Unplugged Activities for K-5

Printing Options

We've compiled a list of all of our unplugged lessons for you to use in your classroom. Now you can teach the fundamentals of computer science, whether you have computers in your classroom or not!

We refer to lessons in which students are not working on a computer as “unplugged.” Students will often work with pencil and paper or physical manipulatives. These are intentionally placed kinesthetic opportunities that help students digest complicated concepts in ways that relate to their own lives.



Unplugged lessons are particularly useful for building and maintaining a collaborative classroom environment, and they are useful touchstone experiences you can refer to when introducing more abstract concepts. Each of these activities can either be used alone or with other computer science lessons on related concepts.

Week 1	Lesson 1: Happy Maps ✂	Lesson 2: Move It, Move It ✂	Lesson 3: My Robotic Friends Jr.	Lesson 4: Graph Paper Programming ✂	
Week 2	Lesson 5: Happy Loops ✂	Lesson 6: Getting Loopy ✂	Lesson 7: My Loopy Robotic Friends Jr.	Lesson 8: The Big Event Jr. ✂	Lesson 9: The Big Event
Week 3	Lesson 10: The Right App ✂	Lesson 11: Designing for Accessibility	Lesson 12: Digital Sharing ✂	Lesson 13: Picturing Data	Lesson 14: Pattern Block Data
Week 4		Lesson 15: Binary Bracelets	Lesson 16: The Internet ✂	Lesson 17: Hardware and Software Skits	Lesson 18: Conditionals with Cards ✂
Week 5	Lesson 19: Songwriting ✂	Lesson 20: Blank Space Stories ✂			

Key	Instructional Lesson ✔ Assessment ✂ Unplugged Lesson

View calendar

Sequencing

Lesson 1: Happy Maps

Grade Level: Kindergarten - 1st Grade

This **context-setting** lesson brings together teams with a simple task: get the "flurb" to the fruit. Students will practice writing precise instructions as they work to translate instructions into the symbols provided.

Unplugged Activity — **Video: Happy Maps - Unplugged Activity**

2 — **Happy Maps - Student Resources**

Lesson 2: Move It, Move It

Grade Level: 1st Grade - 2nd Grade

Students will program their classmates to step carefully around a large grid on the floor in this **context-setting** lesson.

Unplugged Activity — **Move It Move it Video - Unplugged Activity**

2 — **Move It, Move It - Student Resources**

Lesson 3: My Robotic Friends Jr.

Grade Level: 2nd Grade - 3rd Grade

In this **context-setting** lesson, students will use a set of symbols to instruct a "robot" to stack cups in different patterns. Students will take turns participating as the robot, responding only to the algorithm defined by their peers.

Unplugged Activity — **My Robotic Friends - Unplugged Activity**

2 — **My Robotic Friend - Student Resources**

Lesson 4: Graph Paper Programming

Grade Level: 3rd Grade - 4th Grade

In this **context-setting** lesson, students use symbols to instruct each other to color squares on graph paper. By "programming" one another to draw pictures, students get an opportunity to experience some of the core concepts of programming in a fun and accessible way.

Unplugged Activity **Graph Paper Programming - Video**

2 **Graph Paper Programming - Student Resources**

Loops

Lesson 5: Happy Loops

Grade Level: Kindergarten - 1st Grade

This **context-setting** lesson revisits Happy Maps. This time, students will use loops to solve bigger, longer puzzles with their code.

1 **Happy Loops - Unplugged Activity**

2 **Happy Loops - Student Resources**

Lesson 6: Getting Loopy

Grade Level: 1st Grade - 2nd Grade

In this **context-setting** lesson, students will learn to use loops to more easily communicate instructions by looking at the repeated patterns of a dance.

Unplugged Activity **Video: Getting Loopy**

2 **Getting Loopy - Student Handout**

Lesson 7: My Loopy Robotic Friends Jr.

Grade Level: 2nd Grade - 3rd Grade

This **context-setting** lesson builds on the initial "My Robotic Friends" activity, featuring larger and more complicated designs.

Unplugged Activity My Loopy Robotic Friends - Unplugged Activity

2 My Loopy Robotic Friend - Student Resources

Events

Lesson 8: The Big Event Jr.

Grade Level: Kindergarten - 1st Grade

In this **context-setting** lesson, the class will experience the concept of events through a game where they move or shout when you press buttons on a giant remote.

Unplugged Activity The Big Event - Unplugged Activity

2 The Big Event Jr. - Student Resources

Lesson 9: The Big Event

Grade Level: 2nd Grade - 3rd Grade

In this **context-setting** lesson, students will learn that events are a great way to add flexibility to a pre-written algorithm.

Unplugged Activity The Big Event - Unplugged Activity

2 The Big Event - Student Resources

Impacts of Computing

Lesson 10: The Right App

Grade Level: 1st Grade - 2nd Grade

In this **exploratory** lesson, students learn about different computing technologies that exist. They also compare how people lived before the invention of apps and computers with how people live now. The lesson concludes with an activity where students empathize with several fictional

smartphone users to help them design the “right app” that addresses their needs.

Unplugged Activity **The Right App - Unplugged Activity**

2 **The Right App - Student Resource**

Lesson 11: Designing for Accessibility

Grade Level: 4th Grade - 5th Grade

In this **exploratory** lesson, students will learn about accessibility and the value of empathy through brainstorming and designing accessible solutions for hypothetical apps.

Unplugged Activity **Designing for Accessibility**

2 **Designing for Accessibility - Student Discussion**

Lesson 12: Digital Sharing

Grade Level: 4th Grade - 5th Grade

This **exploratory** lesson helps students understand the challenges and benefits of respecting ownership and copyright, particularly in digital environments.

Unplugged Activity **Digital Sharing - Video**

2 **Digital Sharing - Discussion Questions**

Data

Lesson 13: Picturing Data

Grade Level: 2nd Grade - 3rd Grade

Students will collect data from a Play Lab project and visualize it using different graphs in this **exploratory** lesson.

1 **Picturing Data**

Lesson 14: Pattern Block Data

Grade Level: 2nd Grade - 3rd Grade

In this unplugged lesson, students will practice storing, searching, retrieving, modifying and deleting the pattern blocks to simulate what computers do with data.

Unplugged Activity **Pattern Block Data - Unplugged Activity**

2 **Pattern Block Data - Resource Reminder**

Lesson 15: Binary Bracelets

Grade Level: 2nd Grade - 3rd Grade

This **exploratory** lesson introduces the concept of binary which is how computers today store all sorts of information.

Unplugged Activity **Binary Bracelets - Unplugged Activity**

2 **Binary Bracelets - Student Resources**

Lesson 16: The Internet

Grade Level: 5th Grade - 6th Grade

Students will pretend to flow through the internet while learning about connections, URLs, IP Addresses, and DNS in this **exploratory** lesson.

Unplugged Activity **The Internet - Unplugged Activity**

2 **The Internet - Student Resources**

Computer Systems

Lesson 17: Hardware and Software Skits

Grade Level: 3rd Grade - 4th Grade

In this unplugged lesson students will act out the ways internal and external parts of a computer work together. They will take on roles such as: CPU, Keyboard, Monitor, User and more.

Unplugged Activity **Unplugged Activity - Hardware and Software Skits**

2 **Hardware and Software Skits - Student Reminder**

Conditionals

Lesson 18: Conditionals with Cards

Grade Level: 3rd Grade - 4th Grade

In this **context-setting** lesson, students will write *conditional (if/else) statements* to state the rules of simple card games.

Unplugged Activity **Conditionals with Cards - Video**

2 **Conditionals with Cards - Student Resources**

Functions

Lesson 19: Songwriting

Grade Level: 4th Grade - 5th Grade

This **context-setting** lesson will help students understand why combining chunks of code into functions can be a helpful practice.

Unplugged Activity **Songwriting - Functions Video**

2 **Songwriting - Student Resources**

Variables

Lesson 20: Blank Space Stories

Grade Level: 5th Grade - 6th Grade

In this **context-setting** lesson, students will use fill-in-the-blank stories (similar to Mad Libs®) as a context for understanding how computers take and store input from a user, then use it later as a program runs.

Unplugged Activity **Video: Unplugged Variables in Envelopes**

2 **Blank Space Stories - Student Resources**

Lesson 1: Happy Maps

53 minutes

Overview

Grade Level: Kindergarten - 1st Grade

This **context-setting** lesson brings together teams with a simple task: get the "flurb" to the fruit. Students will practice writing precise instructions as they work to translate instructions into the symbols provided.

Purpose

The bridge from algorithms to programming can be a short one if students understand the difference between planning out a sequence and encoding that sequence into the appropriate language. This activity will help students gain experience reading and writing in shorthand code.

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **AP** - Algorithms & Programming

Agenda

Warm Up (5 minutes)
- **Introduction**

Main Activity (40 minutes)
- **Happy Maps Programming**

Wrap Up (8 minutes)

Extended Learning

Objectives

Students will be able to:

- Articulate an algorithm for a daily task
- Decode and run a program created by someone else
- Identify and address bugs or errors in sequenced instructions
- Translate an algorithm into a program

Preparation

- Print one ***Worksheet** and one set of ***Manipulatives** per group of 2-3.
- Make sure each student has a Reflection Journal.

Links

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

For the teachers

- **Happy Map Cards** - Worksheet
  Answer Key ▾ Make a Copy
- **Happy Maps** - Slides (**Download**)
  ▾ Make a Copy

For the students

- **Feeling Faces - Emotion Image** - Resource ▾ Make a Copy
- **Happy Map Cards** - Worksheet
  ▾ Make a Copy
- **Happy Map Game Pieces** - Manipulatives ▾ Make a Copy
- **Happy Maps** - Unplugged Video (**Download**)

Vocabulary

- **Algorithm** - A list of steps to finish a task.
- **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.

Teaching Guide

Warm Up (5 minutes)

Introduction

Display: Show "Step-by-step directions" slides

This warm-up is meant to get the class thinking about how to break a big problem down into a list of individual steps.

Do This: Before students give you any instructions, ask them to describe the **goal** of the program ("Walk to the chalkboard and draw a smiley face") and the **expected outcome** ("A smiley face will be drawn on the chalkboard"). Remind students that it is good practice to articulate the goal and outcome of the programs they will create before beginning to write their algorithms.

Model: Start by asking the class for step-by-step directions on how to get to the chalkboard. Make sure that you ask students to break apart any large instructions, like "Walk to the chalkboard," into smaller instructions like "Step forward."

When you reach the board, ask for instructions to draw a smiley face and try to get them to keep their instructions equally small.

Say: *Well done! You just gave me a list of steps to finish a task. In computer science, that's called an* ***Algorithm****!* (Follow up with your typical method of introducing vocabulary: word wall, repeat-after-me, etc.)

Teaching Tip

If the class starts shouting simultaneously, explain that you can only hear one instruction at a time. Call on students individually if that helps.

Display: Show "Vocabulary" slide

Say:

- **Algorithm** - A list of steps to finish a task.
- **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.

Main Activity (40 minutes)

Happy Maps Programming

Unplugged Activity **Video: Happy Maps - Unplugged Activity**

2 **Happy Maps - Student Resources**

In this exercise, the class will get map cards that have a pre-defined start space (flurb) and end space (fruit). Students will need to get the flurbs to the fruit on each card, using the arrows provided.

Model: Select one of the intermediate maps from the Happy Maps Cards worksheet (e.g., #3). Display it for the class and work through the puzzle together.

Have students look at the puzzle, then think-pair-share their solution for how they would get the flurb to the fruit.

Display: Show “Decompose it!” slide

Remind students to break each puzzle down into smaller steps.

Display: Show "Get the Flurb to the banana" slide

Think: This flurb needs to take two steps to get to the fruit. Work with your elbow partner to decide what you think those two steps are.

Pair: Have students discuss with neighbors for about 90 seconds.

Share: Ask a few students to describe their algorithm to the class. Move your finger along the displayed map as the students read their steps. Once you have a solution, ask if anyone else came up with a different idea that also works.

Display: Show "Using Symbols" slide

Now, share with the students that the magic step of changing an algorithm into a program happens when the code is written down using symbols. Do the students see any symbols on the display?

Think: Challenge students to encode the algorithm that they came up with before into symbols, and to write those symbols down in their journals (or on a piece of paper.)

Pair: Once students have written down their symbols, ask them to swap with a partner to see if they can follow each others' instructions.

Share: Ask for volunteers to come draw their arrows on the board. If the original code doesn't work, spend some time debugging as a class. Students should be familiar with the idea of "debugging" from previous lessons, so be sure to use the vocabulary to get them comfortable with it.

Once the code has been successfully written on the board, congratulate the class on writing their first program together!

Students' Turn

Display: Show "Your Turn" slide

Group: If your class is comfortable, place students into small groups of 2-3. Otherwise, you can continue solving these problems as a class and having them think-pair-share to write programs.

Distribute: Pass out one of the images from the ***Worksheet** to each group as needed.
(Optional) If you start noticing that students are ready for more, use the provided ***Manipulatives** and let students choose their own start and finish destinations on the blank map.

Encourage the students to follow these steps:

- Discuss an algorithm to get the flurb to the fruit.
- Encode the algorithm into arrows in their journals.
- Try their code to see if everything works as expected.
- Debug any issues and fix their code until it works correctly.

Share: When the lesson is done, offer to let groups share out the most difficult maps that they solved. If you have time, ask them to share their solutions as well.

Wrap Up (8 minutes)

Flash Chat

Display: Show "Flash Chat" slide

Discuss: When it's time to wind down, ask students if they can tell the difference between an algorithm and a program. Both are a list of steps, but a program (*code*) has been encoded in a way that can be run by a machine (or a kindergartener!)

- *Do you think that someone who speaks another language would be able to run your program?*
- *Why or why not?*

Reflection

Display: Show "Reflection" slide

Prompts:

- What was today's lesson about?
- 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 your own flurb map?
- What would the code be to solve your map?

Extended Learning

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

Stuffie Maps

- Create a life-size grid on the rug with tape and have student bring stuffed animals to school.
- Now students can program friends to move their actual stuffies as directed in the programs.

Create Your Own

- Have students create their own maps.
- Have other students solve them using programs.


This work 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: Move It, Move It

50 minutes

Overview

Grade Level: 1st Grade - 2nd Grade

Students will program their classmates to step carefully around a large grid on the floor in this **context-setting** lesson.

Purpose

By using physical movement to program their classmates, students will run into issues and emotions similar to what they will feel when they begin coding on a computer. Encountering those stresses in a playful and open environment will help to alleviate intensity and allow students to practice necessary skills before they run into problems on their own.

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

▶ **AP** - Algorithms & Programming

Agenda

Warm Up (20 minutes)
- **Vocabulary**
- **Where did I go wrong?**

Main Activity (20 minutes)
- **Move It, Move It**

Wrap-up (10 minutes)
- **Reflection**

Extended Learning

Objectives

Students will be able to:

- Define a list of steps (algorithm) to get a friend from their starting position to their goal
- Identify and fix errors in the execution of an algorithm
- Translate a list of steps into a series of physical actions

Preparation

- Watch the ***Teacher Video**
- Print (or otherwise prepare to display) one ***Worksheet Answer Key**
- Print one ***Map Activity** and one ***Worksheet** per group of 2-3 students
- Prepare blank papers to fill out the rest of the walking grid (4-7 needed per group)

Links

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

For the teachers

- **Move It, Move It** - Worksheet Answer Key ▾ Make a Copy
- **Move It, Move It** - Slides (**Download**) ▾ Make a Copy
- **Move it, Move it** - Teacher Video

For the students

- **Feeling Faces Emotion Image** - Resource ▾ Make a Copy
- **Move It, Move It** - Map Activity ▾ Make a Copy

- **Move It, Move It** - Worksheet
  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.

Teaching Guide

Warm Up (20 minutes)

Vocabulary

Display: Show "Vocabulary" slide

Say:

- **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.

Where did I go wrong?

Display: Show "Where did I go wrong?" slide

Goal: In this lesson, we want to help students learn to identify and fix bugs in their own programs. The easiest way to do that is to first present students with a program that contains bugs that are not their fault.

Display: Show "How does it feel when you can't figure out an answer?" slide

Goal: Once students have helped you fix "your" program, share with them how frustrating it can be to make mistakes, and help them see that those feelings are completely normal and they shouldn't feel embarrassed by them.

Teaching Tip

If your class has not already learned cardinal directions, it will be worth covering them before they begin Course B. This conversion will come in handy for nearly all of the online puzzles aimed at first grade, as well as several of the unplugged activities.

Let students know that they will continue to see those directions in the online programs next to the direction arrows.

Display: Show "Debugging Worksheet" slide

Display: Display the image below so that everyone can see it.

Discuss: Get the attention of the class and let them know that you are stuck! You have this challenge, and you thought you had solved it, but it doesn't seem to be working. You want to walk to your desk but your program has a bug. Can your students help you fix it?

Take a moment to walk them through the rules:

- Start at the compass rose
- Follow the instructions step-by-step
- End at the desk

Do This: Before students give you any instructions, ask them to describe the **goal** of the program ("Walk from the starting point to the desk") and the **expected outcome** ("Arrive at the desk"). Remind students that it is good practice to articulate the goal and outcome of the programs they will create before beginning to write their algorithms.

(*Optional:* Walk through your program using your fingers on top of the map, under the document camera. Express frustration when your fingers end up off the map, instead of at the treasure.)

Think: *My program says "East, East, North". Can you figure out why my program doesn't work?*

Pair: Let students work together to see if they can figure out what the program is supposed to say.

Share: Ask students if anyone was able to figure out a way to solve the problem. When you get a correct answer, let the students know that they are great at debugging!

Discuss: Ask the students if they could tell how you were feeling when you couldn't figure out the answer. They might suggest that you were "mad" or "sad". Instead of telling them "no", describe that you were feeling a little bit mad, a little bit sad, and a little bit confused. When you put all of those emotions together, it makes a feeling called "frustration". When you are "frustrated" you might think you are mad, sad, or confused -- and you might be tempted to give up -- but frustration is a natural feeling and it's a big hint that you are about to learn something! Instead of quitting, practice persistence. Keep trying over and over again. After a few times, you will start to understand how to debug your problems!

Teaching Tip

Take a moment to cover tips on frustration and persistence as a class:

Tips to Help With Frustration

- Count to 10
- Take deep breaths
- Journal about them
- Talk to a partner about them
- Ask for help

Tips for Being Persistent

- Keep track of what you have already tried
- Describe what is happening
- Describe what is supposed to happen
- What does that tell you?
- Make a change and try again

Distribute: To make sure that students understand the idea of finding and fixing errors (debugging) pass out the ***Worksheet** and have students complete the task in pairs.

(*Optional:* If you want to move the activity along more quickly, feel free to complete these as a class, instead.)

Transition: *Now it's time to play the game!*

Main Activity (20 minutes)

Unplugged Activity **Move It Move it Video - Unplugged Activity**

2 **Move It, Move It - Student Resources**

Move It, Move It

Display: Show “Map Activity” slide

Distribute: Hand each group of 2-3 students a ***Map Activity**, as well as the blank papers for the grid on the ground. Allow students to either cut the halves of each map apart, or fold the sheets in half so that each map is clearly visible (without distraction.)

Set-Up: In each group, each player will get a task.

- Player 1: Choose/set-up the map to play
- Player 2: Programmer
- Player 3: Walking Machine

Directions for Class:

1. Decide who will take each job.
2. Have player 1 set a grid on the floor made up of pieces of paper (as shown on one of the Move It Maps) except with the gem paper facing the ground.
3. Player 3 will start by standing on the page with the compass rose.
4. Player 2 will then guide player 3 step-by-step through the paper maze using the provided arm signals.
5. When player 2 gives the signal to “STOP”, player 3 will flip over the page that they are on. If that page is a gem, then the maze was a success!
6. If there is time, let everyone rotate positions and go again!

Note that the rules are not the most important thing here. Feel free to clarify if the students have questions, but if the students are playing a bit differently than described, you don't need to hold them to the letter of the game. The crucial bit is that they are moving from immediate instructions to giving two or three instructions before the Walking Machine moves.

