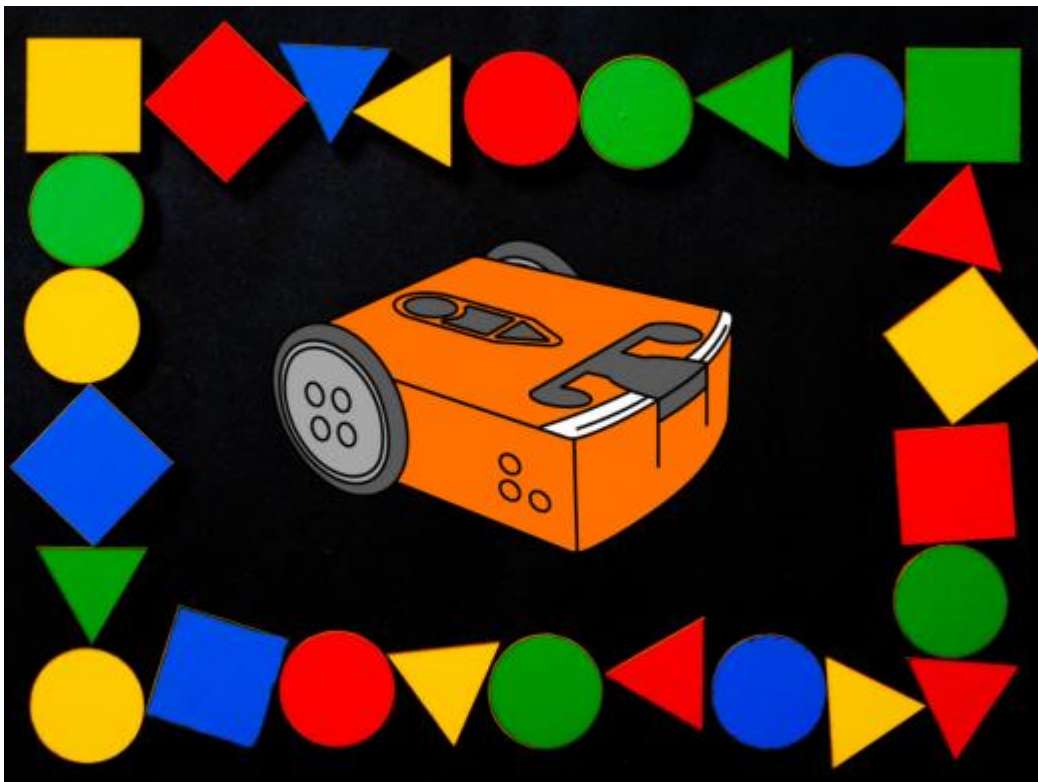


It's hip to be a square....

Teacher's notes



The [*It's hip to be a square...*](#) lesson set by Emma Grace, Emma Dewar, Kat Kennewell 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: Drive a square	9
Part 4: Drive a triangle	10
Part 5: Drive a hexagon	11
Part 6: Challenge: Choose your shape.....	12

About this lesson and guide

This guide offers teachers and instructors overview information, facilitation recommendations and other supporting information for the *It's hip to be a square* lesson available at <https://meet Edison.com/hip-to-be-a-square>.

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 *It's hip to be a square* 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](https://creativecommons.org/licenses/by-sa/4.0/).

Developed and written by: Kat Kennewell and Emma Grace

Illustrations by: Jin Peng, Emma Dewar and Emma Grace

The *It's hip to be a square* lesson set was developed using resources from the [EdScratch Lesson Plans Set](#).

Lesson overview

The computational thinking concept of repetition is the primary focus of this lesson. Using Edison and the [Scratch-based programming language EdScratch](#), students explore repetition in programming and the idea of making code more efficient by using the computer programming structure of definite loops.

Students start with having their robot create a square shape, and are then introduced to the definite loop to make their code more efficient.

They are then asked to have their robot create a triangle and hexagon shape and understand the pattern between each.

The final task in this activity includes a semi open-ended programming challenge where students can select their own shape to navigate.

Grade levels	Difficulty	Duration
Year 5+	Intermediate	60 minutes

Prerequisite knowledge

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

- Have previously used block-based coding applications
- Understand sequence and sequential programming

Computer science and computational thinking topics covered

- Sequence
- Definite loops
- Inputs-outputs

Tie-ins to other subjects

- 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

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>

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>

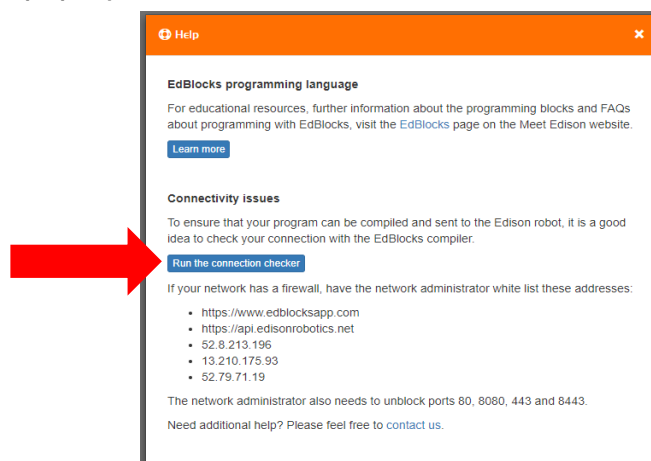
Setting up your programming devices for EdScratch

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: Drive a square

Overview

Students first write a program to have Edison drive in a square using only sequential programming, observing the repeating pattern. Definite loops are then introduced, and students recreate their 'drive in a square' program using this more efficient coding method.

Delivery recommendations

- Recommended delivery time: 20 minutes

Tips and tricks

- If students are struggling to understand the loop concept, it can be helpful to draw out what is happening in this program on paper or the whiteboard. Write down the 'drive' and 'turn' commands in sequential order and wrap them in a loop. Write the numbers 1, 2, 3, and 4 next to the loop. Use arrows to show that the robot will do the two commands in order, then move back to the top of the loop. Cross out the number 1 and then repeat the process until you have moved through all four repetitions of the loop.
- Loops make programming more efficient, but that doesn't mean that programs that don't use loops are wrong. Programming both the eight-block and three block program, observing that both programs work equally well at getting Edison to drive in a square, is a good way for students to experience the idea that 'multiple solutions are possible in coding' for themselves.
- By first writing the eight-block program, then the three-block program, students can see how programming structures can make creating programs easier and more efficient.
- If Edison's wheels catch the edge of the activity sheet's paper, this can throw the robot off slightly. You can fix this by taping the activity sheet down or replicating the activity sheet's pattern on a larger piece of paper.
- Due to minor mechanical differences in the motors and encoders inside different Edison robots, some robots may not turn to exactly 90 degrees when

given the input of 90. Encourage students to try different values around 90 (e.g. 87 or 93) to find the input that works best for their Edison.

- When using programs that run multiple ‘drive’ commands consecutively, students may note that their Edison robot’s accuracy is reduced as more and more commands run. Adding a pause (using either a ‘wait’ with a very short input value or a ‘stop motors’ block as appropriate) between ‘drive’ commands allows the motors to fully stop moving, which will increase accuracy.
- Students may also enjoy having Edison draw their shapes by using either our [EdSketch kit](#) or LEGO compatible bricks to build a pen holder (examples are available [here](#)).

Answer key

Question	Type	Answer/Sample answer	Marking notes
1	EA	8	If students only have seven blocks, their robot likely doesn’t complete the final turn, which returns it to the starting position, as instructed on the worksheet.
2	SE	<i>There is a pattern to the blocks. Only two block types are being used with the same input parameters, and they repeat four times.</i>	Students should note a pattern of repetition regarding the blocks being used and/or the order of the blocks in the program.
3	EA	4	
4	SE	<i>The value needs to be 4 because a square has four sides and four corners. So, the code needs to drive each side and turn each corner a total of 4 times.</i>	Students will ideally note a correlation between the number of looped actions required and the shape’s sides/angles

Part 4: Drive a triangle

Overview

Students practice using the definite loop programming structure to write an efficient program to get Edison to drive a triangle. This activity is designed to be used in conjunction with the triangle activity page. This will help students master the connection between the number of loops in a definitive loop and the resulting output.

Delivery recommendations

- Recommended delivery time: 10 minutes

Tips and tricks

- This activity lends itself well to a geometry extension lesson.
- It can be tricky for students to work out the value of the three angles (in degrees) inside the triangle. Remind students that the sum of the interior angles of a triangle is 180° and that the triangle on the activity sheet is an equilateral triangle.
- Tape the activity sheet down, or replicate the pattern on a larger piece of paper to prevent Edison's wheels catching on the edge of the paper.

Answer key

Question	Type	Answer/Sample answer	Marking notes
1	SE	3	The minimum number of blocks needed for a successful program is 3: a loop, a drive forward block and a turn block.
2	EA	3	
3	SE	<i>The value needs to be 3 because a triangle has three sides and three corners. So, the code needs to drive each side and turn each corner a total of 3 times</i>	Students will ideally note a correlation between the number of looped actions required and the shape's sides/angles.

Part 5: Drive a hexagon

Overview

Students practice using the definite loop programming structure to write an efficient program to get Edison to drive a hexagon. This activity is designed to be used in conjunction with the hexagon activity page to help students master the connection between the number of loops in a definitive loop and the resulting output.

Delivery recommendations

- Recommended delivery time: 10 minutes

Tips and tricks

- This activity lends itself well to a geometry extension lesson.

- Working out the value of the angle (in degrees) that the robot needs to turn in order to trace the hexagon can be tricky. Remind students that the sum of the interior angles of a hexagon is 720° and that the hexagon on the activity sheet is a regular (meaning that all sides are equal) 6-sided shape.
- Tape the activity sheet down, or replicate the pattern on a larger piece of paper to prevent Edison's wheels catching on the edge of the paper.

Answer key

Question	Type	Answer/Sample answer	Marking notes
1	SE	3	The minimum number of blocks needed for a successful program is 3: a loop, a drive forward block and a turn block.
2	EA	6	
3	SE	<i>The value needs to be 6 because a hexagon has six sides and six corners. So, the code needs to drive each side and turn each corner a total of 6 times</i>	Students will ideally note a correlation between the number of looped actions required and the shape's sides/angles.

Part 6: Challenge: Choose your shape

Overview

Students apply their understanding of the relationship between a definite loop and a regular shape by choosing a shape and writing an efficient program for Edison to drive that shape. This activity asks students to use the patterns they have noticed when using definite loops to draw shapes and then to extrapolate to explore other shapes.

Delivery recommendations

- Recommended delivery time: 15 minutes

Tips and tricks

- To complete a shape using a 3-block program consisting of a definite loop which contains a 'drive' block and a 'turn' block, students must select a regular shape. However, if students decide not to use a regular shape, they can still attempt the challenge

Answer key

Question	Type	Answer/Sample answer	Marking notes
1	EA	12	

2	SE	<i>My shape was an octagon, which has eight sides and angles. You always need to have the input parameter be equal to the number of repeating sides and angles that the shape has, so I knew that I would need an 8 as the repeat block's input parameter.</i>	Students will ideally note a correlation between the number of looped actions required and the shape's sides/angles.