

# Pre Express

Pre-Express was developed with early and pre-readers in mind. Tailored to a novice reading level, this course also assumes limited knowledge of shapes and numbers.

Pre-Express closely parallels Courses A and B in a single course. Students will learn the basics of programming, collaboration techniques, investigation and critical thinking skills, persistence in the face of difficulty, and internet safety. At the end of this course students will create their very own custom game from Play Lab that they can share with a link.

## Chapter 1: Sequencing

### Lesson 1: Learn to Drag and Drop

Sequencing

### Lesson 2: Sequencing with Scrat

Sequencing

### Lesson 3: Programming with Angry Birds

Sequencing

### Lesson 4: Programming with Rey and BB-8

Sequencing

### Lesson 5: Programming with Harvester

Sequencing

## Chapter Commentary

Sequencing

## Chapter 2: Loops

### Lesson 6: Loops with Scrat

Loops

### Lesson 7: Loops with Laurel

Loops

## Lesson 8: Ocean Scene with Loops

Loops

## Lesson 9: Drawing Gardens with Loops

Loops

# Chapter Commentary

Loops

## Chapter 3: Events

### Lesson 10: On the Move with Events

Events

### Lesson 11: A Royal Battle with Events

Events

# Chapter Commentary

Events



This curriculum is available under a  
Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes, **contact us**.

# Lesson 1: Learn to Drag and Drop

## Overview

This lesson will give students an idea of what to expect when they head to the computer lab. It begins with a brief discussion introducing them to computer lab manners, then they will progress into using a computer to complete online puzzles.

## Purpose

The main goal of this lesson is to build students' experience with computers. By covering the most basic computer functions such as clicking, dragging, and dropping, we are creating a more equal playing field in the class for future puzzles. This lesson also provides a great opportunity to introduce basic computer hardware terminology, potentially including "mouse", "trackpad" or "touchscreen", depending on your devices.

## Agenda

### Warm Up (10 min)

Teaching this course as a class?  
Behaving in the Computer Lab  
Discuss

### Preview Online Puzzles (5 min)

### Main Activity (20 - 30 min)

Learn to Drag and Drop

### Wrap Up (5 - 10 min)

Journaling

### Extension Activities

### View on Code Studio

## Objectives

Students will be able to:

- Recognize what is expected when students transition into the computer lab.
- Use appropriate terminology when referring to a computer mouse, trackpad, or touchscreen.

## Preparation

- Create your own class section on Code.org and make sure every student has a card with their **passcode** on it.
- Make sure students will be able to access Lesson 2 from their devices. Consider whether you want to hide future lessons to prevent students from moving ahead too quickly.
- Have the school IT person add a quick link for your class section to the computer desktop.
- Review the Common Sense Education website for more online safety content.

## Links

**Heads Up!** Please make a copy of any documents you plan to share with students.

### For the Teachers

- **How to Make a Class Section on Code.org** - Teacher Video

### For the Students

- **Think Spot Journal** - Reflection Journal  
[Make a Copy](#)
- **Pair Programming** - Student Video
- **Feeling Faces** - Emotion Images  
[Make a Copy](#)

# Teaching Guide

## Warm Up (10 min)

### Behaving in the Computer Lab

🎓 Goal: This discussion will teach students what to expect and how to behave when they enter the computer lab.

### Discuss

Have a good discussion around the computer lab expectations to make sure that students understand the rules. Some topics of discussion might include:

- Is running in the computer lab okay?
- How loudly should we walk when we are in the computer lab?
- What should you do if you get stuck on a puzzle?
- If you get frustrated, will it help to hit the computer?
- When we're about to go to the computer lab, how should we get ready?

## Preview Online Puzzles (5 min)

Project a puzzle from lesson 1. Show the class how to click on the picture and place it in the correct spot by dragging and dropping. Purposely make mistakes such as clicking the background or dropping the image before it's at the right spot. Ask for help from volunteers in the class when you run into these problems.

## Main Activity (20 - 30 min)

### Learn to Drag and Drop

Goal: This will teach students how to use Code.org to complete online puzzles.

This stage was designed to give students the opportunity to practice hand-eye coordination, clicking, and drag & drop skills. Students will also play with sequence.

🎓 Take some time to explicitly teach how to click, drag, and drop. Take time to introduce the language around the devices students will be using when they work on the puzzles. If you have tablets, students will be using a *touch screen*. If you have laptops, they will likely be using a *trackpad*. Desktop computers like you might find in a lab will rely on the use of the *mouse*.

Place kids in pairs and have them watch the pair programming video as a class or at their stations. This should help students start off in the right direction.

### Content Corner

### Teaching this course as a class?

Our grade-aligned CS Fundamentals courses use unplugged lessons to build community and introduce tricky computer science concepts, including **digital citizenship**. Check out the lesson **Going Places Safely** from **Course A!**

### Discussion Goals:

- Use calm bodies in the lab
- Remember not to chew gum or candy
- Sanitize your hands
- Sit with your partner at one computer
- Make sure that the first "driver" can reach the mouse
- When you get frustrated, don't hit or shake the computer or monitor
- Follow the **20/20/20 - Website** rule
- How to deal with the **Wiggles** every 20-30 minutes (requires a free login on GoNoodle)
- Ask your partner before you ask the teacher
- Keep volume down so everyone else can hear their partners
- Use your journal for keeping track of feelings and solutions

🔗 Teachers play a vital role in computer science education and supporting a collaborative and vibrant classroom environment. During online puzzles, the role of the teacher is primarily one of encouragement and support. Online puzzles are meant to be student-centered, so teachers should avoid stepping in when students get stuck. Some ideas on how to do this are:

- Utilize pair programming whenever possible during the activity.
- Encourage students with questions/challenges to start by asking their partner.
- Unanswered questions can be escalated to a nearby group, who might already know the solution.
- Remind students to use the debugging process before you approach.
- Have students describe the problem that they're seeing. What is it supposed to do? What does it do? What does that tell you?
- Remind frustrated students that frustration is a step on the path to learning, and that persistence will pay off.
- If a student is still stuck after all of this, ask leading questions to get the student to spot an error on their own.