Display: Show “Decompose it!” slide

Remind students to break each puzzle down into smaller steps.

Wrap-up (10 minutes)

Reflection

Display: Show "Reflection” slide

Prompts:

- What were the four directions on the compass rose?
  - What tricks can we use to remember North, South, East and West?
- Draw another way we could we have given instructions without using our arms.
- Draw your favorite part about that game.

Teaching Tip

Reflection prompts like these can be completed in a variety of styles. Journaling or discussion (with groups or partners) are great options!

Extended Learning

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

X's and O's

- Draw a tic-tac-toe board for the class.
- Place a single X and a single O somewhere on the board.
- Ask the class if they can get the X to the O using arm gestures as a class.

X's, O's, and Arrows

- Similar to the activity above, but have the students write their programs in advance using arrows instead of hand gestures.
  - This can be done in groups.
  - Groups can share their solutions for the class.



This work 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: My Robotic Friends Jr.

55 minutes

Overview

Grade Level: 2nd Grade - 3rd Grade

In this **context-setting** lesson, students will use a set of symbols to instruct a "robot" to stack cups in different patterns. Students will take turns participating as the robot, responding only to the algorithm defined by their peers.

Purpose

This unplugged lesson brings the class together as a team with a simple task to complete: get a "robot" to stack cups in a specific design. This activity lays the groundwork for the programming that students will do throughout the course as they learn the importance of defining a clearly communicated algorithm.

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **AP** - Algorithms & Programming

Agenda

Warm Up (15 minutes)
- **Reflect**
- **Vocabulary**
- **Introduction**

Activity (30 minutes)
- **My Robotic Friends**

Wrap Up (10 minutes)
- **Reflection**

Objectives

Students will be able to:

- Attend to precision when creating instructions
- Identify and address bugs or errors in sequenced instructions
- Name daily processes where you follow an algorithm of steps to complete a task.

Preparation

- Prepare a stack of 20 paper cups (or paper trapezoids) for each group of 2-3 students.
- Display the symbols from ***Symbol Key** where students can reference throughout the lesson.
- (Optional) Print out one ***Cup Stacking Ideas** per group of 2-3 students.

Links

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

For the teachers

- **My Robotic Friends Jr.** - Slides (**Download**) ▾ Make a Copy

For the students

- **My Robotic Friends** - Cup Spacing ▾ Make a Copy
- **My Robotic Friends** - Symbol Key ▾ Make a Copy
- **My Robotic Friends** - Unplugged Video (**Download**)

- **My Robotic Friends** - Cup Stacking Ideas ▾ Make a Copy
- **My Robotic Friends** - Paper Trapezoid Template ▾ 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.

Teaching Guide

Warm Up (15 minutes)

Reflect

Display: Show "Reflect" slide

Discuss: *When would you need to complete a list of steps in order?*

Discussion Goal: Students will think of examples of when they would need to complete a list of steps in order. Such as steps to turn on the light in your classroom or to get a drink of water.

Vocabulary

Display: Show "Vocabulary" slide

- **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

Introduction

Talking to Robots

Display: Show "Introduction" slide

Watch one of the videos below to give students context for the types of things that robots can do:

- **Asimo by Honda** (3:58)
- **Egg Drawing Robot** (3:15)
- **Dancing Lego Robot** (1:35)

Display: Show “Discuss” slide

Discuss: Refer to the video that you chose and ask students:

- *How do you think that the robot knew what to do?*
- *Does a robot really “understand” what you say?*
- *Is it worried about getting in trouble if it doesn't do what it's told?*

The goal of this quick discussion is to call out that while robots may seem to behave like people, they're actually responding only to their programming. Students will likely refer to robots from movies and TV that behave more like humans. Push them to consider robots that they've seen or heard of in real life, like *Roombas*, or even digital assistants like *Amazon Alexa*.

Say: *Robots can only do what they've been told to do, but we don't just tell them using words. In order to do something, a robot needs to have a list of steps that it can read. Today, we are going to learn what it takes to make that happen.*

Activity (30 minutes)

My Robotic Friends

Unplugged Activity **My Robotic Friends - Unplugged Activity**

2 **My Robotic Friend - Student Resources**

Unplugged

My Robotic Friends

Symbol Key (Course B)



Pick Up Cup



Put Down Cup



Step Forward



Step Backward

Introduction and Modeling

Set Up: Have stacks of cups or cut paper trapezoids available for groups.

Teaching Tip

Handy Rules:

- **Up** means that the cup automatically goes up as high as it needs to.
- **Down** means that it automatically goes down until it lands on something.
  - The hand automatically returns to cup stack after setting down a cup.

- **Forward** means the robot moves one step (1/2 cup width) forward.
- **Backward** means the robot moves one step (1/2 cup width) backward.
  - Note: Students may not use backward at this age unless they want to build the cup stacks in reverse (which is also okay).
- Programmers are not allowed to talk when the robot is working. This includes blurting out answers or pointing out when the robot has done something wrong.
- Programmers should raise their hand if they see a bug.

Differentiation Options:

Simplify: Does this all feel a little complicated for your students?

Don't forget to model this in front of the class until students understand all of the rules. If it's still confusing, try running this whole activity together as a classroom using volunteers as robots, instead of breaking up into groups!

Intensify: Are your students more advanced? Do you want this lesson to relate more closely to the online puzzles? Here are some modifications that you can make:

- One arrow corresponds to one movement:
  - When a cup is removed from the stack, it returns to table-level before moving.
  - Students need to use multiple "up" arrows to lift the cup multiple levels.
  - Students need to use multiple "down" arrows to lower the cups multiple levels.
  - Students need to use the "back" arrows to get back to the cup stack.

🖽 **Display**: Show “Symbol Key” slide

Display: *Symbol Key or write the allowed actions on the board - make sure these are in a place where they can be seen for the whole activity. Explain to the class that these will be the only four actions that they can use for this exercise. For this task, they will instruct their “robot” friend to build a specific cup stack using only the commands listed on the key.

🖽 **Display**: Show “Three Cup Pattern” slide

Model: In order to explain how the instructions are intended to work, model for the class how to create and follow an algorithm for replicating a simple pattern. Place a single stack of cups in front of you to start.

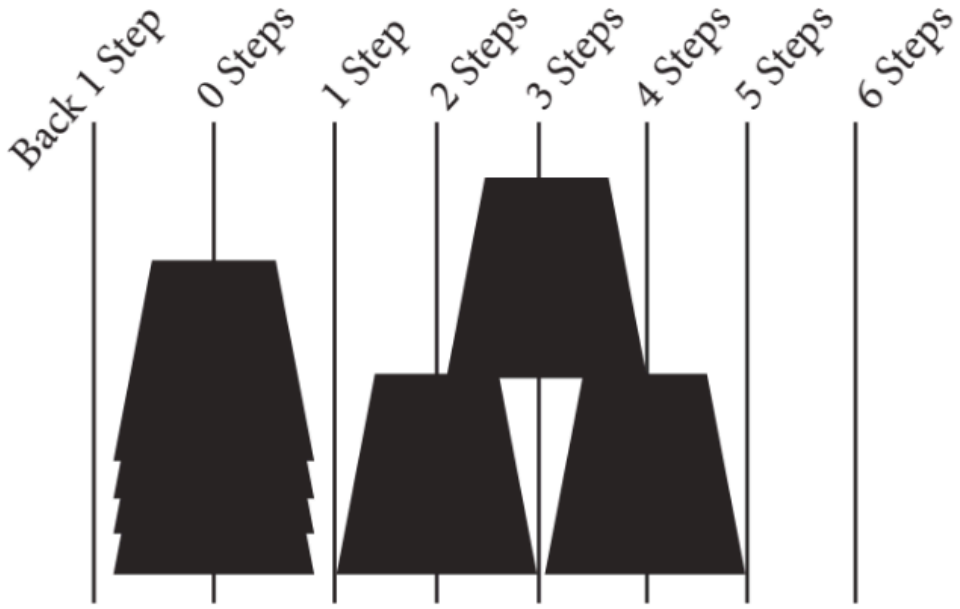
Display: Hold up the pattern you plan to model. A simple three cup pattern is a great place to start.



Display: Show “List of Steps” slide

Prompt: Ask the class what the first instruction should be, using only the four instructions allowed. The first move should be to "pick up cup." If students suggest something else from the list, perform that action and allow them to see their error. If they suggest something not from the list, make a clear malfunction reaction and let them know that the command is not understood.

With cup in hand, ask the class to continue giving you instructions until the first cup is placed. This is a great place to clarify that "step forward" and "step backward" each imply moving half a cup width. See the image below for reference.



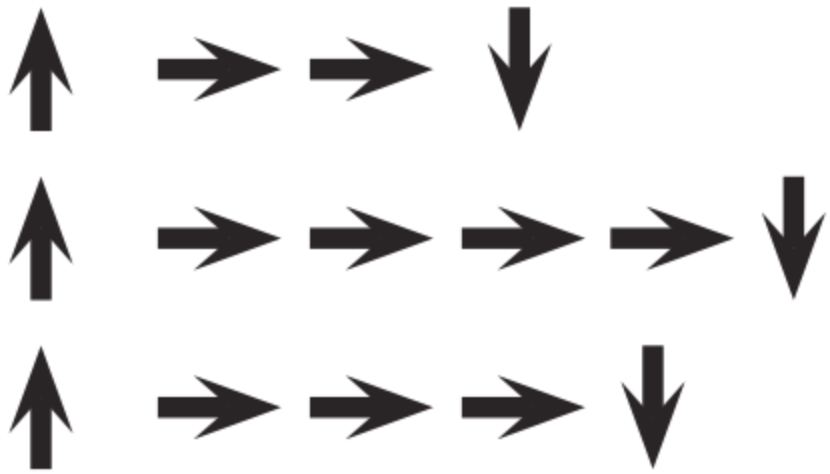
Continue asking for instructions from the classroom until you have completed the entire design.

Once your stack is complete, point out that they just gave you a list of steps for completing a task. That's an algorithm. Algorithms are great for sharing ideas, but spelling them out word by word can take a long time. That's what the symbols are for! When you change an algorithm into symbols that a robot (or computer) understands, that's called programming.

Ask the class to help you write the "program" for that first move by changing the text into an arrow. Then work with them to write down the rest of the moves necessary to complete the pattern. Depending on the confidence of your students, you might switch back and forth frequently between acting as the "robot" and writing down the code, or you might push them to write the whole program before you will implement it.

Display: Show “Answer Key” slide

Say: *One possible solution looks like this:*



Volunteer: Once the class has completed the model program, ask one of the students to come up and act as the "robot" to ensure that the program really works. Encourage them to say the instructions out loud as they "run" the code.

Programming Your Robots

Display: Show “Programming Your Robots” slide

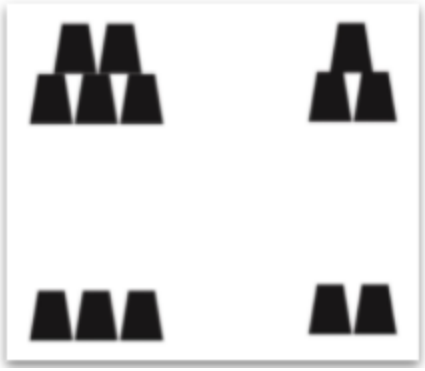
Group: Place students into groups of 4. Each group should then further break down into two pairs - each pair will develop their own program to be "run" by the other pair.

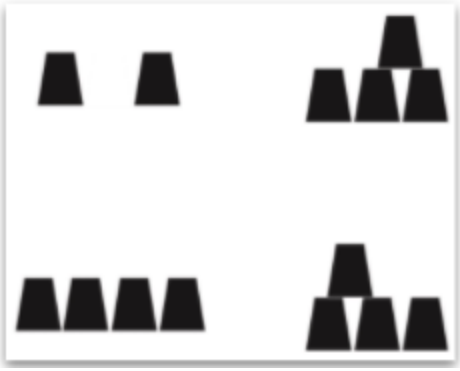
Distribute: Give each group one stack of cups or paper cutouts.

Display: *Cup Stacking Ideas to the class or hand out individual copies for groups to use. Have each pair (not group) choose which design they would like their robots to do. Try to push for an easier design for the first time, then have them choose a more complex design later on. Encourage pairs to keep their choice secret from the other half of their group.

Say:

- *Once you are in your group, divide into partners.*
- *Look at Cup Stacking Ideas.*
- *With your partner, choose one design for your robot to do.*





Display: Show "The "Program" slide

Discuss:

- *In pairs, discuss how the stack should be built, using only arrows.*
- *Write down your algorithm on a piece of paper.*

Display: Show "Running and Debugging Your Robots" slide

Say:

- *Now it's time to be a robot! Trade algorithms with the other partners in your group.*
- *Be on the lookout for bugs. If you see one, have the robot finish the instructions as best as it can.*
- *Discuss the bug with your partners and come up with a solution.*
- *Keep testing the algorithm until it works!*

Circulate: Look for groups who are trying to take shortcuts by adding extra things (like numbers) to their code. Praise them for their ingenuity, but remind them that for this exercise, the robots do not understand *anything* but the provided symbols. If you like, you can hint that they should save their brilliant solution for the next time they play this game, since they might get the chance to use their invention soon!

Iterate: Depending on the time available, mix up the pairs and give them a chance to do a different pattern. Each time groups repeat the process, encourage them to choose a more challenging pattern.

Display: Show "Discuss" slide

Discuss: After everyone has had a chance to be the robot, bring the class back together to discuss their experience. The goal of this discussion is to give students space to make sense of their experience both as robot and programmer. The questions are intentionally broad, but designed to get students thinking about the challenges of writing a clear program and the constraints of a robot or computer in interpreting your instructions. In particular, discuss as a class:

- *What was the most difficult part of coming up with the instructions?*
- *When you were the robot, what was the hardest part of following the instructions you were given?*

Display: Show "Discuss (continued)" slide

- *Did anyone find a bug in your instructions once your robot started following them?*
- *What was the bug?*
- *Why do you think you didn't notice it when writing the program?*

Wrap Up (10 minutes)

Reflection

Teaching Tip

Reflection prompts like these can be completed in a variety of styles. Journaling or discussion (with groups or partners) are great options!

Display: Show "Reflect" slide

Reflect: *Draw your own stack of cups that you would like to see a robot build. Can you create a program for that cup stack?*

This work 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: Graph Paper Programming

55 minutes

Overview

Grade Level: 3rd Grade - 4th Grade

In this **context-setting** lesson, students use symbols to instruct each other to color squares on graph paper. By "programming" one another to draw pictures, students get an opportunity to experience some of the core concepts of programming in a fun and accessible way.

Purpose

The goal of this activity is to build critical thinking skills and excitement for the course, while introducing some of the fundamental programming concepts that will be used throughout the course. By introducing basic concepts like sequencing and algorithms to the class in an unplugged activity, students who are intimidated by computers can still build a foundation of understanding on these topics. In this lesson, students will learn how to develop an algorithm and encode it into a program.

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

▶ **AP** - Algorithms & Programming

Agenda

Warm Up (15 minutes)
- **Reflect**
- **Vocabulary**
- **Introduction**

Main Activity (30 minutes)
- **Graph Paper Programming**

- **Optional Assessment (10 minutes)**

Wrap Up (10 minutes)
- **Reflection**

Extended Learning

Objectives

Students will be able to:

- Explain constraints of translating problems from human language to machine language
- Reframe a sequence of steps as an encoded program
- Try out various roles to collaborate with peers on an algorithm.

Preparation

- (Optional) Watch the ***Lesson in Action** video.
- Print out one ***worksheet** and ***assessment** for each student.
- **Note:** You may want to teach an unplugged lesson on debugging before moving onto the coding lessons that follow this lesson. Have your students practice their debugging skills with an unplugged **Relay Programming lesson.**

Links

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

For the teachers

- **Graph Paper Programming** - Lesson in Action Video
- **Graph Paper Programming** - Worksheet Answer Key
  ▼ Make a Copy
- **Graph Paper Programming** - Assessment Answer Key
  ▼ Make a Copy

- **Graph Paper Programming** - Slides (**Download**) ▾ Make a Copy

For the students

- **Graph Paper Programming** - Activity Worksheet ▾ Make a Copy
- **Graph Paper Programming** - Unplugged Video (**Download**)
- **Graph Paper Programming** - Assessment ▾ Make a Copy

Vocabulary

- **Algorithm** - A list of steps to finish a task.
- **Collaborate** - Work together on an activity
- **Program** - An algorithm that has been coded into something that can be run by a machine.

Teaching Guide

Warm Up (15 minutes)

Reflect

Display: Show "Reflect" slide

Reflect: *When would you need to complete a list of steps in order?*

Vocabulary

Display: Show "Vocabulary" slide

- **Algorithm** - A list of steps to finish a task.
- **Program** - An algorithm that has been coded into something that can be run by a machine.

Introduction

Display: Show "Introduction" slide

Watch one of the videos below to give students context for the types of things that robots can do:

- **Asimo by Honda** (3:58)
- **Egg Drawing Robot** (3:15)
- **Dancing Lego Robot** (1:35)

In the Main Activity, students will encode instructions to guide each other toward making drawings without letting the rest of their group see the original image. This warm-up frames the activity for the class.

Display: Show “Discuss” slide

Discuss: *How do you suppose that robots know how to do the things that they do? Do they have brains that work the same way that ours do?*

Discussion Goal: Call out that while robots may seem to behave like people, they're actually responding only to their programming. Students will likely refer to robots from movies and TV that behave more like humans. Push them to consider robots that they've seen or heard of in real life, like *Roombas*, or even digital assistants like *Amazon Alexa*.

Work this into a discussion on how people have to program robots to do specific things, using specific commands.

Main Activity (30 minutes)

Graph Paper Programming

Unplugged Activity **Graph Paper Programming - Video**

2 **Graph Paper Programming - Student Resources**

In this activity, students will act as both programmers and robots, coloring in squares according to programs that they have written for one another.

Distribute: Students will use 4x4 grids (or sheets of graph paper with 4x4 boxes sectioned off). They will also need the image worksheet.

Display: Show “Graph Paper Programming - Commands” slide

Do This: Project these commands, or write them on the board. They won't persist long, but they will help students make the transition from Algorithm to Program.

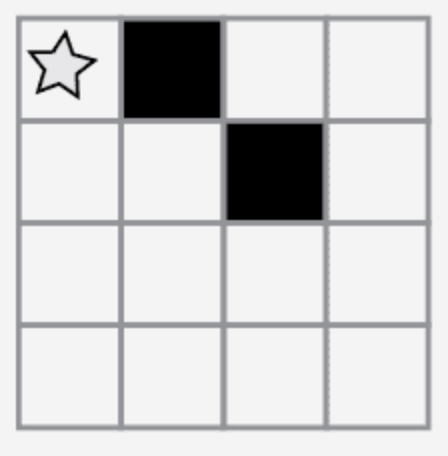
```
Move one square right
Move one square left
Move one square up
Move one square down
Fill in square with color
```

Remarks

> Today we all get to program robots...and they're already here in the room! It's you! We're going to write programs using symbols with special meanings to help each other recreate a picture. First, we'll practice together as if I am the robot and you are the programmers, then we can break up into groups so that everyone gets a turn.

Display: Show “Graph Paper Programming - Puzzle #1” slide

Display both the image that you are going to have the students walk you through, and a blank grid that you will fill-in with your ARM. Make sure that the instructions, grid, and image remain visible at the same time. When the direction “fill in square with color” is given, drag one of the two squares on the slide onto the grid to represent a square filled in with color.



Remarks

> *Here is an image. Pretend that I am the robot with an Automatic Realization Machine (ARM). These are the only instructions that I understand.*
>
> *Starting at the upper left-hand corner, guide my ARM out loud with your words.*

Model: The class might give you instructions like these below. As you hear an instruction that you intend to follow, make sure to repeat it out loud so that the students can keep track of what you are doing.

```
Move One Square Right
Fill In Square with Color
Move One Square Right
Move One Square Down
Fill In Square with Color
```

Continue with the activity until you have completed your sample square.

