GRADE 3 Repeating Events | GRADE 4 Nested Events | GRADE 5 Conditional Statements

{CODING} IN THE CLASSROOM

GRADE 4 TEACHERS' GUIDE

Coding with nested events

Welcome to NII Explore's *Coding in the Classroom* program for Grade 4 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 4 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 Grade 3, students learn about repeating events. A repeating event, called a **loop**, makes something happen multiple times.

SCHEDULE AT A GLANCE

WEEK 1

- **Online Lesson 1**
  Artist 1 & 2
- **Offline Activity 1**
  Loop, Paper, Scissors

WEEK 2

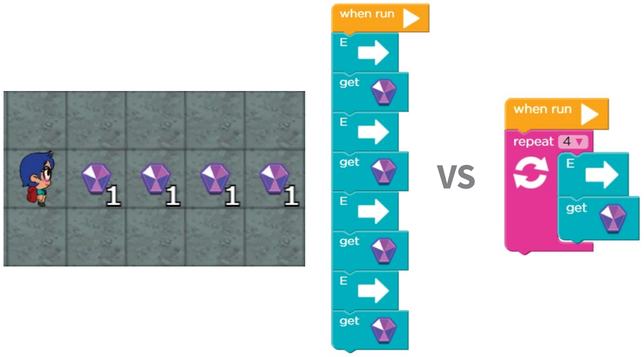
- **Online Lesson 2**
  Bee: Nested Loops
- **Offline Activity 2**
  Nested Loop Dances

WEEK 3

- **Online Lesson 3**
  Artist: Nested Loops
- **Offline Activity 3**
  The Art of Coding

WEEK 4

- **Final Project**
  Making Art with Code

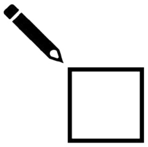


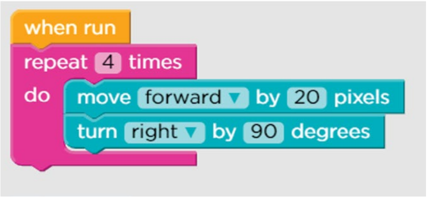
***Loop example.** Both algorithms will solve the maze, but the pink repeat loop makes the code on the right shorter and more efficient.*

Loops make our code shorter and therefore easier to read and write. Instead of writing the same steps over and over, we can use a loop.

In **GRADE 4**, students start coding with **nested events**. Nested events are when loops are "nested" inside of another loop. *It's like loop-ception!*

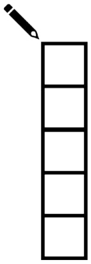
Here's an example. This code uses a repeat loop to draw a square:

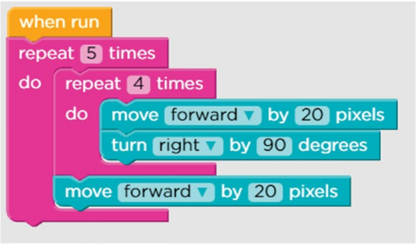




```
when run
repeat 4 times
do  move forward by 20 pixels
    turn right by 90 degrees
```

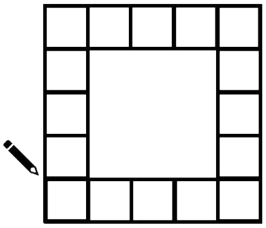
But what if you wanted to draw 5 squares? You could put the previous repeat loop inside another repeat loop like this:

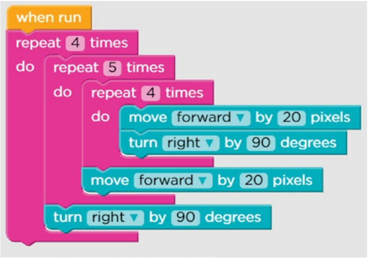




```
when run
repeat 5 times
do  repeat 4 times
    do  move forward by 20 pixels
        turn right by 90 degrees
    move forward by 20 pixels
```

This is a nested event – we have a repeat loop nested inside another loop. And we can keep going! Here's a loop inside a loop inside a loop:

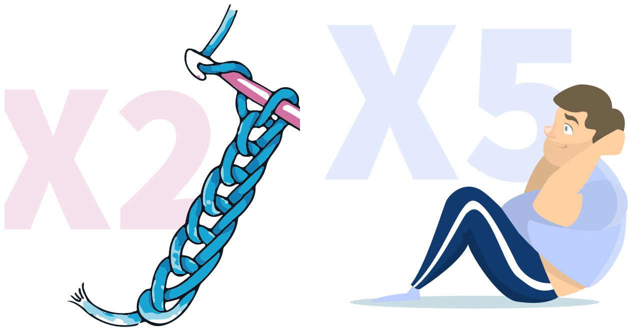




```
when run
repeat 4 times
do  repeat 5 times
    do  repeat 4 times
        do  move forward by 20 pixels
            turn right by 90 degrees
        move forward by 20 pixels
    turn right by 90 degrees
```

By nesting our events, we can execute complicated sets of instructions using just a few lines of code. In the previous example, we made our computer draw 100 individual lines, but we did it using only 8 lines of code (each coloured block represents one line of code).

Nested events appear all the time in computer code because they are so efficient. We can even find examples of nested events in our everyday lives!



Nested loops in everyday life. *Crochet patterns and exercise routines are both examples of nested loops. We repeat actions (e.g., crochet 2 stitches or do 5 sit-ups) and then repeat those repeats (e.g., do 5 double crochets, do 2 sets of 5 sit-ups).*

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 – Artist 1 & Artist 2

Students will write code using simple loops, nested loops, and basic geometry (side lengths and angles) to create digital art.

PREP Log students onto computers and open activity link.

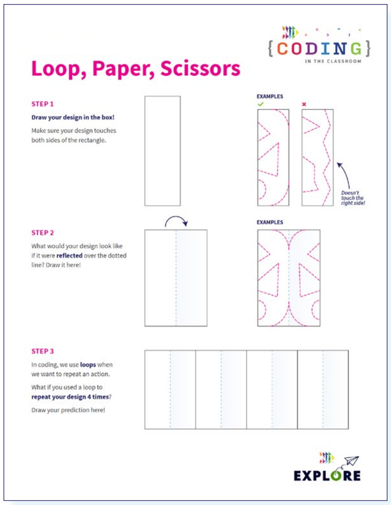
POST Complete “Offline Activity 1 – *Loop, Paper, Scissors*” before next online session.

See **Page 4.5** for lesson instructions.

Offline Activity 1 – Loop, Paper, Scissors

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

See **Page 4.8** for activity instructions.