### 📖 Content Corner

Considering having students break down the steps of dragging and dropping and record them on the board. For example:

1. Move the arrow to the block.
2. Click and hold the mouse button.
3. Move the mouse.
4. Let go of the button.

Doing this will give students practice with creating an *algorithm*, which is a concept that will be explored in upcoming lessons.

### 💡 Teacher Tip

Show the students the right way to help classmates:

- Don't sit in the classmate's chair
- Don't use the classmate's keyboard
- Don't touch the classmate's mouse
- Make sure the classmate can describe the solution to you out loud before you walk away

## Wrap Up (5 - 10 min)

### Journaling

Goal: Help students reflect on the things they learned in this lesson

Give the students a journal prompt to help them process some of the things that they encountered during the day.

Journal prompts could include:

- Draw one of the feeling faces that shows how you felt about today's lesson in the corner of your journal page.
- Can you draw a sequence for getting ready to go to the computer lab?
- Draw a computer lab "Do" and a "Don't"
  - Draw and label the name of the computer part you used when clicking and dragging during the puzzles. (mouse button, touch screen, trackpad)

## Extension Activities

If students complete the puzzles from Stage 4 early, have them spend some time trying to come up with their own puzzles in their **Think Spot Journal - Reflection Journal**.

## Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **AP** - Algorithms & Programming
- ▶ **CS** - Computing Systems



This curriculum is available under a  
Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes, **contact us**.

# Lesson 2: Sequencing with Scrat

## Overview

Using Scrat from the Ice Age franchise, students will develop sequential algorithms to move a squirrel character from one side of a maze to the acorn at the other side. To do this they will stack code blocks together in a linear sequence.

## Purpose

In this lesson, students will develop programming and debugging skills on a computer platform. The block-based format of these puzzles help students learn about sequence and concepts, without having to worry about perfecting syntax.

## Agenda

### Warm Up (10 min)

Behaving in the Computer Lab

### Previewing Online Puzzles as a Class (10 - 15 min)

Teaching this course as a class?

### Main Activity (20 - 30 min)

Online Puzzles

### Wrap Up (5 - 10 min)

Journaling

### Extension Activities

### View on Code Studio

## Objectives

Students will be able to:

- Model proper computer lab behaviors
- Experiment with standard block-based programming actions such as: clicking, drag and drop, etc.

## Preparation

- Watch the **How to Make a Class Section on Code.org - Teacher Video**. Create a class section and make sure every student has a card with their passcode on it
- Have the school IT person add a quick link for your class section to the computer desktop
- Make sure each student has a **Think Spot Journal - Reflection Journal**

## Links

**Heads Up!** Please make a copy of any documents you plan to share with students.

### For the Students

- **Feeling Faces** - Emotion Images  
[Make a Copy](#)
- **Pair Programming** - Student Video
- **Think Spot Journal** - Reflection Journal  
[Make a Copy](#)

## Vocabulary

- **Click** - Press the mouse button
- **Double-Click** - Press the mouse button very quickly
- **Drag** - Click your mouse button and hold as you move the mouse pointer to a new location
- **Drop** - Release your mouse button to "let go" of an item that you are dragging

# Teaching Guide

## Warm Up (10 min)

### Behaving in the Computer Lab

Review expectations and how to behave when they enter the computer lab.

#### Discuss:

Have a good discussion around your computer lab expectations to make sure that students understand the rules. Some topics of discussion might include:

- Is running in the computer lab okay?
- How loudly should we walk when we are in the computer lab?
- What should you do if you get stuck on a puzzle?
- If you get frustrated, will it help to hit the computer?
- When we're about to go to the computer lab, how should we get ready?

## Previewing Online Puzzles as a Class (10 - 15 min)

🎓 Project a puzzle from the lesson. Show the class how to click on a block and place it in the correct spot by dragging and dropping. Purposely make mistakes such as clicking the background or dropping the image before it's at the right spot. Ask for help from volunteers in the class when you run into these problems, and help them use the skills that they developed in the last unplugged lesson to make things right.

## Main Activity (20 - 30 min)

### Online Puzzles

This will teach students how to use Code.org to complete online puzzles.

This stage was designed to give students the opportunity to practice hand-eye coordination, clicking, and drag & drop skills. Students will also play with sequence.

The vocabulary introduced in this lesson becomes relevant during this activity. Take some time to explicitly teach how to click, double-click, drag, and drop. It might work better for you to cover these

#### 💡 Some possible things to cover:

- Use calm bodies in the lab
- Remember not to chew gum or candy
- Sanitize your hands
- Sit with your partner at one computer
- Make sure that the first "driver" can reach the mouse
- When you get frustrated, don't hit or shake the computer or monitor
- Follow the **20/20/20 - Website** rule
- How to deal with the **Wiggles** every 20-30 minutes (requires a free login on GoNoodle)
- Ask your partner before you ask the teacher
- Keep volume down so everyone else can hear their partners
- Use your journal for keeping track of feelings and solutions

#### 🎓 Content Corner

### Teaching this course as a class?

Our grade-aligned CS Fundamentals courses use unplugged lessons to build community and introduce tricky computer science concepts, including **sequencing**. Check out the lesson **Happy Maps** from **Course A** or **Move It, Move It!** from **Course B!**

#### 💡 Teacher Tip

Show the students the right way to help classmates:

- Don't sit in the classmate's chair
- Don't use the classmate's keyboard
- Don't touch the classmate's mouse
- Make sure the classmate can describe the solution to you out loud before you walk away

words in the classroom environment where you can lead by example -- or it might make more sense to teach the words individually as students work on their puzzles in the lab. You will need to decide what you believe is best for your class.

