GRADE 2 Concurrent Events | GRADE 3 Repeating Events | GRADE 4 Nested Events

{CODING} IN THE CLASSROOM

GRADE 3 TEACHERS' GUIDE

Coding with repeating events

Welcome to NII Explore's *Coding in the Classroom* program for Grade 3 students. During the 4-week program, you and your class will complete:

- 3 online lessons
- 3 offline activities
- A final coding project

This teacher guide includes everything you need to get started!

THE GRADE 3 CODING CURRICULUM

As of 2020, Ontario's math curriculum includes coding expectations. Put simply, coding is when we make instructions, or "code", for a computer to follow. There are two core expectations that run through every grade level of the coding curriculum.

1. **Writing and executing code**
2. **Reading and altering existing code**

Each grade level introduces students to a new coding skill. Students can practice this new skill while also using the skills learned in previous grades. In Grades 1 and 2, students learn about **sequential events** (things that happen in a specific order) and **concurrent events** (things that happen at the same time).

SCHEDULE AT A GLANCE

WEEK 1

- **Online Lesson 1**
  Loops with Laurel
- **Offline Activity 1**
  Human Robots

WEEK 2

- **Online Lesson 2**
  Bee: Loops
- **Offline Activity 2**
  Repeat Loop Dance

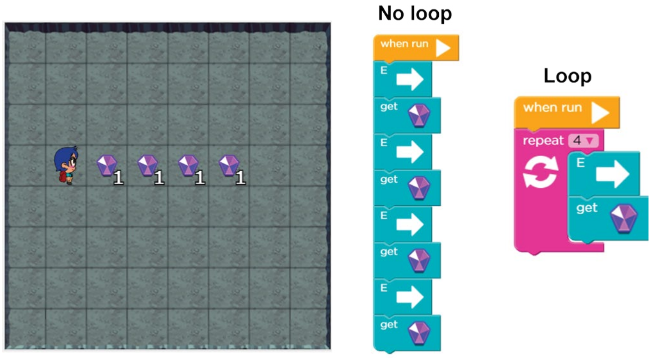
WEEK 3

- **Online Lesson 3**
  While Loops in Farmer
- **Offline Activity 3**
  Loop, Paper, Scissors

WEEK 4

- **Final Project**
  Drawing with Loops

In **GRADE 3**, students are introduced to **repeating events**. A repeating event, also called a **loop**, is something that happens multiple times in a row. Loops help make our code shorter and easier to read. Instead of writing the same code over and over, we can use a loop like in the following example:



No loop

```
when run
E ➔
get
E ➔
get
E ➔
get
E ➔
get
```

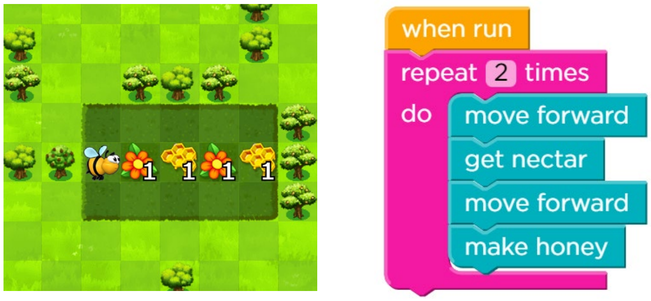
Loop

```
when run
repeat 4
    E ➔
    get
```

Loop example. *Both algorithms will solve the maze, but the pink repeat loop makes the code on the right shorter and more efficient.*

The maze above has an A-B-A-B pattern of (A) moving east and (B) getting a gem. Any time we notice a pattern like this, we can simplify our code using a repeat loop.

Repeat loops can be used for any pattern type. In the next example, the bee moves forward, then collects nectar, moves forward, then makes honey. This A-B-A-C pattern can be coded with a single loop:



```
when run
repeat 2 times
do  move forward
    get nectar
    move forward
    make honey
```

Repeat loop example. *Loops can be used whenever we notice a pattern. In this case, the puzzle has an A-B-A-C pattern.*

In coding, there are three types of loop that we might use. **For loops** repeat a section of code *for* a certain number of times. The examples we've seen so far are for loops. **Until loops** repeat code *until* a certain condition is met. **While loops** will repeat code *while* a certain condition is true.

We can understand the difference between the loop types using an example from our everyday lives. Imagine you are making a cake – *how long should you bake it?*

If the recipe uses a for loop, it will say something like this:

BAKE CAKE FOR 50 MINUTES

But some recipes use until loops instead:

BAKE CAKE UNTIL CAKE TESTER COMES OUT CLEAN

And it may be less practical in this situation, but we could also use a while loop here:

BAKE CAKE WHILE IT IS STILL RAW

Until loops and while loops are both examples of *conditional* loops and most programming languages will only use one or the other. *Code.org*, the platform NII Explore uses with Grade 3 classes, has while loops. *Scratch*, the platform we use with Grades 5 to 8, has until loops.

Loops are a computer's specialty. Humans get tired and bored when we have to do the same thing over and over, but computers don't have that problem. We can program a computer to perform millions of operations each second, and with loops, we only need a few lines of code to do it.

PROGRAM SCHEDULE

The *Coding in the Classroom* program will last four weeks. Here is a detailed guide of what you will be doing each week.

BEFORE WEEK 1

Read through this teacher guide, including the instructions for the three online lessons. If you have time, you may want to try the online activities for yourself.

Make sure your class has access to devices (laptops or tablets) for each of the online lessons. Your class will also need devices for the final project.

WEEK 1

Online Lesson 1 – Loops with Laurel

This lesson introduces students to block-based coding and repeating events called "loops." Students will write and edit code to help Laurel find and collect treasure.

PREP Log students onto computers and open activity link.

POST Complete "Offline Activity 1 – *Human Robots*" before next online session.

See **Page 3.5** for lesson instructions.

Offline Activity 1 – Human Robots

Coders will practice giving step-by-step instructions to their human "robots."

See **Page 3.9** for activity instructions.

WEEK 2

Online Lesson 2 – Bee: Loops

Students will continue coding with loops as they help a bee collect nectar and make honey.

PREP Same as Week 1.

POST Complete "Offline Activity 2 – *Repeat Loop Dance*" before next online session.

See **Page 3.11** for lesson instructions.

Offline Activity 2 – Repeat Loop Dance

Your class will practice reading and writing code with loops by creating their own dance routines.

See **Page 3.14** for activity instructions.

WEEK 3

Online Lesson 3 – While Loops in Farmer

This lesson introduces students to a new type of loop: while loops. At the end of the class, discuss the final coding project.

PREP Same as Week 1.