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 1** Loop, Paper, Scissors*

WEEK 2

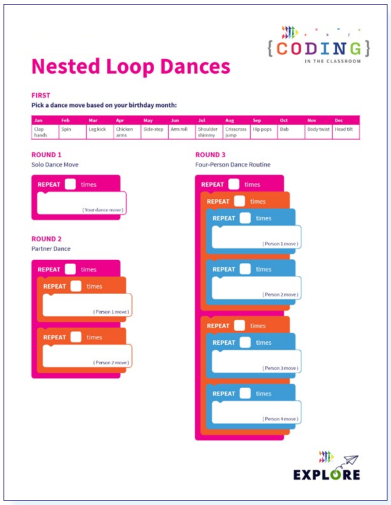
Online Lesson 2 – Bee: Nested Loops

Students will code with loops and nested events to solve bee-themed puzzles. The lesson will also discuss some applications of coding.

PREP Same as Week 1.

POST Complete “Offline Activity 2 – *Nested Loop Dances*” before next online session.

See **Page 4.12** for lesson instructions.



Nested Loop Dances

FIRST

Pick a dance move based on your birthday month:

Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Clap hands	Spin	Leg kick	Chicken arms	Side-step	Arm roll	Shoulder shimmy	Crisscross jump	Hip pops	Dab	Body twist	Head tilt

ROUND 1

Solo Dance Move

```
REPEAT □ times
    ( Your dance move )
```

ROUND 2

Partner Dance

```
REPEAT □ times
    REPEAT □ times
        ( Person 1 move )
    REPEAT □ times
        ( Person 2 move )
```

ROUND 3

Four-Person Dance Routine

```
REPEAT □ times
    REPEAT □ times
        REPEAT □ times
            ( Person 1 move )
        REPEAT □ times
            ( Person 2 move )
    REPEAT □ times
        REPEAT □ times
            ( Person 3 move )
        REPEAT □ times
            ( Person 4 move )
```

***Week 2** Nested Loop Dances*

Offline Activity 2 – Nested Loop Dances

Your class will learn more about loops and nested loops while coming up with collaborative dances.

See **Page 4.15** for activity instructions.

WEEK 3

Online Lesson 3 – Artist: Nested Loops

This lesson reinforces concepts from the previous Artist activities, this time with a focus on nested loops. Discuss the final project at the end of the class.

PREP Same as Week 1.

POST Complete “Offline Activity 3 – *The Art of Coding*” and the Final Project.

See **Page 4.18** for lesson instructions.

Offline Activity 3 – The Art of Coding

Students will plan the piece of digital art that they will make in the Final Project.

See **Page 4.21** for activity instructions.

WEEK 4

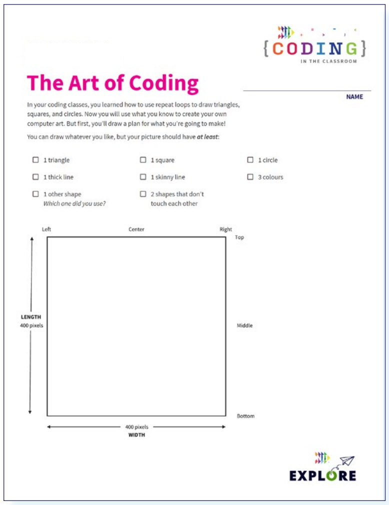
Final Project – Making Art with Code

Students will apply their learning to create personalized pieces of digital art using the same coding platform as the online lessons. When they’re finished, students will save their artwork and code for you to review.

See **Page 4.25** for project instructions and **Page 4.27** for assessment criteria.

AFTER WEEK 4

Keep the coding going with the additional resources on **Page 4.30**!



{CODING} IN THE CLASSROOM

The Art of Coding

NAME

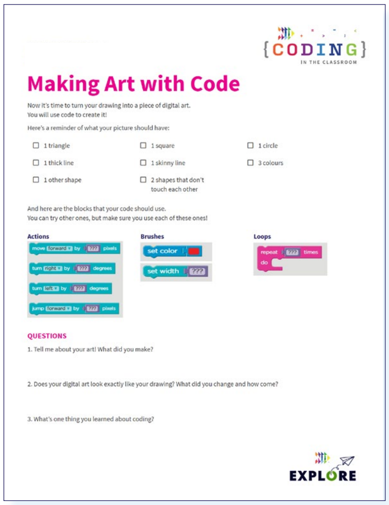
In your coding classes, you learned how to use repeat loops to draw triangles, squares, and circles. Now you will use what you know to create your own computer art. But first, you’ll draw a plan for what you’re going to make!

You can draw whatever you like, but your picture should have *at least*:

- □ 1 triangle
- □ 1 thick line
- □ 1 other shape *Which one did you use?*
- □ 1 square
- □ 1 skinny line
- □ 2 shapes that don’t touch each other
- □ 1 circle
- □ 3 colours

EXPLORE

***Week 3** The Art of Coding*



{CODING} IN THE CLASSROOM

Making Art with Code

Now it’s time to turn your drawing into a piece of digital art.
You will use code to create it!

Here’s a reminder of what your picture should have:

- □ 1 triangle
- □ 1 thick line
- □ 1 other shape
- □ 1 square
- □ 1 skinny line
- □ 2 shapes that don’t touch each other
- □ 1 circle
- □ 3 colours

And here are the blocks that your code should use.
You can try other ones, but make sure you use each of these ones!

Actions

```
move forward by ??? pixels
turn right by ??? degrees
turn left by ??? degrees
jump forward by ??? pixels
```

Brushes

```
set color
set width ???
```

Loops

```
repeat ??? times
do
```

QUESTIONS

1. Tell me about your art! What did you make?

2. Does your digital art look exactly like your drawing? What did you change and how come?

3. What’s one thing you learned about coding?

EXPLORE

***Week 4** Final Project*

ONLINE LESSON 1

Artist 1 & Artist 2

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.

The two Artist activities have students use simple loops, nested loops, and basic geometry (length and angles) to draw shapes. Artist 1 is a good way to practice using angles. If your class finishes Artist 1 quickly, you can move on to Artist 2.

QUICK LINKS

Artist 1
studio.code.org/s/20-hour/lessons/5/levels/1

Artist 2
studio.code.org/s/20-hour/lessons/7/levels/1

PowerPoint
Grade 4 – Week 1 – The Artist

CURRICULUM CONNECTIONS

CODING SKILLS

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

SPATIAL SENSE

- **E1.1** - identify geometric properties of rectangles, including the number of right angles, parallel and perpendicular sides, and lines of symmetry
- **E2.4** - identify angles and classify them as right, straight, acute, or obtuse

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. If a video starts to auto-play, have them mute their computers and/or close the video. They may also 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 the slides to explain what coding is. "Coding is when we give instructions to a computer."