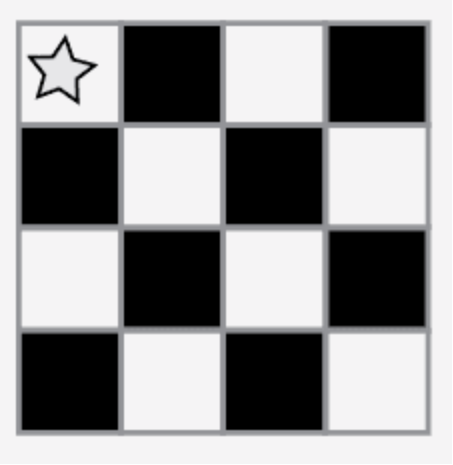
Capture: Write each of the commands down so that students can see all of the steps that went into the one image.

```
Move One Square Right
Fill In Square with Color
Move One Square Right
Move One Square Down
Fill In Square with Color
```

Say: *You just gave me a list of steps to finish a task. In programming, they call that an algorithm. Algorithms are great, because they are easy for you to understand as the programmer. BUT, what happens when we want to write down the algorithm for a drawing like this?*

Display: Show “Graph Paper Programming - Puzzle #2” slide

Show the students a more complicated image, like the one below. Feel free to use the "List of steps" table in the slide to write down the steps students suggest.



Next, begin writing down some of the instructions that it would take to replicate that image. Hopefully, students will see that writing everything out longhand would quickly become a bit of a nightmare.

```
Move One Square Right
Fill In Square with Color
Move One Square Right
Move One Square Right
Fill In Square with Color
Move One Square Down
Move One Square Left
Fill In Square with Color
Move One Square Left
Move One Square Left
Fill In Square with Color
PLUS 12 MORE INSTRUCTIONS!
```

Display: Show next slide

Show the students this list of symbols.



Discuss: *How could we use these symbols to make our instructions easier?*

Draw out ideas that relate to transitioning from the verbal instructions to the symbols. Once the students get to that place, point out that this text:

- “Move one square right, Move one square right, Fill-in square with color”

would now correspond to the program:



Discussion Goal: Get at the idea that students can use symbols to stand for entire phrases. Once they understand that, share with them that making the switch from listing steps in detail to encoding them is called **"programming."**

Display: Show next slide

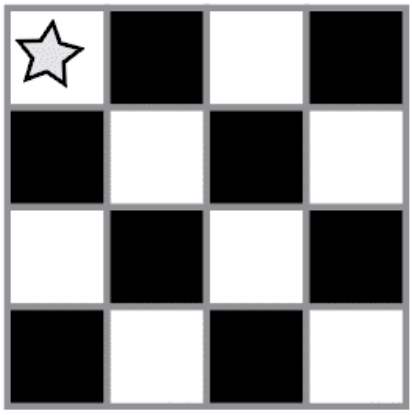
Model: Now, have the class help you draw the larger image using only symbols. Do not worry about unnecessary steps for now. If their final program works to create the image, consider it a win. (The symbols are infinitely cloned on this slide for your use in modeling this puzzle.)

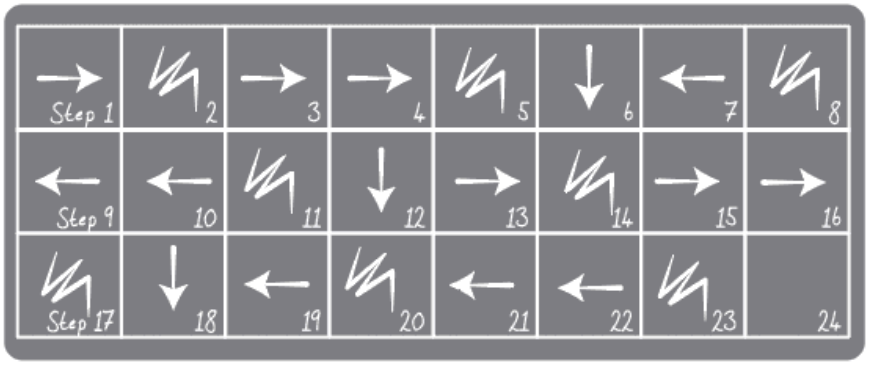
Teaching Tip

Notice that we have written our program from left to right like you would read a book in English. Some students prefer this method. Others like to start each line of the grid on a new line of paper. The way they write their program doesn't matter as much as whether the other people in their groups can follow along!

The classroom may be buzzing with suggestions by this point. If the class gets the gist of the exercise, this is a good place to discuss alternate ways of filling out the same grid. If there is still confusion, save that piece for another day and work with another example.

Display: Show "Sample Solution" slide





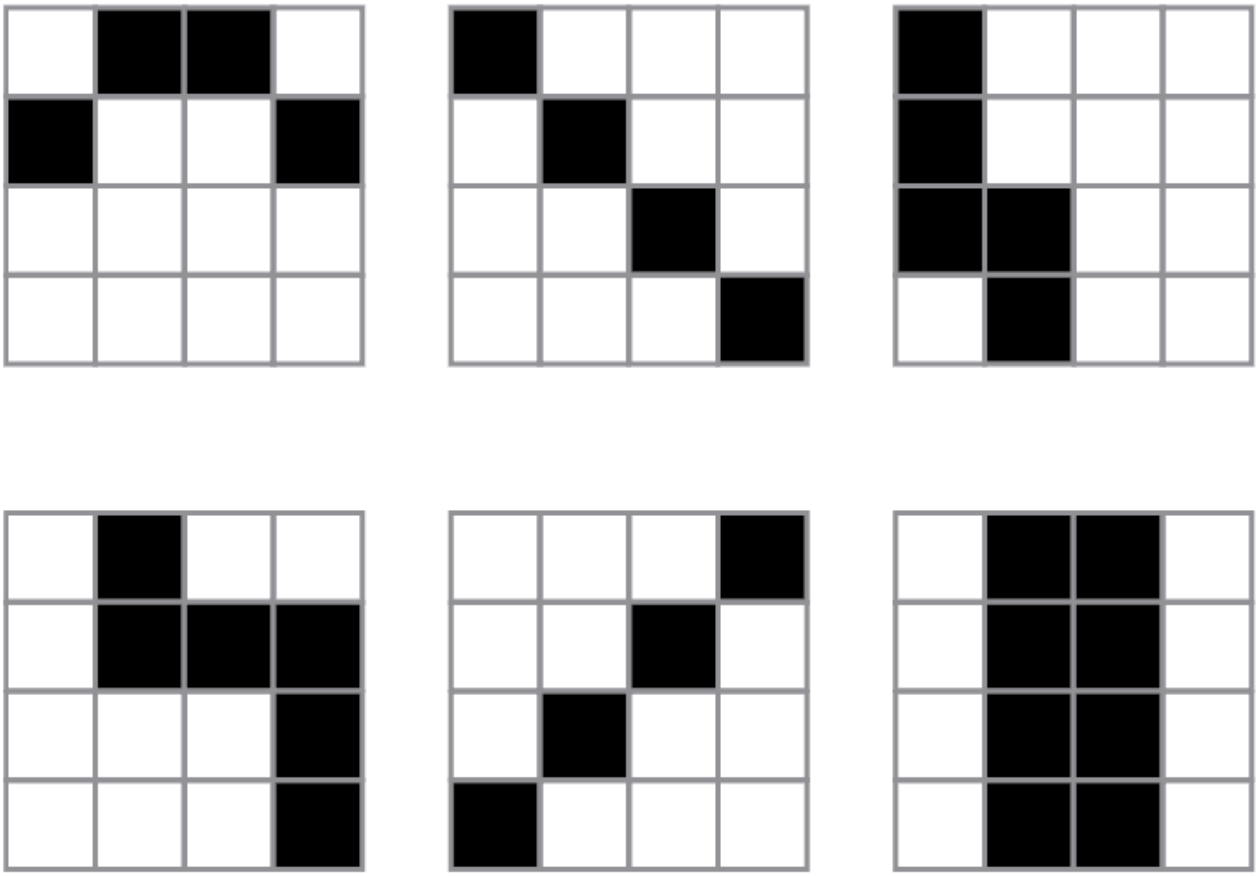
Display: Show "Your Turn!" slide.

The Students' Turn

Display: Show "Activity Sheet" slide

Group: Divide students into pairs or small groups.

- Have each pair/group choose an image from the worksheet.
- Discuss the algorithm to draw chosen image with partner(s).
- Convert algorithm into a program using symbols.
- Trade programs with another pair/group and draw one another's image.
- Choose another image and go again!



Optional Assessment (10 minutes)

Display: Show next slide if using the Optional Assessment

- Hand out ***Graph Paper Programming - Assessment** and allow students to complete the activity independently after the instructions have been well explained.
- This should feel familiar, thanks to the previous activities.

Wrap Up (10 minutes)

Reflection

Display: Show "Reflect" slide

Reflect:

- *Draw another image that you could code. Can you write the program that goes with this drawing?*
- *What other types of robots could we program if we changed what the arrows meant?*

Teaching Tip

Reflection prompts like these can be completed in a variety of styles. Journaling or discussion (with groups or partners) are great options!

Extended Learning

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

Better and Better

- Have your class try making up their own images.
- Can they figure out how to program the images that they create?

Class Challenge

- As the teacher, draw an image on a 5x5 grid.
- Can the class code that up along with you?



This work 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: Happy Loops

33 minutes

Overview

Grade Level: Kindergarten - 1st Grade

This **context-setting** lesson revisits Happy Maps. This time, students will use loops to solve bigger, longer puzzles with their code.

Purpose

This lesson serves as an introduction to loops. Loops allow for students to simplify their code by grouping commands that need to be repeated. Students will develop critical thinking skills by noticing repetition in movements of their classmates and determining how many times to repeat commands. By seeing "Happy Maps" again, students will get the chance to relate old concepts such as sequencing to the new concept of repeat loops.

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

▶ **AP** - Algorithms & Programming

Agenda

Warm-Up (5 minutes)

Main Activity (20 minutes)
- **Happy Loops**

Wrap Up (8 minutes)
- **Reflection**

Extended Learning

Objectives

Students will be able to:

- Identify repeating code and use numbers to shorten multiple actions into a single loop.
- Interpret a program with loops as a series of multiple actions.

Preparation

- Print out a ***Worksheet** to display for the class
- Per each group of 2-3, print one of each: ***Game pieces, *XL Worksheet, and *Bonus Pack Manipulatives**.
- Make sure each student has a Reflection Journal.

Links

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

For the teachers

- **Happy Loops** - Slides (**Download**) ▾ Make a Copy

For the students

- **Feeling Faces Emotion Image** - Resource ▾ Make a Copy
- **Happy Map Cards** - Worksheet ▾ Make a Copy
- **Happy Map Cards XL** - Worksheet ▾ Make a Copy
- **Happy Map Game Pieces** - Manipulatives ▾ Make a Copy
- **Happy Map Game Pieces Bonus Pack** - Manipulatives ▾ Make a Copy

Vocabulary

- **Algorithm** - A list of steps to finish a task.
- **Loop** - The action of doing something over and over again.
- **Repeat** - To do something again.

Teaching Guide

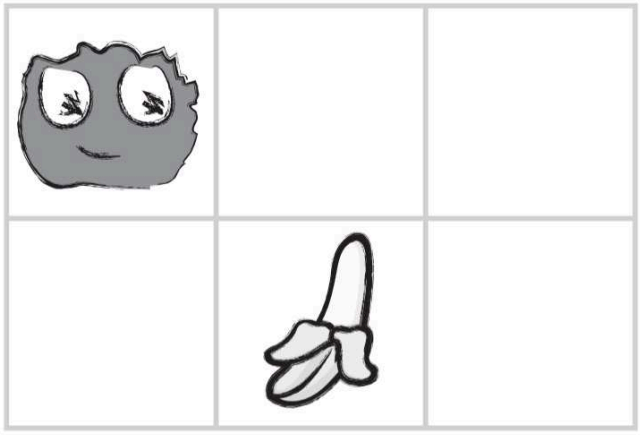
Warm-Up (5 minutes)

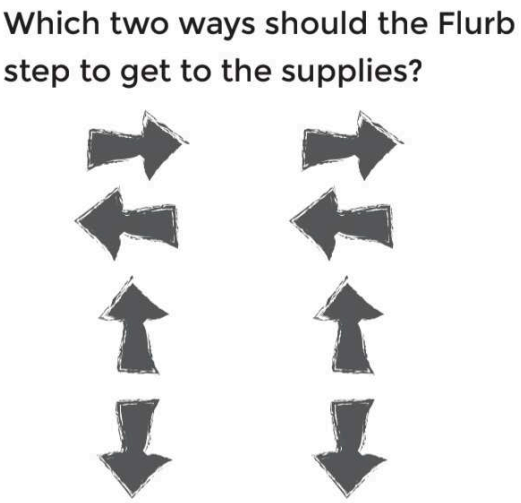
Happy Maps Review

Display: Show "Happy Maps Review" slide

This lesson builds off of the Happy Maps activity from earlier in the year. Students might benefit from a quick refresher before you hop into the difficult stuff.

Display:





Model: Select one of the maps that your class covered together in the last lesson.

Have students look at the puzzle, then think-pair-share their solution for how they would get the flurb to the fruit (using the symbols from last time).

Think: *Do you remember writing programs to get the flurbs to the fruit? What would this program look like coded with arrows?*

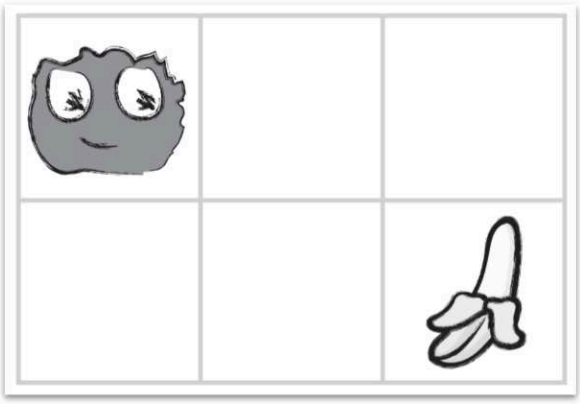
Pair: Have students discuss for about 90 seconds.

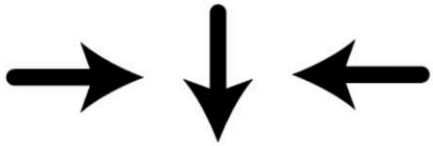
Share: Ask a student to use their fingers to point in the direction of the arrows that they chose to solve this puzzle.

Debugging Review

Display: Show "Debugging Review" slide

This activity is designed to help students review how to debug.





There's a bug with this code. Can you figure out how to fix the error?

Display:

Model: Ask students if they can spot the error with the code. Ask them where the Flurb ends up with the current algorithm.

Think: *What should we change in the algorithm to get the Flurb to the banana?*

Pair: Have students discuss for about 30 seconds.

Share: Ask a student to use their fingers to point in the direction of the arrows that they chose to solve this puzzle.

Display: Show "Vocabulary" slide

Say:

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

Main Activity (20 minutes)

Happy Loops

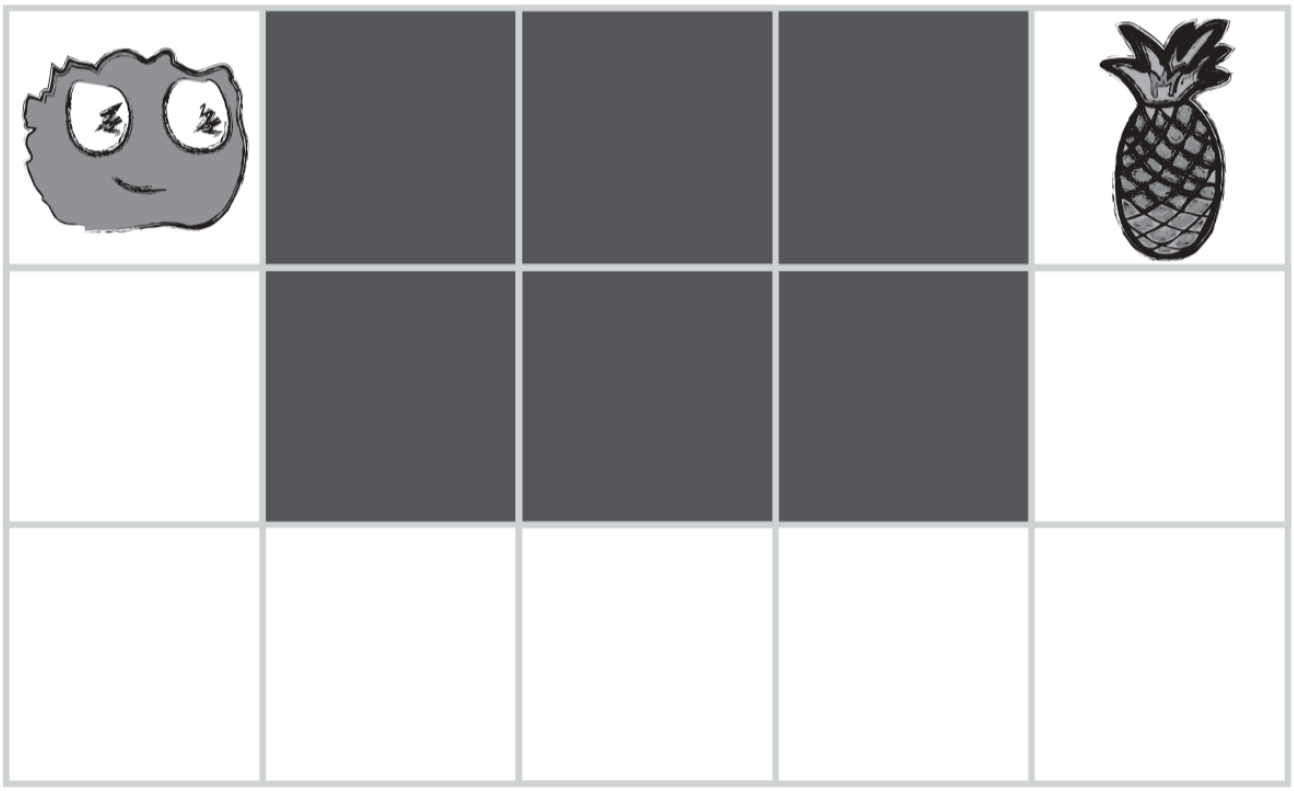
Display: Show "Happy Maps XL" slides

1 **Happy Loops - Unplugged Activity**

2 **Happy Loops - Student Resources**

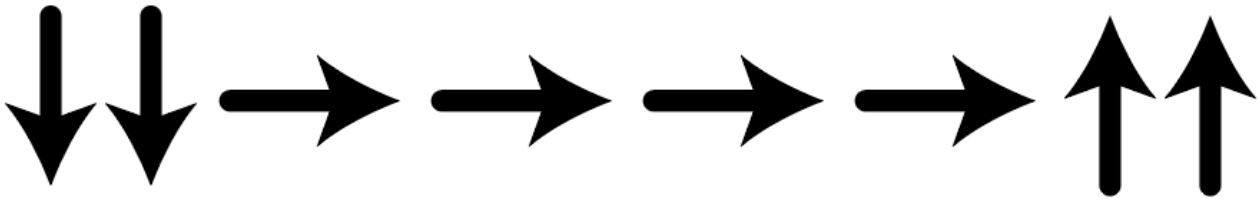
This portion of the lesson should help students see that there is an easier way to handle repetitive code than to brute force a solution with dozens of the same symbols.

Display:



Model: Once you are confident that your students remember Happy Maps, pull up one of the new -- and much longer -- *Happy Maps XL*.

