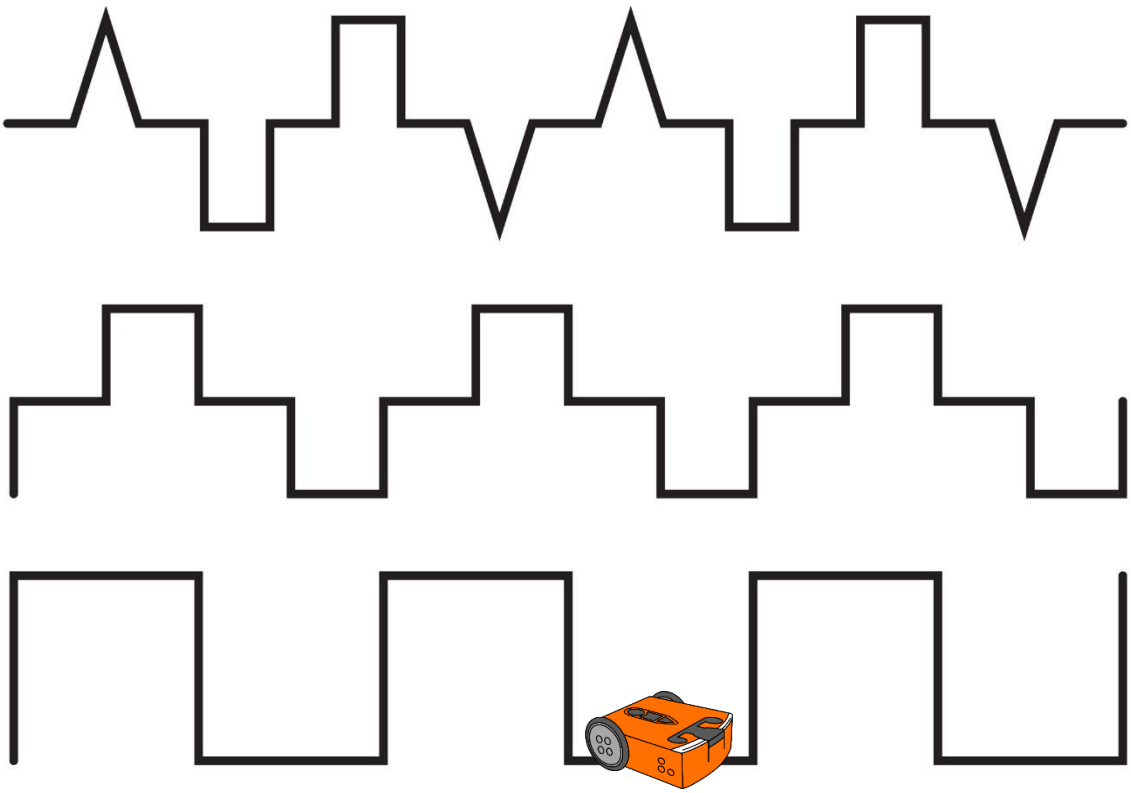
Edison the designer

Teacher's notes



The *Edison the designer* lesson set by 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
Part 1: What is a loop? 5
Part 2: What are nested loops? 6
Part 3: How do you program a pattern inside a pattern? 7

Go ahead – show off!

We love seeing how classrooms use Edison! If you and your students want to share your Hour of Code Edison EdVenture, be sure to tag us into the fun!

@meetedison twitter.com/meetedison

@meet_edison instagram.com/meet_edison

@meetedison facebook.com/meetedison

EdScratch is developed by Microbric Pty Ltd using open source software created and maintained by the Scratch Foundation. The Scratch Foundation does not sponsor, endorse, or authorize this content. See scratch.mit.edu for more information.

About this lesson and guide

This guide offers teachers and instructors overview information, facilitation recommendations and other supporting information for the *Edison the designer* lesson available at https://meetedison.com/robotics-lesson-plans/edison-designer

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

As long as 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 key computer science concepts and practicing skills along the way. 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.

Answer key
Solutions or examples to the included questions.

Creative Commons licence attribution details

The *Edison the designer* lesson set is comprised of the student sheets and this guide. This set was developed using resources from the EdScratch Lesson Plans Set[1] and is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License[2].

Developed and written by: Kat Kennewell
Illustrations by: Jin Peng

[1] https://meetedison.com/robot-programming-software/edscratch/#EdScratch-resources
[2] http://creativecommons.org/licenses/by-sa/4.0/

Lesson overview

Introduce the key computational concepts of loops, nested loops and sequential order using Edison robots and the Scratch-based programming language EdScratch[3]. This lesson centres on the idea of patterns, including geometric shapes, and explores how loops can be used in programs which contain repeating code.