SLIDES 3 TO 8 - HISTORY MINUTE: ANIMATION

Use these slides to highlight animation as an application of coding. The history minute discusses the history of Pixar and computer animation. See the slide notes for talking points.

SLIDE 9 - THE CODER’S CODE

Have students recite the “Coder’s Code” by repeating after you. This helps set expectations for the students.

SLIDE 10 - READY TO START

Switch your screen over to code.org.

ARTIST 1 - LEVEL 1

Explain the premise of the activity. "We will be using code to draw different shapes."

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.

Go over the Artist controls as a class. The Artist can **move forward** by a certain number of pixels (pixels are how we measure distance on a computer screen) and **turn left** or **turn right** by a number of degrees. Students may need a reminder about what degrees mean.

Let students use trial and error to solve the first level.

LEVEL 2

Give students time to try this level on their own then go over the answer as a class.

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 the Artist appears) by dragging the right side of the window.

LEVEL 3

Show students the new pink block (called a **repeat loop**). They may have learned about repeat loops in Grade 3, but students will probably need a refresher. Loops allow us to repeat the same series of steps multiple times without rewriting the same code.

Show students how to place blocks inside the repeat loop. Let them figure out which two blocks need to go into the loop.

LEVELS 4 TO 6

Give students time to work at their own pace.

Encourage them to help others if they finish quickly.

Tell students about target block numbers (e.g., “Workspace: **2/4 blocks**” means they’ve used 2 out of the 4 blocks allowed) and to read the instructions.

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.

LEVEL 7

Challenge level: Help students work through this together if they need extra support.

Take up level together.

Point out that students can draw their own pictures using code, just like this example.

LEVEL 8

Point out the new “move backward” command.

LEVEL 9

Remind or teach students that there are 360 degrees in a circle. “If we turn 1 degree each time and a circle has 360 degrees, how many repeats do we need?”

LEVEL 10

If students finish quickly, have them switch over to Artist 2 and try a few levels there.

ARTIST 2 - LEVEL 1

Remind students about using repeat loops.

Let them try to figure out the correct angle for drawing triangles.

LEVELS 2+

Give students the rest of the time to work at their own pace. It’s very unlikely that they will finish all the levels in an hour.

SLIDE 11 - RECAP

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

Ask students "What did we learn about today? How do loops help us when we code? How do we measure the size of shapes on a computer screen? (Pixels) How many degrees are in a circle?" (360).

SLIDE 12 - 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 13 - 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

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 lessons.
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

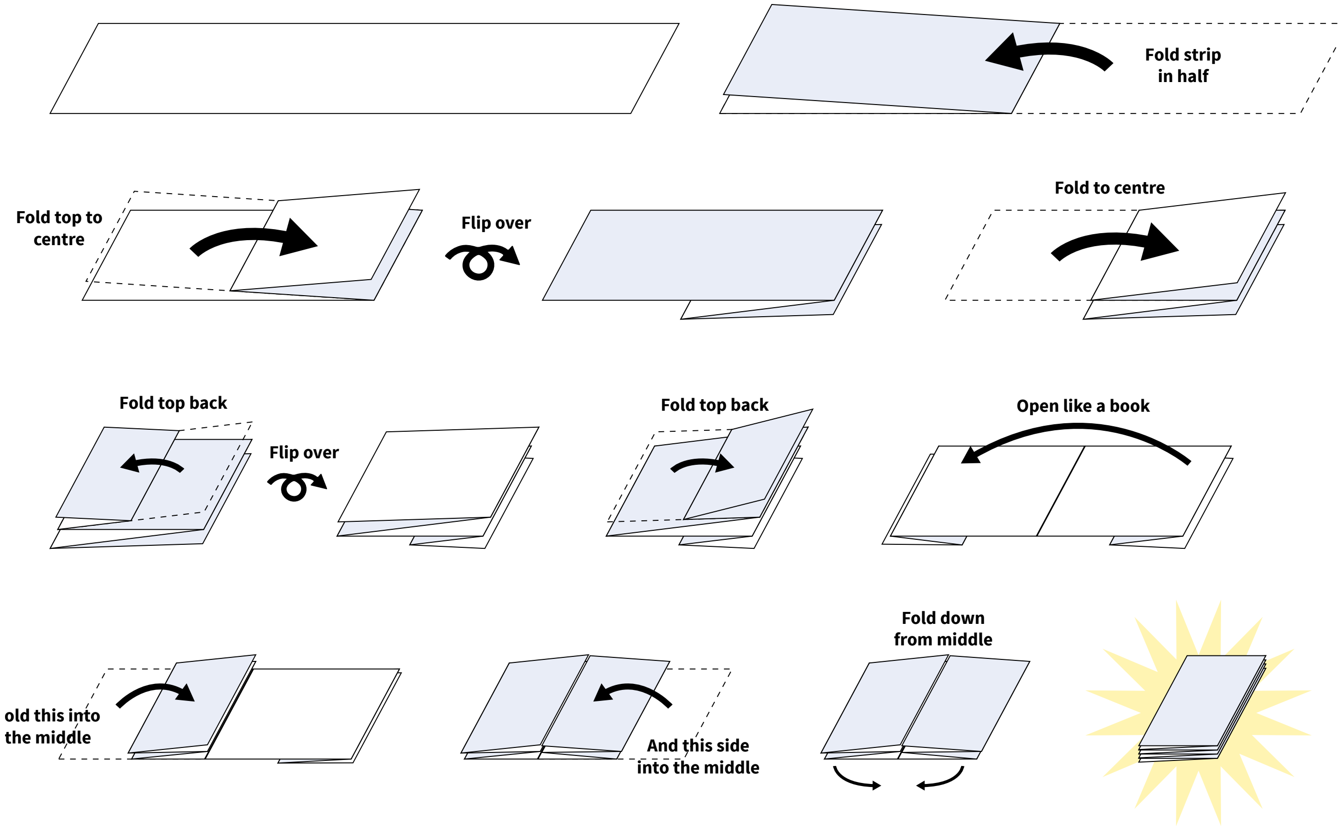
1 blank piece of paper

Scissors

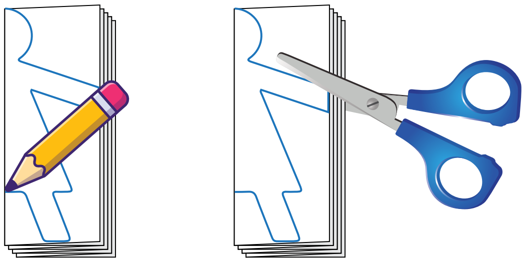
5. Students will finish their worksheet by predicting what their pattern would look like if it were repeated 4 times. Have them draw their prediction in the third section of their sheet.