Can your students help you program these maps? The resulting code might look something like this:

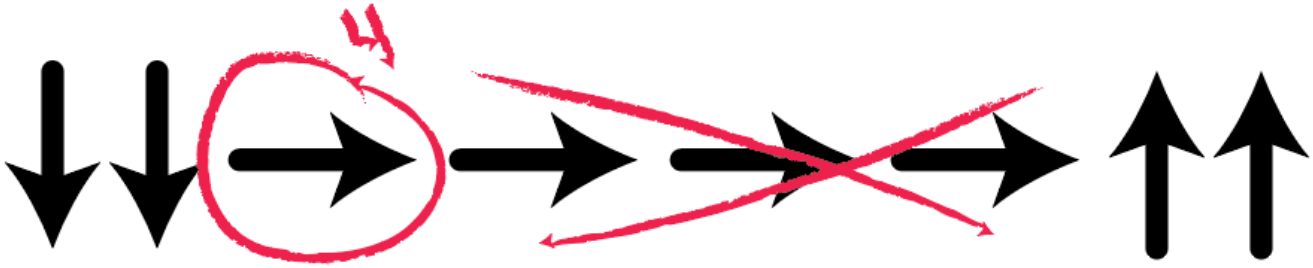


It's a bit longer, isn't it?

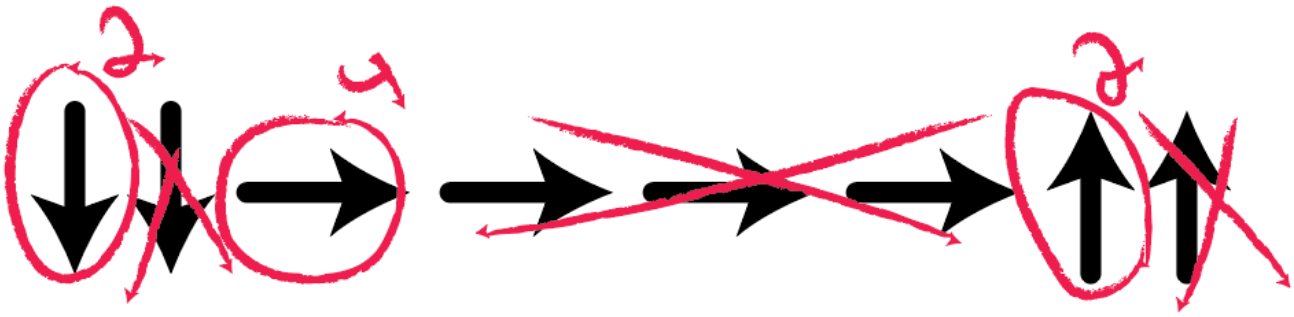
Discuss The point of this discussion is to get students to see that, sometimes, many symbols are repeated and they can lump all of those movements into a single icon.

Give students the opportunity to brainstorm shorter ways to relay the code that they're creating. (This bit can be skipped over if your students start saying things like: "Move forward 6 times." Since that will open the discussion about how to show "six times" with symbols.)

Ideally, students will land on a method that looks like this one:



Which can be reduced even further to this:



Eventually, students can write programs like this on the fly:



Define: Once students have put together the idea of repeating code, give them the vocabulary around it. Make sure to share with them that often the terms “repeat something” and “loop something” will be used interchangeably in Code Studio.

Happy Loops

Display: Show "Let’s do one more together!" slide

Now that students are familiar with the ability to repeat lots of code using a single loop, select an XL map and let them help you write code for the situation. Do this as many times together as a class as you need, then set students off in groups to solve some problems on their own. You may also need to add the ***Game Pieces Bonus Pack** to adapt this activity for loops.

Circulate: Make sure to walk around and have students run through their code with you watching. Are there any bugs? Use the debugging questions to help them find a solution.

- *What does it do?*
- *What is it supposed to do?*
- *What does that tell you?*
- *Does it work at the first step?*
- *Does it work at the second step?*
- *Where does it stop working?*

Wrap Up (8 minutes)

Reflection

Display: Show "Reflection" slide

Prompts:

- Draw a face that shows how you felt about today's lesson in the corner of your journal page.
- Have the students write or draw something in their journal that will remind them later what loops are. Prompts include:
  - What does "repeat" mean to you?
  - Draw a picture of you repeating something.

Extended Learning

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

- Create a life-size grid on the rug with tape and have student bring stuffies to school. Now students can program friends to move their actual stuffies as directed in the programs.
- Have students create their own maps for other students to solve using loops.
- Draw a program on the board that uses several sets of repeated commands and have students take turns coming to the front to swap symbols for repeat loops.



This work 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: Getting Loopy

45 minutes

Overview

Grade Level: 1st Grade - 2nd Grade

In this **context-setting** lesson, students will learn to use loops to more easily communicate instructions by looking at the repeated patterns of a dance.

Purpose

At this point in the course, students should have developed comfort with programming a set of linear instructions. Frequently the linear set of instructions includes patterns that are repeated multiple times and as students want to write more complex and interesting programs, manually duplicating that code becomes cumbersome and inefficient. To enable students to write more powerful programs, we'll need to rely on structures that break out of the that single linear list. Loops allow for students to structure their code in a way that repeats. In this lesson, we will focus on identifying patterns in physical movement before moving back onto the computer to look for patterns in our code.

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

▶ **AP** - Algorithms & Programming

Agenda

Warm Up (5 minutes)
- **Vocabulary**
- **Repeat After Me**

Main Activity (15 minutes)
- **Getting Loopy**

Assessment (10 minutes)

Wrap-Up (15 minutes)
- **Reflection**

Extended Learning

Objectives

Students will be able to:

- Convert a series of multiple actions into a single loop.
- Repeat actions initiated by the instructor.
- Translate a picture program into a real-world dance.

Preparation

- Prepare to display the ***Worksheet** to students.
- Print one ***Assessment** per student.
- Make sure every student has a reflection journal.

Links

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

For the teachers

- **Getting Loopy** - Assessment Answer Key ▾ Make a Copy
- **Getting Loopy** - Slides (**Download**) ▾ Make a Copy

For the students

- **Getting Loopy** - Unplugged Video (**Download**)
- **Getting Loopy** - Worksheet ▾ Make a Copy
- **Getting Loopy** - Assessment ▾ Make a Copy

Vocabulary

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

Teaching Guide

Warm Up (5 minutes)

Vocabulary

Display: Show "Vocabulary" slide

Say:

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

Repeat After Me

Display: Show "Repeat After Me" slide

Model: Ask for a volunteer and have them stand.

- Instruct your volunteer to walk around the table (or their chair, or a friend).
- When they finish, instruct them to do it again, using the exact same words you did before.
- When they finish, instruct again.
- Then again.

Prompt: *Would it have been easier for me to just ask you to go around the table four times?*

Think:

- *What if I wanted you to do it ten times?*
- *How would you reword my instructions so that they were more efficient and I didn't have to repeat myself so much?*

Feel free to write your instructions down on a piece of scrap paper.

Share: Ask a few students to share their instructions with the class, pointing out how each approach has simplified the overall approach to giving instructions.

Remarks

> Today we're going to work on finding ways to make giving lots of instructions easier, especially when those instructions repeat themselves a lot. This will be really useful when we go back to the computers and have to write lots of instructions in our programs.

Main Activity (15 minutes)

Display: Show “Video: Getting Loopy” slide

Unplugged Activity **Video: Getting Loopy**

2 **Getting Loopy - Student Handout**

Getting Loopy

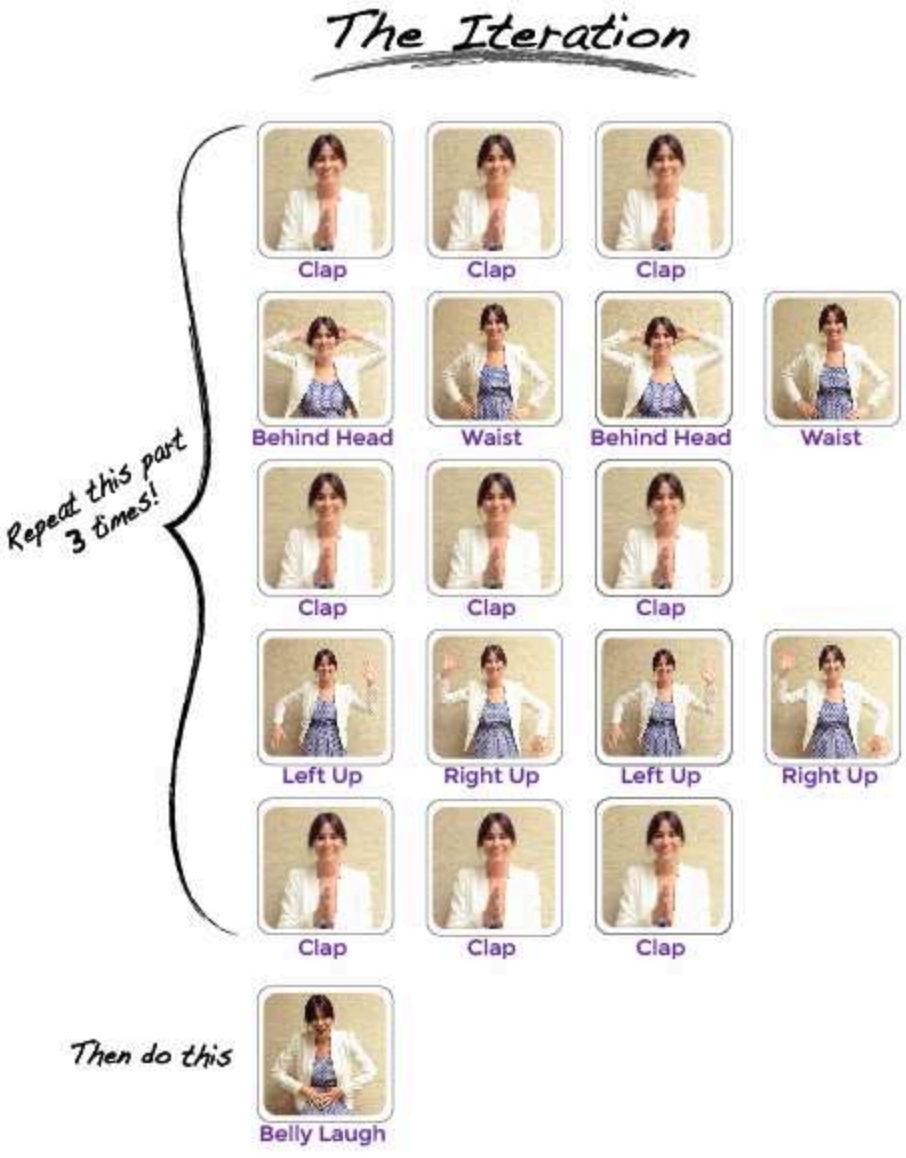
Display: Show “Decompose it!” slide

Remind students to break each section down into smaller steps.

Display: Show “Dance Party” slide

Introduce the main activity by letting the class know that we will be having a dance party. In order to have that party, we'll need to know what all of the steps in the dance are, and how many times we should do them.

Display: Show the ***Worksheet** so that all students can see it. Talk through the different sections of the dance as a class. Point out the section that repeats, in particular.



Model: Show the class what the entire dance looks like done at full-speed. Then run through the dance slowly, asking a different student to call out each line of instructions. Next, have the students perform the dance along with you, saying the instructions aloud as they get to each move.

Prompt: Ask students to work with a neighbor to find all of the sections of the dance that repeat.

Share: Ask a few students to share the repeating patterns that they found. As a class, talk through how you might rework the instructions to be even shorter by repeating those patterns.

Finally, help them understand a symbology for capturing these loops on their picture program, since the assessment will utilize this same method. Here is an example:



Teaching Tip

Looking for some good music? Here are some great places to find some:

- **Radio Disney**
- **Nick Radio**
- **Kidz Bop Radio**

Please be advised that some of these stations may display ads with third-party content. If you find that displayed ads are inappropriate, you may want to direct students to a different site, or research ad-blockers that can prevent this content.

Assessment (10 minutes)

Ending with an assessment sheet will help solidify this lesson for your students.

Distribute: Hand out the ***Assessment** to each student. Allow students to complete the activity independently after the instructions have been well explained. This should feel familiar, thanks to the previous activities.

Wrap-Up (15 minutes)

Display: Show "Assessment" slide

Do This: Have students:

- Circle the action that can be grouped into a loop
- Cross out the actions that aren't necessary anymore
- Write a number next to each circle to show how many times to repeat the action

Display: Present the vocab for this lesson, loop Ask the class to point out the main loop that was in the dance. Why do you think we call it a loop?

Flash Chat: What did we learn?

- Do you think it is easier to add more pictures to the screen or change the number of times we loop?
  - Would your answer be the same if we wanted to loop 100 times?

- Could we use these same loops with different dance moves?
- Do you know any dances that are done inside a loop?
- What was your favorite part about that activity?

Reflection

Display: Show "Reflection" slide

Prompts:

- What was today's lesson about?
- How did you feel during today's lesson?
- Draw a picture of you dancing today. Draw the loops that you did, like clapping three times.
- What else can you use a loop for?

Extended Learning

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

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!

This work 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: My Loopy Robotic Friends Jr.

45 minutes

Overview

Grade Level: 2nd Grade - 3rd Grade

This **context-setting** lesson builds on the initial "My Robotic Friends" activity, featuring larger and more complicated designs.

Purpose

This lesson serves as a reintroduction to loops, using the now familiar set of "robot" programming instructions. Students will develop critical thinking skills by looking for patterns of repetition in the movements of classmates and determining how to simplify those repeated patterns using loops.

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **AP** - Algorithms & Programming

Agenda

Warm Up (10 minutes)
- **Reflect**
- **Vocabulary**
- **Introduction**

Activity (30 minutes)
- **My Loopy Robotic Friends**

Wrap Up (5 minutes)
- **Reflection**

Extended Learning

Objectives

Students will be able to:

- Identify repeated patterns in code that could be replaced with a loop
- Write instructions that use loops to repeat patterns

Preparation

- Prepare a stack of 20 paper cups (or ***paper trapezoids**) for each group of 4 students.
- Display the ***symbols** where students can reference throughout the lesson.
- (Optional) Print out one ***Cup Stacking Ideas** per group of 4 students.

Links

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

For the teachers

- **My Loopy Robotic Friends Jr.** - Slides (**Download**) ▾ Make a Copy

For the students

- **My Robotic Friends** - Cup Spacing ▾ Make a Copy
- **My Robotic Friends** - Symbol Key ▾ Make a Copy
- **My Robotic Friends** - Cup Stacking Ideas ▾ Make a Copy
- **My Robotic Friends** - Paper Trapezoid Template ▾ Make a Copy

Vocabulary

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

Teaching Guide

Warm Up (10 minutes)

Reflect

Display: Show “Reflect” slide

Reflect: *In the previous lesson, which shapes repeated lines of code? What lines of code did they repeat?*

Vocabulary

Display: Show “Vocabulary” slide

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

Introduction

My Robotic Friends Review

Display: Show “Remember This?” slide

Goal: This review will refresh the students’ minds about how quickly programs for the "My Robotic Friends" activity can get intense.

Display: Show the ***Symbol Key** that we used in "My Robotic Friends". For each of the four symbols, ask students to show you what it looks like for a robot to follow that instruction.

My Robotic Friends

Unplugged

Symbol Key (Course B)



Pick Up Cup



Put Down Cup



Step Forward



Step Backward

Model: With the class together as a group, pull an easy puzzle from the ***Cup Stacking Ideas** and program with each other as a reminder of rules and terminology.

Display: Show "Now Try This!" slide

Next, pull a puzzle that's slightly harder, but also requires a lot of steps like the one below.



Volunteer: Ask a volunteer (or a group of volunteers) to come forward to help program this one on the board. If you make them stick strictly to the "no symbols other than those on the key" rule, it will probably take a while!

Display: Show "One More Time!" slide

Display: Now, bring up this image:



What is the reaction of the class?

Prompt: Give students the opportunity to brainstorm shorter ways to relay the code that they're about to create. (This bit can be skipped over if your students start saying things like: "Move forward 6 times." Since that will open the discussion about how to show "six times" with symbols.)

Once students have put together the idea of "repeating" code, give them the vocabulary around it. Make sure to share with them that often the terms "repeat something" and "loop something" are often used interchangeably.

Activity (30 minutes)

My Loopy Robotic Friends

Display: Show “Decompose it!” slide

Remind students to break each puzzle down into smaller steps.

Unplugged Activity **My Loopy Robotic Friends - Unplugged Activity**

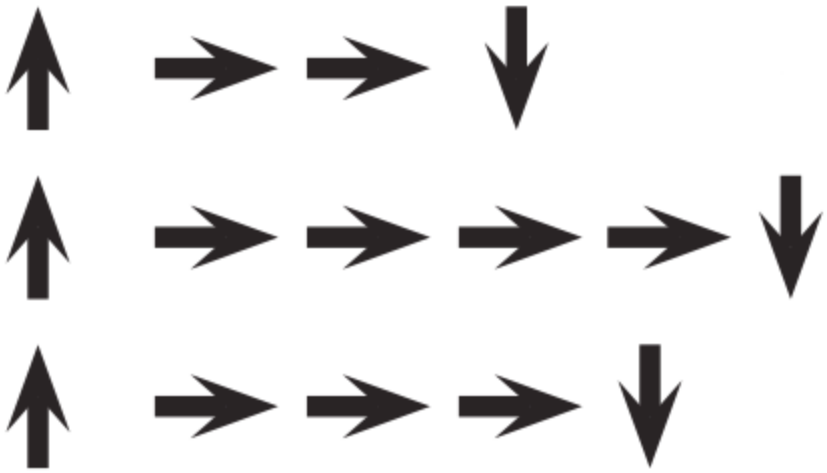
2 **My Loopy Robotic Friend - Student Resources**

Introduction and Modeling

Display: Show “My Loopy Robotic Friends” slide

Set Up: Have stacks of cups or cut paper trapezoids available for groups.

Display: Take the program from one of your previous cup stacks and display it for the class, or use the one below.



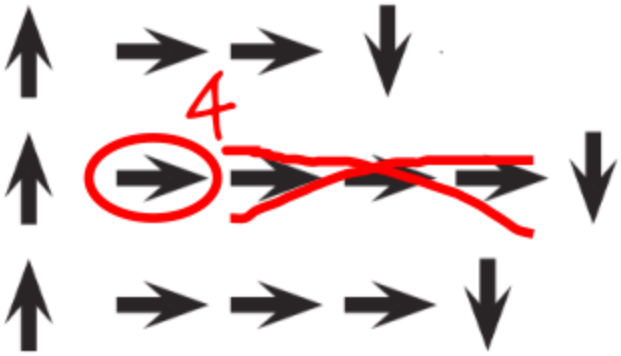
Think: Ask students to think quietly about where in this program they can find a pattern of instructions that repeat uninterrupted (one arrow repeated over and over).

Pair: Turn to a neighbor and share one of the repeating patterns you found.

Share: Ask a few students to share out the patterns they identified. Try to pull out different approaches to grouping patterns. For each pattern, ask students to identify how many times the pattern repeats.

Display: Show “Sample Answer” slide

Model: Using one of the repeating patterns that the class identified, model how to Circle the instruction or pattern that repeats, write the number of loops near that circle, then cross out the rest of the arrows.



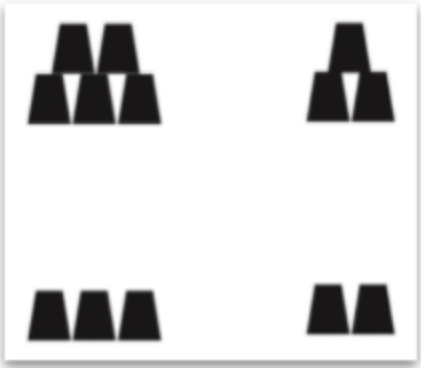
Repeat this until the entire program has been shortened, then re-write the program in a way where students can see how much simpler the resulting instructions are.

