never use lithium rechargeable, heavy-duty disposable, super heavy-duty disposable or carbon zinc batteries.

3. To turn Edison on, flip the robot over. Slide the power switch to the 'on' position, as shown in the picture. This will turn Edison on, and the red LED lights will start flashing.



Ensure the batteries are in the right way.



Push the switch towards the 'on' symbol.

Please note: While Edison will turn off automatically if not used after five minutes, we recommend you turn the robots off manually when not in use.

Part 1: What is an Edison robot?

Overview

New to Edison robots? Spend just a few minutes to ‘meet’ the Edison robot and learn about the robot’s key features.

Delivery recommendations

- Recommended time: 5 minutes
- *Already familiar with Edison and EdScratch? Have students start at part 3 instead!*

Tips and tricks

- Have the time to do more than just 1 hour of coding? Using Edison’s various sensors in programs can really bring robotics to life. Choose one of Edison’s sensors and run another lesson using that sensor from the free [EdScratch Lesson Plans Set](#).

Part 2: How do you use EdScratch with Edison?

Overview

This part of the lesson draws attention to the hardware-software connection that exists in all computer systems, then introduces the online application EdScratch to students, including the names of the main areas inside the programming environment. Students will practice downloading a program from EdScratch to their Edison in the next section, so it is important that they are familiar with the user interface in this section.

Delivery recommendations

- Recommended time: 5 minutes
- *Already familiar with Edison and EdScratch? Have students start at part 3 instead!*

Tips and tricks

- Some devices, especially tablets, automatically lower the volume when they detect that an audio device, such as headphones, has been connected to the audio jack. The programming device may read the EdComm cable as ‘headphones’. Make sure the volume on the computer or tablet is still turned all the way up after the EdComm cable is plugged in.
- While a program is downloading to Edison, Edison makes a whirring sound, similar to an old dial-up modem. Once the program downloads successfully, Edison will make a chirping beep. If the program fails while downloading, Edison will make a ‘fail sound’. You can hear what the success and fail noises sound like at <https://meetedison.com/edison-robot-support/trouble-shooting>

- Make sure students do not unplug the EdComm cable until after they hear the 'success' sound.
- When many students are downloading programs at the same time, you may experience slower internet speeds, causing the program to take longer to create the 'Program Edison' button in the pop-up box and for the program to download to Edison. Remind students to listen for the success sound before unplugging the EdComm cable to ensure they wait until the program fully downloads.

Part 3: Let's try to make Edison draw a square!

Overview

This section is primarily a refresher for how to use EdScratch, it takes some basic blocks of code and puts them into a loop, but it is nevertheless important to always practice writing good code and pseudocode.

This lesson presents a good opportunity to review geometry concepts, like shapes and angles to solve a tangible problem, whilst demonstrating to students to how to effectively use loops in EdScratch.

Encourage students to move slowly through this section. Understanding the individual concepts (angles, shapes and loops) will be important for students to successfully complete the final activity of the lesson.

Delivery recommendations

- Recommended delivery time: 10 minutes
- Provide a refresher to the entire class on the different shapes, and encourage students think of shapes other than the square.
- Use the shapes you have thought of to create different shape combinations.
- As a class, use joint construction to design pseudocode for a variety of shapes. Students can be prompted to offer different coding solutions for producing those shapes.
- Ask students to challenge themselves to use the fewest number of blocks as possible to make the code more efficient.

Part 4: What are variables and iteration?

The concept of variables, one of the fundamental components of code, is introduced in this section. The student sheets explain how variables work and how they can be manipulated using *set* and *increment* blocks. Students are then asked to trace a program that they will use to construct their program for the final drawing task, which requires them to understand how a variable can be used to trigger *conditional events*.

Delivery recommendations

- Recommended time: 10 minutes
- To ensure students are understanding the concept of variables, you can ask them to answer the following two questions as a part of this section:
 1. Can you look at the code and how changing the variable will change the output?
 2. Look at the code in the program. Explain what the output of the code is and how it can be changed. What in the code makes the robot beep and why?