Watch the **Pair Programming - Student Video** with your students, then assign them to pairs. This should help students start off in the right direction.

Teachers play a vital role in computer science education and supporting a collaborative and vibrant classroom environment. During online activities, the role of the teacher is primarily one of encouragement and support. Online lessons are meant to be student-centered, so teachers should avoid stepping in when students get stuck. Some ideas on how to do this are:

- Utilize pair programming whenever possible during the activity.
- Encourage students with questions/challenges to start by asking their partner.
- Unanswered questions can be escalated to a nearby group, who might already know the solution.
- Remind students to use the debugging process before you approach.
- Have students describe the problem that they're seeing. What is it supposed to do? What does it do? What does that tell you?
- Remind frustrated students that frustration is a step on the path to learning, and that persistence will pay off.
- If a student is still stuck after all of this, ask leading questions to get the student to spot an error on their own.

## Wrap Up (5 - 10 min)

### Journaling

Give the students a journal prompt to help them process some of the things that they encountered during the day.

**Journal Prompts:**

- Can you draw a sequence for getting ready to go to the computer lab?
- Draw a computer lab "Do" and a "Don't"
- Draw one of the **Feeling Faces - Emotion Images** that shows how you felt about today's lesson in the corner of your journal page.

## Extension Activities

If students complete the puzzles from this lesson early, have them spend some time trying to come up with their own puzzles in their **Think Spot Journal - Reflection Journal**.

## Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **AP** - Algorithms & Programming



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes, **contact us**.

# Lesson 3: Programming with Angry Birds

## Overview

Using characters from the game Angry Birds, students will develop sequential algorithms to move a bird from one side of a maze to the pig at the other side. To do this they will stack code blocks together in a linear sequence.

## Purpose

In this lesson, students will develop programming and debugging skills on a computer platform. The block-based format of these puzzles help students learn about sequence and concepts, without having to worry about perfecting syntax.

## Agenda

**Warm Up: Preview Programming in Maze (3 min)**

**Main Activity (30 min)**

Online Puzzles

**Wrap Up (5 - 10 min)**

Journaling

**Extended Learning**

[View on Code Studio](#)

## Objectives

Students will be able to:

- Translate movements into a series of commands.
- Construct a program by reorganizing sequential movements
- Build a computer program from a set of written instructions

## Preparation

- Play through the lesson to find any potential problem areas for your class.
- (Optional) Pick a couple of puzzles to do as a group with your class.
- Make sure each student has a journal.

## Vocabulary

- **Algorithm** - A list of steps to finish a task.
- **Bug** - Part of a program that does not work correctly.
- **Debugging** - Finding and fixing problems in an algorithm or program.
- **Program** - An algorithm that has been coded into something that can be run by a machine.
- **Programming** - The art of creating a program.

# Teaching Guide

## Warm Up: Preview Programming in Maze (3 min)

To finish the connection, preview an online puzzle (or two) as a class.

**Model:** Reveal an entire online puzzle from the progression to come. We recommend Puzzle 5. Do they see any similarities to the exercise that they just did? What are the big differences?

Work with your class to drag code into the workspace in such a way that the bird (eventually) gets to the pig.

**Transition:** Students should now be ready to transition to computers to complete online puzzles on their own.

## Main Activity (30 min)

### Online Puzzles

**Circulate:** Teachers play a vital role in computer science education and supporting a collaborative and vibrant classroom environment. During online activities, the role of the teacher is primarily one of encouragement and support. Online lessons are meant to be student-centered, so teachers should avoid stepping in when students get stuck. Some ideas on how to do this are:

- Utilize **Pair Programming - Student Video** whenever possible
- Encourage students with questions/challenges to start by asking their partner
- Unanswered questions can be escalated to a nearby group, who might already know the solution
- Remind students to use the debugging process before you approach
- Have students describe the problem that they're seeing. What is it supposed to do? What does it do? What does that tell you?
- Remind frustrated students that frustration is a step on the path to learning, and that persistence will pay off
- If a student is still stuck after all of this, ask leading questions to get the student to spot an error on their own

#### Teacher Tip:

Show the students the *right* way to help classmates by:

- Don't sit in the classmate's chair
- Don't use the classmate's keyboard
- Don't touch the classmate's mouse
- Make sure the classmate can describe the solution to you out loud before you walk away

## Wrap Up (5 - 10 min)

### Journaling

Having students write about what they learned, why it's useful, and how they feel about it can help solidify any knowledge they obtained today and build a review sheet for them to look to in the future.

**Journal Prompts:**

- What was today's lesson about?
- Draw one of the **Feeling Faces - Emotion Images** that shows how you felt about today's lesson in the corner of your journal page.

## Extended Learning

In small groups, let students design their own mazes on paper and challenge other students or groups to write programs to solve them. For added fun, make life-size mazes with students as the pig and bird.

# Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▶ **AP** - Algorithms & Programming



This curriculum is available under a  
Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes, **contact us**.

# Lesson 4: Programming with Rey and BB-8

## Overview

In this lesson, students will use their newfound programming skills in more complicated ways to navigate a tricky course with BB-8.