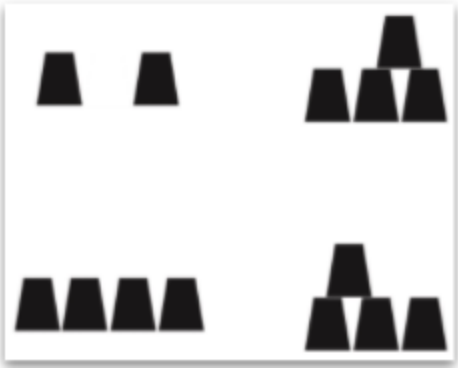
Looping Your Robots

Display: Show "Looping Your Robots" slide

Say:

- Once you are in your group, divide into partners.
- Look at Cup Stacking Ideas.
- With your partner, choose one design for your robot to do.





Display: Show "The "Program"" slide

Say:

- *Discuss how the stack should be built, using only arrows.*
- *Write down your algorithm on a piece of paper.*
- *Replace any repeated actions with a loop.*

Teaching Tip

Looking for Loops: Be sure to keep your eyes open for students using loops. Try to avoid correcting their overall algorithms or prescribing a solution, but feel free to direct students towards patterns that could be shortened by using a repeat circle.

Watch students as they run through the code. Are there any bugs? Use the debugging questions to help them find a solution.

- What does it do?
- What is it supposed to do?
- What does that tell you?

- Does it work at the first step?
- Does it work at the second step?
- Where does it stop working?

Display: Show "Running and Debugging Your Robots" slide

Say:

- *Now it's time to be a robot!*
- *Trade algorithms with the other partners in your group.*
- *Be on the lookout for bugs. If you see one, have the robot finish the instructions as best as it can.*
- *Discuss the bug with your partners and come up with a solution.*
- *Keep testing the algorithm until it works!*

Display: Show "Discuss" slide

Discuss: When all of the pairs have had a chance to run their programs, ask a few to share their solutions with the class. Use this opportunity to discuss how groups came up with different solutions to the same puzzle. In particular, you might ask of each program:

- *Did you identify any loops?*
- *Are there other ways those loops could have been written? How?*

Wrap Up (5 minutes)

Reflection

Display: Show "Reflect" slide

Reflect: *Is it easier to understand programs with loops, or programs written out longhand? Why?*

Extended Learning

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

- Have students draw their own cup stacking creations for someone else to code.
- Provide students with algorithms that utilize repeats, then have them expand the program back out to a full step-by-step version.

This work 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: The Big Event Jr.

45 minutes

Overview

Grade Level: Kindergarten - 1st Grade

In this **context-setting** lesson, the class will experience the concept of events through a game where they move or shout when you press buttons on a giant remote.

Purpose

Today, students will learn to distinguish events from actions. The students will see activities interrupted by having a "button" pressed on a paper remote. When seeing this event, the class will react with a unique action. Events are widely used in programming and should be easily recognizable after this lesson.

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **AP** - Algorithms & Programming

Agenda

Warm Up (15 minutes)
- **Introduction**

Main Activity (15 minutes)
- **The Big Event**

Wrap Up (15 minutes)
- **Reflection**
- **Assessment**

Extended Learning

Objectives

Students will be able to:

- Practice differentiating pre-defined actions and event-driven ones.
- Recognize actions of the teacher as signals to initiate commands.
- Repeat commands given by an instructor.

Preparation

- Prepare to project ***The Controller Image**.
- Print one ***Assessment** per student.
- Make sure each student has a reflection journal.

Links

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

For the teachers

- **The Big Event** - Assessment Answer Key ▾ Make a Copy
- **The Big Event Jr** - Slides (**Download**) ▾ Make a Copy

For the students

- **Feeling Faces - Emotion Image** - Resource ▾ Make a Copy
- **The Big Event** - Unplugged Video (**Download**)
- **The Big Event** - Assessment ▾ Make a Copy
- **The Big Event (Courses A, B)** - Controller Image ▾ Make a Copy

Vocabulary

- **Algorithm** - A list of steps to finish a task.
- **Event** - An action that causes something to happen.

Teaching Guide

Warm Up (15 minutes)

Introduction

Vocabulary

Display: Show "Vocabulary" slide

This lesson has one new and important vocabulary word:

- **Event** - *Say it with me: E-vent.*

An event is an action that causes something to happen.

A Series of Events

Display: Show "Patterns" slide

Let's see if we can find a pattern here:

- *When you flip a switch, the lights turn on.*
- *When you tap on a device, an app starts.*
- *When the alarm goes off, you get out of bed.*
- *When (event), (action).*

Display: Show "Events" slide

In computer science, **events** cause other actions to happen. Our new vocabulary word today is **event**. Normally when we hear "event" we think of:

- Field trip
- Competition
- Birthday Party

We're learning a new meaning for the word event today. Let's focus on events that cause other actions to happen like when flipping a switch causes the lights to turn on or pressing a button to make a character in a game move.

Main Activity (15 minutes)

Display: Show "Games have events in their programs" slide

Unplugged Activity **The Big Event - Unplugged Activity**

The Big Event

- *Do you remember helping the flurbs find fruit?*
  - *In that exercise, you knew in advance exactly where you wanted your flurb to end up, so you could make a program that took them from start to finish without any interruptions.*
  - *In most real programs, we can't do that because we want to have options, depending on what the user needs.*
    - *Say that I only want my character to move when my finger is on the screen of my phone. I would need to program the character to only move when I put my finger on the screen of my phone.*
    - *Putting my finger on the screen would then become an event that tells my character to move.*

Teaching Tip

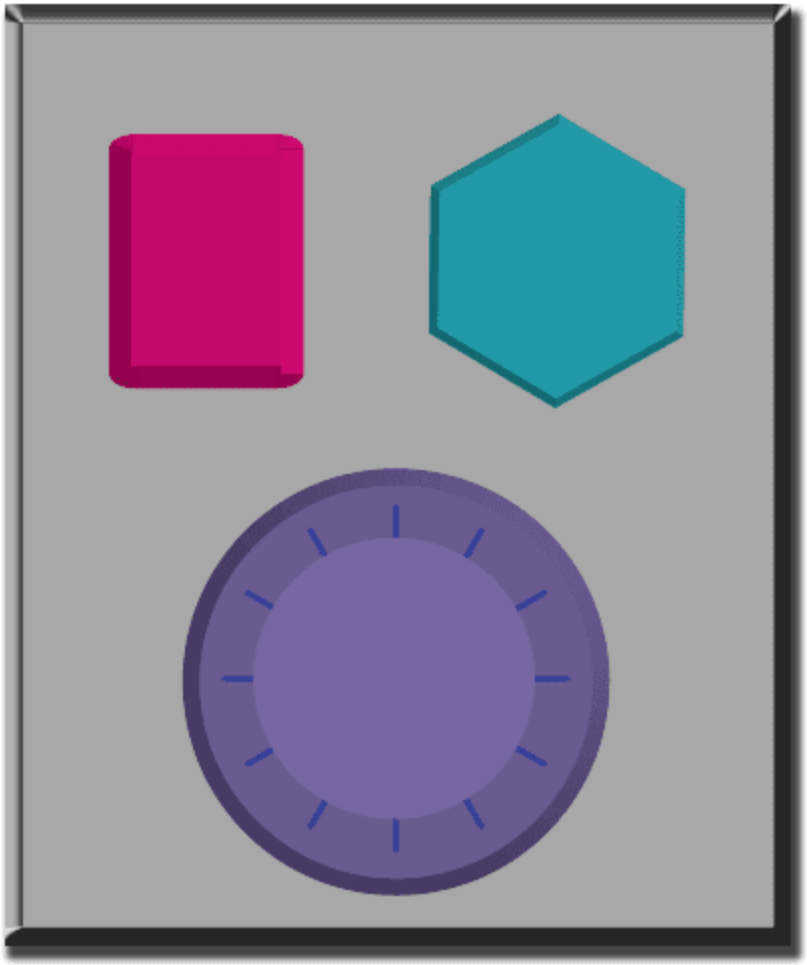
If your students seem confused, talk about their favorite games and all of the ways that they let the characters know what they're supposed to do. Point out how the game would be really boring if it ran from start to finish without any events required.

In earlier lessons, we created algorithms that allowed us to control a friend or flurb for several steps at a time. It was fun and useful, but what happens when you don't know everything that you want your friend to do in advance? This is where events come in!

Directions:

Display: Show "Programming Buttons" slide

- **Project *The Big Event (Courses A, B) - Controller Image** onto your classroom screen.



- Decide with your class what each button does. We suggest:
  - Pink Button -> Say “Wooooo!”
  - Teal Button -> “Yeah!”
  - Purple Dial -> “Boom!”
- Practice tapping the buttons on the screen and having your class react.
- Add some button sequences into the mix and have the students try to keep up with their sounds.
- Let your class know that every time you push a button, it is an event that lets them know what they are expected to do next.
- Have students practice working backwards. Give them 3-4 sounds and ask what buttons they would need to press to get the sounds to play in that order. For example:
  - What buttons would you need to press to get "Boom! Boom! Yeah!"?
- Get the class started on a planned task before interrupting them again with the buttons. We suggest:
  - Counting to 10
  - Singing “Old MacDonald”
- Once their plan is underway, interject button presses sporadically.
- Continue the blend until they understand the difference between actions that are guided by a plan and those that are event driven.

Display: Show “Decompose it!” slide

Remind students to break each puzzle down into smaller steps.

Wrap Up (15 minutes)

Reflection (10 minutes)

Display: Show "Reflection" slide

Discuss: Ask students to reflect on what they have learned through the following prompts:

- *What did we learn?*
- *What are some examples of events?*

Prompts:

- What was today's lesson about?
- Draw a face that shows how you felt about today's lesson in the corner of your journal page.
- Draw an event that caused an action today.
- Draw an action that was caused by an event that happened today.

Teaching Tip

Students should demonstrate understanding that events are something that can cause other actions to happen. Refer back to earlier examples from the lesson or encourage students to come up with their own.

Assessment (5 minutes)

Distribute: Hand out one ***Assessment** to each student and allow them to complete it independently after the instructions have been well explained. This should feel familiar, thanks to the previous activities.

Extended Learning

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

One Person's Event is Another One's Reaction

Assign each student an event to watch out for, and an appropriate reaction to that event. Chain the actions so that each child's reaction becomes an event that triggers the reaction of another student. Keep assigning until everyone has something to do and everyone makes someone react.

Eventopalooza

Break the class up into groups. Using the Events Controller, assign each group a different reaction to the same button. Do this for all three buttons, then watch the chaos!

This work 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: The Big Event

45 minutes

Overview

Grade Level: 2nd Grade - 3rd Grade

In this **context-setting** lesson, students will learn that events are a great way to add flexibility to a pre-written algorithm.

Purpose

Today, students will learn to distinguish events and actions. The students will see activities interrupted by having a "button" pressed on a paper remote. When seeing this event, the class will react with a unique action. Events are widely used in programming and should be easily recognizable after this lesson.

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

▶ **AP** - Algorithms & Programming

Agenda

Warm Up (15 minutes)
- **Reflect**
- **Introduction**
- **Vocabulary**

Main Activity (15 minutes)
- **The Big Event**
- **Optional Assessment (5 minutes)**

Wrap Up (15 minutes)
- **Reflection**

Extended Learning

Objectives

Students will be able to:

- Practice differentiating pre-defined actions and event-driven ones.
- Recognize movements of the teacher as signals to initiate commands.
- Repeat commands given by an instructor.

Preparation

- Print or display the ***Event Controller**.
- Print one ***assessment** for each student.

Links

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

For the teachers

- **The Big Event** - Assessment Answer Key ▾ Make a Copy
- **The Big Event** - Slides (**Download**) ▾ Make a Copy

For the students

- **The Big Event** - Unplugged Video (**Download**)
- **The Big Event** - Assessment ▾ Make a Copy
- **The Big Event (Course C)** - Event Controller ▾ Make a Copy

Vocabulary

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

Teaching Guide

Warm Up (15 minutes)

Reflect

Display: Show “Reflect” slide

Reflect: *When have you caused something to happen because of an action you took? For example, you flipped a switch, the lights turned on.*

Introduction

A Series of Events

Display: Show “Let’s see if we can find a pattern here” slide

Say: *Let’s see if we can find a pattern here:*

- *When you tap on a device, an app starts.*
- *When the alarm goes off, you get out of bed.*
- *When (event), (action).*

Display: Show “Next” slide

Say: *Normally when we hear “event” we think of:*

- *Field trip*
- *Competition*
- *Birthday Party*

We’re learning a new meaning for the word "event" today.

Vocabulary

Display: Show “Vocabulary” slide

This lesson has one new and important vocabulary word:

Event - An event is an action that causes something to happen.

Main Activity (15 minutes)

The Big Event

Unplugged Activity **The Big Event - Unplugged Activity**

2 **The Big Event - Student Resources**

Display: Show “Remember This?” slide

Say: *Do you remember guiding Red from "Angry Birds" to the pig in the Maze puzzles? You wrote a program that took Red from start to finish without any interruptions.*

Display: Show “Giving the User Options” slide

Say: *In most real programs we want more options. What if I only want my character to move when my finger is on the screen of my phone?*

Display: Show “Controlling the Game” slide

Say: *I need to program the character to only move when my finger is on the screen of my phone. Putting my finger on the screen would then become an event that tells my character to move.*

Display: Show “When Do We Use Events” slide

Say: *In earlier lessons, we created algorithms that allowed us to control a friend or other character for several steps at a time.*

It was fun and useful, but what happens when you don’t know everything that you want your friend to do in advance?

This is where events come in!

Teaching Tip

If your students seem confused, talk about their favorite games and all of the ways that they let the characters know what they're supposed to do. Point out how the game would be really boring if it ran from start to finish without any events required.

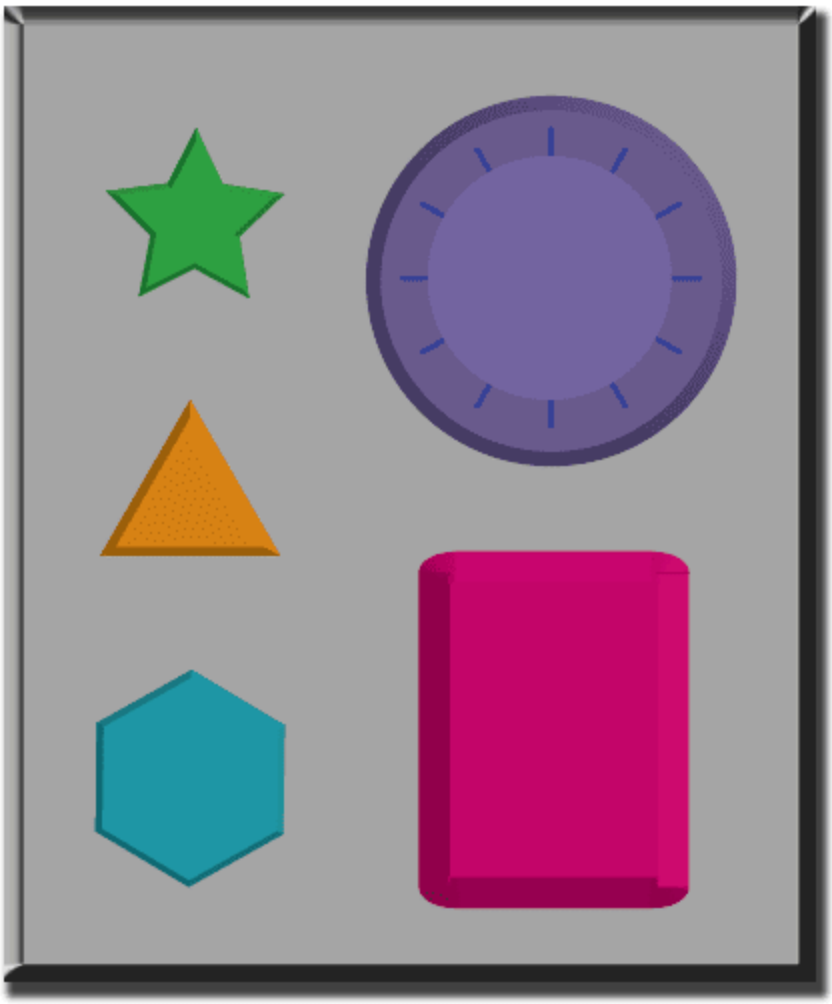
Display: Show “Decompose it!” slide

Remind students to break each puzzle down into smaller steps.

Display: Show “Event Controller” slide

Directions:

- Project the ***Event Controller** onto your classroom screen.



- Decide with your class what each button does. We suggest:
  - Green Star -> Clap
  - Orange Triangle -> Hands in the air
  - Blue Hexagon -> Hands on hips
  - Purple Circle -> Twist
  - Pink Rectangle -> Stomp
- Practice tapping the buttons on the screen and having your class react.
- Add some button sequences into the mix and have the students try to keep up with their actions.
- Let your class know that every time you push a button, it is an event that lets them know what they are expected to do next.
- Have students practice working backwards. Give them 3-4 sounds and ask what buttons they would need to press to get the sounds to play in that order. For example:
  - What buttons would you need to press to get "Twist, Twist, Clap, Stomp"?
- To make things more challenging, get the class started on a separate task, then begin interrupting them again with the buttons. We suggest:
  - Counting to 10
  - Singing “Old MacDonald”
- Practice until students seem comfortable with the idea that each button press causes them to perform an action.

Optional Assessment (5 minutes)

Display: Show “Assessment” slide

- Hand out ***Assessment** and allow students to complete the activity independently after the instructions have been well explained.
- This should feel familiar, thanks to the previous activities.

Wrap Up (15 minutes)

Reflection (10 minutes)

Display: Show "Reflect" slide

Reflect: *What are some examples of events?*

Teaching Tip

Students should demonstrate understanding that events are something that can cause other actions to happen. Refer back to earlier examples from the lesson or encourage students to come up with their own.

Extended Learning

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

One Person's Event is Another One's Action

- Assign each student an event to watch out for, and an appropriate reaction to that event. Chain the actions so that each child's action becomes an event that triggers the reaction of another student. Keep assigning until everyone has something to do and everyone makes someone react.

Eventopalooza

- Break the class up into groups. Using the Events Controller, assign each group a different reaction to the same button. Do this for all three buttons, then watch the chaos!

This work 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: The Right App

45 minutes

Overview

Grade Level: 1st Grade - 2nd Grade

In this **exploratory** lesson, students learn about different computing technologies that exist. They also compare how people lived before the invention of apps and computers with how people live now. The lesson concludes with an activity where students empathize with several fictional smartphone users to help them design the "right app" that addresses their needs.

Purpose

Today, computing is more accessible than ever before. People from all walks of life use software on many different devices, particularly smartphones, for information, communication, and entertainment. Because people have such diverse experiences and needs, it is important for budding computer scientists to empathize with people and identify solutions with them in mind.

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **CS** - Computing Systems
- ▶ **IC** - Impacts of Computing

Cross-Curricular Opportunities

Next Generation Science Standards

- ▶ **ETS** - Engineering in the Sciences

Agenda

Warm Up (10 minutes)
- **Vocabulary**
- **Question of the Day**
- **Fill in the Blank**

Activity (30 minutes)
- **What is an App?**
- **Station Activities**

Objectives

Students will be able to:

- Compare daily life before and after the invention of certain technologies
- List several different examples of smartphone apps and computing technologies.

Preparation

- Read through the speaker notes in the ***slide deck**.
- Cut out the Stations Activities resources in advance
- Prepare enough sketching/drawing supplies for all students.
- Make sure each student has a reflection journal.

Links

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

For the teachers

- **The Right App** - Slides (**Download**)
  Make a Copy
- **The Right App - Stations Activities** - Resource (**Download**)
  Make a Copy
- **[KEY] The Right App - Student Guide** - Answer Key (**Download**)
  Make a Copy
- **[Optional] The Right App - Stations Activities - Words** - Resource (**Download**) Make a Copy

For the students

Design an App

Wrap Up (5 minutes)

Reflection

- **The Right App - Student Guide** - Activity Guide (**Download**)
  ▾ Make a Copy

Vocabulary

- **App** - An application that a person downloads to a phone or tablet.
- **Computer** - An electronic device that can input, output, store, and process information

Teaching Guide

Warm Up (10 minutes)

Vocabulary (1 minutes)

Display: Show “Vocabulary” slide

Question of the Day (1 minutes)

Display: Show “Question of the Day” slide

Say: *What was life like before and after the invention of different types of computers?* We will revisit this question at the end of the lesson.