POST Complete "Offline Activity 3 – *Loop, Paper, Scissors*" and the Final Project.

See **Page 3.17** for lesson instructions.

Offline Activity 3 – Loop, Paper, Scissors

Your class will explore the power of repeat loops while making some paper art.

See **Page 3.20** for the activity instructions.

WEEK 4

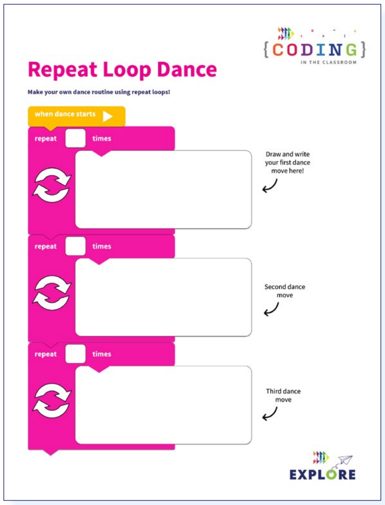
Final Project – Drawing with Loops

Students will apply their learning to complete part of a *Code.org* lesson on their own. They will then be given a blank canvas to create their own art using code.

See **Page 3.24** for project instructions and **Page 3.26** for assessment criteria.

AFTER WEEK 4

Keep the coding going with the additional resources on **Page 3.28**!



Repeat Loop Dance

Make your own dance routine using repeat loops!

when dance starts

repeat □ times

Draw and write your first dance move here!

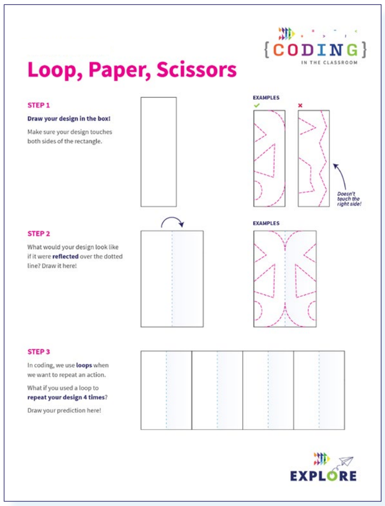
repeat □ times

Second dance move

repeat □ times

Third dance move

Week 2 *Repeat Loop Dance*



Loop, Paper, Scissors

STEP 1

Draw your design in the box!

Make sure your design touches both sides of the rectangle.

EXAMPLES

✓ ✗

Doesn't touch the right side!

STEP 2

What would your design look like if it were **reflected** over the dotted line? Draw it here!

EXAMPLES

STEP 3

In coding, we use **loops** when we want to repeat an action.

What if you used a loop to **repeat your design 4 times**?

Draw your prediction here!

Week 3 *Loop, Paper, Scissors*

ONLINE LESSON 1

Loops with Laurel

60 MINUTES

The three online lessons and final project all use **code.org** and block-based coding. If you are new to code.org, you can check out their **teacher resources** and try the lessons for yourself before each class.

In this lesson, students will use arrow blocks and repeat loops to help Laurel collect treasure. It is one of the simpler loop lessons on code.org, so it will be a good starting point if some of your students have never coded before. There is a mix of writing new code and editing existing code.

QUICK LINKS

Student Activity Link
studio.code.org/s courseb-2022/lessons/8/ levels/1

PowerPoint
Grade 3 – Week 1 – Intro to Coding

CURRICULUM CONNECTIONS

CODING SKILLS

- **Math C3.1** – solve problems and create computational representations of mathematical situations by writing and executing code, including code that involves sequential, concurrent, and repeating events
- **Math C3.2** – read and alter existing code that involves sequential, concurrent, and repeating events, and describe how changes to the code affect the outcomes

PATTERNS

- **Math C1.1** - identify and describe repeating elements and operations in a variety of patterns, including patterns found in real-life contexts
- **Math C1.2** - create and translate patterns that have repeating elements, movements, or operations using various representations, including shapes, numbers, and tables of values

LOCATION AND MOVEMENT

- **Math E1.4** - give and follow multi-step instructions involving movement from one location to another, including distances and half- and quarter-turns

LESSON BREAKDOWN

SLIDE 1 - SET UP AND INTRODUCTION

Open the PowerPoint slides and the code.org activity on your own computer. Project for the students to see.

Get the students logged onto their computers with the activity link open. They may get a pop-up message that says: "You are not signed in". They can dismiss this message by clicking anywhere else on the screen. When everyone is ready, have them turn their attention to you for a quick discussion before the coding begins.

SLIDE 2 - WHAT IS CODING?

Check if your students have coding experience. Ask them what coding means.

Use slides to explain what coding is. "Coding is when we give instructions to a computer."

SLIDE 3 - WHAT CAN YOU DO WITH CODE?

Ask students what they think – Why are we learning to code? What can we use it for?

Use slides to share some possible answers.

SLIDE 4 - THE CODER'S CODE

Have students recite the "Coder's Code" by repeating after you.

The Coder's Code will help set expectations for the students.

SLIDE 5 - READY TO START

Switch your screen over to code.org.

LEVEL 1

Watch the introductory video together (Level 1 of the lesson).

LEVEL 2

Proceed to Level 2. Give students a tour of the code.org layout – where do we find everything?

Levels are listed in the bar along top, preview window is on the left, blocks are in the middle, workspace is on the right, instructions are above the blocks and workspace. You should take some time to familiarize yourself with the layout before you teach these lessons.

NOTE: You will probably want students to turn down their computer volume or mute their speakers entirely. You can also make your preview window larger (i.e., the area where Laurel appears) by dragging the right side of the window.

LEVEL 2 - CONTINUED

Complete the level together.

Explain the goal of the level and read the instructions as a class.

Demo how to click and drag blocks, and how to run the program.

NOTE: This is a practice level. You only need to collect a single gem.

LEVEL 3

Read the instructions together and give students a chance to try it.

Take up the solution all together to make sure students understand what to do before moving on.

NOTE: It gives you the option to use a repeat loop, but don't use it yet.

LEVEL 4

Watch the video explaining loops (i.e., repeating events).

LEVEL 5

Read the instructions together.

Have students notice the pattern (Move East → get gem → Move East → get gem …).

Remind students what repeat blocks (called "loops") do and compare the two solutions side by side (one without the loop and one with the loop). Which one is shorter to write? Why do you think we use repeat loops?

LEVEL 6

Explain that grey blocks can't be deleted. We are trying to fix or "debug" the code that is already here.

Show the target number of blocks. (e.g., 6/8 blocks). We are trying to solve this level using only 8 blocks.

Let students try the level on their own, offering help as needed.