## Purpose

With transfer of knowledge in mind, this lesson gives students a new environment to practice the skills that they have been cultivating. Star Wars fans will jump for joy when they see these puzzles. Each puzzle in this series has been added to provide a deeper understanding of the basic concepts that they will be using throughout the rest of this course.

## Agenda

### Warm Up (15 min)

Introduction

### Main Activity (30 min)

Online Puzzles

### Wrap Up (15 min)

Journaling

### View on Code Studio

## Objectives

Students will be able to:

- Sequence commands in a logical order.
- Recognize problems or "bugs" in a program and develop a plan to resolve the issues.

## Preparation

- Play through the lesson to find any potential problem areas for your class.
- Make sure every student has a journal.

## Links

**Heads Up!** Please make a copy of any documents you plan to share with students.

### For the Students

- **Feeling Faces** - Emotion Images  
[Make a Copy](#)
- **Think Spot Journal** - Reflection Journal  
[Make a Copy](#)

## Vocabulary

- **Algorithm** - A list of steps to finish a task.
- **Bug** - Part of a program that does not work correctly.
- **Debugging** - Finding and fixing problems in an algorithm or program.
- **Program** - An algorithm that has been coded into something that can be run by a machine.
- **Programming** - The art of creating a program.

# Teaching Guide

## Warm Up (15 min)

### Introduction

Ask the students how they felt about the last lesson.

- Which puzzles were too hard or too easy?
- Which puzzles were frustrating or a lot of fun?
- If they were to teach the lesson to a friend, which part of the lesson would they want to review?

Use these questions to form a brief review of programming and debugging. If you think the class could benefit from it, you can go over the vocabulary words and definitions from the last lesson.

If you feel comfortable, also give a brief introduction to BB-8 from Star Wars. Many students may already be familiar with the lovable robot, but the introduction will surely build excitement.

## Main Activity (30 min)

### Online Puzzles

As we mentioned in the last lesson, we highly recommend viewing and using **Pair Programming - Student Video** as a class. Pair programming stimulates a discussion that can answer questions, review basic concepts, and build confidence with the subject.

## Wrap Up (15 min)

### Journaling

Having students write about what they learned, why it's useful, and how they feel about it can help solidify any knowledge they obtained today and build a review sheet for them to look to in the future.

**Journal Prompts:**

- What was today's lesson about?
- Draw one of the **Feeling Faces - Emotion Images** that shows how you felt about today's lesson in the corner of your journal page.
- Draw a picture of BB-8 you guided through the maze today and add a list of the commands that you used.

## Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ AP - Algorithms & Programming



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes, **contact us**.

# Lesson 5: Programming with Harvester

## Overview

Students will apply the programming concepts that they have learned to the Harvester environment. Now, instead of just getting the character to a goal, students have to collect corn using a new block. Students will continue to develop sequential algorithm skills and start using the debugging process.

## Purpose

In this lesson, students will develop debugging skills and will continue developing their programming skills.

## Agenda

### Main Activity (30 min)

Online Puzzles

### Wrap Up (5 - 10 min)

Journaling

### View on Code Studio

## Objectives

Students will be able to:

- Translate movements into a series of commands.
- Identify and locate bugs in a program.

## Preparation

- Play through the puzzles to find any potential problem areas for your class
- Locate or reprint supplies for Happy Maps
- Make sure each student has a **Think Spot Journal - Reflection Journal**

## Links

**Heads Up!** Please make a copy of any documents you plan to share with students.

### For the Students

- **Feeling Faces** - Emotion Images  
[Make a Copy](#)
- **Think Spot Journal** - Reflection Journal  
[Make a Copy](#)

## Vocabulary

- **Algorithm** - A list of steps to finish a task.
- **Bug** - Part of a program that does not work correctly.
- **Debugging** - Finding and fixing problems in an algorithm or program.
- **Program** - An algorithm that has been coded into something that can be run by a machine.
- **Programming** - The art of creating a program.

# Teaching Guide

## Main Activity (30 min)

### Online Puzzles

At this point, students should already be familiar with the programming environment. Some new things to look out for in this lesson are confusion about the debugging process or not remembering to use the `pick corn` block when the harvester reaches corn.

**Circulate:** During online activities, the role of the teacher is primarily one of encouragement and support. In addition to the ideas listed in the last lesson, some more ideas on how to do this are:

- Remind students to use the debugging process before you approach.
- Have students describe the problem that they're seeing. What is it supposed to do? What does it do? What does that tell you?
- Remind frustrated students that frustration is a step on the path to learning, and that persistence will pay off.
- If a student is still stuck after all of this, ask leading questions to get the student to spot an error on their own.

**Transition:** Have students grab their Thinkspot Journals and take a moment to leave lessons for themselves.

## Wrap Up (5 - 10 min)

### Journaling

Having students write about what they learned, why it's useful, and how they feel about it can help solidify any knowledge they obtained today and build a review sheet for them to look to in the future.

Journal Prompts:

- What was today's lesson about?
- Draw one of the **Feeling Faces - Emotion Images** that shows how you felt about today's lesson in the corner of your journal page.
- Draw a time you found a bug in your code.

## Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **AP** - Algorithms & Programming

### Teacher Tip:

Show the students the *right* way to help classmates by:

- Don't sit in their chair
- Don't use their keyboard
- Don't touch their mouse
- Make sure the classmate can describe the solution before you walk away



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes, **contact us**.

# Lesson 6: Loops with Scrat

## Overview

Building on the concept of repeating instructions from "My Loopy Robotic Friends," this stage will have students using loops to get to the acorn more efficiently on Code.org.

## Purpose

In this lesson, students will be learning more about loops and how to implement them in Blockly code. Using *loops* is an important skill in programming because manually repeating commands is tedious and inefficient. With these Code.org puzzles, students will learn to add instructions to existing loops, gather repeated code into loops, and recognize patterns that need to be repeated.