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

   - Fold and then cut the 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.
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.

Example of a finished piece of paper art.

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? How does this relate to 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 and 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 you think 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?

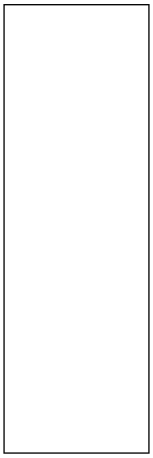
ACTIVITY 1 WORKSHEET

Loop, paper, scissors

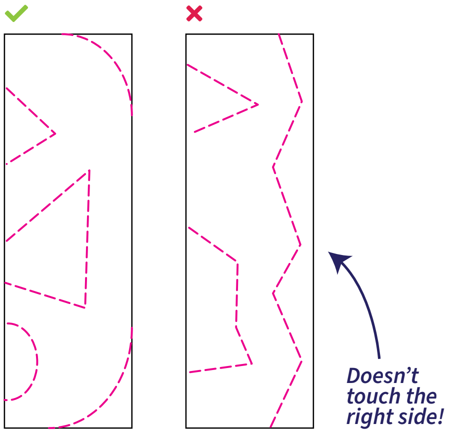
STEP 1

Draw your design in the box!

Make sure your design touches both sides of the rectangle.

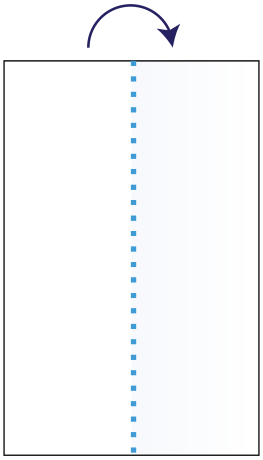


EXAMPLES

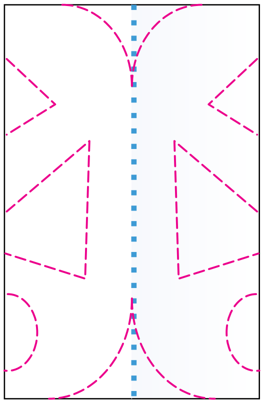


STEP 2

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



EXAMPLE

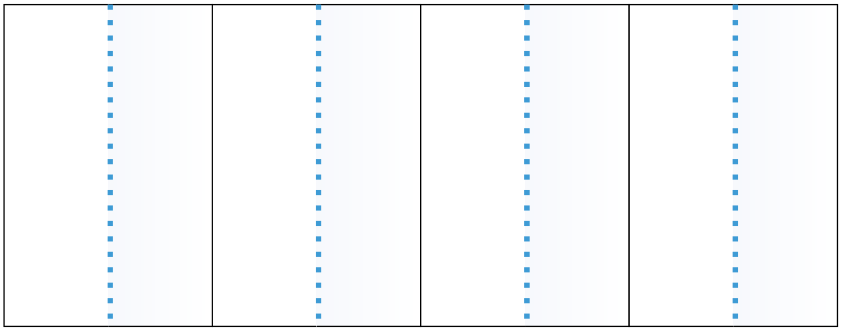


STEP 3

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!



ONLINE LESSON 2

Bee: Nested loops

60 MINUTES

In this lesson, students will use loops to help the bee collect nectar and make honey. The early levels emphasize nested loops, and the later levels introduce "while" loops.

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, repeating, and nested events
- **Math C3.2** – read and alter existing code that involves sequential, concurrent, repeating, and nested events, and describe how changes to the code affect the outcomes

PATTERNS AND RELATIONSHIPS

- **Math C1.1** - identify and describe repeating and growing patterns, including patterns found in real-life contexts
- **Math C1.3** - determine pattern rules and use them to extend patterns, make and justify predictions, and identify missing elements in repeating and growing patterns

NUMBER SENSE

- **Math B1.2** – compare and order whole numbers up to and including 10 000, in various contexts

QUICK LINKS

Student Activity Link
studio.code.org/s/course3/lessons/13/levels/1

PowerPoint
Grade 4 – Week 2 – Nested Loops

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.)

SLIDE 2 - 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 potential answers.

SLIDES 3 & 4 - 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 five flowers).

LEVEL 1

Remind students how to click and drag blocks, run program, use repeat loops.

Challenge students to solve the level using a repeat loop.

LEVEL 2

Watch the video together as a class – it introduces the concept of **nested loops**. Students can skip the video on their own devices.

Ask students to identify the pattern in this puzzle (move forward, get nectar...). “Whenever you see a pattern, it means we can use a loop.”

Point out the “-/5 blocks” and remind them about the target number of blocks. “We can solve this level using only 5 coding blocks.”

See if anyone can solve the puzzle using only 5 blocks.

Demo the solution and explain nested loops . “There is a loop nested inside of another loop. It’s loop-ception! We call these nested loops.”

LEVEL 3

Let students know that grey blocks cannot be deleted, only moved around.

Read through the code together and try to figure out what it does.

Have students read the instructions at the top to solve the level.

LEVELS 4 & 5

Let students try these by themselves or with a partner.

Remind them to check the target number of blocks.

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 nested loop.

LEVEL 6

Explain while loops. “There are different kinds of loops. So far, we’ve been using loops where you have to say exactly how many times to repeat. It’s sometimes called a **for loop**. The other type is a **while loop** – it will repeat while a certain thing is true, but the actual number of repeats can change.”

Real-world examples:

For loop - Clean room FOR 15 minutes (we need to specify an exact number) vs. **While loop** - Clean room WHILE room is dirty (doesn’t give an exact number).

Do level together.

LEVEL 7

Encourage students to try on their own, but be ready to clarify while loops.

LEVELS 8 TO 11

Give students time to work at their own pace.

Encourage them to help others if they finish quickly.

Remind students about target block numbers and to read the instructions.

These levels are trickier than Week 1, so your class probably won’t get through all the levels.

LEVELS 12 & 13

Tell students that this is a mini “quiz” to see what they learned.

Encourage students to work independently and not share answers until the end.

SLIDE 5 - 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? (A repeat block – it makes actions repeat) What’s it called when we put a loop inside another loop? (A nested loop) What’s a while loop?" (A loop that repeats while a certain thing is true).