NOTE: You can monitor students' progress by checking the bar at the top of their screen. A solid green dot shows that they fully completed a level. A partially filled dot means that they solved the level but didn't use the intended method. For example, they may have solved the level without using a repeat loop.

LEVELS 7 TO 9

If students are getting the hang of it, let them work at their own pace through these next few levels. Remind students to check the target number of blocks for each level.

LEVEL 10

This level is advertised as a "Challenge" level. Encourage students to try it out but let them know it's okay to ask for help if they need it.

NOTE: If there are any super keen students, challenge them to solve this level using only 8 blocks instead of the target 13. It can be done using a nested loop (i.e., loop within another loop) which they will learn more about in Grade 4.

LEVELS 11 & 12

Give students time to try these levels on their own, offering support when needed.

LEVEL 13

Advertise this to the students as a final challenge. If any students finish early, ask them to share their solution. Compare solutions between students – is your code identical? Encourage students to look for the most efficient (i.e., fewest blocks) solution to the level.

SLIDE 6 - RECAP

With about 5 minutes left in the class, switch back to the PowerPoint slides.

Ask students → What did we learn about today? What does coding mean? (Giving instructions to computers) What's a loop? (A command that makes code repeat) Why do we use loops when we code? (Means we don't have to write the same thing over and over)

SLIDE 7 - POEM OF THE DAY

Each slideshow ends with a Poem of the Day to recap the lesson. Introduce the concept of the Poem of the Day then read the poem together.

SLIDE 8 - WHAT'S NEXT?

Let students know when you will be coding again. We recommend alternating between the online lessons and the offline activities. It requires less screen time for your class and will give students more time to absorb the new information.

OFFLINE ACTIVITY 1

Human robots

30 MINUTES

Coders will practice giving step-by-step instructions to their human "robots."

MATERIALS

None required

LEARNING OBJECTIVES

- Give and follow sequential instructions
- Perform movement skills safely

CURRICULUM CONNECTIONS

- Math C3.1 & C3.2
- Health and Physical Education C1

SET-UP

This activity requires some space so consider moving to a gym or outside.

INSTRUCTIONS

SUMMARY

Students will work in groups of three. Student A (the "Coder") will give instructions to help Student B (the "Robot") find Student C (the "Target").

1. Ask students to recall what they learned about in the online class.
   ASK What is coding? (Giving instructions to a computer)
2. Explain the premise of today's activity. "Today, we are going to do some coding, but instead of writing instructions for a computer, we will be giving instructions to each other."
3. Move to a large, open area with few tripping hazards like a gym or schoolyard. Divide students into groups of three. Within each group, assign one student to be the Coder, one to be the Robot, and one to be the Target.

4. Explain the rules of the game. The Robot will close their eyes while the Target moves to stand somewhere within the playing area. When the Target is ready, the Robot keeps their eyes closed and the Coder gives them step-by-step instructions to find the Target (e.g., take two steps forward then turn left). You may want to play a practice round with volunteers to help you explain.
5. Begin the game. Ask the Targets not to move too far for the first round. For safety, remind the Robots to move slowly and walk with their arms out. Tell the Coders to warn their Robot if they are about to run into something. You can also monitor the playing area to see if students are about to bump into a wall or another student.
6. The round ends when all the Robots have found their Target. Have the students switch roles (e.g., Coder becomes the Robot, Robot becomes the Target, Target becomes the Coder). Play more rounds until each student has had a chance to try each role, or until time runs out.
7. Wrap up the activity using some of the discussion questions as a guide.

DISCUSSION QUESTIONS

Which role did you find the most fun? Which was the most challenging?

Do you find it easier to give instructions to a person like we did today or to a computer like we did in coding class? Why?

Possible answers: Computers because they will do exactly what you say. Humans because they can figure out what you mean if you forget to say something.

When we find and fix mistakes in computer code, it’s called debugging. Did your group make any mistakes? How did you fix it?

ONLINE LESSON 2

Bee: Loops

60 MINUTES

Students will continue coding with loops as they help a bee collect nectar and make honey. The activity will require some pattern recognition and includes a multiple-choice question to test students' understanding.

QUICK LINKS

Student Activity Link
studio.code.org/s/course2/lessons/8/levels/1

PowerPoint
Grade 3 – Week 2 – History of Coding

CURRICULUM CONNECTIONS

CODING

- **Math C3.1** – solve problems and create computational representations of mathematical situations by writing and executing code, including code that involves sequential, concurrent, and repeating events
- **Math C3.2** – read and alter existing code that involves sequential, concurrent, and repeating events, and describe how changes to the code affect the outcomes

PATTERNS

- **Math C1.1** - identify and describe repeating elements and operations in a variety of patterns, including patterns found in real-life contexts
- **Math C1.2** - create and translate patterns that have repeating elements, movements, or operations using various representations, including shapes, numbers, and tables of values
- **Math C1.3** - determine pattern rules and use them to extend patterns, make and justify predictions, and identify missing elements in patterns that have repeating elements, movements, or operations

LOCATION AND MOVEMENT

- **Math E1.4** - give and follow multi-step instructions involving movement from one location to another, including distances and half- and quarter-turns

LESSON BREAKDOWN

SET UP

Open the PowerPoint slides and the code.org activity on your own computer. Project for the students to see.

Get the students logged onto their computers with the activity link open.

SLIDE 1 - WEEK 1 RECAP

Ask students to recall what they learned about in the first coding lesson. What are the pink blocks called? (Repeat loops) Why do we use them? (We can repeat a step without having to write the same instructions again and again).

SLIDES 2 & 3 - READY TO START

OPTIONAL: Choose a point from the Coder's Code to emphasize this week. The slide is hidden in the PowerPoint file if you want to use it.

Switch your screen to code.org. Make sure students have the right lesson open (bee with two flowers).

LEVEL 1

Read instructions together.

Review what's different than last week (move forward and turn blocks instead of direction arrows).

Do level together and take up the answer before moving on.

LEVEL 2

Watch video to remind students how to use loops. When do we use them? (Any time we notice a pattern)

Give students time to try solving this level using a loop. Go over solution together.

LEVEL 3

Try level together.

Point out the target number of blocks (1/5 blocks).

If students seem to have the hang of it, let them work on their own. Remind them to check the target number of blocks for each level and that each level will probably require a loop at some point.

LEVEL 4

Introduce the new "make honey" block to the students.

LEVEL 5

Introduce turning. Ask students to show you with their hands which way is left and which way is right. Explain that the directions are from the perspective of the bee. Students often find this part the trickiest. If they can't figure it out, they can always use trial and error to figure out which way to turn.

LEVELS 6 & 7

Let students try these levels on their own.