## Agenda

### Online Foundation: Preview Loops in Ice Age

Teaching this course as a class?

### Main Activity (30 min)

Online Puzzles

### Wrap Up (5 - 10 min)

Journaling

### Extended Learning

### View on Code Studio

## Objectives

Students will be able to:

- Construct a program using structures that repeat areas of code
- Improve existing code by finding areas of repetition and moving them into looping structures

## Preparation

- (Optional) Pick a couple of puzzles to do as a group with your class.
- Make sure each student has a journal.

## Links

**Heads Up!** Please make a copy of any documents you plan to share with students.

### For the Students

- **Feeling Faces** - Emotion Images  
[Make a Copy](#)
- **Think Spot Journal** - Reflection Journal  
[Make a Copy](#)

## Vocabulary

- **Loop** - The action of doing something over and over again.
- **Repeat** - To do something again.

# Teaching Guide

## Online Foundation: Preview Loops in Ice Age

To finish the connection, preview an online puzzle (or two) as a class.

**Model:** Reveal an entire online puzzle from the progression to come. Work with your class to drag code into the workspace in such a way that Scrat (eventually) gets to the acorn. Make sure to highlight the use of repeat blocks.

**Transition:** Students should now be ready to transition to computers to complete online puzzles on their own.

### Main Activity (30 min)

As students work through the puzzles, see if they can figure out how many blocks they use with a loop vs. without a loop.

### Online Puzzles

**Circulate:** Teachers play a vital role in computer science education and supporting a collaborative and vibrant classroom environment. During online activities, the role of the teacher is primarily one of encouragement and support. Online lessons are meant to be student-centered, so teachers should avoid stepping in when students get stuck. Some ideas on how to do this are:

- Utilize **Pair Programming - Student Video** whenever possible
- Encourage students with questions/challenges to start by asking their partner
- Unanswered questions can be escalated to a nearby group, who might already know the solution
- Remind students to use the debugging process before you approach
- Have students describe the problem that they're seeing. What is it supposed to do? What does it do? What does that tell you?
- Remind frustrated students that frustration is a step on the path to learning, and that persistence will pay off.
- If a student is still stuck after all of this, ask leading questions to get the student to spot an error on their own.

### Wrap Up (5 - 10 min)

#### Journaling

Having students write about what they learned, why it's useful, and how they feel about it can help solidify any knowledge they obtained today and build a review sheet for them to look to in the future.

Journal Prompts:

- What was today's lesson about?
- Draw one of the **Feeling Faces - Emotion Images** that shows how you felt about today's lesson in the corner of your journal page.
- Draw Scrat and an acorn.

#### Content Corner

### Teaching this course as a class?

Our grade-aligned CS Fundamentals courses use unplugged lessons to build community and introduce tricky computer science concepts, including **loops**. Check out the lesson **Happy Loops** from **Course A** and **Getting Loopy** from **Course B**!

#### Teacher Tip:

Show the students the *right* way to help classmates by:

- Don't sit in the classmate's chair
- Don't use the classmate's keyboard
- Don't touch the classmate's mouse
- Make sure the classmate can describe the solution to you out loud before you walk away

- Draw yourself using a loop to do an everyday activity, like brushing your teeth.

## Extended Learning

### So Moving

- Give the students pictures of actions or dance moves that they can do.
  - Have students arrange moves and add loops to choreograph their own dance.
- Share the dances with the rest of the class.

### Connect It Back

- Find some YouTube videos of popular dances that repeat themselves.
- Can your class find the loops?
- Try the same thing with songs!

## Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **AP** - Algorithms & Programming



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes, **contact us**.

# Lesson 7: Loops with Laurel

## Overview

In this lesson, students continue learning the concept of loops. Here, Laurel the Adventurer uses loops to collect treasure in open cave spaces. A new `get_treasure` block is introduced to help her on her journey.

## Purpose

This lesson gives students more practice with loops and encourages them to put multiple blocks inside of a `repeat` as they try to collect as much treasure as possible.

## Agenda

### Warm Up (10 min)

Introduction

### Online Foundation: Preview Loops in Collector

### Main Activity (30 min)

Teacher Demonstration

Online Puzzles

### Wrap Up (5 - 10 min)

Journaling

### View on Code Studio

## Objectives

Students will be able to:

- Identify the benefits of using a loop structure instead of manual repetition.
- Break down a long sequence of instructions into the smallest repeatable sequence possible.

## Preparation

- Play through the puzzles to find any potential problem areas for your class.
- (Optional) Pick a couple of puzzles to do as a group with your class.
- Make sure each student has a journal.

## Links

**Heads Up!** Please make a copy of any documents you plan to share with students.

### For the Students

- **Feeling Faces** - Emotion Images  
[Make a Copy](#)
- **Think Spot Journal** - Reflection Journal  
[Make a Copy](#)

## Vocabulary

- **Loop** - The action of doing something over and over again.
- **Repeat** - To do something again.

# Teaching Guide

## Warm Up (10 min)

### Introduction

Quickly review the definition of a loop, the action of doing something over and over again.

- What are loops?
- Why do we use them?

## Online Foundation: Preview Loops in Collector

To introduce Laurel the Collector, preview an online puzzle (or two) as a class.

**Model:** Reveal an entire online puzzle from the progression to come. We recommend Lesson 9, Puzzle 8. Do students see any similarities to the last set of exercises that they did? What are the big differences? When should the `get_treasure_block` be used?

Work with your class to drag code into the workspace in such a way that Laurel (eventually) collects all of the treasure.