Tips and tricks

- Only variable names using legal characters will be able to be compiled and sent to Edison. If students get error messages when they try to download a program with variables, check the variable's name.
- For a deeper exploration on using Edison's 'set motors' blocks, refer to activity U2-2.5 *Let's explore Edison's motors* in the free [EdScratch Lesson Plans Set](#).
- When it comes to expressions and any mathematics in EdScratch programs, it is important to note that Edison can only handle integers.
- If an operator input parameter is left blank in an EdScratch program, it will be read as '0' (zero) by the robot.
 - The operator may display NaN in this case, which stands for 'not a number'.
- Due to minor mechanical differences in the motors and encoders inside different Edison robots, students may need to adjust the input parameters of the drive block related to the turn in this program to suit their robots. To improve accuracy, students can also add either a 'stop motors' block or a 'wait' block with a sort input value (e.g. .2 secs) in-between the drive commands (including at the bottom of the code inside the loop). This will be particularly important to achieving accurate turns in order for the shapes to look how they are supposed to. Even with accurate degrees in the turning blocks of code, if there is still momentum from Edison driving forward when it goes to turn the turns may be inaccurate.

Part 5: Your turn! Designing blueprints!

Overview

Building on the programming concepts explored in parts 3 and 4, this section asks students to develop a program that will allow them to draw shapes with Edison. This will require students to not only come up with the code that will allow Edison to draw the shapes, but also code that will let them iterate through a variable and choose which shape to draw according to the value of the variable.

This activity is designed to encourage students to combine their mathematical knowledge and understanding of geometry with their programming and problem-solving skills to use Edison in a creative way. It challenges students to recall what they know about shapes and build advanced sequences that require students to consider the logic behind design choices that engineers and architects make when designing structures.

The extra ‘bonus challenge’ attached to this activity offers students an opportunity to mix computer programming and physical engineering to turn their blueprints into real structures that can be tested with their Edisons!

Delivery recommendations

- Recommended time: 20 minutes
- An example programming solution can be seen at <https://cloud.edscratchapp.com?share=5DM116oD>.
- If there is extra lesson time available, as a class, look at some blueprints and spot the different shapes that are used throughout for inspiration and real-world connection.
- If students are struggling to conceptualise the steps necessary for Edison to draw a shape, encourage them to try drawing a shape by hand, without lifting their pencil. Each time they change direction with their pencil can be thought of as line of code they will need to provide Edison with to draw.

Tips and tricks

- If students are feeling stuck, suggest they review the program in section 3 and 4 to consider how shapes can be drawn, and to revisit the blocks they will need to select their shapes.
- Remind students to use good variable names and avoid illegal characters in their variable’s name.

What should I do if a student finishes the lesson early?

- Encourage students to make more blueprints and structures using variables and Edison.

- Or, ask students who finish early to help classmates who are having trouble with the activity.

Cheat sheet for shapes coding

Square:



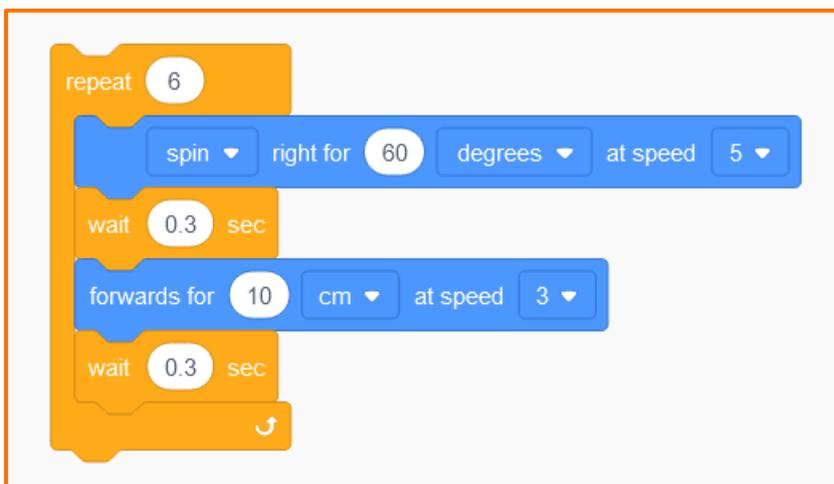
```
repeat (4)
  forwards for (10) cm at speed (3)
  wait (0.3) sec
  spin left for (90) degrees at speed (3)
  wait (0.3) sec
```

Triangle:



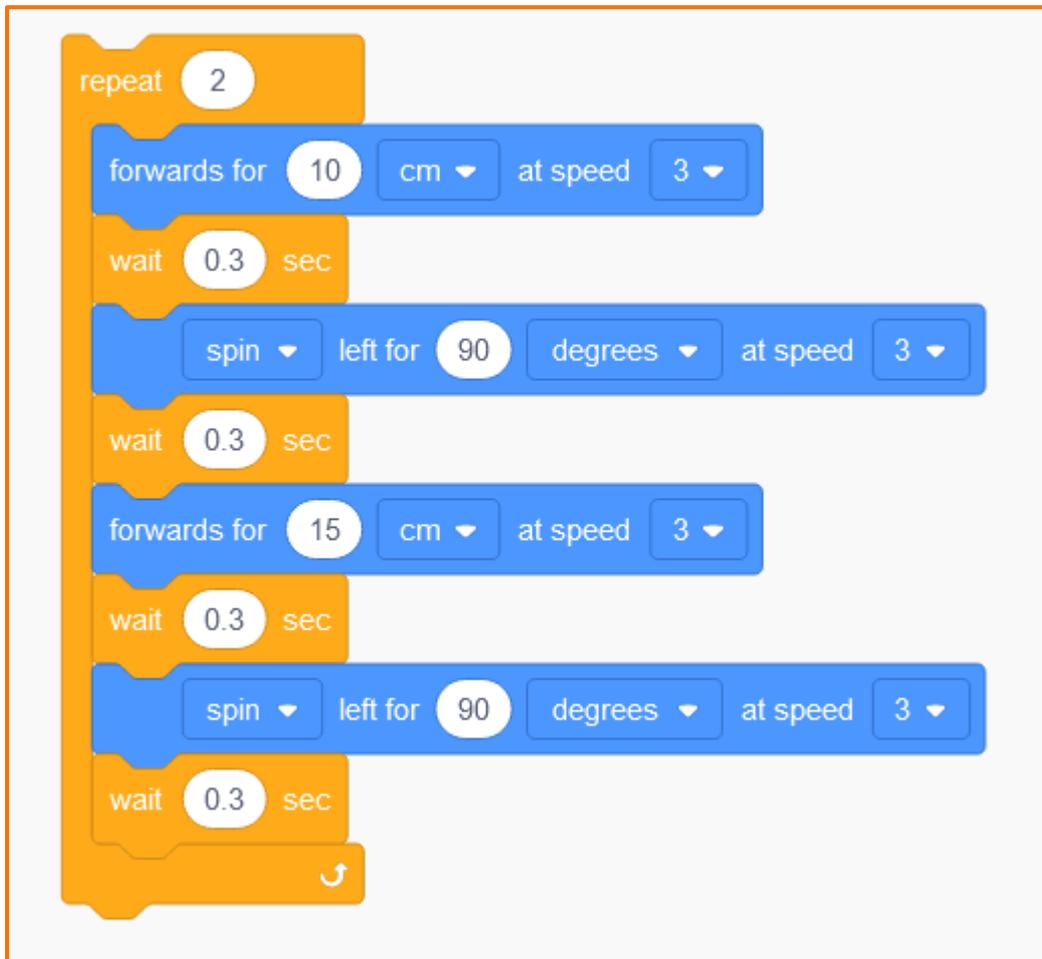
```
repeat (3)
  forwards for (10) cm at speed (3)
  wait (0.3) sec
  spin left for (120) degrees at speed (7)
  wait (0.3) sec
```

Hexagon:



```
repeat (6)
  spin right for (60) degrees at speed (5)
  wait (0.3) sec
  forwards for (10) cm at speed (3)
  wait (0.3) sec
```

Rectangle:



```
repeat (2)
  forwards for (10) cm at speed (3)
  wait (0.3) sec
  spin left for (90) degrees at speed (3)
  wait (0.3) sec
  forwards for (15) cm at speed (3)
  wait (0.3) sec
  spin left for (90) degrees at speed (3)
  wait (0.3) sec
```

The code block is a Scratch script for drawing a rectangle. It starts with a 'repeat' block set to 2 iterations. Inside the repeat block, the sequence of actions is: 'forwards for 10 cm at speed 3', 'wait 0.3 sec', 'spin left for 90 degrees at speed 3', 'wait 0.3 sec', 'forwards for 15 cm at speed 3', 'wait 0.3 sec', 'spin left for 90 degrees at speed 3', and 'wait 0.3 sec'. The repeat block ends with a circular arrow icon.