SLIDE 6 - POEM OF THE DAY

Share this week’s Poem of the Day.

SLIDE 7 - WHAT’S NEXT?

Give a quick preview of next lesson. “We will use loops and nested loops to make more digital art."

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

OFFLINE ACTIVITY 2

Nested loop dances

45 MINUTES

Learn about loops and nested loops by coming up with collaborative dances!

MATERIALS

Each student will need:

Pencil or pen

1 worksheet

LEARNING OBJECTIVES

- Write pseudocode with nested loops
- Create and present a dance routine

CURRICULUM CONNECTIONS

- Math C3.1 & C3.2 (Coding)
- Arts A1 (Dance)

SET-UP

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

INSTRUCTIONS

SUMMARY

Each student will have a dance move. They will repeat their move a certain number of times. Later, they will combine their moves with other students to make a more complex dance routine.

1. Ask students to recall what they learned about nested loops in the online class.
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 write instructions for a dance routine."
3. Demonstrate each of the 12 dance moves on the worksheet. It's not super important what the moves look like as long as they're all distinct. Students are assigned a dance move using their birthdays.

4. **ROUND 1** Have each student pick a number of times that they will repeat their dance move (e.g., spin 4 times). **ASK** "What is it called in coding when we repeat an action several times?" (A loop) They will record their dance move and number of repeats in the first repeat loop on their worksheet.
5. **ROUND 2** Match everyone with a partner. Each pair will now combine their individual dance moves to create a partner dance. They should repeat each person's section at least twice (e.g., Clap 5 times then arm roll 3 times, repeat whole thing 3 times). They will record their dance routine as code in the Round 2 section of their worksheet.
6. **ROUND 3** After each pair has had time to practice, match them with another pair to form a group of 4. These two groups will now combine their two dances to form one big dance routine. Give them time to practice. They will fill in their final routine in the Round 3 section of the worksheet. Their "code" will look like a nested loop.
7. Take turns having each group perform their dance for the class. **OPTIONAL** Set the dance to music.

DISCUSSION QUESTIONS

What did you notice each time we added a repeat loop?

How long would your dance get if we added a fourth repeat loop?

How do repeat loops help us when we're writing instructions?

What did you find the trickiest? What made it tricky?

We used repeat loops to write a dance routine. What other instructions could we write using loops?

ACTIVITY 2 WORKSHEET

Nested loop dances

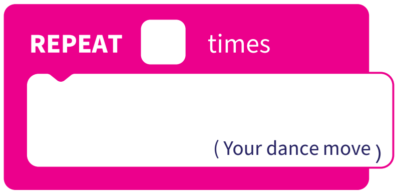
FIRST

Pick a dance move based on your birthday month:

Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Clap hands	Spin	Leg kick	Chicken arms	Side-step	Arm roll	Shoulder shimmy	Crisscross jump	Hip pops	Dab	Floss	Head tilt

ROUND 1

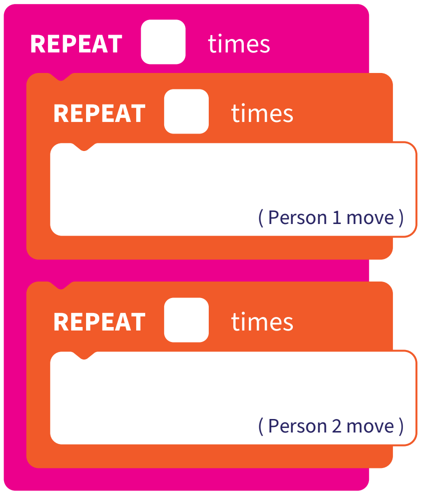
Solo Dance Move



```
REPEAT □ times
    ( Your dance move )
```

ROUND 2

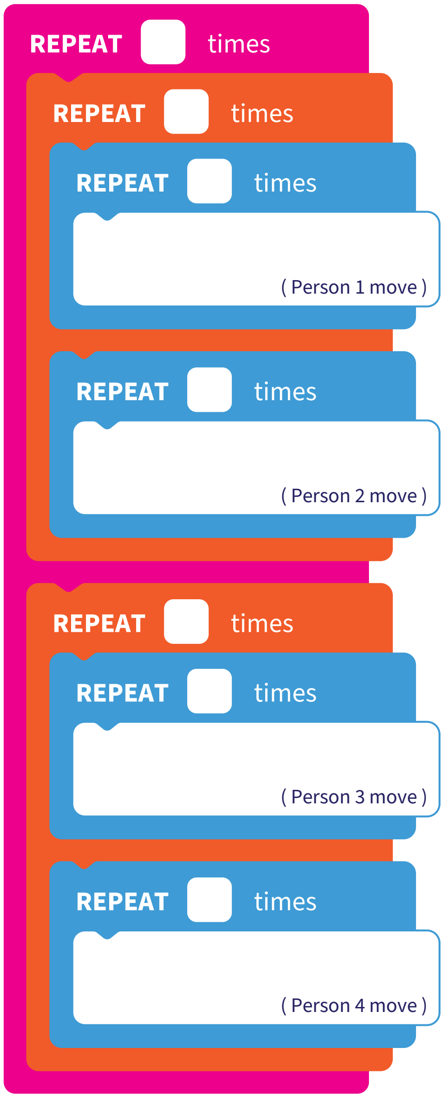
Partner Dance



```
REPEAT □ times
    REPEAT □ times
        ( Person 1 move )
    REPEAT □ times
        ( Person 2 move )
```

ROUND 3

Four-Person Dance Routine



```
REPEAT □ times
    REPEAT □ times
        REPEAT □ times
            ( Person 1 move )
        REPEAT □ times
            ( Person 2 move )
    REPEAT □ times
        REPEAT □ times
            ( Person 3 move )
        REPEAT □ times
            ( Person 4 move )
```

ONLINE LESSON 3

Artist: Nested loops

60 MINUTES

This lesson reinforces concepts from the previous Artist activities, but this time the focus is on nested loops. Students will combine simple shapes to create pictures, and the final level is a blank canvas where students can experiment in preparation for their final project.

CURRICULUM CONNECTIONS

CODING

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

SPATIAL SENSE

- **E1.1** - identify geometric properties of rectangles, including the number of right angles, parallel and perpendicular sides, and lines of symmetry
- **E2.4** - identify angles and classify them as right, straight, acute, or obtuse

QUICK LINKS

Student Activity Link
studio.code.org/s/course2/lessons/19/levels/1

PowerPoint
Grade 4 – Week 3 – Artist and Final Project Preview

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.