**Transition:** Students should now be ready to transition to computers to complete online puzzles on their own.

## Main Activity (30 min)

### Teacher Demonstration

We've included some multiple choice prediction levels that are difficult for non-readers. These levels are optional for you to review with your class to help prepare for the puzzles to come. Alternatively, these could be used after finishing the stage as a review for the class.

Prediction Levels:

- **Loops in Collector**

### Online Puzzles

As students work through the puzzles, see if they can figure out how many blocks they use with a loop vs. without a loop.

## Wrap Up (5 - 10 min)

### Journaling

Having students write about what they learned, why it's useful, and how they feel about it can help solidify any knowledge they obtained today and build a review sheet for them to look to in the future.

Journal Prompts:

- What was today's lesson about?
- Draw one of the **Feeling Faces - Emotion Images** that shows how you felt about today's lesson in the corner of your journal page.
- Draw a line of treasure that Laurel could collect.
- Draw something that uses loops.

# Standards Alignment

CSTA K-12 Computer Science Standards (2017)

▶ **AP** - Algorithms & Programming



This curriculum is available under a  
Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes, **contact us**.

# Lesson 8: Ocean Scene with Loops

## Overview

Returning to loops, students learn to draw images by looping simple sequences of instructions. In the previous plugged lesson, loops were used to traverse a maze and collect treasure. Here, loops are creating patterns. At the end of this stage, students will be given the opportunity to create their own images using loops.

## Purpose

This lesson gives a different perspective on how loops can create things in programming. Students can also reflect on the inefficiency of programming without loops here because of how many blocks the program would require without the help of repeat loops.

## Agenda

### Warm Up (10 min)

Introduction

### Main Activity (30 min)

Online Puzzles

### Wrap Up (5 - 10 min)

Journaling

### View on Code Studio

## Objectives

Students will be able to:

- Count the number of times an action should be repeated and represent it as a loop.
- Decompose a shape into its largest repeatable sequence.
- Create a program that draws complex shapes by repeating simple sequences.

## Preparation

- Play through the puzzles to find any potential problem areas for your class.
- Make sure each student has a **Think Spot Journal - Reflection Journal**.

## Links

**Heads Up!** Please make a copy of any documents you plan to share with students.

### For the Teachers

- **Pause and Think Online** - Video

### For the Students

- **Feeling Faces** - Emotion Images  
[Make a Copy](#)
- **Think Spot Journal** - Reflection Journal  
[Make a Copy](#)

## Vocabulary

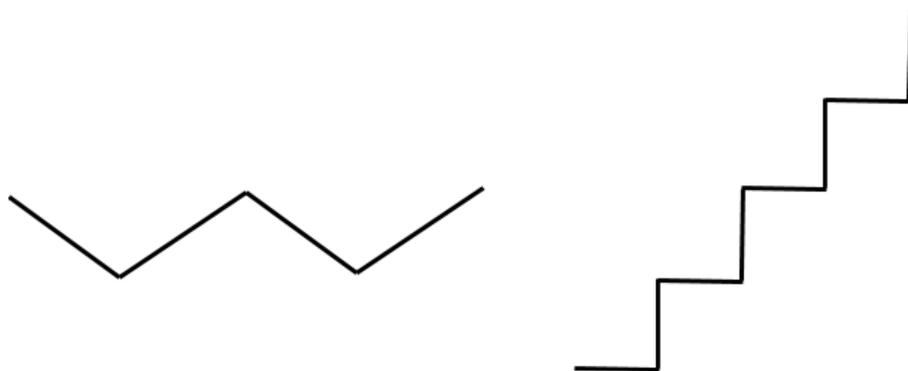
- **Loop** - The action of doing something over and over again.
- **Repeat** - To do something again.

# Teaching Guide

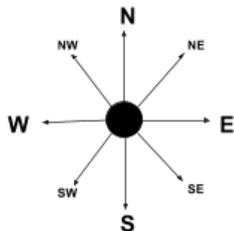
## Warm Up (10 min)

### Introduction

- Quickly review the definition of a loop, the action of doing something over and over again.
- Discuss different patterns like zigzags and stairsteps.
  - How would you explain to someone how to draw that pattern?
  - How could you draw this using a loop?



In the artist levels students will be using 45 degree angles described as northwest, northeast, southwest, southeast. We recommend briefly discussing these directions with the class and drawing an image for students to refer back to.



## Main Activity (30 min)

### Online Puzzles

#### Teacher Demonstration

We've included some multiple choice prediction levels that are difficult for non-readers. These levels are optional for you to review with your class to help prepare for the puzzles to come. Alternatively, these could be used after finishing the stage as a review for the class.

Prediction Levels:

- **Loops in Artist**

#### Teacher Tip

Remind the students to only share their work with their close friends or family. For more information watch or show the class **Pause and Think Online - Video**.

## Wrap Up (5 - 10 min)

### Journaling

Having students write about what they learned, why it's useful, and how they feel about it can help solidify any knowledge they obtained today and build a review sheet for them to look to in the future.

### Journal Prompts:

- What was today's lesson about?
- Draw one of the **Feeling Faces - Emotion Images** that shows how you felt about today's lesson in the corner of your journal page.
- Draw some stairs. Imagine the loop needed to draw this.
- Draw something else in your life that uses loops.

## Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **AP** - Algorithms & Programming



This curriculum is available under a  
Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes, **contact us**.

# Lesson 9: Drawing Gardens with Loops

## Overview

Returning to loops, students learn to draw images by looping simple sequences of instructions. In the previous online lesson, loops were used to traverse a maze and collect treasure. Here, students use loops to create patterns. At the end of this stage, students will be given the opportunity to create their own images using loops.