Fill in the Blank (8 minutes)

Display: Show “Fill-in-the-blank” slide

Say: Look at the two pictures on the top of the screen. Which of the three computing technologies at the bottom would come next in the timeline. Fill in the blank: a self-driving car, a robot dog, or a laptop? Explain your answer.

Display: Show “Think about it” slide

Say: These pictures in the timeline represent different forms of transportation. People used to travel by horse and carriage before the invention of the car. Now people mostly travel in cars. Did you know some car companies just invented a car that can drive itself? It’s called a self-driving car or autonomous vehicle.

Discuss: What problem did each of these inventions solve?

Discussion Goal: Students should understand that each of these images represents a technological advancement in travel. As our technology advances, we are able to create more efficient means of transportation. The horse and carriage replaced walking or traveling on horseback and being exposed to the elements. Cars replaced having to rely on animals for transportation. Self-driving cars replace the need for a human driver with the goal to be safer than human-driven cars.

Display: Show “Fill-in-the-blank” slide

Say: Look at the two pictures on the top of the screen and see if you can figure out which of the 3 apps at the bottom would come next. Would it make sense to fill in the blank with a texting app, a maps app, or a music app?

Display: Show “Think about it” slide

Discuss: What can a music app do that a radio or piano can't do? Why might someone choose to play music on a piano and not a music app?

Discussion Goal: Students should understand that a music app represents an advancement in computing technology. We now have the ability to play any song we want on demand from our phone or computer. A radio can only play stations that are currently broadcasting. And a piano can only play piano music (and only if the person knows how to play).

Someone might choose to play music from a piano when they want to play music themselves or for a live audience. They also might prefer the sound of piano music or wish to learn a new instrument.

Activity (30 minutes)

Unplugged Activity **The Right App - Unplugged Activity**

2 **The Right App - Student Resource**

What is an App? (5 minutes)

Display: Show “What’s an app?” slide

Discuss: Which apps do you recognize? What do people use these apps for?

Discussion Goal: Students should be able to identify some of the following apps: Calendar, Weather, Maps, Photos, Clock, YouTube, Safari, Camera, Calculator, Messages, Music, FaceTime/video calls.

Have students choose one app to explain how it works or when people might use it. For example, people would use a calendar app to keep track of important events, such as going on a trip or meeting up with a friend.

Display: Show “Different examples of apps” slide

Say: Here are some more examples of apps that you might recognize. It will be important to know how these apps work for the first station. Are there any apps that you have a question about?

Do this: Answer any questions students might have about apps before moving on.

Station Activities (10 minutes)

Display: Show “Station 1: Matching” slide

Say: At Station One, you’ll be playing a matching game. Look at the pictures on the left and figure out what they represent. Then look at the apps on the right and find the app that has replaced the activity on the left.

[Optional] Do This: In the teacher resources section, there is a set of matching cards with words instead of images. If students are older and would like more practice reading, use the word matching cards instead.

Display: Show “Station 2: Sorting” slide

Say: At Station Two, you’ll be sorting the images on the right into one of two categories: Computer or Not a computer. Make sure to look at the definition of computer at the bottom of the screen.

Do This: Do an example as a class by asking students to classify one of the images on the right. Then drag it under the corresponding category on the left.

Discussion Goal: Some items might be easily sorted, whereas others like the calculator might inspire some discussion among students.

Do This: If students have extra time at this station, encourage them to play a game of charades where they have to act out one of the images and have their classmates guess what they're acting out.

Design an App (15 minutes)

Display: Show "Choose a character and design their app" slide

Say: The final activity will be done in your student guide. Choose one of the following characters and design an app for them. Here are the characters:

- **Adventurous Hedgehog** wants a Travel App to keep track of all the countries they visit
- **Peaceful Turtle** wants a Yoga App to learn new yoga poses
- **Reader Rabbit** wants a Book App to choose a new book to read
- **Daydreamer Fox** wants a Dream App with a place to write down dreams

Wrap Up (5 minutes)

Reflection (5 minutes)

Display: Show "Reflection" slide

Discuss: Choose one: Think about an app you use. What would you use instead if this app didn't exist? What type of app or computer technology do you wish existed?

Discussion goal: Have students consider what their lives would be like if they didn't have access to an app (or other computing technology) that they use on a regular basis. Get them to consider how different their life might be. Then have them brainstorm a computing technology or app they wish existed. Encourage them to think of ways that an app could make their life easier.



This work 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: Designing for Accessibility

45 minutes

Overview

Grade Level: 4th Grade - 5th Grade

In this **exploratory** lesson, students will learn about accessibility and the value of empathy through brainstorming and designing accessible solutions for hypothetical apps.

Purpose

Through learning about accessibility, students recognize the impacts of computing beyond their own lives. Accessibility might not seem like a relevant CS topic, but creating technology that is accessible for underserved users helps make tech better for everyone else as well.

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **AP** - Algorithms & Programming
- ▶ **CS** - Computing Systems
- ▶ **IC** - Impacts of Computing

Agenda

Warm Up (5 minutes)
Context-Setting

Main Activity (35 minutes)
Designing for Accessibility

Wrap Up (5 minutes)
Reflection

Objectives

Students will be able to:

- Describe the impact of mobile apps on the modern world.
- Explain why accessibility is an important part of designing an app for users.
- Improve upon an existing app design by addressing the accessibility needs of users.

Preparation

- For context, read about the ***types of disabilities**.
- Read through the speaker notes in the ***slide deck**.
- Prepare enough sketching/drawing supplies for all students.

Links

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

For the teachers

- **Designing for Accessibility** - Slides (**Download**) ▾ Make a Copy
- **Types of Disabilities** - Resource

Teaching Guide

Warm Up (5 minutes)

Context-Setting

Display: Show "Reflect" slide

Reflect: *Ask students to define the term "app" in their own words.*

- *What is an app?*
- *What kinds of apps exist?*
- *When do we use apps?*

Discussion: Students should understand that apps can address a variety of needs. For example, apps can be used to request rides, recommend movies, or communicate with friends and family. Many people have access to a mobile device (e.g., smartphone) which makes it possible to use apps just about anywhere.

While discussing, show screenshots or examples of apps on your smartphone or tablet if possible.

Ensure that all students are on the same page regarding this topic by introducing key terms and concepts with the remarks below.

Display: Show "What are apps?" slide

Say:

- *In this course, we have been learning how to code. Code is used to create computer applications. Games, web browsers, media players... the list of applications goes on and on!*
- *App is short for application. Usually, apps more specifically refer to applications running on small computers called mobile devices.*
- *Smartphones and tablets are examples of mobile devices.*

Introduce accessibility with the remarks below.

Display: Show "What is accessibility?" slide

Say:

- *As apps become used by more and more people, app developers often come across issues that they had not considered, likely because they have never experienced those issues in their own lives.*
- *Many of these issues are related to accessibility. In the world of computer science, accessibility is about creating technology for people with disabilities.*
- *Disabilities include physical, auditory, visual, and many others.*
- *Disabilities are diverse and impact people in different ways. For example, someone who is slightly visually impaired might wear glasses. Someone who is severely visually impaired might be considered "blind" and will need more assistance.*

Transition: Tell students you'd like to introduce them to a few kids who are just like them even though they have disabilities.

Main Activity (35 minutes)

Designing for Accessibility

2 **Designing for Accessibility - Student Discussion**

Scenarios (10 min)

Display: Display each scenario from the Designing for Accessibility scenarios. While displaying a scenario image, read its accompanying script to the class (see below). Discussion questions are included in relevant slides.

Designing for Accessibility Scenarios

Display: Show "Meet Wei" slide

- Wei is a talented kid who LOVES music. She plays piano and sings very well!
- Wei is visually impaired. While she is not completely blind, it is very difficult for her to see things, even if they are near her eyes.
- Wei has had to learn piano differently from kids who aren't visually impaired, but that hasn't stopped her from becoming a great musician!
- Wei recently began creating her own music. She is very proud of it! She would like to record and share her music with her friends.

Display: Show "Introducing Grammaphone!" slide

- Grammaphone is a popular new app that allows people to upload and share their own music with friends.

Discuss: What do you think about this app? How do you think this app works?

Display: Show "Wei and Grammaphone" slide

- Wei needs to use Grammaphone to share her music. After all, it's the most popular music-sharing app!
- However, she's having a difficult time using the app. Why do you think that is?

Display: Show "Meet Roger" slide

- Roger is an energetic kid who loves video games! He especially likes playing sports games with his friends.
- Things have been hard for Roger lately, though. A few years ago, Roger's hand had to be "amputated" (surgically removed) due to an accident.
- Thankfully, computer scientists, engineers, and doctors are working hard to provide Roger with a cool-looking robotic "prosthetic hand," which will be awesome!
- But for now, Roger wants to play sports video games with his friends!

Display: Show "Introducing Football Frenzy!" slide

- Football Frenzy is a new mobile game that everyone loves!
- It has cool graphics, awesome sound effects, tight controls, and amazing action!

Discuss: What do you think about this app? How do you think this game is played?

Display: Show "Roger and Football Frenzy" slide

- All of Roger's friends play Football Frenzy. It's connected online so that you can play it anywhere!
- Roger wants to play Football Frenzy with them, but he is having a difficult time. Why do you think that is?

Display: Show "Designing for Accessibility" slide

- Neither of these apps was designed with people like Wei and Roger in mind.
- When it comes to these apps, what specific features do you think Wei and Roger wish for?

- Let's redesign these apps!

App Redesign Activity (25 min)

Display: "Your turn!" slide

The final scenario allows students to sketch a redesign of one app from the scenarios ("Grammaphone" for Wei or "Football Frenzy" for Roger).

Students should primarily focus their efforts on making their redesigned app more accessible for either of them, along with any other accessibility features they can think of. Students can design as many screens for the app as they wish, and they do not need to make their app look anything like the original examples.

If there is time, students may redesign apps for both characters.

Distribute: Pass out crayons/pencils/etc. to each student as they open their reflection journals.

Display: "Share Out" slide

Share: After designing, students share their app design with a neighbor, or if there is time, some students can share with the whole class. Each student should answer how their app design addresses the need of the user in question.

Wrap Up (5 minutes)

Reflection

Display: "Reflect" slide

Reflect:

- *What are real-world examples of how apps and other technology can be made more accessible?*
- *How can making something more accessible also make it better for others?*

Discuss: Discuss real-world examples of how apps and other technology can be made more accessible. Ask students how they think making something more accessible can also make it better for others. One example is wheelchair ramps near building entrances. These were originally made for people with wheelchairs, but they are also convenient for people with baby strollers, dollies, and other wheeled equipment.



This work 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 12: Digital Sharing

55 minutes

Overview

Grade Level: 4th Grade - 5th Grade

This **exploratory** lesson helps students understand the challenges and benefits of respecting ownership and copyright, particularly in digital environments.

Purpose

Students will soon be creating projects to share and most of these projects will contain either code or imagery that students did not create themselves. This lesson is here to show students the proper way to handle the use of content that is not their own.

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **AP** - Algorithms & Programming
- ▶ **IC** - Impacts of Computing

Agenda

Warm-Up (Optional) (15 minutes)
- **Write a Character Sketch**

Main Activity (30 minutes)
- **Digital Sharing**

Wrap-Up (10 minutes)
- **Reflection**

Extended Learning

Objectives

Students will be able to:

- Interpret ethical sharing of copyrighted material vs. sharing that is not ethical.
- Understand their own rights regarding materials that they have created

Preparation

- Locate the ***Copyright Sharing video** in "Links" below
- Download and review the complete ***Digital Sharing Lesson Plan** from Copyright and Creativity
- As the teacher, create a piece of art for the lesson (picture, song, slideshow, etc.)
- Make sure every student has a reflection journal.
- You will need a tablet or smart phone to replicate the sharing of that item

Links

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

For the teachers

- **Digital Sharing** - Slides (**Download**) ▾ Make a Copy
- **Digital Sharing Lesson Plan** - Resource ▾ Make a Copy

For the students

- **Digital Sharing Ethics (Video)** - Video

Vocabulary

- **Copyright** - the exclusive legal right to print, publish, perform, film, or record literary, artistic, or musical material, and to authorize others to do the same

Teaching Guide

Warm-Up (Optional) (15 minutes)

Display: Show "Reflect" slide

Reflect: *What are examples of projects or artwork that people share online?*

The following writing exercises are designed to create context, help students engage with the topic, and prepare them for the lesson discussion.

Display: Show "Digital Sharing" video slide

Have students watch the one-minute ***Digital Sharing video**. You may need to play it two or three times.

Write a Character Sketch

Display: Show "Write a Character Sketch" slide

Say: *Write a character sketch about one or both of the characters in the video. Be as creative as you can. There are no wrong answers. Give these characters a life of their own, whatever you want it to be.*

Prompt with questions:

- *Who is your character?*
- *What is their name?*
- *Who are their friends?*
- *How long have they known each other?*
- *Who are the people in your character's family? What are they like?*
- *What's your character's backstory?*
  - *Where do they live?*
  - *Where did they used to live?*
  - *What exciting thing might have happened to them back in kindergarten, first grade, etc.?*
- *What is your character looking forward to?*
- *What is your character afraid of?*

Teaching Tip

These exercises may also be done orally as a class discussion before the copyright lesson. Write the story or character sketch on the board as students contribute ideas.

Vocabulary

Display: Show "Vocabulary" slide

- **Copyright** - the exclusive legal right to print, publish, perform, film, or record literary, artistic, or musical material, and to authorize others to do the same.

Main Activity (30 minutes)

Digital Sharing

Unplugged Activity **Digital Sharing - Video**

2 **Digital Sharing - Discussion Questions**

Ethical Sharing

Display: Show "Ethical Sharing" slide

Activity: Have all of your students stand up. Begin reading the list of ways to create content below. Instruct students to sit down as soon as they can answer "Yes" to one of the prompts.

- *Have you ever made a video (on a camera, phone, iPad, or computer) and sent it to a family member or posted it online?*
- *Have you taken a photo and sent it to a family member or posted it online?*
- *Have you created a piece of art to share with your family and friends?*
- *Have you made up a song to make your friends laugh? Or a sad song to make them cry?*
- *Have you written a poem for your mom or dad on their birthday?*

Keep asking similar questions until the entire class is seated.

Display: Show "Discuss" slide.

Discuss:

- *How did it feel to produce something creative?*
- *How did you feel when you were able to share your creation with others?*
- *How do you feel when you view or listen to other people's creations?*

Encourage all responses.

Help students feel the joy of creating something. Creating can be a lot of hard work, but it is one of the most rewarding things we do. Sharing what we create is fun, and it can encourage more creativity and art. As we get older, we have more and more opportunities to share our work and explore media and art that other people have created. We want to make sure we are always fair when using others' art and creative work.

Display: Show "Copyright" slide

Say: *Remember, copyright protects all kinds of creative work so that artists/creators can get paid for their effort. This includes, original writing (stories), art, photographs, audio, images, music, song lyrics, even the doodle you drew on your napkin at lunch. It doesn't matter if it was created by a famous artist or by you. When you make an original work, you get to decide who can:*

- *make copies*
- *distribute copies*
- *display or perform the work in public*
- *make spin-offs; we call these derivatives (for example, like a book being made into a movie)*

These rights are given to artists and creators to encourage them to make even more creative work.

Display: Show "Think-Pair-Share" slide

Think: Have students think back to the character sketch they created at the beginning of class. Inform them that they created an original work with legal protection. Congratulations!

Pair:

- *How might you know if something is copyrighted?*
  - The circle © indicates copyright, but copyright protection exists even without the symbol. Creators have ownership over their work, unless they sell it to someone else.
- *Where have you seen the copyright symbol?*
  - At the front of books, in movies, on images, posters, etc.

Share: Encourage students to share their answers with the class.

Demonstrate: Show students how to draw a copyright symbol and write the year next to it. When you make something creative like this, it's automatically protected by copyright, even without the copyright symbol.

Display: Show "Discuss" slide

Discuss:

- *How does it feel when you share your things with someone else?*
- *What does it feel like when someone takes your things and shares them without your permission?*

Say: *The same principles of respect and fairness apply when we share our work or someone else's work online.*

To Share or Not to Share?

Demonstrate: Pull out the instructor's creative work: picture, song, story, video, recipe.

Display: Show "Digital Sharing Ethics" video

Watch: Have students watch the one-minute ***Digital Sharing Ethics** video. *You may need to play it two or three times.*

Discussion Goal: Students should understand when it is okay and when it is not okay to share our digital materials and those of others.

Not Okay to Share

Display: Show "To Share or Not to Share" slide

- *Have you ever transferred songs to your friend's MP3 player? Is that OK?*
  - If it's a song you hear on the radio, it's most likely protected by copyright and NOT OK to share, copy, and give away.
- *What if your friend invites you to his house to watch a movie that just came out on DVD? This is one of your favorite movies. You want it on your phone, so you can watch it whenever you want. So, you take out your phone and record the movie. Is this a fair way to get a copy of the movie?*
  - No. This is not OK to share/copy. Why? Because you don't own the right to make a copy and give it away.
  - *How else could you get an authorized (legal) copy of the movie for your phone?*
    - iTunes or Amazon sell movies legally.

Okay to Share

Display: Show next "To Share or Not to Share" slide

- *I made this. That means I own it. I think I'm going to share it. I'm going to take out my phone (iPad, camera, etc.) to get a picture. I think I want to share it on my blog where I might make some money advertising on[Use the site of your choice.] Is this fair?*
  - Yes, this is OK to share because I made it — I own the copyright.
- *What about a song I wrote? Can I share that? . . . Who gets to decide?*
  - "That's right, I do."
- *What will happen when I share it?*
  - Take responses: "It's fun, . . . I'll get a bunch of 'likes' . . . People will want to use it for mashups."
- *Let's say you draw a picture to sell at a school art show. The money from the art sale will go to buy new library books. Is this a good share . . . is it ethical?*
  - *Yes.*
  - *Why is this share OK?*
    - Take responses: "It's yours!", "You made it. You own it. You can choose to share it."

Discuss: *What did you think of that? How do you think you would feel if you wrote a song and people shared it without asking for your permission? When we share digital files by:*

- sending pictures or songs through email
- copying songs from our MP3 player to our friend's computer
- copying a movie from a DVD to all our friends' computers

That's not just sharing, it's making new copies!

Teaching Tip

Demonstrate the following by handing a book to a student:

Sharing a digital file is different from face-to-face sharing. If I hand you my book to share it with you, you have the book and I don't—that's sharing. If I hand you my iPod, so you can listen to my music, that is sharing. If I share a digital file with you—like a song or a movie or computer game—we both end up with the file. In that case, we made a copy. If I copy my songs for you to put on your iPod, that is not sharing—it's copying. Making copies of copyrighted work hurts the artist/creators. In addition, P2P sharing and torrent sites can put your computer at risk for bad stuff: malware, ads, and worse.

Display: Show "Discuss" slide

Discuss:

- *If you were one of these characters in the video, what could you do to share fairly?*
- *What about other sharing situations?*
- *What other ethical considerations are there?*

Some acceptable responses:

- Send your friend a link to the artist's YouTube channel where she can listen to the song.
- Help your friend buy the song from an online store that you can trust because it's used by a large online community, like iTunes or Amazon.

Ask yourself: Who owns this? Do I have permission to share? Do I have a right to make a copy? Am I being fair to everyone involved?

Wrap-Up (10 minutes)

Reflection

Remember, copyright is a protection given to writers and artists for a limited time to let them receive payment for their work. It's intended to foster more creativity. As we share and use, we need to respect each other's work and the laws of copyright. Just because we own a copy of something does not mean we have the right to make more copies to give or sell to other people. Copyright gives us some protection over how our art will be used and shared by others.

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.