SLIDES 1 & 2 - WEEKS 1 & 2 RECAP

Ask students to recall what they learned about in Weeks 1 and 2 (e.g., pixels, loops, nested loops, angles, circles).

SLIDES 3 & 4 - 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 (Artist with three triangles).

LEVEL 1

Read the instructions together and see if students can figure out the answer.

Ask them to remember what a loop within a loop is called (a nested loop).

Demo what the tortoise and hare slide bar does. It controls the speed of the drawing.

LEVEL 2

Have students read instructions and try level for themselves.

Ask if they notice anything about the number of repeats versus the turn angle. “We turn 60 degrees and we repeat that 6 times. That’s 6 x 60 which equals 360. Where have we seen that number before? (Number of degrees in a circle) Why do you think that shows up here? (We have to do a full rotation to draw this shape and end up back where we started).

LEVEL 3

Let students read the instructions and try it for themselves.

Offer to work through it together if students need help with the math.

HINT: 12 repeats x 30 degree turn = 360 degrees

LEVEL 4

“So far we’ve been focused on drawing simple shapes. Now we are going to use those shapes to create entire pictures.”

Read instructions with students, remind them that they can peek at their code from previous levels to remember how to do it.

LEVEL 5

Introduce students to the new “jump forward” command and then let them try the level for themselves.

LEVEL 6

Let students work by themselves, be available to help as needed.

LEVELS 7 TO 9

"For the next three levels, we will be applying what we know to draw an entire picture."

Let students work through all three levels.

LEVELS 10 TO 13

If time permits, allow students to keep working independently.

SLIDE 5 - RECAP

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

Ask students: What's something you learned about coding? What shapes did we learn how to draw? What's one part of our Coder's Code?

Use slides to recap all three lessons.

SLIDES 6 & 7 - EXPLAIN FINAL PROJECT

In the third offline activity, students will be drawing pictures using simple shapes. Then they will make their drawings using code as part of the final project.

When students are done, they will share their work for you to evaluate.

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 8 - POEM OF THE DAY

Share the final Poem of the Day.

SLIDE 9 - 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

The art of coding

30 MINUTES

Students will prepare for their final coding project by planning a piece of digital art.

LEARNING OBJECTIVES

- Know different geometric shapes
- Identify pixels as a unit of geometric measurement
- Compose a piece of art

CURRICULUM CONNECTIONS

- Math E2 (Measurement)
- Media Literacy 3.4
- Arts D1 (Visual Arts)

SET-UP

Print the Activity 3 and Final Project worksheets on the front and back of a single page. They will use the front for this activity and the back for their final projects.

INSTRUCTIONS

1. Ask students to recall what they learned in the online coding classes.
2. Explain the premise of the activity. “You will be planning a piece of art using different shapes. The next time we have the computers you will be using code to create a digital version of your picture.”
3. Hand out worksheets and review the instructions together. There are 8 things that they should be sure to include in their drawing. Remind them that whatever they draw is what they will be recreating using code so it might be hard to draw fine details (e.g., think stick figures rather than real-looking people). **See right** for example.

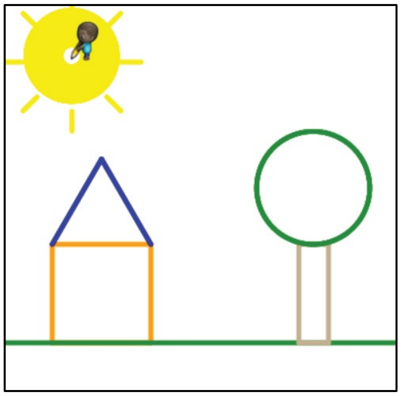
MATERIALS

Each student will need:

Pencil or pen

Pencil crayons or markers

1 worksheet



4. If students ask, you can explain pixels and the labels on their drawing box. **SAY** "Pixels are how we measure sizes on a computer screen. You can think of it like a single dot on the screen. When you code your art, you'll get a page that is 400 pixels by 400 pixels."

   The pixel scale will be useful for their final project to help them figure out how big to make their shapes.

5. Give students time to work on their drawings. Circle the room and give assistance when needed. Remind students to check that their pictures use all 8 items from the checklist.

6. When students are finished their drawings, ask them to decide what order they will have their Artist draw each shape. **ASK** Which shape will you draw first? When will you need to switch colours? Are there any shapes that you don't know how to code yet?