SLIDES 4 TO 7 - INTERMISSION: HISTORY MINUTE

Take a break to share the history minute about Ada Lovelace (switch back to slides). See the slide notes for talking points.

It's a good opportunity to check how students are doing and re-focus their attention if they are getting restless.

LEVEL 8

Talk about patterns.

Move forward, get nectar, move forward, get honey → This is an ABAC pattern. You may have talked about this in math class before.

Point out the connection between patterns and repeat loops. We can use repeat loops whenever there is a pattern involved.

LEVELS 9 TO 13

Give students the rest of the time to work through these levels at their own pace. Let them know it's ok if they don't finish all of them.

LEVEL 14

At the end of the class, have everyone try the question in Level 14 – they may need to skip ahead a few levels. Read the question together and let them all try to answer on their own. Take up the answer together.

SLIDE 8 - RECAP

With about 5 minutes left in the class, switch back to the PowerPoint slides.

Ask students "What did we learn about today? What's a loop? When would you use a loop?".

SLIDE 9 - POEM OF THE DAY

Share this week's Poem of the Day.

SLIDE 10 - WHAT'S NEXT?

Give a quick preview of next lesson. "We will solve some more puzzles and learn about a new type of loop."

You can also try **Offline Activity 2 – Repeat Loop Dance** before the next online lesson.

OFFLINE ACTIVITY 2

Repeat loop dance

30 MINUTES

Practice reading and writing code with loops by creating a dance routine.

LEARNING OBJECTIVES

- Read and write instructions that use repeat loops
- Create and present a dance routine

CURRICULUM CONNECTIONS

- Math C3.1 & C3.2
- Arts A1

SET-UP

This activity requires some space, so consider moving to a gym or outside. Print and hand out worksheets.

INSTRUCTIONS

SUMMARY

Students will use repeat loops to write and draw instructions for a dance routine. Students will trade instructions with a partner to see if they can follow the steps.

1. Ask students to recall what they learned about loops in the online class. **ASK** What do we call the pink blocks that make an action repeat? (A loop) Why do we use loops? (They make our code shorter and easier to follow)
2. Explain the premise of today's activity. "Today, we are going to write instructions for a dance routine using repeat loops. Then we will trade our instructions with a partner and see if they can figure out how to do our dance."
3. Students will create their dance routines by completing their worksheet. Their dances will include three moves. For each move, students will write the name of their move, draw a picture of what the move looks like, and choose a number of times that the move will be repeated.

MATERIALS

Each student will need:

- Pencil or pen
- 1 worksheet

4. Give students time to test out dance moves and fill out their worksheet. If students need help, you can suggest some simple dance moves like clapping, spinning, or jumping side to side.
5. When students are finished creating their dance routines, assign partners and have them trade worksheets. Students will now take turns reading their partner's dance routine and trying it for themselves.
6. If time allows, have students trade dance routines with another partner or ask for volunteers to perform their dances for the class.
7. Wrap up the activity using some of the discussion questions below as a guide.

DISCUSSION QUESTIONS

Why did we use repeat loops when writing our dance routines?

Answer: They make our instructions shorter.

Did your partner perform the dance the way you thought they would? What was different?

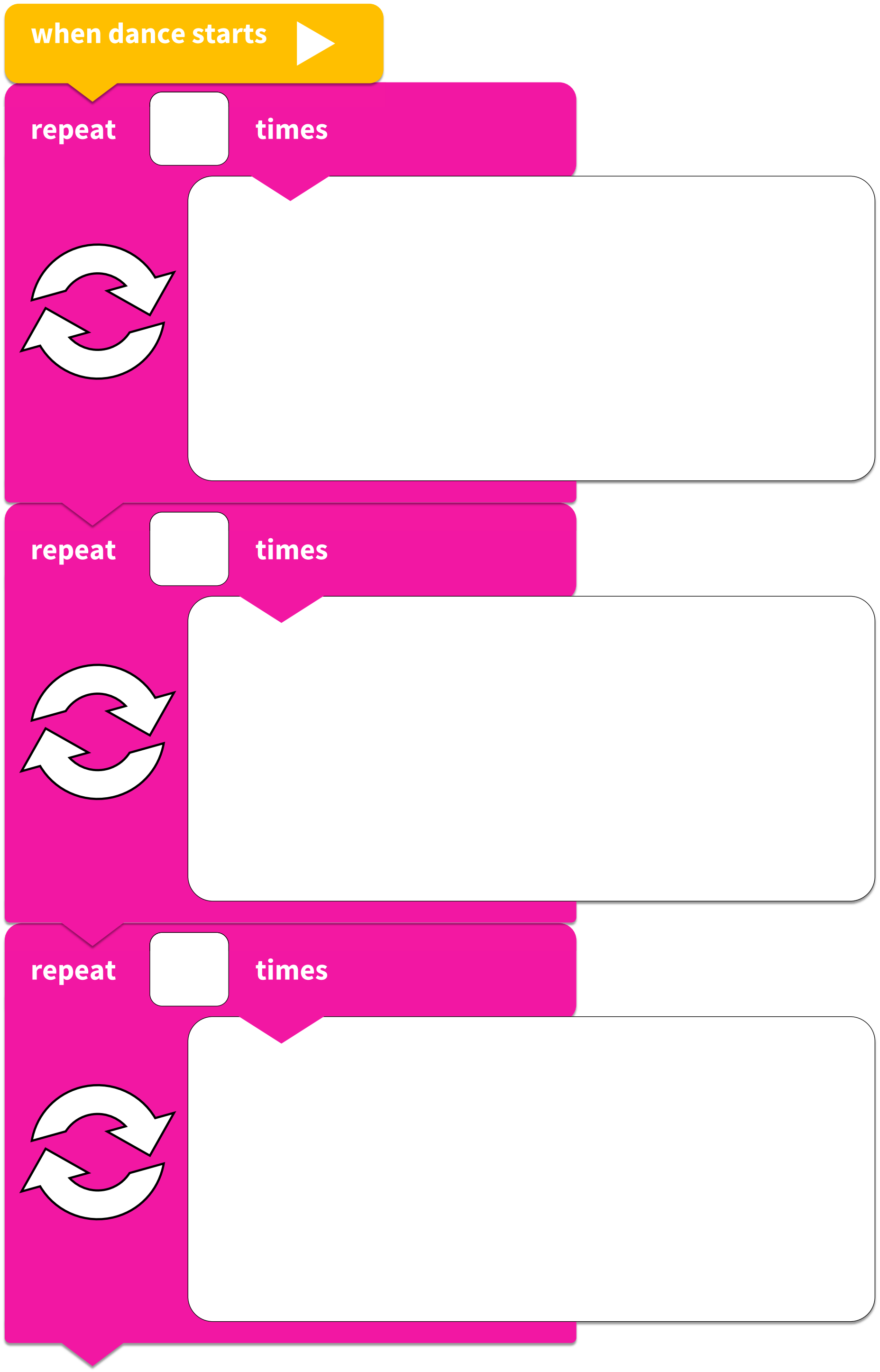
In coding, we use loops whenever we want to repeat an action multiple times. Can you think of a time in our everyday lives when we could use a repeat loop?

