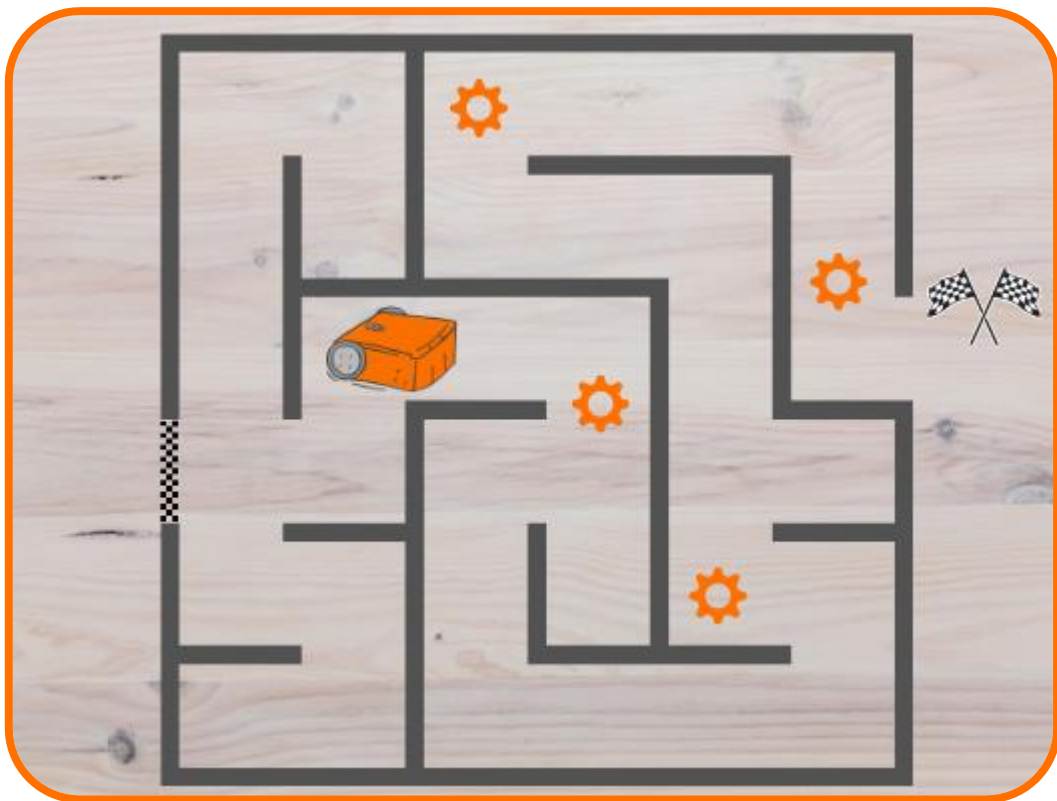




# Edison the maze racer

## Teacher's notes



The *Edison the maze racer* lesson set by [Robert Spargo](#) is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](#).

## Contents

About this lesson and guide .....	3
Lesson overview .....	4
Section 1 – What is a Euler Path?.....	5
Section 2 – What is an input?.....	5
Section 3 – What is an output? .....	6

### 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](https://twitter.com/meetedison)



@meet\_edison [instagram.com/meet\\_edison](https://www.instagram.com/meet_edison)



@meetedison [facebook.com/meetedison](https://www.facebook.com/meetedison)

## About this lesson and guide

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

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<sup>1</sup>, you can start learning along with your students! The student sheets for this lesson have been designed to allow students to work through the stages 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 section of the lesson is included in this guide along with any relevant supporting information for that section. Supporting information is divided into the following sections:

### How it works

Information about the technology and computer science concepts being highlighted in that section.

### 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 maze racer* lesson set is comprised of the student sheets and this guide. This set was developed using resources from the [EdBlocks Lesson Activities Collection](#)<sup>2</sup> and the [EdScratch Lesson Plans Set](#)<sup>3</sup> and is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](#)<sup>4</sup>.

Developed by: Robert Spargo

Original illustrations by: Emma Dewar

---

<sup>1</sup> The *Getting started with Edison and EdBlocks* set available at [Edison the maze racer](#) has step-by-step help for setting up your robots and programming devices. If you are new to Edison or EdBlocks, it is recommended you start with that guide.

<sup>2</sup> <https://meetiedison.com/robot-programming-software/edblocks/#edblocks-resources>

<sup>3</sup> <https://meetiedison.com/robot-programming-software/edscratch/#EdScratch-resources>

<sup>4</sup> <http://creativecommons.org/licenses/by-sa/4.0/>

## LESSON OVERVIEW

Introduce the key computational concepts of inputs and outputs using Edison robots and the [EdBlocks programming language](#)<sup>5</sup>. This lesson utilises the Edison robot’s drive function through a set of progressive programming tasks.

For students to create programs for the Edison robot to ‘trace’ (ie. drive in the pattern of) the Euler path shape mazes of choice requires the application of sequential thinking and an understanding of the input-process-output cycle.

Grade levels	Difficulty	Prerequisites	Duration
Year 2 – Year 6	Beginner	None <sup>6</sup>	40-50minutes

### Computer science and computational thinking topics

- Sequential programming
- Inputs (including input parameters)
- Outputs
- Pattern recognition
- Decomposition and problem solving

## Supplies you need

- Full set of Edison robots and EdComm programming cables
- Full set of prepared programming devices (computers or tablets)
- 4x AAA batteries per robot
- Print-outs or digital copies of the student sheets
- Print outs of the Euler path maze activity sheets
- Blank paper and black markers for students to create their own mazes
- EdSketch kit or Lego compatible bricks to build a pen holder

### 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! (^\_^)***

<sup>5</sup> <https://meetiedison.com/robot-programming-software/edblocks/>

<sup>6</sup> You and your students should be familiar with Edison and EdBlocks already, however. The *Getting started with Edison and EdBlocks* set available at [Edison the maze racer](#) has a step-by-step activity for introducing Edison and EdBlocks. If this is your first-time using Edison or EdBlocks, start with that guide and activity.