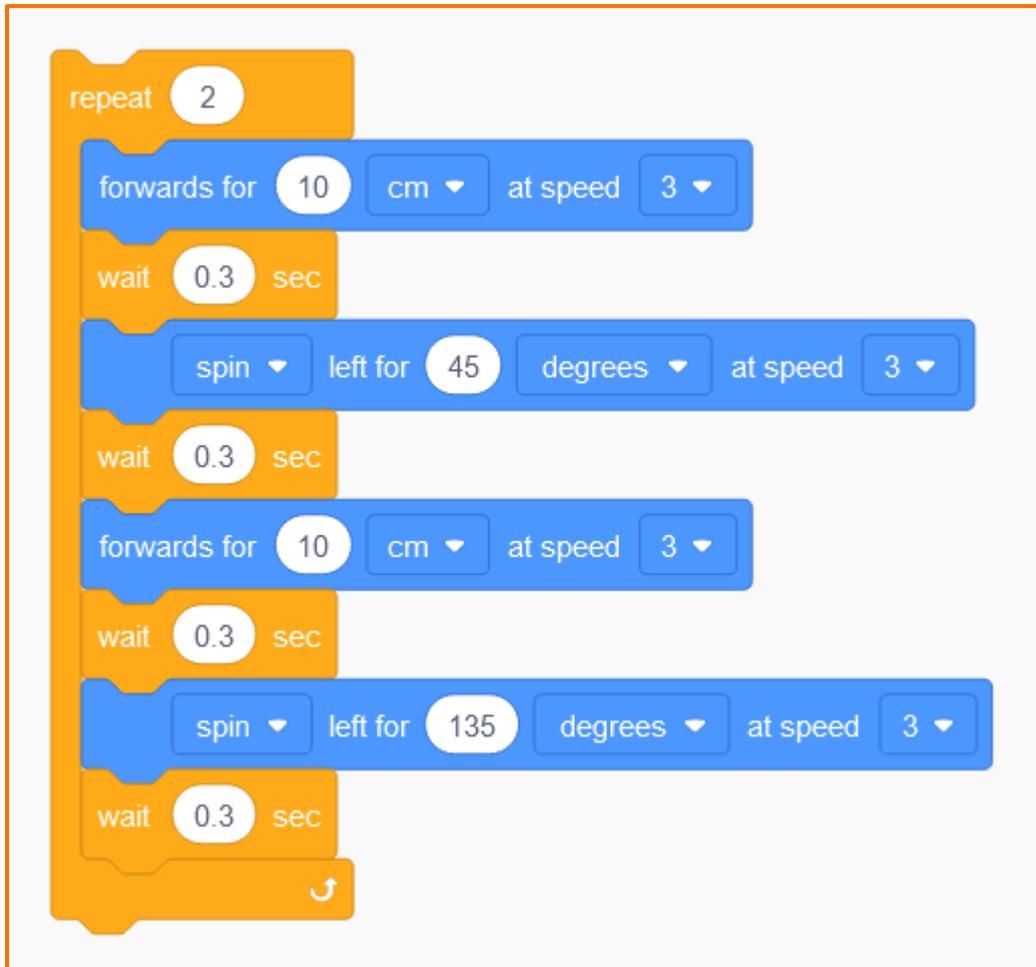
Octagon:



```
repeat (8)
  spin right for (45) degrees at speed (5)
  wait (0.3) sec
  forwards for (10) cm at speed (3)
  wait (0.3) sec
```

The code block is a Scratch script for drawing an octagon. It starts with a 'repeat' block set to 8 iterations. Inside the repeat block, the sequence of actions is: 'spin right for 45 degrees at speed 5', 'wait 0.3 sec', 'forwards for 10 cm at speed 3', and 'wait 0.3 sec'. The repeat block ends with a circular arrow icon.

Rhombus (Diamond):



```
repeat (2)
  forwards for (10) cm at speed (3)
  wait (0.3) sec
  spin left for (45) degrees at speed (3)
  wait (0.3) sec
  forwards for (10) cm at speed (3)
  wait (0.3) sec
  spin left for (135) degrees at speed (3)
  wait (0.3) sec
```

The image shows a Scratch script for drawing a rhombus. It starts with a 'repeat' block set to 2. Inside the repeat loop, there are seven blocks: a 'forwards for' block (10 cm, speed 3), a 'wait' block (0.3 sec), a 'spin left for' block (45 degrees, speed 3), another 'wait' block (0.3 sec), a second 'forwards for' block (10 cm, speed 3), a third 'wait' block (0.3 sec), and a final 'spin left for' block (135 degrees, speed 3). The script ends with a 'wait' block (0.3 sec) and a refresh icon.