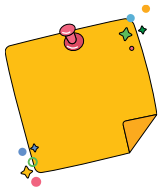
Name _____

Do you want to know a secret about working with computers and writing code?

Here it is:

SOMETHING ALWAYS GOES WRONG!

Okay, so that isn't actually a secret, but it is really important. A major part of **computational thinking** and working with computers is problem-solving.



Don't forget

Computational thinking means thinking about a problem or task in a way that is similar to how a computer thinks. It is a way of planning, problem-solving and analysing information the same way a computer does.

When things don't work the way you want, remember that is okay! It just means it is time to problem-solve. One of the main types of problem-solving you do in computer science is finding and fixing **bugs.**

Debugging is a major part of coding and using robots. People who work as computer scientists, computer programmers or roboticists professionally spend a lot of time debugging. Probably even more time than they do writing their code!

Jargon buster

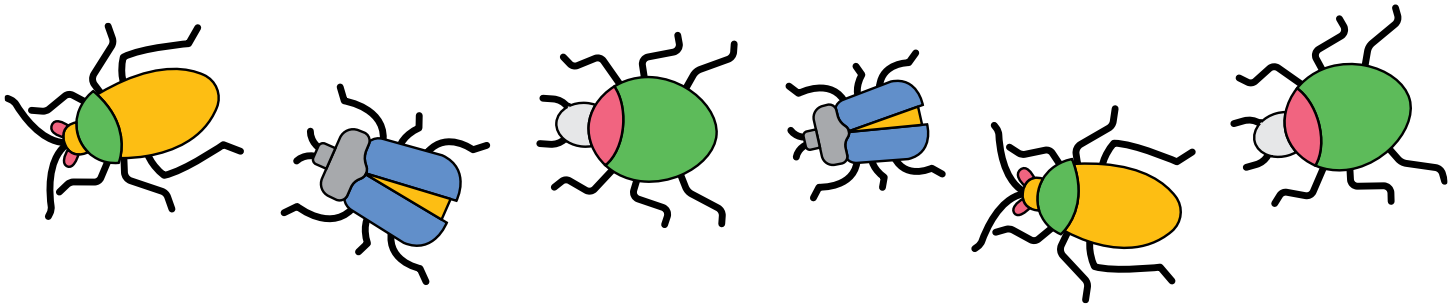
When something isn't working in a computer program, it is called a **bug.**
Finding and fixing bugs in a computer program is called **debugging.**

Why is that?

Calling something that's going wrong a 'bug' might seem a bit strange, but that's really what these errors are called.

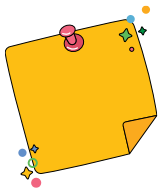
Why are computer problems called **bugs**? A woman named Grace Murray Hopper, who is one of the inventors of modern computer programming, came up with this term. She once discovered that the issue causing her computer to malfunction was an actual moth that had gotten into the hardware! She fixed the problem by literally **debugging** her computer.

The name stuck and now problems with software or hardware in computers are called **bugs** and fixing them is called **debugging**!



Debugging in EdScratch

The bug box in the EdScratch programming environment is a special feature to help you find and fix any bugs in your code.



Don't forget

The **bug box** is the area below the block pallet and programming area in EdScratch. If there is a bug or if it seems like something isn't quite right in a program, a warning message will show up in the bug box.

The warning messages that appear in the bug box are there to give you information about any problems or potential problems in your code. These messages are EdScratch's way of saying that there is an error in your code, or, that you might find an error when you run your program in Edison. In coding, there are two main types of errors: **syntax errors** and **logical errors.**

Jargon buster

Syntax is the rules of how a programming language works. All languages have rules. For languages people speak, like English or Chinese, there are rules about spelling and grammar and how to write the letters or characters in that language. Syntax is the same thing but for a computer language.

Syntax errors are caused by problems in how you wrote your code which break the rules of the language. These errors are sort of like typos or spelling mistakes in writing.

Logic is an organised way of thinking that makes sense to a computer. Logic determines the flow of a program, how you order things inside a program and what inputs you use to generate the outputs you want.

Logical errors are problems with the logic of a program. Logical errors are basically problems with the way of thinking in a program. Programs that do not make sense to the computer and programs that ask a computer to do something that it cannot do are examples of logical errors. If a program works differently than you expected it to work, there is a good chance that there's a logical error somewhere.

Understanding if a bug is the result of a syntax error or a logical error can help you fix the problem. If there is a syntax error in your EdScratch program, you will get a red warning message in the bug box, and you won't be able to download the program to Edison. If you can download your program and run it in Edison, but it doesn't do what you want, that probably means there is a logical error.

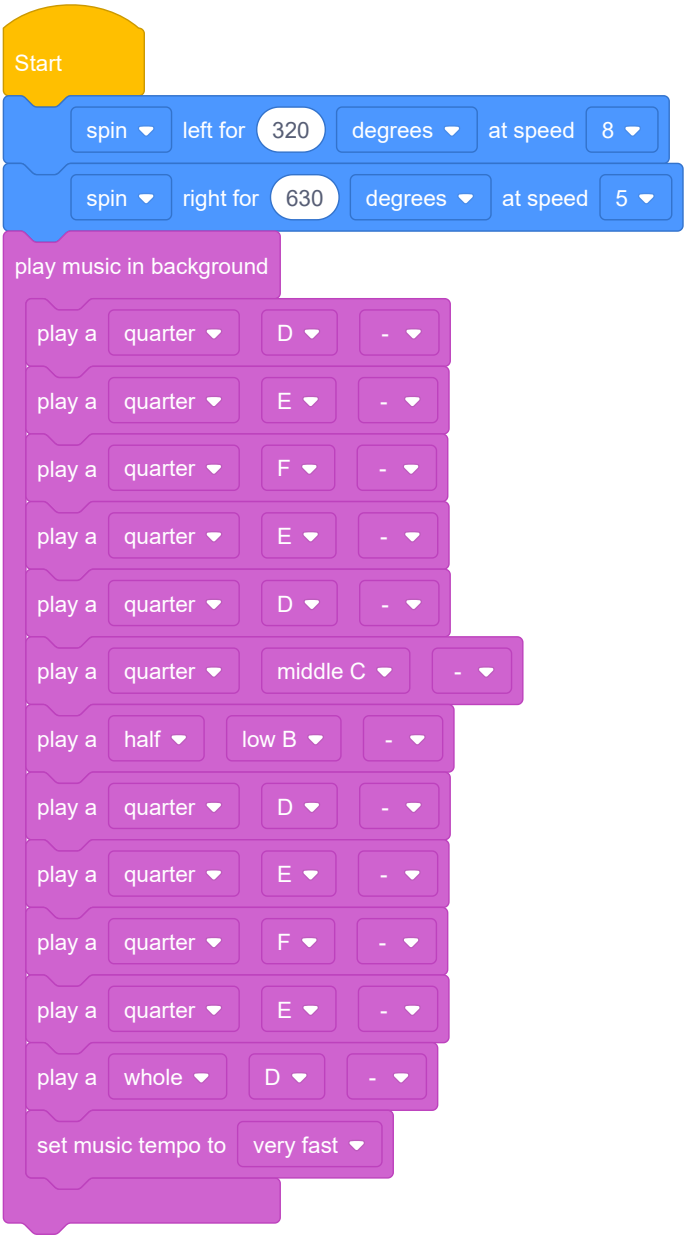
Hint

One of the best ways to see what's going wrong – and right – with a program is to test it out using EdScratch and Edison!

Name _____

Try it out

Look at the program below, it is full of bugs. It is your job to fix it!



This is what the programmer who wrote this code said about what the program is meant to do:

"I want the Edison robot to play a tune very fast. While the music is playing, I want the robot to spin left for one full circle (360 degrees), then spin back right the same distance at the same speed."

The programmer who wrote this program made two errors with the **set music tempo** block. One error is a **syntax error**, and one error is a **logical error.**

Write an explanation of each of these two errors to explain them to the programmer.

Hint: the programmer wants the music to play at a very fast tempo.

Syntax error:

Logical error:

The program does not work the way the programmer wants. You need to debug the program and get it working for the programmer. Test your program using Edison and keep debugging until the program works just the way the programmer explained. Describe the bugs you found and what you did to fix them.

Name _____

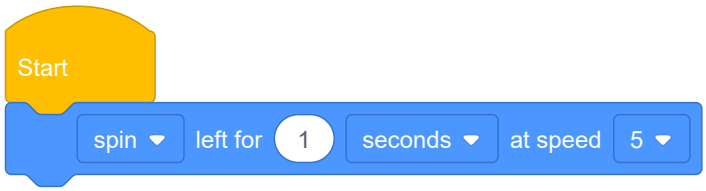
Let's explore Edison's motors

Edison robots have two motors: one on the left side and one on the right side. Outputs using these motors are one of your Edison robot's three main types of outputs. In EdScratch, the blocks related to motor outputs are in the **Drive** category.

When you write a program for Edison using blocks from the **Drive** category in EdScratch, you are telling the motors what to do. Most of the blocks control both of Edison's motors. Does that mean that both the motors do the same thing?

Spin that robot

If you wanted to write code to tell your robot to 'spin left' you can make a simple one-block program like this:



The input parameters in that block tell Edison the direction, the distance, the distance units and the speed you want the robot to use in the program.
The direction input parameter that has been selected is **spin,** which means the whole direction input is **spin left**. What is that input telling Edison's motors to output?
In EdScratch, write the program using the same input parameters as the one in the picture. Download the program and run it with Edison on the desk or floor.

Now run the program again, but this time hold Edison in your hands.
Feel how the wheels are moving. What do you notice?

Which direction is the left wheel moving? _____

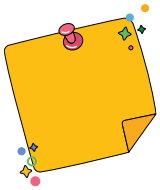
Which direction in the right wheel moving? _____

Edison's motors don't have to both do the same thing at the same time. Does that mean you could write a program moving just one of the motors? Can you write a program that tells each motor what to do separately?

Open the EdScratch app and look at the **Drive** category in the block pallet. Look at the different blocks and see if you can discover blocks you could use if you only wanted an output from one of Edison's motors.

Which blocks do you think only use one of Edison's motors? Why do you think that?

You can use Edison to build and invent lots of different things. Imagine you need to create something using Edison which only uses one of Edison's motors. What could you build? How would your creation use the one motor?



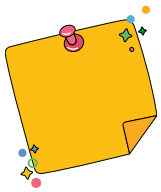
Don't forget

The wheels of your Edison robot can be removed from the powered sockets they sit in. These sockets are what Edison's motors actually move.

Direction = forward

To get Edison to work the way you want, you have to make sure you give the robot all the information it needs. If the robot doesn't have all the inputs and instructions it needs, the program probably won't work the way you want. This kind of logical error can be frustrating, especially if you think you have given the robot all the information necessary.

One of the main ways you give information to the robot using EdScratch is through input parameters.



Don't forget

Each input parameter in a block gives a different piece of information to Edison that the robot will need to be able to run that command. Input parameters are sort of like the answers to questions the robot has about what you are asking it to do.

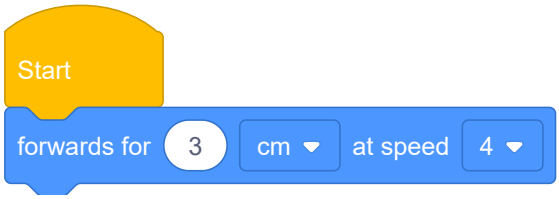
In EdScratch, some blocks get all the information they need from their own input parameters. Other blocks get information from inside their block, but also need information from somewhere else in the program as well.

Let's make a program for Edison to get the robot to move its motors. For this program to work, there are four questions you need to make sure your program answers:

Question 1: What direction do you want the robot to go?
Question 2: How far do you want the robot to go?
Question 3: What units are you using to measure distance?
Question 4: How fast do you want the robot to go?

Your program needs to give the robot an answer to all of those questions.

Look at this program:



If you ran this program in an Edison robot, would the robot have all the information it needed to know what to do? In other words, does this program tell the robot the direction, distance, distance-units and speed to move the motors?

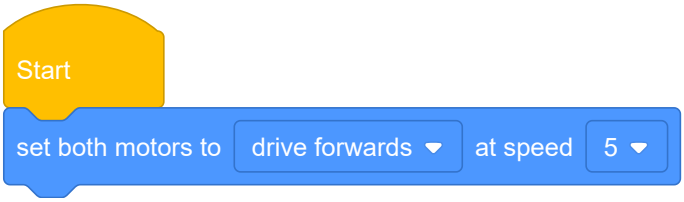
Fill in this chart:
If the information is in the program, write the value of that input in the 'value' column. For example, the value of 'distance' is the answer to the question, 'how far is this program telling the robot to go?'

Information	In the program?	Value
Direction		
Distance		
Distance units		
Speed		

Write the program in EdScratch, download it and run it in your robot.
What did the robot do when you ran the program?

Name _____

Now look at this next program:



Does this program give the robot all the information it needs?

Fill in this chart:
If the information is in the program, write the value of that input.

Information	In the program?	Value
Direction		
Distance		
Distance units		
Speed		

Write the program in EdScratch, download it and run it in your robot.

What did the robot do when you ran the program? Why did it behave that way?

How can you give the robot the rest of the information it needs so that the robot moves its motors? Experiment in EdScratch to see if you can write a program that uses the **set both motors** block but no other blocks from the **Drive** category and gets the robot to move forward.

What did you try? Did it work? Write what you did, what worked, what didn't work and how experimenting made you feel.

Hint

Don't give up! Experimenting, testing and problem solving is how you learn new things in coding.

Feeling stuck? Think about what it is you are trying to do a different way. Look at different blocks in EdScratch and ask yourself, 'if I use this block, will it fix the problem I am trying to solve?'

If you aren't sure, try it and see what happens! Be sure to check the bug box for hints too!