Display: Show "Reflect" slide

Reflect:

- *Give an example of a way that you have seen other creators share or remix someone's work.*
- *Do you believe that was fair? Why or why not?*

Extended Learning

Please be sure to visit **Copyright & Creativity** to find more lessons on digital sharing and creative rights.



This work 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 13: Picturing Data

45 minutes

Overview

Grade Level: 2nd Grade - 3rd Grade

Students will collect data from a Play Lab project and visualize it using different graphs in this **exploratory** lesson.

Purpose

Computers were created to help process data. There is an increasing amount of data in the world, so being able to read and analyze it is important. This lesson is here to make sure students have the basic experience of collecting, visualizing, and analyzing a simple set of data.

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **DA** - Data & Analysis

Cross-Curricular Opportunities

Common Core Math Standards

- ▶ **MD** - Using Probability To Make Decisions

Agenda

Warm Up (5 minutes)
- **Reflect**
- **Vocabulary**
- **Introduction**

Main Activity (35 minutes)
- **Picturing Data**

Wrap Up (5 minutes)
- **Reflection**

Objectives

Students will be able to:

- Collect and record data about quantities of real objects, or characters on a screen
- Create a bar graph and pie chart to represent simple data.
- Make comparisons between data visualizations made by others and use them to make a prediction.

Preparation

- Print out one ***worksheet** for each student.
- Try today's lesson on Code Studio. This is meant to be used as a tool for today's activity. Be prepared to project it to the class, or otherwise allow students to visit it on their own computers.

Links

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

For the teachers

- **Graphing Data from Play Lab** - Worksheet ▾ Make a Copy
- **Picturing Data** - Slides (**Download**) ▾ Make a Copy

Vocabulary

- **Data** - A collection of information

- **Graph** - A picture or a diagram that represents data or values in an organized manner

Teaching Guide

Warm Up (5 minutes)

Reflect

Display: Show “Reflect” slide

Reflect: *Let’s say you want to find out the most popular ice cream flavors among your classmates. How would you figure this information out?*

Vocabulary

Display: Show “Vocabulary” slide

- **Data** - A collection of information
- **Graph** - A picture or a diagram that represents data or values in an organized manner

Introduction

The Need for Visualization

Display: Show “Favorite Season” slide

Say: *On the count of 3, tell me your favorite season. 1, 2, 3*

Goal: Students should all answer at once. Students should see first-hand how this is an inefficient, or even inaccurate way of collecting data.

Display: Show “Discuss” slide

Discuss:

- *Can we tell which season most of the class likes?*
- *Why do you believe that?*
- *What could we do to be more certain?*

Take ideas from the class, but lead them towards writing the data down using tally marks on a board or chart paper.

Display: Show “Tally Marks” slide

Remarks

> *Tally marks like these help us to keep track of everyone's answers, which is great. It still takes time to count them to know which option was chosen by the most or fewest students. Imagine if someone showed us tally marks for the entire school. It would take some time to count them! It is much easier to see the data written out as numbers rather than tally marks.*

Main Activity (35 minutes)

Picturing Data

Display: Show "Level 1 - Picturing Data" slide .

Distribute: Pass out the ***handout** along with pencils and coloring instruments of some kind. Direct students to Code Studio or project the level on the board.

1 **Picturing Data**

Teaching Tip

This program generates a random assortment of pets each time it is run. If you are projecting the program for the whole class to use, consider running it multiple times and asking different groups of students to track the numbers each time so that there will be variety in their responses for later discussion.

This activity is broken into two parts.

Display: Show "Counting Data" slide

On the first page students use tally marks to keep track of how many times each pet appears in the program.

Display: Show "Graphing Data" slide

On the second page they use a variety of graphs to visualize the data they collected.

Display: Show "Share" slide

Share: After finishing, encourage students can share their results with a neighbor and look for similarities and differences. When the whole class is ready, bring everyone together.

Display: Show "Discuss" slide

Discuss: Ask the class to discuss some of the following questions (Bolded questions appear in the slides. Feel free to add any others that would work for you):

- **Which graph shows you the most useful information? Why?**
- Which pets did you see the most? Is that true for everyone?
- **What do you think would happen if this program was run for 100 animals? 1000?**
- **What questions could you answer with your data?**

The goal of this discussion is to get students thinking about the usefulness of data in identifying or answering questions, and to start them thinking about why they might want to use different kinds of visualizations (graphs in this case).

Summary

Display: Show "Summary" slide

Remarks

> *Collecting and graphing data can help us ask and answer interesting questions.*

> *For example, if you play soccer, you might want to know if goals are scored more often from the left, the right, or center. What data would you want to collect to answer that question? How would you show it?*

Wrap Up (5 minutes)

Reflection

Display: Show "Reflect" slide

Reflect: *What is one question that you would like to answer? What data would you want to collect to answer it?*



This work 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 14: Pattern Block Data

45 minutes

Overview

Grade Level: 2nd Grade - 3rd Grade

In this unplugged lesson, students will practice storing, searching, retrieving, modifying and deleting the pattern blocks to simulate what computers do with data.

Purpose

In this Unplugged lesson, students will use pattern blocks inside a box as an analogy for data inside a computer. Students will practice storing, searching, retrieving, modifying and deleting the (data) pattern blocks. At the end of the lesson, students will create various bar graphs based on the attributes of their (data) pattern blocks.

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **DA** - Data & Analysis

Common Core Math Standards

- ▶ **MD** - Using Probability To Make Decisions

Agenda

Warm Up (10 minutes)
- **Which one doesn't belong?**

Main Activity (25 minutes)
- **Sorting Data**
- **Graphing Data**

Wrap Up (10 minutes)
- **Reflection**

[Optional] Assessment
- **Check For Understanding**

Objectives

Students will be able to:

- Classify and sort data into predefined categories based on different attributes
- Create bar graphs to present data
- Search for and accurately retrieve data based on specific criteria

Preparation

The data analogy can be made with various objects. Select an alternate from the list below if you don't have pattern blocks:

- Toy cars
- Toy animals
- Letter tiles
- Counting bears
- Unifix cubes
- Buttons or beads
- Other small items with sortable attributes

Links

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

For the teachers

- **CSF - Course C - Slides** - Slides (**Download**) ▾ Make a Copy
- **Pattern Block Data** - Slides (**Download**) ▾ Make a Copy

Vocabulary

- **Data** - A collection of information

- **Delete** - To remove or take something away
- **Modify** - To change or make something different
- **Retrieve** - To get back something you stored or put away
- **Search** - To look for something you need
- **Sort** - To put things in groups
- **Store** - To keep something in a safe place

Teaching Guide

Warm Up (10 minutes)

Which one doesn't belong? (10 minutes)

Display: Show "Warm Up" slide

Discuss: Which one doesn't belong?

Discussion Goal: Students discuss the attributes of the object in each square. The teacher labels each square with its shape, color and what it is. Students form opinions about which one doesn't belong. There is no right answer. The focus is on students explaining their thinking and justifying their choice.

Remarks

> Which one doesn't belong? I'm going to give you some silent think time. Show a thumbs up if you have an idea.

Do This: First, have students turn and talk to a neighbor about which one doesn't belong. Provide a sentence frame to encourage students to justify their choice. "The ____ doesn't belong because ____." Then, have students share their ideas with the whole group. Make sure to call on students so that each square is discussed as to why it might not belong.

Main Activity (25 minutes)

Sorting Data (15 minutes)

Remarks

> Today we will be using boxes and pattern blocks to represent data in a computer. Pattern blocks have many different attributes just like ones we were just discussing! You could describe the pattern blocks using colors, sizes and shapes.

Do This: Students will be working in groups of about four. Pass out "computers" and "data" to each group. Define each of the items in this way. The computer is a box. The data is pattern blocks. Each group should have around 20 pattern blocks, make sure there is a random assortment of shapes for each group. As you pass out the materials, prompt students to store the data in their computer.

Remarks

> First we're going to sort the data in the computer by **color**. Work with your group to organize the data inside your computer.

Do This: Ask students to search for and retrieve pieces of data from their computer.

Say:

- Find a red block and hold it up.
- Find a blue block and hold it up.
- Find a yellow block and hold it up.

Note: You can continue based on other colors of pattern blocks.

Remarks

> We are going to modify the data and sort it in a new way. This time sort the data in the computer by **shape**. Work with your group to organize the data inside your computer.

Do This: Ask students to search for and retrieve pieces of data from their computer.

Say:

- Find a triangle and hold it up.
- Find a square and hold it up.
- Find a hexagon and hold it up.

Note: You can continue based on other shapes of pattern blocks.

Remarks

> We are going to delete some of the data. This box here is the trash. Bring me all of the squares.
>
> Now, we are going to modify the data and sort it one last way. This time sort the data in the computer by **size**. Work with your group to organize the data inside your computer.

Do This: Ask students to search for and retrieve pieces of data from their computer.

Say:

- Find a big shape and hold it up.
- Find a medium shape and hold it up.
- Find a small shape and hold it up.

Unplugged Activity **Pattern Block Data - Unplugged Activity**

2 **Pattern Block Data - Resource Reminder**

Graphing Data (10 minutes)

Do This: Each group of students will be assigned a different graph to create: color, shape, size. Students should create a title for their graph, labels on the x axis and then record their data by coloring in squares. After explaining the instructions, pass out the graphing templates to the groups.

Wrap Up (10 minutes)

Reflection

Do This: Post each group's data on the whiteboard.

Discuss:

- *What was the process you used to create your graph?*
- *Why was it easier to create a graph once the data was sorted?*
- *Observe the different graphs: shape, size, color. What is the same and what is different?*

[Optional] Assessment

Check For Understanding (5 minutes)

Question: What actions can computers do with data?

Discussion Goal: Review the activities that students did today with their pattern blocks and relate it to computers: store, sort, search, retrieve, modify and delete.



This work 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 15: Binary Bracelets

40 minutes

Overview

Grade Level: 2nd Grade - 3rd Grade

This **exploratory** lesson introduces the concept of binary which is how computers today store all sorts of information.

Purpose

In this lesson students will learn how information is represented in a way such that a computer can interpret and store it. When learning binary, students will have the opportunity to write codes and share them with peers as secret messages. This can then be related back to how computers read a program, translate it to binary, use the information in some way, then reply back in a way humans can understand. For example, when we type a sentence into a document then press save, a computer translates the sentence into binary, stores the information, then posts a message indicating the document has been saved.

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

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

Agenda

Warm Up (15 minutes)
- **Reflect**
- **Vocabulary**
- **Introduction**

Main Activity (20 minutes)
- **Binary Bracelets**

Optional Assessment (15 minutes)

Wrap Up (5 minutes)
- **Reflection**

Extended Learning

Objectives

Students will be able to:

- Decode binary back to letters.
- Encode letters into binary.
- Relate the idea of storing letters on paper to the idea of storing information in a computer.

Preparation

- (Optional) Watch the ***Lesson in Action** video.
- Gather markers for the bracelets. Other decorations like beads and pipecleaners are optional.
- Print one ***worksheet** and one ***assessment** per student.
- (Optional) Write a short message on the board in binary.

Links

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

For the teachers

- **Binary Bracelets** - Lesson in Action Video
- **Binary Bracelets** - Assessment Answer Key ▾ Make a Copy
- **Binary Bracelets** - Slides (**Download**) ▾ Make a Copy

For the students

- **Binary Bracelets** - Unplugged Video (**Download**)
- **Binary Bracelets** - Worksheet ▾ Make a Copy

- **Binary Bracelets** - Assessment
  Make a Copy
- **Bits Versus Bytes** - Student Video

Vocabulary

- **Binary** - A way of representing information using only two options.

Teaching Guide

Warm Up (15 minutes)

Reflect

Display: Show “Reflect” slide

Reflect: *What is inside a computer?*

Vocabulary

Display: Show “Vocabulary” slide

Say: *This lesson has one new and important word:*

- **Binary** - A way of representing information using only two options

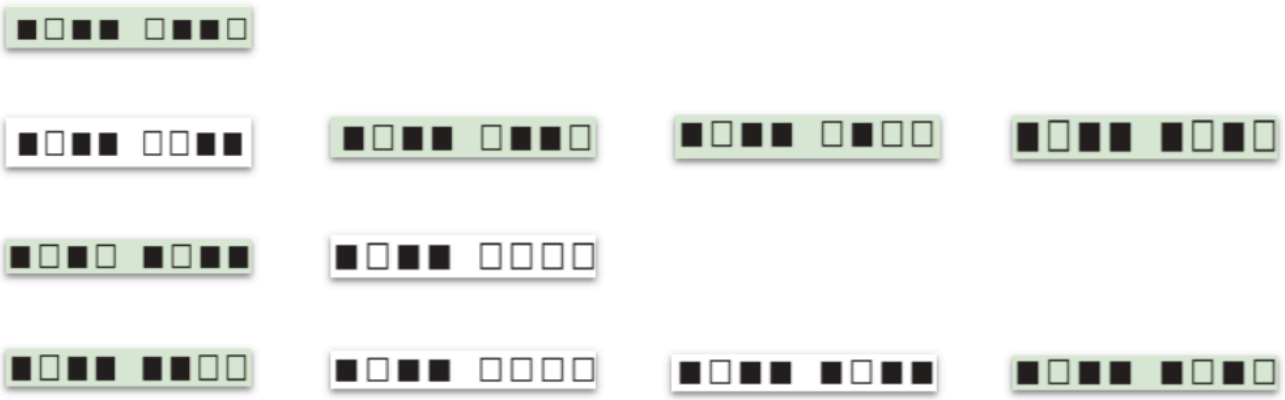
Introduction

Off and On

Display: Show “Can you read this message” slide

Do This: If you've written a short message on the board in binary, call the students' attention to it and ask if anyone knows what it is or what it means.

Can you read this message?



Put the message aside and move on to prepping for the activity.

Display: Show “What’s in here?” slide

- You can start by asking the class if they have ever seen inside a computer.
  - What's in there?
  - This is a good place to actually show them the inside of a computer (or pictures of the inside of a computer).



Display: Show “Off and On” slide

Remarks

> - Wires carry information through the machine in the form of electricity.
>   - The two options that a computer uses with respect to this electrical information are "off" and "on." Just like the lights in this room!
>   - When computers represent information using only two options, it's called "Binary."
>   - That theme of two options doesn't stop when the information gets to its destination.

Display: Show “Computer Storage” slide

Remarks

> - Computers also store information using binary. Binary isn't always off and on:
>   - Hard Disk Drives store information using magnetic positive and magnetic negative.
>   - DVDs store information as either reflective or non-reflective.

Display: Show “Binary” slide

Remarks

> How do you suppose we can convert real-life things that we want to store in a computer into binary?

Display: Show “Letters in Binary” slide

Say: Let's start with letters.

Do This:

- This is a good time to mention that each spot where you have a binary option is called a "binary digit" or "bit" for short.
- Ask if anyone knows what a grouping of eight bits is called (it's a byte.)
- Fun fact: A grouping of four bits is called a nibble.

Display: Show “Bits Versus Bytes” video

Do This: Watch the ***Bits Versus Bytes - Student Video** (~1 minute)

Display: Show “What Letter is This?” slide

Do This: Drag the blocks on the slide to make different letters. Show students how to break down the 8-bit sequence into the first 4 bits and the last 4 bits to read it more easily. Within each 4 bit sequence, have students break it down further by looking at the bits from left to right to determine if the bit is on or off.

- Go over a few examples of converting letters into binary, then back.
- Afterward, write an encoded letter and give the class a few seconds to figure out what it is.
- When the class can figure out that encoded letter on their own, you can move on to the activity.

Main Activity (20 minutes)

Binary Bracelets

Display: Show “Decompose it!” slide

Remind students to break each puzzle down into smaller steps.

Unplugged Activity **Binary Bracelets - Unplugged Activity**

2 **Binary Bracelets - Student Resources**

Teaching Tip

You know your classroom best. As the teacher, decide if students should do this individually or if students should work in pairs or small groups.

Binary Bracelets - Worksheet

You do not need to cover the whole of binary, like counting and converting numbers back and forth from decimal. This lesson is intended to be a fun introduction to how computers store information, not a frustrating lesson in bases.

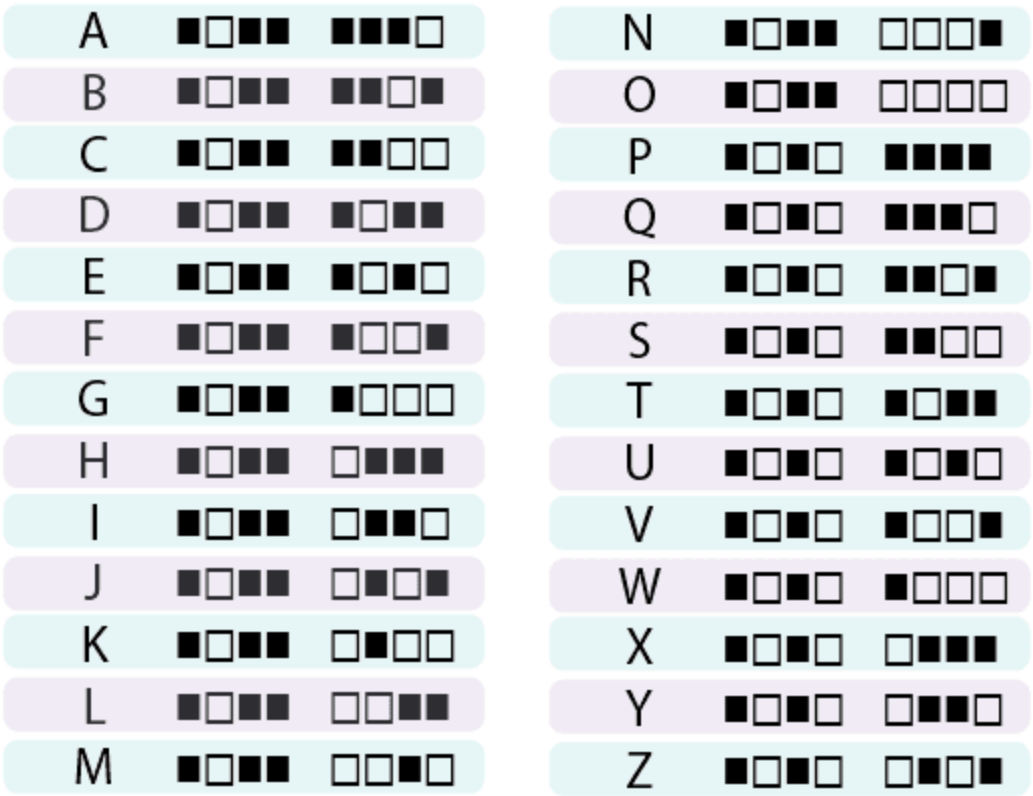
Display: Show “Binary Bracelets” slide

Directions:

- *Find the first letter of your first name on the activity sheet.*
- *Fill in the squares of a bracelet to match the pattern of the squares next to the letter that you selected.*

Display: Show next slide

- *Cut the bracelet out.*
- *Tape the bracelet around your wrist to wear it!*
- *Share your bracelet with your classmates to see if they can figure out your letter.*



Teaching Tip

If your class has extra budget for materials, try doing this exercise using thread (or pipe cleaners) and beads to create the binary bracelets instead of pen and paper. You can provide any combination of two colors in beads to the students, but black and white tend to be easiest, given the way that the key is done.

Display: Show “Can you read this message now?” slide

After the activity, revisit the message that was on the board and see if your class can decypher it using what they've learned.

Display: Show “Answer” slide

Walk students through the steps needed to read the message.

Optional Assessment (15 minutes)

Display: Show “Assessment” slide

- Hand out the ***assessment** and allow students to complete it independently after the instructions have been well explained.
- This should feel familiar, thanks to the previous activities.

Wrap Up (5 minutes)

Reflection

Flash Chat: What did we learn?

Display: Show “Flash Chat” slide

- What else do you think is represented as binary inside of a computer?
- How else might you represent binary instead of boxes that are filled or not filled?
- What was your favorite part about that activity?