7. Collect the worksheets. You will hand them back when you have computer time for the final project.

ACTIVITY 3 WORKSHEET

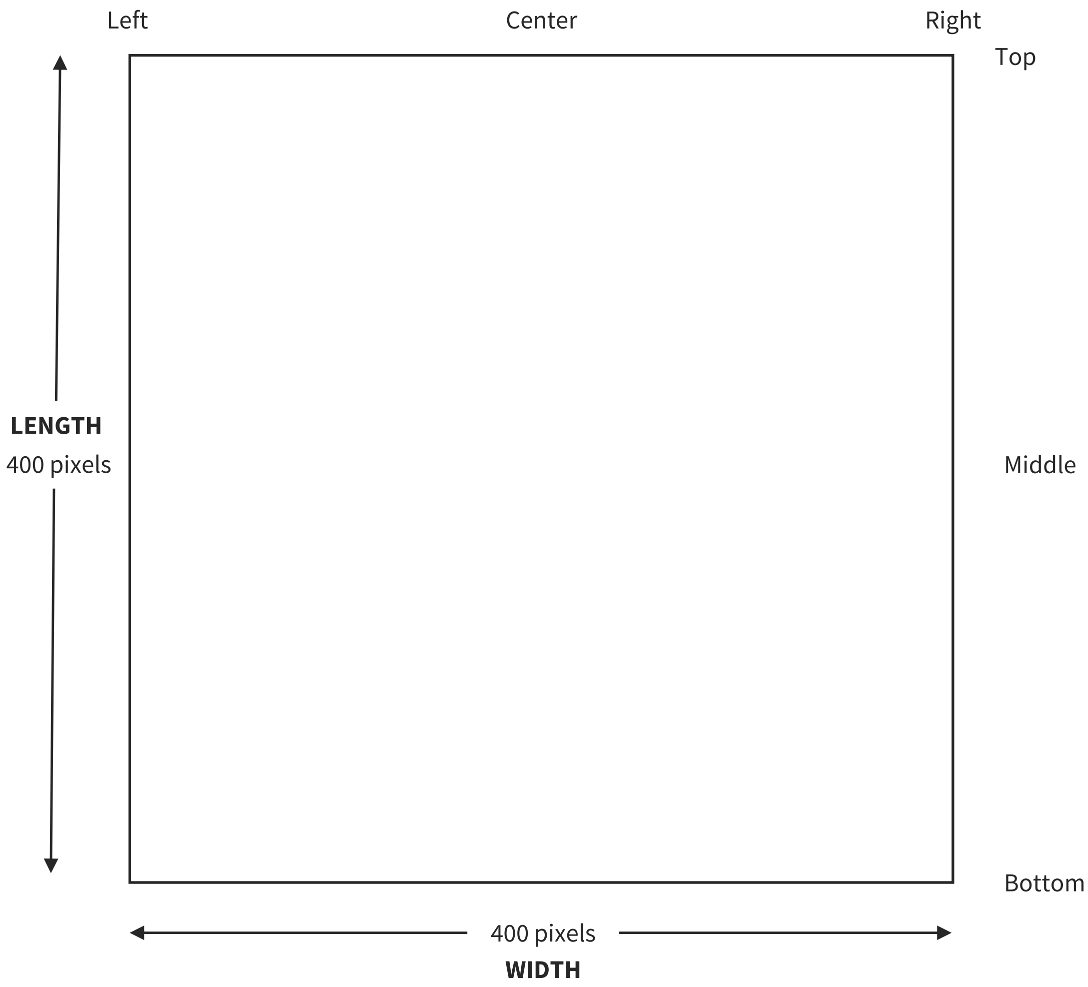
The art of coding

NAME

In your coding classes, you learned how to use repeat loops to draw triangles, squares, and circles. Now you will use what you know to create your own computer art. But first, you'll draw a plan for what you're going to make!

You can draw whatever you like, but your picture should have ***at least***:

- □ 1 triangle
- □ 1 thick line
- □ 1 other shape
  Which one did you use?
- □ 1 square
- □ 1 skinny line
- □ 2 shapes that don't touch each other
- □ 1 circle
- □ 3 colours



FINAL PROJECT WORKSHEET

Making art with code

Now it’s time to turn your drawing into a piece of digital art. You will use code to create it!

Here’s a reminder of what your picture should have:

□ 1 triangle

□ 1 thick line

□ 1 other shape

□ 1 square

□ 1 skinny line

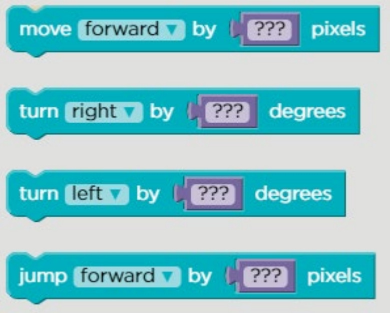
□ 2 shapes that don’t touch each other

□ 1 circle

□ 3 colours

And here are the blocks that your code should use.
You can try other ones, but make sure you use each of these ones!

Actions



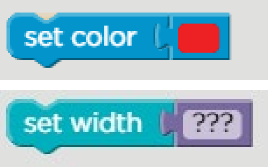
move forward ▾ by ??? pixels

turn right ▾ by ??? degrees

turn left ▾ by ??? degrees

jump forward ▾ by ??? pixels

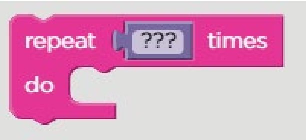
Brushes



set color 🟥

set width ???

Loops



repeat ??? times
do

QUESTIONS

1. Tell me about your art! What did you make?

2. Does your digital art look exactly like your drawing? What did you change and how come?

3. What’s one thing you learned about coding?

FINAL PROJECT

Making art with code

60 MINUTES

Students will apply their coding knowledge to make a piece of digital art. It's an opportunity to demonstrate what they learned and a chance to assess their work.

QUICK LINKS

Student Activity Link
code.org

LEARNING OBJECTIVES

- Write original code to create a digital representation of an image
- Alter and debug existing code

CURRICULUM CONNECTIONS

- Math C3.1 & C3.2 (Coding)
- Media Literacy 3.4
- Arts D1

SET-UP

View this **video tutorial** about the project. Hand back student worksheets from Offline Activity 3 – The Art of Coding. Have students log onto computers and open activity link. This should take them to the **code.org** homepage. From there, select "Projects" from the top menu then choose "Artist" under "Start a new project".

INSTRUCTIONS

SUMMARY

Students will be using their drawings from Activity 3 and turning them into digital art using code. At the end, students will answer questions on their worksheet, and save their code and art for you to assess. See **ASSESSMENT** on **Page 4.27**.

1. Explain challenge to students. “You will be using what we’ve learned about coding to recreate your art on the computer.”
2. Review instructions and worksheet as a class. The sheet outlines what elements their picture should include and which code blocks they need to use.
3. Students will start by naming their project. Choose “Rename” in the top left. They can title their work and add their initials e.g., “Spring Flowers by DY”.
4. Next, students will begin adding code to their workspace. All the blocks they will need are found under **Actions, Brushes,** and **Loops** in the Blocks section. The program will automatically draw an image using whatever is in their workspace. Students should use trial and error to draw any shapes that they are unsure about.

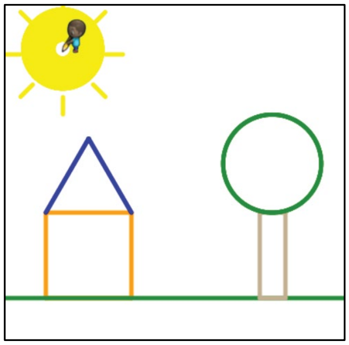
   NOTE: This Code.org lesson has more controls than we used in previous weeks. All the controls students need are found under “Actions”, “Brushes”, and “Loops”. Rewatch our video tutorial if you need assistance.
5. Give students time to work on their drawings. Encourage them to work through problems on their own when they get stuck instead of immediately asking for help.
6. Students may be unable to draw the picture they had originally planned. If that’s the case, encourage them to be flexible and change their plan as needed. Every student should have something to show by the end, even if it isn’t perfect.
7. Leave time at the end of the class for students to save and submit their work (about 15 minutes). Students will:
   - Name their project if they haven’t already done so.
   - Click the “Share” button at the top then “Copy link to project”.
   - Paste the link in an email or message to you. If you have a virtual classroom (e.g., Google Classroom or Brightspace), you could have them share their links there. Have them open a new email or message, right click in the body of the message, and click “Paste”.
8. Ask students to answer the questions at the end of their worksheet then hand it in to you.