## Purpose

This lesson gives a different perspective on how loops can create things in programming. Students will test their critical thinking skills by evaluating given code and determining what needs to be added in order to solve the puzzle. Students can also reflect on the inefficiency of programming without loops here because of how many blocks the program would require without the help of repeat loops.

## Agenda

### Warm Up (10 min)

Introduction

### Main Activity (30 min)

Online Puzzles

### Wrap Up (5 - 10 min)

Journaling

### View on Code Studio

## Objectives

Students will be able to:

- Count the number of times an action should be repeated and represent it as a loop.
- Decompose a shape into its largest repeatable sequence.
- Create a program that draws complex shapes by repeating simple sequences.

## Preparation

- Play through the lesson to find any potential problem areas for your class.
- Make sure each student has a journal.

## Links

**Heads Up!** Please make a copy of any documents you plan to share with students.

### For the Students

- **Feeling Faces** - Emotion Images  
[Make a Copy](#)
- **Think Spot Journal** - Reflection Journal  
[Make a Copy](#)

## Vocabulary

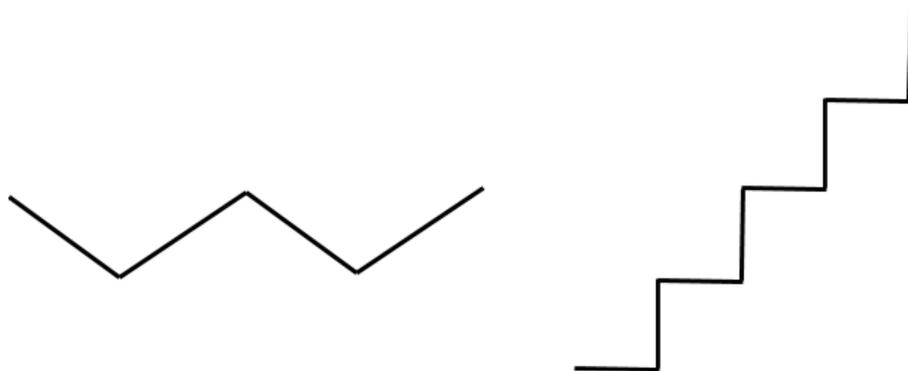
- **Loop** - The action of doing something over and over again.

# Teaching Guide

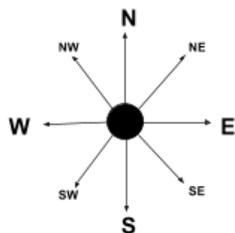
## Warm Up (10 min)

### Introduction

- Quickly review the definition of a loop, the action of doing something over and over again.
- Discuss different patterns like zigzags and stairsteps.
  - How would you explain to someone how to draw that pattern?
  - How could you draw this using a loop?



In the artist levels, students will be using 45 degree angles described as northwest, northeast, southwest, southeast. We recommend briefly discussing these directions with the class and drawing an image for students to refer back to.



## Main Activity (30 min)

### Online Puzzles

#### Teacher Demonstration

We've included some multiple choice prediction levels that are difficult for non-readers. These levels are optional for you to review with your class to help prepare for the puzzles to come. Alternatively, these could be used after finishing the stage as a review for the class.

Prediction Levels:

- **Loops in Artist**

#### Teacher Tip

Remind the students to only share their work with their close friends or family. For more information watch or show the class **Pause and Think Online - Video**.

## Wrap Up (5 - 10 min)

### Journaling

Having students write about what they learned, why it's useful, and how they feel about it can help solidify any knowledge they obtained today and build a review sheet for them to look to in the future.

Journal Prompts:

- What was today's lesson about?
- Draw one of the **Feeling Faces - Emotion Images** that shows how you felt about today's lesson in the corner of your journal page.
- Draw the patterns you made with a loop.
- Draw a pattern that you would like to make with a loop.

## Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **AP** - Algorithms & Programming



This curriculum is available under a  
Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes, **contact us**.

# Lesson 10: On the Move with Events

## Overview

In this online activity, students will have the opportunity to learn how to use events in Play Lab and to apply all of the coding skills they've learned to create an animated game. It's time to get creative and make a story in the Play Lab!

## Purpose

Students will further develop their understanding of events using Play Lab today. Events are very common in most computer programs. In this activity, students will use events to make a character move around the screen, make noises, and change backgrounds based on user-initiated events.

## Agenda

### Warm Up (10 min)

Teaching this course as a class?

Introduction

### Bridging Activity - Events (10 min)

Previewing Online Puzzles as a Class

### Main Activity (30 min)

Online Puzzles and Free Play

### Wrap Up (5 - 10 min)

Journaling

### Extended Learning

### View on Code Studio

## Objectives

Students will be able to:

- Identify actions that correlate to input events.
- Create an animated, interactive story using sequence and event-handlers.
- Share a creative artifact with other students.

## Preparation

- Play through the puzzles to find any potential problem areas for your class.
- (Optional) Pick a couple of puzzles to do as a group with your class.
- Review **CS Fundamentals Main Activity Tips - Lesson Recommendations**.
- Make sure every student has a **Think Spot Journal - Reflection Journal**.

## Links

**Heads Up!** Please make a copy of any documents you plan to share with students.

### For the Students

- **Feeling Faces** - Emotion Images  
[Make a Copy](#)
- **Think Spot Journal** - Reflection Journal  
[Make a Copy](#)

## Vocabulary

- **Event** - An action that causes something to happen.

# Teaching Guide

## Warm Up (10 min)

### Introduction

🎓 Now we're going to add events to our code. Specifically, we're going to have an event for when two characters touch each other.

- When have you seen two characters touch each other as an event in games?