Possible answers: Tie left shoe then repeat for right shoe, use cookie cutter then repeat 12 times, repeat "Trick or Treat" at every house, read a page then repeat until end of book.

ACTIVITY 2 WORKSHEET

Repeat loop dance

Make your own dance routine using repeat loops!



when dance starts ▶

repeat □ times

Draw and write your first dance move here!

repeat □ times

Second dance move

repeat □ times

Third dance move

ONLINE LESSON 3

While loops in farmer

60 MINUTES

This lesson introduces students to a new type of loop – while loops – as they help a farmer dig and fill holes. Students will write code with while loops and make predictions about what their code will do. This activity is a bit more difficult than the previous two, so be prepared to support students as needed.

QUICK LINKS

Student Activity Link
studio.code.org/s/coursed-2022/lessons/15/levels/1

PowerPoint
Grade 3 – Week 3 – Final Project

CURRICULUM CONNECTIONS

CODING

- **Math C3.1** – solve problems and create computational representations of mathematical situations by writing and executing code, including code that involves sequential, concurrent, and repeating events
- **Math C3.2** – read and alter existing code that involves sequential, concurrent, and repeating events, and describe how changes to the code affect the outcomes

PATTERNS

- **Math C1.1** - identify and describe repeating elements and operations in a variety of patterns, including patterns found in real-life contexts
- **Math C1.2** - create and translate patterns that have repeating elements, movements, or operations using various representations, including shapes, numbers, and tables of values
- **Math C1.3** - determine pattern rules and use them to extend patterns, make and justify predictions, and identify missing elements in patterns that have repeating elements, movements, or operations

LOCATION AND MOVEMENT

- **Math E1.4** - give and follow multistep instructions involving movement from one location to another, including distances and half- and quarter-turns

LESSON BREAKDOWN

SET UP

Open the PowerPoint slides and the code.org activity on your own computer. Project for the students to see.

NOTE: Some of the levels in this lesson are quite tricky, so we recommend trying them ahead of time.

Get the students logged onto their computers with the activity link open.

SLIDE 1 - WEEK 2 RECAP

Ask the students to recall what they learned about in Week 2 (repeat loops, patterns, turning left and right, etc.).

SLIDES 2 & 3 - READY TO START

OPTIONAL: Choose a point from the Coder's Code to emphasize this week. The slide is hidden in the PowerPoint file if you want to use it.

Make sure students have the right lesson open (farmer with a shovel on a dirt path).

LEVEL 1

Read instructions and do level together.

Remind students about using the repeat block.
Ask: Why do we use it? (It means we don't have to write as much code).

LEVEL 2

Introduce the “fill” block then let students try this level on their own.

LEVEL 3

Let students try this level on their own. Remind them about the target number of blocks at the top. We’re trying to solve this level using only 4 blocks or lines of code.

LEVEL 4

Watch the video together as a class. It explains **while loops**. Students can skip the video on their own devices.

Introduce while loops as a new type of loop. Instead of saying how many times to repeat, a while loop repeats *while* a certain thing is true. Let’s say you have to clean your room. I could give you an exact number like “clean your room for 30 minutes” or I could say “clean your room while it is still dirty.”

LEVEL 5

Read the question and code together then have students make their predictions. Poll the class by having them raise their hands for each option (C is the correct answer).

LEVEL 6

Explain the problem to the students but give them time to try figuring it out for themselves. If they need a hint, tell them to use the while loop.

Take up the solution together to make sure students understand what the while loop does and when to use it.

LEVEL 7

Let students try this one with some support.

For the rest of these levels, don't worry too much if they are hitting their target block numbers – the more important thing is if they understand what the while loops do.

LEVEL 8

Introduce students to the "while there is a hole" option.

LEVEL 9

Introduce students to the "while path ahead" option.

LEVELS 10 TO 14

Let students try these levels if time allows, but you probably won't get through all of them. Note that some of these levels are tricky, even for adults.

SLIDE 4 - RECAP

With about 5 minutes left in the class, switch back to the PowerPoint slides.

Ask students "What did we learn about in our coding classes? What's a loop? (A repeat block) When can we use loops? (When we spot a pattern) What's the new kind of loop we used today?" (While loop).

SLIDE 5 - EXPLAIN FINAL PROJECT

Students will try one more code.org lesson, this time without much help.

The lesson is about drawing art using code. Students will control "the Artist" and tell him what to draw. They will complete a few levels on their own as practice then they will get a blank canvas to draw whatever they'd like!

SLIDE 6 - NII EXPLORE ART CONTEST

When students are done, they can show their pictures to you.

As a bonus, NII Explore is always accepting submissions to our digital art contest. Email your students' work to **explore@nii.ca** - we hand out prizes periodically for some of our favourite art!

SLIDE 7 - POEM OF THE DAY

Share the final Poem of the Day.

SLIDE 8 - WHAT'S NEXT?

Complete the final project in the next couple weeks while this lesson is still fresh. If you haven't already, try out the offline activities with your class. Happy coding!

OFFLINE ACTIVITY 3

Loop, paper, scissors

30 MINUTES

Explore the power of repeat loops by making your own paper artwork.

LEARNING OBJECTIVES

- Understand the purpose of loops
- Predict the result of a pattern

CURRICULUM CONNECTIONS

- Math C3.1 & C3.2 (Coding)
- Math C1.1 & C1.3 (Patterns)
- Arts D1

SET-UP

Print worksheets and check that students have access to the required materials.

INSTRUCTIONS

SUMMARY
Students will create a pattern and then predict what it would look like if it were reflected and then repeated. Students will test their prediction by cutting out their pattern in paper.

1. Ask students to recall what they learned about repeat loops in the online class.
2. Explain the premise of today's activity. **SAY** "Today, we're going to see another example of repeat loops. We will use a shortcut to repeat the same pattern many times. By the end of the class, we will all make a piece of paper art."
3. Have students design their initial pattern in the first section of their worksheet. Remind them that their design needs to touch both edges of the rectangle. They may refer to the examples for assistance.
4. Next, students will predict what their pattern would look like if it were reflected. They will draw their prediction in the second section of the worksheet.