FINAL PROJECT

Assessment and evaluation

When students have sent you the link to their work from Code.org, you can view their picture by clicking the link. Their finished picture will open in a new window. You can start your assessment by checking that the students used the following elements in their artwork:

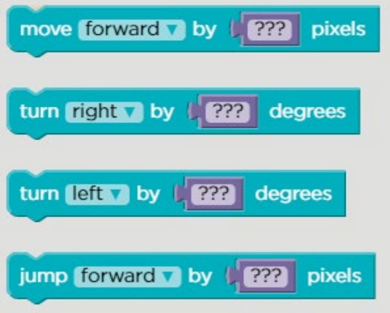
- □ 1 triangle
- □ 1 circle
- □ 1 skinny line
- □ 3 different colours
- □ 1 square
- □ 1 other shape
- □ 1 thick line
- □ 2 shapes that don't touch each other



An example that meets all these criteria

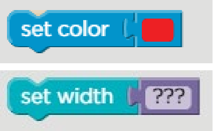
To view the code that created the artwork, click "How It Works". You will now be able to see the student's code in the Workspace. Read through their code to see if they used these blocks:

Actions



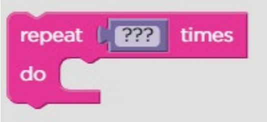
```
move forward ▾ by ??? pixels
turn right ▾ by ??? degrees
turn left ▾ by ??? degrees
jump forward ▾ by ??? pixels
```

Brushes



```
set color 🟥
set width ???
```

Loops



```
repeat ??? times
do
```

Students may have used more blocks than the ones listed above, but these are the ones specified on their worksheet and the ones used in the online lessons.

Continue to the **ASSESSMENT FRAMEWORK** on the next page.

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 required block types in final code Does not answer questions during online classes, even with assistance	Uses some required block types in final code Answers questions during online classes with some assistance	Uses most required block types in final code Answers some questions during online classes	Uses all required block types in final code Answers many questions during online classes
Understanding of content	Rarely uses repeat loops when appropriate to make code more efficient Rarely uses nested loops when appropriate	Sometimes uses repeat loops when appropriate to make code more efficient Sometimes uses nested loops when appropriate	Often uses repeat loops when appropriate to make code more efficient Often uses nested loops when appropriate	Always uses repeat loops when appropriate to make code more efficient Always uses nested loops when appropriate

THINKING

	LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4
Use of planning skills	Creates an initial design with few of the necessary shapes/ elements	Creates an initial design with some of the necessary shapes/ elements	Creates an initial design with most of the necessary shapes/ elements	Creates an initial design with all of the necessary shapes/elements
Use of processing skills	Uses code to convert initial drawing to digital image with limited effectiveness	Uses code to convert initial drawing to digital image with some effectiveness	Uses code to convert initial drawing to digital image with considerable effectiveness	Uses code to convert initial drawing to digital image with high degree of effectiveness
Use of critical/ creative thinking processes	Troubleshoots and "debugs" code with much assistance Does not experiment with blocks	Troubleshoots and "debugs" code with assistance Experiments with required block types	Troubleshoots and "debugs" code with some assistance Experiments with some new block types	Troubleshoots and "debugs" code with little assistance Experiments with several new block types

COMMUNICATION

	LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4
Expression and organization of ideas and information in oral, visual, and/or written forms	Creates digital art that expresses few required elements	Creates digital art that expresses some required elements	Creates digital art that expresses most required elements	Creates digital art that expresses all required elements
Communication for different audiences and purposes in oral, visual, and/or written forms	Explains code and digital art, either orally or in writing, with limited effectiveness	Explains code and digital art, either orally or in writing, with some effectiveness	Explains code and digital art, either orally or in writing, with considerable effectiveness	Explains code and digital art, 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 new coding levels with much assistance	Completes new coding levels with assistance	Completes new coding levels with some assistance	Completes new coding levels with little or no assistance
Application of knowledge and skills to new contexts	Transfers coding knowledge to make original art with much assistance	Transfers coding knowledge to make original art with assistance	Transfers coding knowledge to make original art with some assistance	Transfers coding knowledge to make original art 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

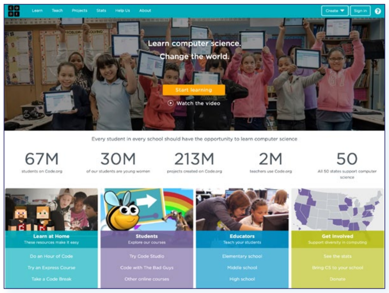
{CODING} IN THE CLASSROOM

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



Learn computer science. Change the world.

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

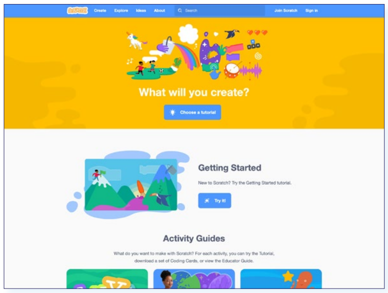
Learn at Home	Students	Educators	Get Involved
These resources make it easy	Explore our courses	Teach your students	Support diversity in computing
Do an Hour of Code	Try Code Studio	Elementary school	See the stats
Try an Express Course	Code with The Bad Guys	Middle school	Bring CS to your school
Take a Code Break	Other online courses	High school	Donate

Code.org

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



What will you create?

Getting Started

New to Scratch? Try the Getting Started tutorial.

Activity Guides

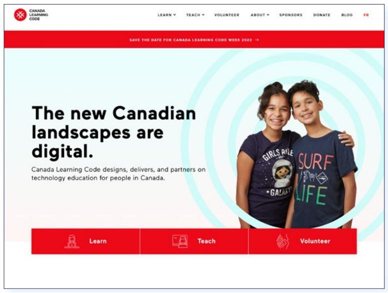
What do you want to make with Scratch? For each activity, you can try the Tutorial, download a set of Coding Cards, or view the Educator Guide.

Scratch

CANADA LEARNING CODE

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

canadalearningcode.ca



SAVE THE DATE FOR CANADA LEARNING CODE WEEK 2022 →

The new Canadian landscapes are digital.

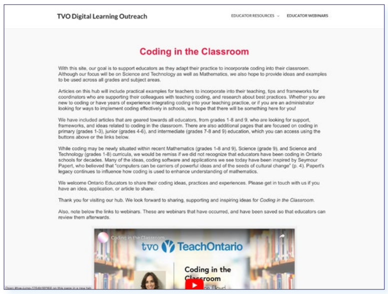
Canada Learning Code designs, delivers, and partners on technology education for people in Canada.

Canada Learning Code

TVO CODING IN THE CLASSROOM

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

outreach.tvolearn.com/codingintheclassroom



TVO Digital Learning Outreach

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

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