## Section 1 – What is a Euler Path?

Start off the lesson with a quick exploration of Euler Paths and Euler Circuits. Euler circuits and Euler's formula are very important for mechanical engineering and are great to introduce to students early, but the complex formulae may be confusing for younger students.

### How it works

Ask students if they have ever tried drawing shapes or letters without lifting their pencil. A Euler path shape can be explained by showing a variety of different shapes and seeing which ones can be traced without lifting your pencil or backtracking. Discuss what the shapes have in common with each other and see if your students can identify the characteristics of a Euler path shape. As a class, come up with a few definitions! You can show a few different Euler path shapes on the whiteboard, there are examples attached to this guide.

### Delivery recommendations

- Recommended time: 5-10 minutes

### Tips and tricks

- If you have time, letting students trace or draw some shapes themselves would be a good way to plan and familiarise themselves with the instructions they will need to provide Edison with.

## Section 2 – What is an input?

Before starting to program Edison, students should have a grasp of sequence and what an input is.

### How it works

Understanding the input-process-output cycle and the roll of sequence in structuring programs is one of the most important aspects of learning to code.

### Delivery recommendations

- Recommended time: 10 minutes
- If students are new to programming, you may want to work through an example by describing the sequential order required to move in the pattern of a shape (square or triangle should be relatively simple). You can have students then translate this pattern into code blocks and discuss why it's important that the blocks are arranged in a particular order to get the outputs you are looking for.
- If students are new to EdBlocks, you may want to showcase each of the three drive blocks (drive forward, turn left and turn right) to the whole class.

## Tips and tricks

- If this is your student's first-time using sequence, try asking them to develop a simple set of sequence instructions for a task such as making a sandwich and have another student act out the steps. This is a great way to show students the importance of sequence and encourages them to not overlook steps.

## Section 3 – What is an output?

Explore the output of code input in action. Students create their own program to drive a Euler path maze. The mazes are attached to the student handouts and will need to be printed in at least A3.

### How it works

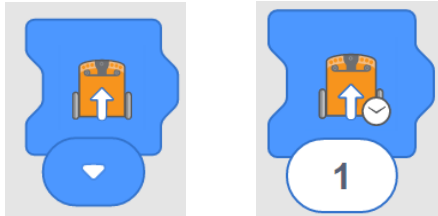
This section introduces the concept of an output. Students then work through two programming tasks. First, they choose a Euler path maze (using the activity sheets) and use the 'drive' block category in EdBlocks to create a program to direct Edison through the maze. The second task is a semi-open-ended challenge, asking students to design their own Euler path maze (or solve another maze via the activity sheets), then program their robot to drive it. There's no real 'right' or 'wrong' way to complete this programming section. Any program that meets the criteria of 'drive Edison through the maze' is okay. This is an important concept for students to begin to understand. There are many solutions to a problem in programming, which makes it highly creative!

### Delivery recommendations

- Recommended time: 35 minutes
- Ensure the mazes are printed on at least A3 paper, A4 will likely be too small.
- If students are having a difficult time getting Edison to navigate the maze, it may be beneficial for them to change the speed setting to slow, so they can more accurately control the actions of Edison.
- Resilience is key in programming! If students are getting stuck with this section, help them break down what they are trying to do into smaller chunks, then work out a solution for each part one at a time. Practicing decomposition in this way is a valuable computational thinking skill.
- If students are new to EdBlocks, be sure to share with them that any code blocks in the programming area that are NOT attached to the 'start' block will not download to the robot. This is really helpful in this section because students can test sections of their coding solution by attaching just those blocks to the 'start' block in EdBlocks, then downloading and running the program in Edison. They don't need to delete the other blocks. Instead, they can leave the rest of their program in the programming area unattached to the 'start' block. This way students don't need to start over from scratch each time they want to check just one section of their code.

## Tips and tricks

- There are several different types of drive blocks in EdBlocks. For this activity, students need to use both the 'drive forward' and the 'drive [direction] for [time]' blocks:



Remind students to use the right type of 'drive' block in the right spot of the program.

- Tell students they can change the time by clicking on the number and typing in their desired time, anything from 0.01 to 320. Time is in seconds.
- If students are having issues with Edison turning, advise them to add in some wait blocks between movements, as this will allow Edison to stop momentum before attempting a turn.
- The distance parameter may have to be slightly changed after a turn as Edison does not turn perfectly on a spot and instead drives slightly forward. It is very likely the mazes will need to be solved with a decent amount of trial and error.
- 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.
- If you want to highlight the idea that many different solutions are possible, split your students into groups for this activity and give each group the same maze to work with. Have each group demonstrate their solution for the class. Discuss the differences and similarities in each group's program and highlight how, even if they are different, by meeting the criteria, they are still all correct. Even if the students choose the same route through the maze, it is possible that the programs that they wrote will be different.

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

- Encourage students to write another program for Edison in EdBlocks. What do the other blocks do? What can they get their Edison robot to do using code in EdBlocks? Design a maze that incorporates making noises or flashing lights at certain spots.
- Or, ask students who finish early to help classmates who are having trouble with the activity.

Euler Path shape examples:

