Name _____

Let's explore stacking and nesting loops

Writing programs using loops lets you be more efficient because you can get a computer to repeat actions without needing to write out the commands multiple times. Using loops also lets you tell a program to do something forever, which you couldn't do if you had to write out every command one by one.

You can also use more than one loop in a program, and you can use the loops in different ways: by stacking the loops together or nesting loops inside other loops.

Jargon buster

In block-based programming languages like EdScratch, adding blocks together is sometimes called **stacking** blocks and a program is sometimes called a **stack** or a **block stack.** That's why if you use multiple loops together in a program one after another, you can say you are **stacking** the loops.

You can also put a loop block inside another loop block. This is called **nesting** loops.

Why would you use loops in stacks or by nesting them together? Using multiple loops together in this way lets you write programs with repeating patterns. You can even write programs with patterns that repeat inside of other patterns.

Stacking and nesting loops have different uses. By stacking loops, we can write programs to get Edison to do different sets of actions multiple times, then move on to a new set of repeated actions. By nesting loops together, however, we can write programs to get Edison to repeat whole patterns multiple times.

Why is that?

Think about an alarm clock on a mobile phone. The alarm can be set to go off in the morning at 7:00 AM. You can set the phone to repeat that alarm every day. When the alarm goes off, it beeps on and off a set number of times. If you snooze the alarm, it stops for a certain amount of time, then comes back on, beeping on and off again for a set number of times.

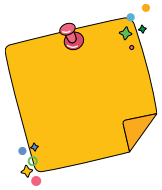
Can you see how there are repeating patterns inside of other repeating patterns?

This is an example where using stacked and nested loops to write a program would be very helpful. That's because stacking and nesting loops lets you repeat whole sets of commands inside your program.

Stacking and nesting loops have different uses. By stacking loops, we can write programs to get Edison to do different sets of actions multiple times, then move on to a new set of repeated actions. By nesting loops together, however, we can write programs to get Edison to repeat whole patterns multiple times.

What's going to happen?

Programs that use multiple loops, especially nested loops, might seem a bit confusing at first. To understand what the program is going to do, you need to think about each action that is going to happen in sequence.



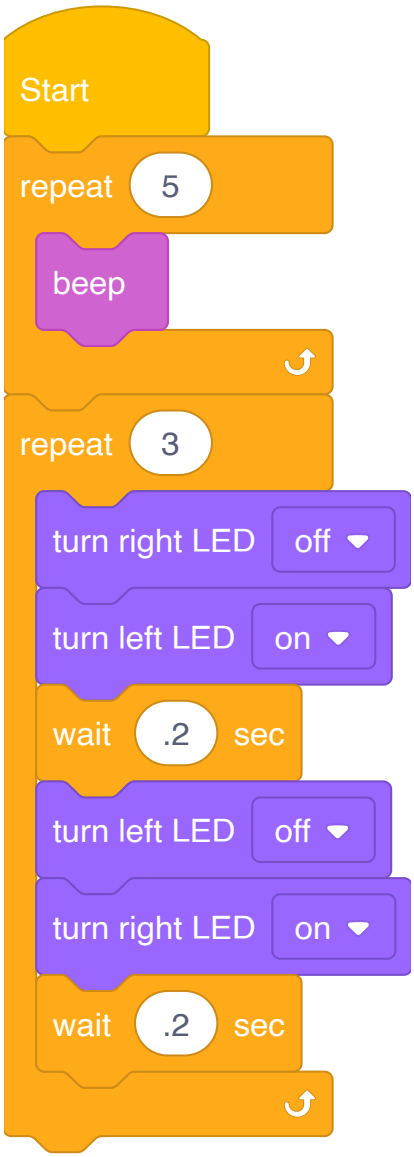
Don't forget

When you make a program for Edison in EdScratch, the robot will start with the top block and do each action one by one. Once it completes a block, it will move to the next block. This is true of all EdScratch programs – whether there are zero loops, one loop or multiple loops!

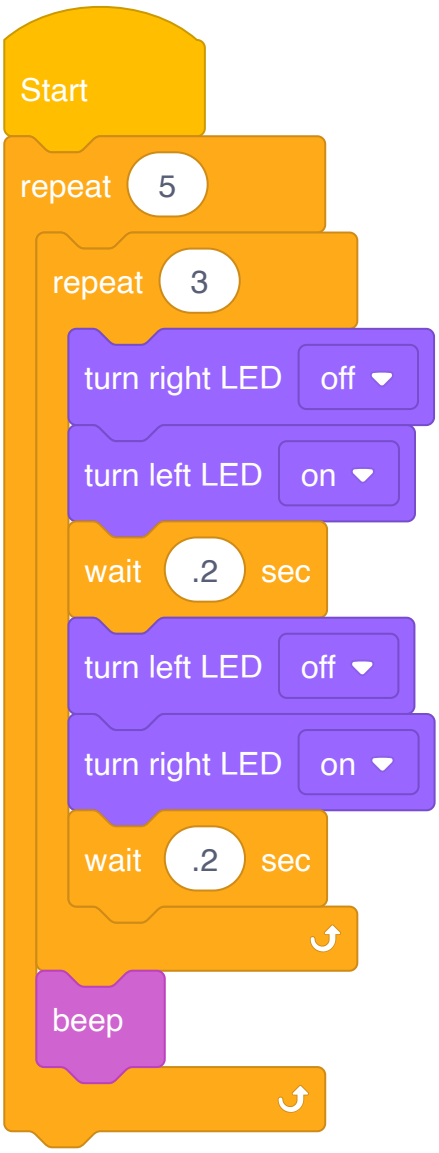
Name _____

Look at the following programs and answer the questions about what is going to happen in each program.

Program 1:



Program 2:



In program 1, how many times will the right LED turn on?

In program 1, which will finish first: all of the beeps or all of the LED flashes?

In program 2, how many times will the right LED turn on?

In program 2, which will finish first: all of the beeps or all of the LED flashes?

Hint

Can you follow along with what is happening in each program? If you want to double check your answers, try writing each program in EdScratch, then download it to your Edison robot. Run the program to see what happens.

Name _____

Drive the pattern

Just like a loop lets you repeat a pattern multiple times, nesting loops allows you to repeat multiple patterns! For this activity, you need to use the activity on page 87.

Look at the pattern on the activity sheet. How would you describe the pattern?

You need to write a program so that Edison will drive the pattern on the page 87 activity sheet.
Your program can have Edison go across the same line more than once, but the robot must touch all the lines. How do you think you can use a nested loop to help you write an efficient program for your Edison robot to drive the pattern on the activity sheet?

Try writing an EdScratch program so that you get your Edison robot to drive the pattern on the activity sheet on page 87. Test your idea for using a nested loop to see if it works.

Hint

You can write a program that completes the activity sheet using just five EdScratch blocks!

Change it up: Edison the designer

Lots of things that run using computer programs have repeating patterns. There are also many programs that have patterns that repeat inside of other patterns. These programs often use nested loops to repeat whole sets of commands inside a program.

What to do

Try using loops to write a program for your robot which makes Edison drive a pattern. If that design has a pattern with a repeating pattern inside of it, try using a nested loop.

Look at activity sheet on page 88 and choose one of the designs to use. For this activity, you will need to create a workspace to test your program. Make a workspace that is large enough to test your program with Edison. You could draw the pattern onto a large sheet of paper or mark it out using dark coloured tape on the floor. Copy out the design onto your workspace. Then write a program in EdScratch that gets Edison to drive that design.

Hint

Stuck? Try breaking down the pattern into smaller sections and writing code to get Edison to drive each part of the pattern. Link all of the chunks together to get Edison to drive the whole pattern. This can help you to find places where the code repeats. Make your program more efficient by replacing repeating code with loops.

If there is a pattern inside a pattern, be sure to try a nested loop!

Mini Challenge

If you want, you can also design your own pattern to use for this activity. Make sure your design has a pattern repeating inside another pattern. Test your design by writing a program that gets Edison to drive your pattern.

Name

Write your "Edison the Designer" program in the text box below or take a photo of your program and add it to the image box.

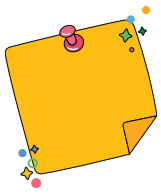
Name _____

Let's explore interrupting the main program

Different computer programming languages have different syntaxes, or rules, which make them look and feel a bit different from each other. No matter the syntax, however, all computer languages work using the same underlying logic. This is why all computer programs behave in similar ways and follow core programming logic, like sequence.

In EdScratch, the logical flow of a program is to start with the top block and complete each action one block at a time. Programs with loops also follow sequence. When the program sees a loop block, it executes the commands inside that loop in order. When it gets to the bottom of the loop, it goes back to the top of the loop and starts again. Even though loop blocks make programs look a bit different, these programs still follow the logical flow of top-to-bottom sequence.

There is a way to interrupt this sequential flow. You can disrupt a computer program's flow by using an interrupt.



Don't forget

Logic is the organised way of doing things that makes sense to a computer. Logic determines the flow of a program.

Syntax is the rules of how a programming language works.

Jargon buster

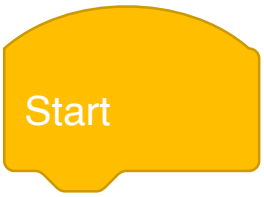
An **interrupt** is a special bit of code that stops the flow of the main code. It's called an interrupt because it interrupts the main code. An interrupt is usually used to pause the main code in order to run a **subroutine.**

A **subroutine** is a distinct set of code that is separate from the main program. You can think of a subroutine as a mini program.

To understand how interrupts work, we need to understand what is being interrupted.

What is the main program?

In EdScratch, the main program is whatever is attached to the yellow start block.

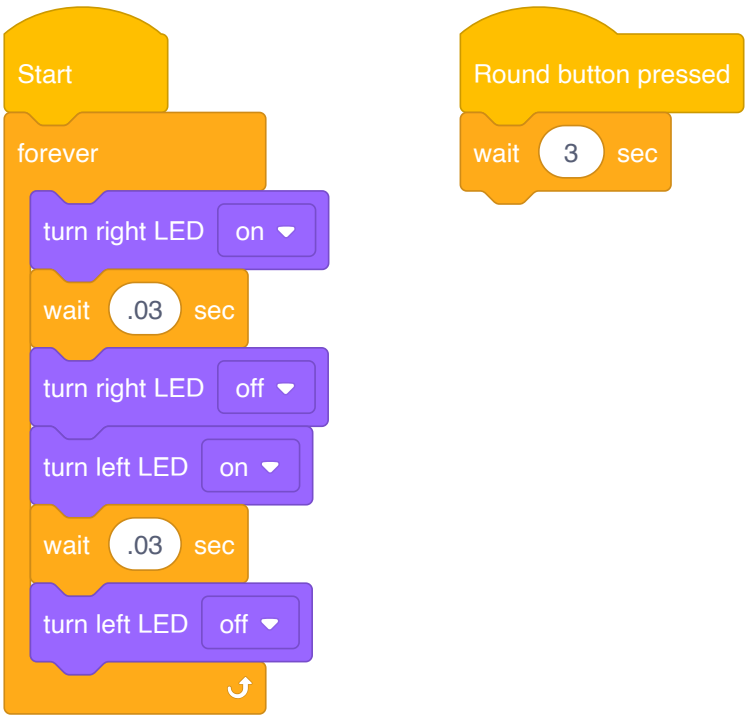


Whenever you write a program for Edison, you need to have at least one block in the main program attached to the start block. When you run a program with your Edison robot by pressing the play (triangle) button, this main program runs block-by-block in sequential order until it reaches the end of the program.

An interrupt can disrupt this flow.

Name _____

Look at the following program:



This program has two parts: the main program and the subroutine.

What does the main program tell Edison to do?
Hint: only the blocks attached to the **start** block are in the main program.

In addition to the main program, there is also a subroutine.
What causes the main program to be interrupted and run the subroutine?

Subroutines will only run when a specific **event** happens.

Jargon buster

In programming, an **event** is something that happens outside of the program code that affects how the program runs. An event might be a button being pressed or information being relayed from a sensor.

In EdScratch, you need to use a block from the **Events** category at the start of any subroutine.

Look at the blocks in the **Events** category in EdScratch.
What do you notice about the shape of these blocks?

The **Event** blocks are interrupts that tell the program to look out for that particular event.
If the event happens, the **Event** block interrupts the main program immediately and runs the subroutine. Once the subroutine code is complete, the program returns to wherever it left off in the main program.

Why is that?

We use interrupts in programming because interrupts allow a program to react to an event at any time while the program is running. By using interrupts, you don't have to predict when an event will occur. Without interrupts, you would have to know exactly when something was going to happen, even when that event is out of your control!

Try it out

Write a program in EdScratch that contains both the main program and the subroutine just as they appear in the picture from earlier in this activity. Download the program to your Edison robot. Press the play (triangle) button on your robot. This will start the main program, causing Edison's LEDs to flash on and off. Now press the round button on the robot. This interrupts the main program and runs the subroutine. This subroutine tells Edison to wait 3 seconds, then return to the main program.

You can use this program to turn your Edison robot into a decider bot!

Think of a question that you can answer with a 'yes' or a 'no'. The decider bot will help you answer that question. If the right LED is lit up when you press the round button, the answer to your question is 'yes', but if the left LED is lit up, then the answer is 'no'.

Why is that?

Remember, an interrupt pauses the main program instantly, so it is possible that neither LED will be lit up when the subroutine runs. If the main program has turned off the right LED but hasn't yet turned on the left LED when the interrupt occurs, then both LEDs will be off. This is a bit like flipping a coin and having it land on its edge – rare, but it can happen!

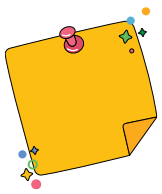
Change it up: Try a clap instead

Remember Edison's sound sensor? This is the special bit of tech that's both a buzzer and a sound sensor. This is the bit of Edison that makes noise, like beeps or musical notes, but can also detect sounds, like a clap.

You can use Edison's sound sensor to trigger an interrupt in a program. That way, you can make a decider bot that responds to the sound of a clap!

What to do

Write a program to turn Edison into a decider bot. Your main program should have Edison flash its two LEDs on and off forever. You also need a subroutine that will interrupt the main program if the robot detects a clap. Your subroutine should tell Edison to wait for a few sounds so that you can see which LED is on and get your answer.



Don't forget

In EdScratch, you need to use a block from the Events category to act as an interrupt and be the first block at the start of any subroutine.

Hint

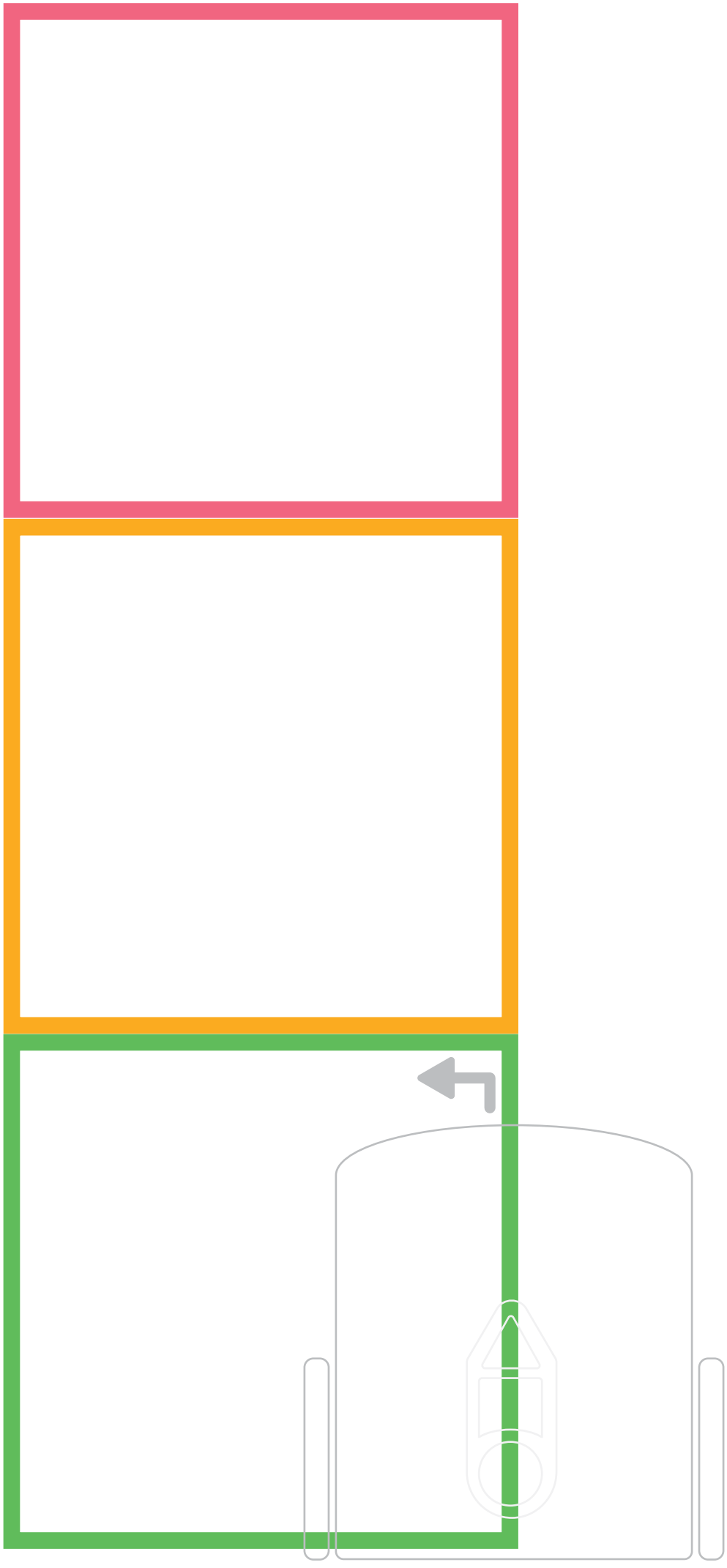
You can use the decider bot program from page 75 as a base for your program.

Download your program to your Edison robot and test it out.

Name _____

Write your "Edison the Decider Bot" program in the text box below or take a photo of your program and add it to the image box.

Activity sheet: Repeating



Name _____

Activity sheet: Driving designs