Grade levels	Difficulty	Prerequisites	Duration
Year 6+	Beginner	None[4]	55+ minutes

Computer science and computational thinking topics	☐ Sequence ☐ Definite loops ☐ Nested loops ☐ Pattern recognition ☐ Decomposition
Tie-ins to other subjects	☐ Mathematics: geometry ☐ Art: visual design

Supplies you need

- Full set of Edison robots
- Full set of prepared programming devices (computers or tablets)
- Print or digital copies of the student sheets and print-outs of the activity sheets
- Space and supplies for test-spaces [such as large paper and markers]
- **Optional**: supplies for the bonus challenge [such as EdCreate kits, LEGO bricks, rubber-bands and other similar materials, plus pens]

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! (^_^)

[3] https://meetedison.com/robot-programming-software/edscratch/

[4] You and your students should be familiar with Edison and EdScratch already, however. The *Getting started with Edison and EdScratch* set available at https://meetedison.com/robotics-lesson-plans/edison-designer has a step-by-step activity for introducing Edison and EdScratch. If this is your first time using Edison or EdScratch, start with that guide and activity.

Part 1: What is a loop?

Overview

Loops, one of the most basic programming control structures, are introduced in this activity. 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 time: 15 minutes
- This informational section provides students with the foundations for the rest of the activity. You may choose to run this section as a facilitator-led demonstration rather than having students complete the tasks independently. Doing so will enable students to spend more time on the creative tasks in Part 2 and Part 3.

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 plus 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 or with tape on the floor.
- 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.

- This section highlights some geometry concepts, such as angles. You may want to highlight this connection to your students.

Answer key

Question	Answer/Sample answer	Notes
1	*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	*There is a pattern to the blocks. Only two block types are being used with the same input parameters, and they repeat four times.*	Students should note a pattern of repetition regarding the blocks being used and/or the order of the blocks in the program.
3	*4*	
4	*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.*	Students will ideally note a correlation between the number of looped actions required and the shape’s sides/angles.

Part 2: What are nested loops?

Overview

Students are introduced to the concept of using multiple loops in a program both by stacking them sequentially and by nesting one loop inside another. Nesting loops may seem like a new way of manipulating the flow of a program, but it really is just layering sequence. Working with programs with stacked and nested loops allows students to apply the computational thinking practices of decomposition (chunking problems into smaller parts) and pattern recognition as well as the application of sequential logic.

Delivery recommendations

- Recommended time: 20 minutes
- Have the time to do more than just 1 hour of coding? Deliver Parts 1 and 2 together as one lesson, then do Part 3 as a separate lesson. This will allow students ample time to both practice the core concepts from Parts 1 and 2 and create a pattern of their own in Part 3 of this lesson.

Tips and tricks

- If students are struggling to understand nested loops, it can be helpful to draw out what is happening on paper or the whiteboard. You can do this by writing down each command one at a time in sequential order with your students or by writing down a program as it appears with nested loops and then use arrows to indicate each looping action.
- 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 or with tape on the floor.
- If students are having trouble getting their robots to turn the exact angles of the pattern, or are finding the accuracy of the robot's driving is reducing, remind them to:
    - try adjusting the degree input to account for the minor mechanical differences in the motors and encoders inside different Edison robots, and
    - add a pause (using either a 'wait' with a very short input value or a 'stop motors' block as appropriate) between 'drive' commands to allow the motors to fully stop moving, which will increase accuracy.

Answer key

Question	Answer/Sample answer	Notes
5	*It's three squares in a row.*	This question is designed to help students see the correlation between the repeating pattern and how a nested loop may help in their program. If students are struggling to get the robots to drive the pattern, ask them "How could you use a nested loop to get Edison to drive a square three times?'

Part 3: How do you program a pattern inside a pattern?

Overview

Give students the chance to put loops and sequence to the test by getting Edison to drive in some fabulous designs. Have students look at designs to identify repeating patterns, decompose the actions required to drive the robot in the pattern, then construct a code solution as effectively as possible using loops and nested loops.

Delivery recommendations

- Recommended time: 20+ minutes

- This section lends itself extremely well to collaboration. Having students work in pairs or small groups will allow them to work together to decompose the patterns and create nested loop programs.
- If students are having trouble getting their robots to turn the exact angles of the patterns, or are finding the accuracy of the robot's driving is reducing, remind them to:
    - try adjusting the degree input to account for the minor mechanical differences in the motors and encoders inside different Edison robots, and
    - add a pause (using either a 'wait' with a very short input value or a 'stop motors' block as appropriate) between 'drive' commands to allow the motors to fully stop moving, which will increase accuracy.
- The bonus challenge brings in engineering and art to physical computing, but can take a while. Students can utilise EdSketch if available or if you have the opportunity, consider making this activity its own lesson to allow students the chance to explore the design-build-test cycle as they design their pen attachment. You can see ideas for how to run this bonus challenge at https://meetedison.com/teach-engineering-design-with-edcreate/

Tips and tricks

- Physically creating the design by marking it out or drawing it can help students recognise the repetitions that occur inside the pattern and better connect these to the loop structures they need inside their EdScratch code solution.
- Having students make their own patterns is a great way to have them think critically about what repetition looks like in practice. Creating a design which has a pattern inside a pattern is a great chance to apply computational thinking to design.

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

- Encourage students to make another pattern and program Edison to drive it using nested loops.
- Even if you don't plan on having students complete the bonus challenge, fast finishers can sketch out or write a description for a design concept for a pen attachment.
- You can always ask students who finish early to help classmates who are having trouble with the activity.