## Bridging Activity - Events (10 min)

### Previewing Online Puzzles as a Class

Pull a puzzle from the corresponding puzzles online. We recommend puzzle 7. Call on different students to make a funny noise when you click on Jorge. Explain this is an event that they are reacting to and Jorge can be coded to make noise when you click on him.

## Main Activity (30 min)

### Online Puzzles and Free Play

At the final level students have the freedom to create a story of their own. You may want to provide structured guidelines around what kind of story to write, particularly for students who are overwhelmed by too many options.

## Wrap Up (5 - 10 min)

### Journaling

Having students write about what they learned, why it's useful, and how they feel about it can help solidify any knowledge they obtained today and build a review sheet for them to look to in the future.

Journal Prompts:

- What was today's lesson about?
- Draw one of the **Feeling Faces - Emotion Images** that shows how you felt about today's lesson in the corner of your journal page.
- Draw an event you used in your program today.
- Imagine that you have a remote controlled robot. What would the remote look like? Draw a picture of what you think you could make the robot do.

## Extended Learning

Use these activities to enhance student learning. They can be used as outside of class activities or other enrichment.

### Look Under the Hood

When you share a link to your story, you also share all of the code that goes behind it. This is a great way for students to learn from each other.

### Content Corner

#### Teaching this course as a class?

Our grade-aligned CS Fundamentals courses use unplugged lessons to build community and introduce tricky computer science concepts, including **events**. Check out the lesson **The Big Event Jr.** from **Course A!**

- Post links to completed stories online.
  - Make a story of your own to share as well!
- When students load up a link, have them click the "How it Works" button to see the code behind the story.
- Discuss as a group the different ways your classmates coded their stories.
  - What surprised you?
  - What would you like to try?
- Choose someone else's story and click `Remix` to build on it. (Don't worry, the original story will be safe.)

## Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **AP** - Algorithms & Programming



This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes, **contact us**.

# Lesson 11: A Royal Battle with Events

## Overview

In this online activity, students will have the opportunity to learn how to use events in Play Lab and apply all of the coding skills that they've learned to create an animated game. It's time to get creative and make a game in Play Lab!

## Purpose

In this online activity, students will learn how to use events in Play Lab. They will start by training the knight to move when an arrow key is pressed, then end with the opportunity to showcase the rest of the skills that they learned throughout this course, including sequence and looping, as part of the final freeplay puzzle.

## Agenda

### Warm Up (10 min)

Introduction

### Bridging Activity - Events (10 min)

Previewing Online Puzzles as a Class

### Main Activity (30 min)

Online Puzzles and Free Play

### Wrap Up (5 - 10 min)

Journaling

### Extended Learning

### View on Code Studio

## Objectives

Students will be able to:

- Identify actions that correlate to input events.
- Create an animated, interactive story using sequences and event-handlers.
- Share a creative artifact with other students.

## Preparation

- Play through the puzzles to find any potential problem areas for your class.
- (Optional) Pick a couple of puzzles to do as a group with your class.
- Make sure every student has a journal.

## Links

**Heads Up!** Please make a copy of any documents you plan to share with students.

### For the Students

- **Feeling Faces** - Emotion Images  
[Make a Copy](#)
- **Think Spot Journal** - Reflection Journal  
[Make a Copy](#)

## Vocabulary

- **Event** - An action that causes something to happen.

# Teaching Guide

## Warm Up (10 min)

### Introduction

Review "The Big Event" activity with students:

- What did we "program" the button events to do?

Now we're going to add events to our code. Specifically, we're going to have an event for when two characters touch each other.

- When have you seen two characters touch each other as an event in games?

## Bridging Activity - Events (10 min)

### Previewing Online Puzzles as a Class

Pull a puzzle from the corresponding online puzzles. We recommend puzzle 4 of this stage. Call on different students to make a funny face representing a mood when you click on Daisy. Explain this is an event that they are reacting to and Daisy can be coded to change moods when you click on her.

#### 💡 Lesson Tip

Students will have the opportunity to share their final product with a link. This is a great opportunity to show your school community the great things your students are doing. Collect all of the links and keep them on your class website for all to see!

Remind the students to only share their work with their close friends or family. For more information watch or show the class **Pause and Think Online - Video**.

## Main Activity (30 min)

### Online Puzzles and Free Play

At the final level students have the freedom to create a story of their own. You may want to provide structured guidelines around what kind of story to write, particularly for students who are overwhelmed by too many options.

## Wrap Up (5 - 10 min)

### Journaling

Having students write about what they learned, why it's useful, and how they feel about it can help solidify any knowledge they obtained today and build a review sheet for them to look to in the future.

Journal Prompts:

- What was today's lesson about?
- Draw one of the **Feeling Faces - Emotion Images** that shows how you felt about today's lesson in the corner of your journal page.
- Draw an event you used in your program today.
- Imagine that you have a remote controlled robot. What would the remote look like? Draw a picture of what you think you could make the robot do.

## Extended Learning

Use these activities to enhance student learning. They can be used as outside of class activities or other enrichment.

### Look Under the Hood

When you share a link to your story, you also share all of the code that goes behind it. This is a great way for students to learn from each other.

- Post links to completed stories online
  - Make a story of your own to share as well!
- When students load up a link, have them click the "How it Works" button to see the code behind the story.
- Discuss as a group the different ways your classmates coded their stories.
  - What surprised you?
  - What would you like to try?
- Choose someone else's story and click `Remix` to build on it. (Don't worry, the original story will be safe.)

## Standards Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **AP** - Algorithms & Programming



This curriculum is available under a  
Creative Commons License (CC BY-NC-SA 4.0).

If you are interested in licensing Code.org materials for commercial purposes, **contact us**.