MATERIALS

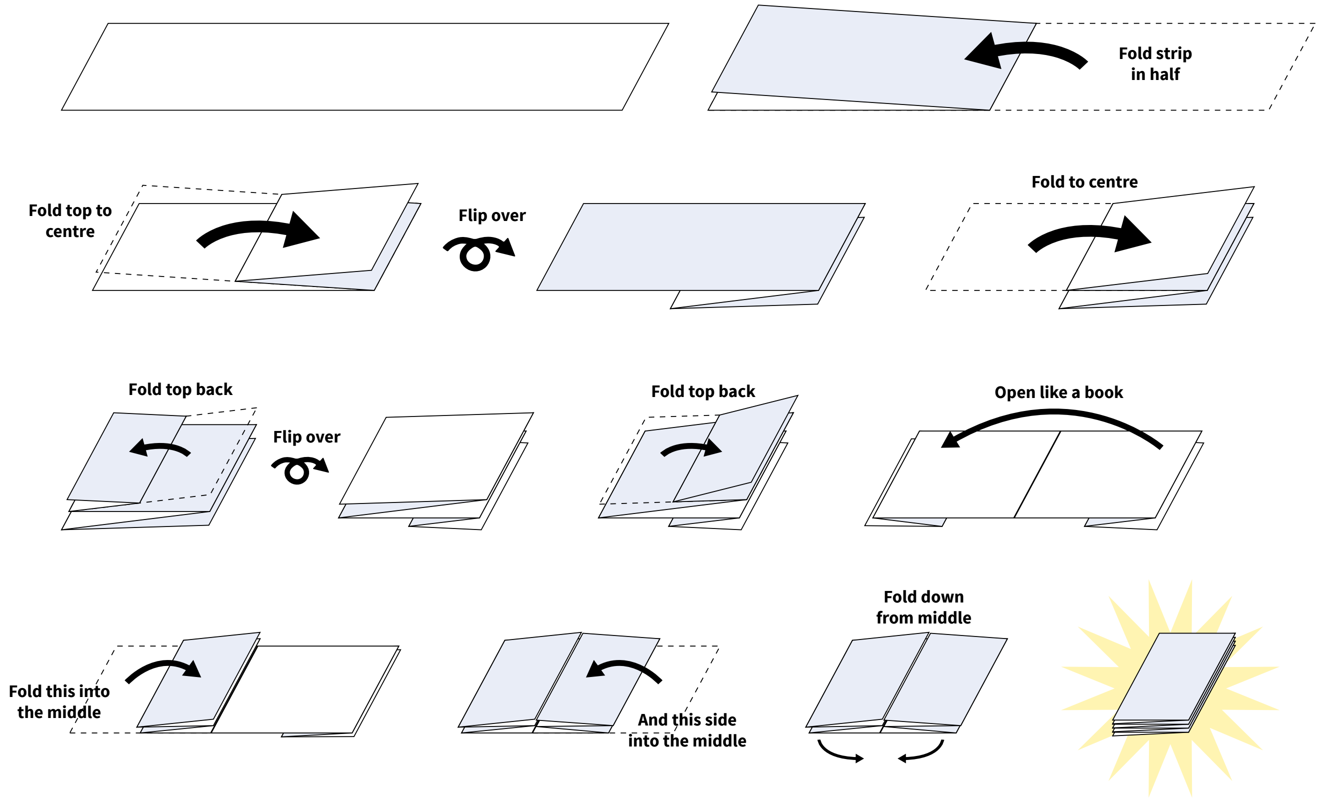
Each student will need:

- Pencil or pen
- 1 worksheet
- 1 blank piece of paper
- Scissors

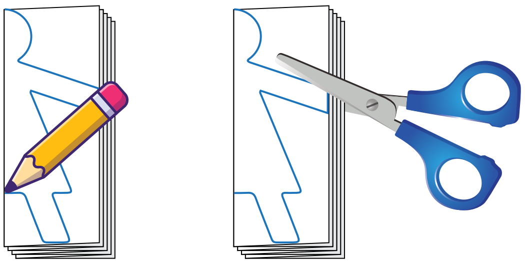
5. Students will finish their worksheet by predicting what their pattern would look like if it were repeated 4 times. Have them draw their prediction in the third section of their sheet.

6. **MAKING PAPER ART** Students will now create their art in paper to see if it matches their prediction.

    - Fold and then cut the blank paper in half lengthwise to make two paper strips.
    - Fold a paper strip into 8 equal sections as shown in the diagram below. When done, the paper strip should unfold like an accordion.



- Draw and then cut out their pattern like in the example shown below. Remind students not to cut off an entire folded edge of their rectangle or the whole thing will fall apart.



7. Have students unfold their paper and check if it matches their predicted image.

Example of a finished piece of paper art.

8. If time permits, have students make another one using the second half of their paper.
9. Wrap up the activity using some of the discussion questions below.

DISCUSSION QUESTIONS

Did your prediction match your final shape? If not, what was different?

What was the advantage of folding our paper before cutting it? What's this similar to in coding?

Answer: It meant we only had to cut once. Repeat loops in coding work the same way – you only need to write the instruction once.

Our original pattern was reflected then repeated 4 times. If we folded our paper one more time before we cut, how many times would our pattern repeat? How do you know?

Answer: 8 times. When you fold paper, it makes it twice as thick, which means you will cut out twice as many layers.

Why do we use repeat loops when we're coding?

Answer: It makes our code more efficient because we don't need to write the same thing over and over.

Can you think of any other shortcuts we use in our everyday lives?

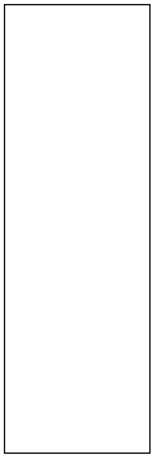
ACTIVITY 3 WORKSHEET

Loop, paper, scissors

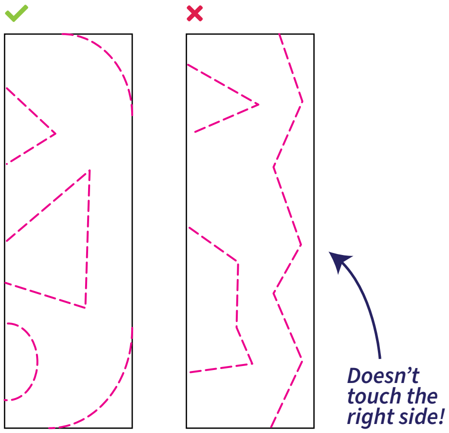
STEP 1

Draw your design in the box!

Make sure your design touches both sides of the rectangle.

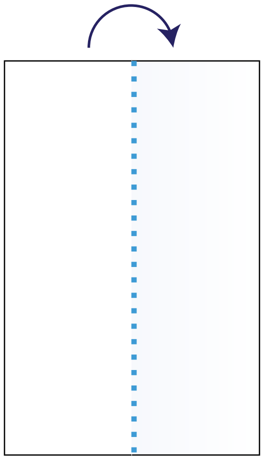


EXAMPLES

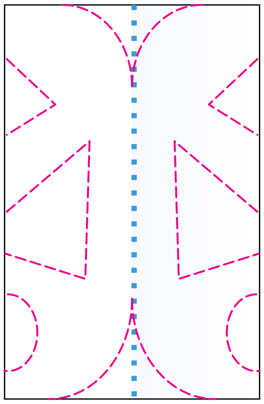


STEP 2

What would your design look like if it were **reflected** over the dotted line? Draw it here!



EXAMPLE

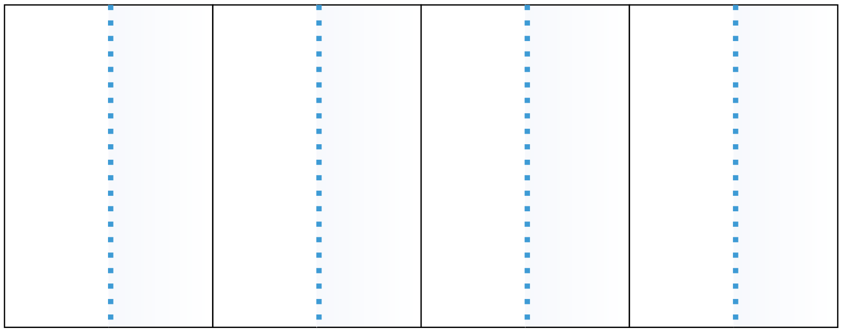


STEP 3

In coding, we use **loops** when we want to repeat an action.

What if you used a loop to **repeat your design 4 times**?

Draw your prediction here!



FINAL PROJECT

Drawing with loops

60 MINUTES +

Students will apply their coding knowledge to complete a *Code.org* lesson on their own or with a partner. For the second half of the lesson, students will draw whatever they'd like using code.

QUICK LINKS

Student Activity Link
studio.code.org/s/courseb-2022/lessons/9/levels/1

LEARNING OBJECTIVES

- Write and execute code, including code with repeat loops
- Read and alter existing code

CURRICULUM CONNECTIONS

- Math C3.1 & C3.2 (Coding)
- Arts D1

SET-UP

Read through the instructions below and try the *code.org* lesson for yourself. It will take about 15 minutes to complete. With your class, have students log onto computers and open the activity link.

INSTRUCTIONS

SUMMARY

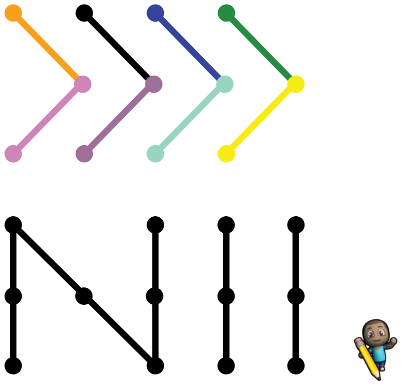
Students will be completing part of a *Code.org* lesson titled “Drawing Gardens with Loops”. Each level requires students to use coding blocks, including repeat loops, to draw simple images. The last level is “free play” where students are given a blank canvas to draw whatever they'd like.

1. Ask students to recall what they learned about in the online classes. **ASK** What do loops do in coding? What do they look like on *Code.org*? Why do we use loops?
2. Start by watching the Level 1 introductory video as a class. The video explains two block types that the students will be using.
3. Give students a minute to try out the draw and jump blocks in Level 2.

4. Watch the Level 3 video together. It reminds students about repeat blocks (loops) and how to use them.

5. Give students time to work through some of Levels 4 to 11 on their own or with a partner. Encourage students to try each puzzle on their own first before asking for help. **NOTE:** Level 9 is listed as a Challenge Puzzle. It is a bit more difficult than the other levels but still quite doable.

6. When there is about 30 minutes left in the class, have students skip ahead to Level 12. You will spend the rest of the class here. This final level is a chance for students to draw whatever they'd like using the coding blocks.
   Here's an example we made:



7. Students can experiment with whichever blocks they'd like, but you can encourage them to try using at least one repeat block (loop) in their code.

8. Circle the room to offer assistance as needed. Ask students to show you what they made.

9. **SAVING THEIR WORK** At the end of the class, help students save a record of what they made. Unfortunately, *Code.org* will not save students' images automatically. We suggest that you either take a screenshot of their window or use a phone camera to take a picture of each student's art.

10. Wrap up the class using some of the suggested discussion questions.

11. After the class, review the **ASSESSMENT FRAMEWORK** on **Page 3.26**. It is meant as a tool to help you evaluate each student's progress throughout the *Coding in the Classroom* program.

DISCUSSION QUESTIONS

Did you make any mistakes today? If so, how did you try fixing it?

What is your favourite thing you've learned about coding so far?

Would you like to continue coding? What would you like to learn how to do?

What do we call an event that repeats itself?

A loop.

Why do we use loops?

They make our code shorter and more efficient which allows our computers to run faster and execute more complicated tasks.

Assessment framework

This chart will help you assess your students' work during the Final Project and the *Coding in the Classroom* program as a whole. It is based on the Ontario Mathematics (2020) curriculum.

KNOWLEDGE AND UNDERSTANDING

	LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4
Knowledge of content	Uses few block types (draw, jump, repeat, set colour, set pattern) in final project Does not answer questions during online classes, even with assistance	Uses some block types (draw, jump, repeat, set colour, set pattern) in final project Answers questions during online classes with some assistance	Uses most block types (draw, jump, repeat, set colour, set pattern) in final project Answers some questions during online classes	Uses many block types (draw, jump, repeat, set colour, set pattern) in final project Answers many questions during online classes
Understanding of content	Rarely uses repeat loops when appropriate to make code more efficient	Sometimes uses repeat loops when appropriate to make code more efficient	Often uses repeat loops when appropriate to make code more efficient	Always uses repeat loops when appropriate to make code more efficient

THINKING

	LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4
Use of planning skills	Rarely predicts which coding blocks are needed to solve a puzzle or draw a picture	Sometimes predicts which coding blocks are needed to solve a puzzle or draw a picture	Often predicts which coding blocks are needed to solve a puzzle or draw a picture	Usually predicts which coding blocks are needed to solve a puzzle or draw a picture
Use of processing skills	Converts ideas into code with limited effectiveness	Converts ideas into code with some effectiveness	Converts ideas into code with considerable effectiveness	Converts ideas into code with high degree of effectiveness
Use of critical/ creative thinking processes	Troubleshoots and "debugs" code with much assistance Does not experiment with new ideas in final project	Troubleshoots and "debugs" code with assistance Experiments with a new idea in final project	Troubleshoots and "debugs" code with some assistance Experiments with a few new ideas in final project	Troubleshoots and "debugs" code with little assistance Experiments with many new ideas in final project

COMMUNICATION

	LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4
Expression and organization of ideas and information in oral, visual, and/or written forms	Rarely organizes code clearly or uses correct terminology	Organizes codes somewhat clearly and sometimes uses correct terminology	Often organizes code clearly and mostly uses correct terminology	Always organizes code clearly and uses correct terminology
Communication for different audiences and purposes in oral, visual, and/or written forms	Explains code and artwork, either orally or in writing, with limited effectiveness	Explains code and artwork, either orally or in writing, with some effectiveness	Explains code and artwork, either orally or in writing, with considerable effectiveness	Explains code and artwork, either orally or in writing, with a high degree of effectiveness

APPLICATION

	LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4
Application of knowledge and skills in familiar contexts	Completes coding levels with much assistance	Completes coding levels with assistance	Completes coding levels with some assistance	Completes coding levels with little or no assistance
Application of knowledge and skills to new contexts	Applies coding knowledge to complete final project with much assistance	Applies coding knowledge to complete final project with assistance	Applies coding knowledge to complete final project with some assistance	Applies coding knowledge to complete final project with little or no assistance
Making connections within and between various contexts	Rarely participates in offline coding activities Rarely makes connections between coding concepts and everyday life	Participates somewhat in offline coding activities Sometimes makes connections between coding concepts and everyday life	Participates in offline coding activities Makes connections between coding concepts and everyday life	Participates fully in offline coding activities Often makes connections between coding concepts and everyday life

Additional resources

CODE.ORG

Code.org is the same platform we used for the online lessons. We recommend any activities about loops or nested loops, but there are hundreds of lessons on a wide range of coding topics.

code.org/student/elementary

SCRATCH

Scratch is the block-based coding platform we use with Grade 5-8 students. It is a lot like Code.org, but it is more open-ended. Check out their "Getting Started" tutorial.

scratch.mit.edu/ideas

CANADA LEARNING CODE

From lesson plans to professional development, this website has a wealth of resources for teaching coding.

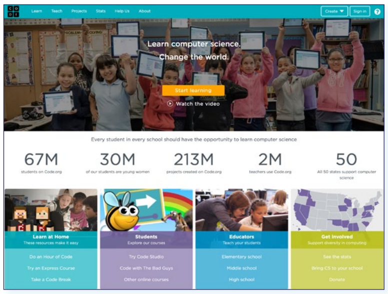
canadalearningcode.ca

TVO CODING IN THE CLASSROOM

Watch webinars and read suggestions for teaching Ontario's coding curriculum.

outreach.tvolearn.com/codingintheclassroom

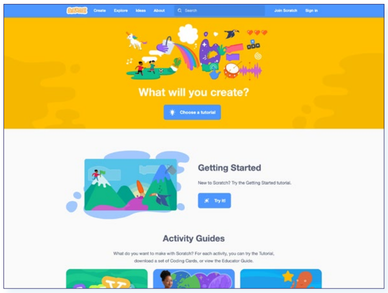
*Coding screenshots are sourced from **code.org**



Every student in every school should have the opportunity to learn computer science

67M	30M	213M	2M	50
students on Code.org	of our students are young women	projects created on Code.org	teachers use Code.org	All 50 states support computer science

Code.org



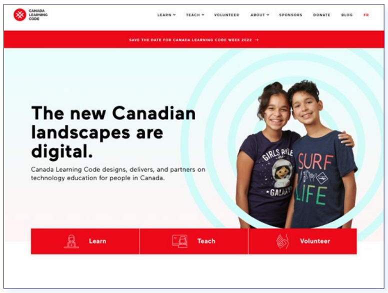
Getting Started

New to Scratch? Try the Getting Started tutorial.

Activity Guides

What do you want to make with Scratch? For each activity, you can try the Tutorial, download a set of Coding Cards, or view the Educator Guide.

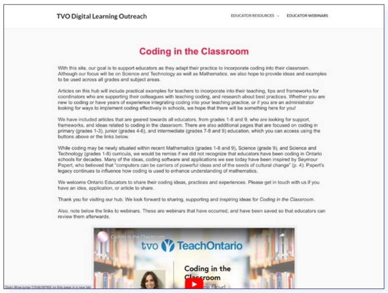
Scratch



The new Canadian landscapes are digital.

Canada Learning Code designs, delivers, and partners on technology education for people in Canada.

Canada Learning Code



Coding in the Classroom

With this site, our goal is to support educators as they adapt their practice to incorporate coding into their classroom. Although our focus will be on Science and Technology as well as Mathematics, we also hope to provide ideas and examples to be used across all grades and subject areas.

Articles on this hub will include practical examples for teachers to incorporate into their teaching, tips and frameworks for coordinators who are supporting their colleagues with teaching coding, and research about best practices. Whether you are new to coding or have years of experience integrating coding into your teaching practice, or if you are an administrator looking for ways to implement coding effectively in schools, we hope that there will be something here for you!

We have included articles that are geared towards all educators, from grades 1-8 and 9, who are looking for support, frameworks, and ideas related to coding in the classroom. There are also additional pages that are focused on coding in primary (grades 1-3), junior (grades 4-6), and intermediate (grades 7-8 and 9) education, which you can access using the buttons above or the links below.

While coding may be newly situated within recent Mathematics (grades 1-8 and 9), Science (grade 9), and Science and Technology (grades 1-8) curricula, we would be remiss if we did not recognize that educators have been coding in Ontario schools for decades. Many of the ideas, coding software and applications we see today have been inspired by Seymour Papert, who believed that "computers can be carriers of powerful ideas and of the seeds of cultural change" (p. 4). Papert's legacy continues to influence how coding is used to enhance understanding of mathematics.

We welcome Ontario Educators to share their coding ideas, practices and experiences. Please get in touch with us if you have an idea, application, or article to share.

Thank you for visiting our hub. We look forward to sharing, supporting and inspiring ideas for *Coding in the Classroom*.

Also, note below the links to webinars. These are webinars that have occurred, and have been saved so that educators can review them afterwards.

TVO Coding in the Classroom