Display: Show “Reflect” slide

Reflect: *Use the activity worksheet to write out the rest of your name or your favorite word in binary.*

Extended Learning

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

Binary Images

- There are several great resources on the web for taking this activity to the next level.
- If your students are interested in how images (or even music) can be represented as binary, you can find more details in Thinkersmith's **Binary Baubles**.



This work 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 16: The Internet

60 minutes

Overview

Grade Level: 5th Grade - 6th Grade

Students will pretend to flow through the internet while learning about connections, URLs, IP Addresses, and DNS in this **exploratory** lesson.

Purpose

If you have been doing every lesson in this course, then each student in your classroom has used the internet... but how many know how it works? Learning more about the internet will help students develop a better understanding of its endless possibilities.

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **NI** - Networks & the Internet

Agenda

Warm Up (20 minutes)
- **Introduction**

Main Activity (20 minutes)
- **The Internet**

Wrap Up (15 minutes)
- **Reflection**

Assessment (5 minutes)

Objectives

Students will be able to:

- Learn about the complexity of sending messages over the internet.
- Translate URLs into IP Addresses.

Preparation

- Watch the ***Internet** - Teacher Video.
- Print enough ***IP Address Cards and Delivery Type Cards** - Manipulatives for each group.
- Print one ***Internet** Assessment - for each student.
- Access to the internet (such as ***get-site-ip.com**.
- Make sure every student has a reflection journal.

Links

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

For the teachers

- **Internet** - Assessment Answer Key
  - Make a Copy
- **The Internet** - Slides (**Download**)
  - Make a Copy
- **get-site-ip-com** - Resource

For the students

- **IP Address Cards and Delivery Type Cards** - Manipulatives
  - Make a Copy
- **Internet** - Assessment
  - Make a Copy
- **The Internet** - Unplugged Video (**Download**)

Vocabulary

- **DNS** - short for Domain Name System, this system translates domain names (like example.com) to IP addresses (like 93.184.216.34)
- **DSL/Cable** - A method of sending information using telephone or television cables.
- **Fiber Optic Cable** - A connection that uses light to transmit information
- **IP Address** - A number assigned to any item that is connected to the Internet.
- **Internet** - A group of computers and servers that are connected to each other.
- **Packets** - Small chunks of information that have been carefully formed from larger chunks of information.
- **Servers** - Computers that exist only to provide things to others.
- **URL** - An easy-to-remember address for calling a web page (like **www.code.org**).
- **Wi-Fi** - A wireless method of sending information using radio waves.

Teaching Guide

Warm Up (20 minutes)

Introduction

Display: Show “Reflect” slide

Reflect: *How does the Internet help connect us to other people?*

Vocabulary:

Display: Show “Key Vocabulary” slide

Teaching Tip

A quick preview is all you need here. These words will all be explained as part of the lesson, so it would be far less confusing to do a brief intro to the words as a "see if you can spot these during the day" type of heads-up.

This lesson has several new and important words:

- **IP Address** - A number assigned to any item that is connected to the Internet
- **DNS (Domain Name Service)** - The service that translates URLs to IP addresses
- **URL (Universal Resource Locator)** - An easy-to-remember address for calling a web page (like www.code.org)
- **Internet** - A group of computers and servers that are connected to each other
- **Servers** - Computers that exist only to provide things to others
- **Fiber Optic Cable** - A connection that uses light to transmit information
- **Wi-Fi** - A wireless method of sending information using radio waves
- **DSL/Cable** - A method of sending information using telephone or television cables
- **Packets** - Small chunks of information that have been carefully formed from larger chunks of information

Teaching Tip

There are some great YouTube videos on this subject that can make this lesson a little easier to understand. You can show them to the class in advance, or just watch them yourself. **Here is one of the videos in the Code.org video series on "How the Internet Works"**. (We recommend watching from 1:44 - 5:13, if possible.) The rest of the playlist is available **here**.

Getting the Message

- It's quite likely that your students are aware of what the internet is, but they may not really understand what the internet does.
  - **Ask:** *What is the internet?*
  - *Is the internet a public place or a private place?*
  - (Truthfully, many people think it can be both, but it should be viewed as a public space no matter what settings you think you've mastered.)
  - *How does information get from place to place?*
- *Let's say that I want to look at the webpage for Code.org. What do you suppose the process would be like for me to send a message to request that page?*
  - *What do I do as a user?*
  - *What do you think happens inside the internet?*

Sending a message over the internet is a lot like sending a message through the mail...if every letter we sent required thousands of envelopes!

Display: Show “Packets” slide

Every message we send through the internet gets chopped up and each piece is wrapped in its own version of an envelope. We call those packets. Packets are specially formed chunks of information that are able to easily flow through any of the internet's channels.

Sometimes, a few of those packets will get lost, because the internet is a crazy place. In that case, the packets need to be resent, and the whole message has to get put on hold until they arrive.

Display: Show “Servers” slide

Where do you think those packets are headed?

- Even if you're sending messages to another person, they first have to go to at least one server.
  - A server is a special computer that is supposed to be always on and ready to send and receive information.
  - Every website has a server.
  - Even email goes through servers.

Display: Show “IP Addresses” slide

Servers don't have names like you and I do. They're actually addressed using numbers. These numbers are called IP addresses, and they look a little strange.

- For example: One of Code.org's IP addresses used to be 54.243.71.82
  - (Please be sure to check this out in advance. Most IP addresses change from time to time and they are then reused for other sites.)

Display: Show “DSL, Cable, Fiber Optic, or WiFi” slide

There are many ways to reach the internet from your house, school, or place of business.

- You can connect directly using a cable (that might be DSL, Cable, or Fiber Optic)
- Or you can connect using radio waves over the air through Wi-Fi.

Direct connections are most reliable, but they can be inconvenient.

- *Can you figure out why?*
  - (You have to be attached to a cable!)

Wi-Fi connections are super convenient, but the aren't always reliable.

- *Can you figure out why not?*
  - (Radio waves bounce all over the place and can get lost.)

Teaching Tip

If you're thinking that this is a lot of text and it would be extremely boring to try to lecture this to a class full of elementary school kids, you're absolutely right! If you're unable to show a YouTube video in class to help explain it all, I highly recommend drawing pictures to explain each idea above, or choosing students as volunteers to act out what you describe while you're explaining. They're not expected to get every detail and definition at this point, only to gain exposure.

Display: Show “IP Addresses & DNS” video

So, if you're used to sending information to URLs (like www.code.org) and the servers actually have IP addresses for names (like 54.243.71.82) how does the Internet change from one to the other? That's what the DNS is for. The DNS (Domain Name Server) has tables that allow the system to go back and forth between URLs and IP addresses. If the Domain Name Servers ever stopped working, it would shut down the internet as we know it!

Display: Show “Create a DNS Table” slide

With that said, let's try to understand what the DNS does by making a little DNS table ourselves.

Pull out a piece of paper and draw a grid similar to that in the internet activity:

Sample of DNS Table:

#	URL	IP Address
1	code.org	54.243.71.82
2		
3		
4		
5		

First, we need to fill in this table.

- Survey the class for their favorite websites and write the URLs in the left column
- Use a site like ***get-site-ip.com** to find the IP addresses for those sites and write them in the corresponding rows of the right column.

Now let's take this DNS Table and pretend to send messages through the internet!

Main Activity (20 minutes)

The Internet

Display: Show “Internet” video

Unplugged Activity **The Internet - Unplugged Activity**

2 **The Internet - Student Resources**

Display: Show “The Internet Activity” slide

The Internet Activity:

- Create your own DNS table, similar to what is shown above.
- Have the class help you fill in the blank spots in the table. Pick your favorite URLs and find their IP addresses using a site like www.get-site-ip.com.
- Divide into groups of 3 to 5.
- Assign each group an IP address from the newly created table, and assign each person in the group a position:
  - The Message Writer
  - The Internet
  - The Server (carries the IP address)
  - The Return Internet (optional)
  - The Message Receiver (optional)

Display: Show “Directions” slide

Directions:

- Each group will draw an ***IP Address Cards and Delivery Type Card** to find out where their message is going and what their method of message delivery (Wi-Fi, Cable/DSL, or Fiber Optic Cable) will be.
- The Message Writer will craft a note to send to the server.
- The Internet will rip the message up into 4 small pieces called packets, then deliver each packet one at a time to the Server with the IP address that was drawn from the IP Address Card stack.
- The Server will make sure that the message arrives in order, then will send each packet off one at a time with the Return Internet (can be the same person or different person than the original Internet).
- The Return Internet will deliver each piece back to the Message Receiver (can be the same person or different person than the Message Writer) and put it back together.
- The Message Receiver will wait for all of the pieces to arrive, then read the message to be sure it arrived correctly!

Display: Show “Rules” slide

Rules:

- The Internet must rip the message into exactly four packets.
- If the Internet drops a packet, they have to pick it up and go back to the start to deliver it again.
- The server has to wait for all of the message pieces to arrive before it can begin to send the message along.

Display: Show “Info” slide

Info:

- Wi-Fi: Convenient, but spotty. Wi-Fi doesn’t require cables, but since the signal bounces all over the place, packets can get lost pretty easily.
  - Simulation: Internet must carry each packet on their shoulder (no hands).
- Cable/DSL: Fairly good at delivering messages, but you must be connected to a wire.
  - Simulation: Internet must carry each packet on the back of one hand and must keep the other hand touching a wall, desk, chair or the floor at all times.
- Fiber Optic Cable: The best at delivering messages, but you must be connected to a wire.
  - Simulation: Internet can carry packets in hand, but must keep the other hand touching a wall, desk, chair or the floor at all times.

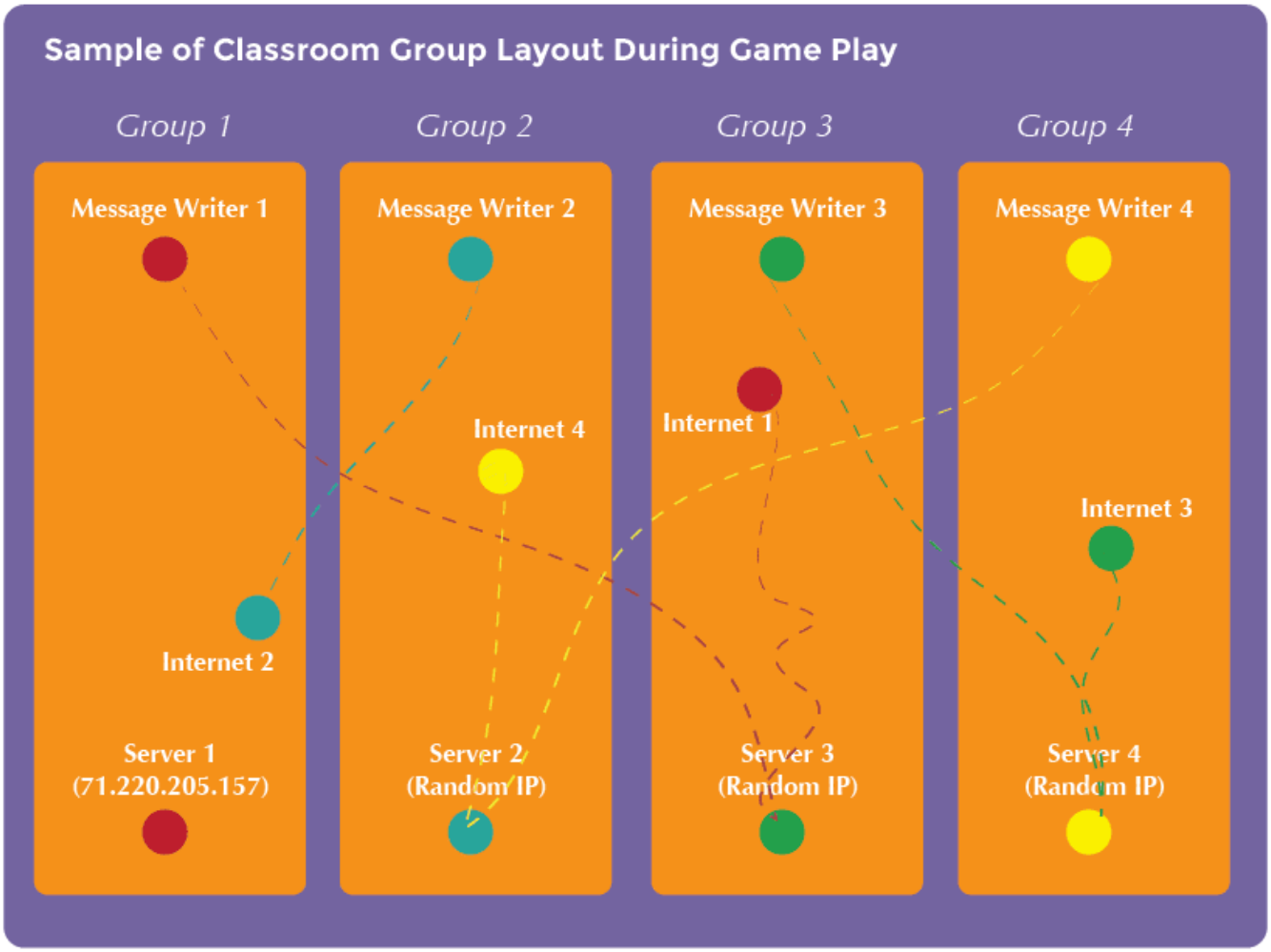
Teaching Tip

If it feels like there are too many rules to explain outright, feel free to post them on the board and just explain the game as you go. You can play multiple rounds until the class really understands.

Display: Show Student Materials slide

To play this game, you can have your groups cluster anywhere, but for the first time it can be less confusing to have groups play in a line.

- Line up the "Servers" on one end of the room (holding their IP addresses). The Return Internet players can be over there as well (if you have that many people in each group).
- Have the everyone else line up across from their server at the other side of the room.
- The Message Senders will likely be sending their messages to a server other than their own, so the Internet players will likely cross over from group to group. It may look something like the diagram below:



Wrap Up (15 minutes)

Reflection

Teaching Tip

Flash Chat questions are intended to spark big-picture thinking about how the lesson relates to the greater world and the students' greater future. Use your knowledge of your classroom to decide if you want to discuss these as a class, in groups, or with an elbow partner.

Flash Chat: What did we learn?

Display: Show "Flash Chat" slide

- What kind of connection would you rather have (Wi-Fi, DSL/Cable, or Fiber Optic)? Why?
- Why might it take your message a long time to get somewhere?

Reflect

Display: Show "Reflect" slide

Reflect: *What's something you learned about the internet today?*

Assessment (5 minutes)

Display: Show “Internet Assessment” slide

Hand out the ***assessment** worksheet and allow students to complete the activity independently after the instructions have been well explained. This should feel familiar, thanks to the previous activities.



This work 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 17: Hardware and Software Skits

45 minutes

Overview

Grade Level: 3rd Grade - 4th Grade

In this unplugged lesson students will act out the ways internal and external parts of a computer work together. They will take on roles such as: CPU, Keyboard, Monitor, User and more.

Purpose

The purpose of this lesson is to understand the ways internal and external parts of a computer work together to perform their functions. Today you will collaborate with a group of students to create short skits showing how various computer parts work together to complete tasks.

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

▶ **CS** - Computing Systems

Agenda

Warm Up (7 minutes)
- **Vocabulary**
- **Looking Inside**

Main Activity (35 minutes)
- **Preparing your Skit**
- **Performing Your Skit**

Wrap Up (3 minutes)
- **Reflection**

[Optional] Assessment
- **Check For Understanding**

Objectives

Students will be able to:

- Collaborate with a group to creatively dramatize a skit outline
- Define various internal and external computer parts
- Understand how internal and external computer parts function as a system

Preparation

- Print the skit outlines and cut them out for each group of students
- Print out one classroom set of the skit roles
- Create heterogeneous groups of students

Links

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

For the teachers

- **Hardware and Software Skits** - Resource (**Download**) Make a Copy
- **Hardware and Software Skits** - Slides (**Download**) Make a Copy
- **Hardware and Software Skits - Roles** - Resource (**Download**) Make a Copy

Vocabulary

- **Central Processing Unit (CPU)** - Controls all functions of the computer and executes all instructions

- **Cloud storage** - Saves your files and other data on the internet
- **Cooling fan** - Hardware inside of a computer that cools down the device
- **Ethernet Cable** - Special cable that allows access to the internet
- **Hardware** - The physical parts of the computer that you can see and touch, like the monitor, the central processing unit (CPU), the keyboard, and the mouse.
- **Keyboard** - A device that allows you to type letters, numbers and other symbols on a computers
- **Laptop** - A personal computer that can be easily moved and used in a variety of locations
- **Monitor** - The screen that displays pictures and text for a computer
- **Mouse** - The hand-held device that allows you to point to a place on a screen and to take action from that place
- **Power cord** - Connects the computer to a power source
- **Software** - Instructions that tells the hardware how to do specific tasks.
- **Stylus** - Used for writing on a screen
- **User** - The person using a device

Teaching Guide

Warm Up (7 minutes)

Vocabulary (2 minutes)

Display: Show “Vocabulary” slide

- Go through some of the terms that might be unfamiliar to students
- Have students refer back to these definitions throughout the main activity as needed

Looking Inside (5 minutes)

Display: Show “Warm Up” slide

Discuss:

- *Think about an electronic device you use often. For example, a laptop, phone or tablet. What do you think it looks like inside?*

Display: Show “Looking Inside” slide

Discuss:

- *What do you notice?*

Discussion Goal: These images show many of the computer parts that students will be using in their skits later in the lesson. This view may be totally new to students and present a new concept they hadn’t considered before - what is inside a computer (or electronic device)? Students may start naming some easily identifiable internal computer parts such as wires. Let students know that by the end of the lesson they will have a much better idea of how everything inside the computer works together!

Main Activity (35 minutes)

Preparing your Skit (15 minutes)

Display: Show “Preparing Your Skit” slide

Do This: Put students into groups of 5 and assign them a skit number from the **skit outline**. Give each student a sheet of paper with their **skit role** on it. First, students should decide who will take on the role of each computer part. Then, students should read through the skit and start brainstorming how they will bring it to life. Finally, students should rehearse their skit and be ready to present it.

Unplugged Activity **Unplugged Activity - Hardware and Software Skits**

2 **Hardware and Software Skits - Student Reminder**

Teaching Tip

- Make sure to encourage student creativity. Students can have the freedom to add details to their skit or prepare quick props. Create a positive environment where students will feel comfortable taking a risk by performing the skit.
- Circulate and provide feedback to students emphasizing both strengths and areas for improvement. Support with time management by routinely letting students know how much time is left.
- If you are unable to create even groups of students, a 6th member of a group could be added. The extra person’s role would be the narrator. They can start off the skit by introducing each character and telling what the computer parts are going to work together to do. It can be at the group’s discretion if any extra narration is needed.
- Alternatively, a smaller group could be formed with only 4 members. The “user” role would be eliminated from the skit. The group would creatively decide how to make up for this missing element.
- If you have more students and require extra skit outlines, two groups can work on the same skit.

Performing Your Skit (20 minutes)

Display: Show “Performing Your Skit” slide

Do This: It is up to your discretion how students will present their skits. You may choose based on your knowledge of your class and time constraints. Some ideas:

- Skits could be recorded via video using a computer device. Flip allows students to record and edit videos and then watch their classmates videos. There are settings that allow students to comment and like each others videos.
- Groups could be paired up and show each other their skits. In this option, students perform in front of another small group of peers. The audience is small so stakes are lower. Then they have the opportunity to watch another skit live.
- Groups could volunteer to present to the whole class. Before calling students back to the rug, have groups discuss whether they would like to present in front of everyone. Tell groups that all group members must agree that they wish to present.

Wrap Up (3 minutes)

Reflection (3 minutes)

Display: Show "Reflection" slide

Do This: Have students share a glow (positive), grow (constructive feedback), and question they have for another group who performed.

[Optional] Assessment

Check For Understanding (5 minutes)

Do This: This can be used for an exit ticket or other formative assessment.

Question:

- *What is something new you figured out about an internal or external computer part?*

This work 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 18: Conditionals with Cards

50 minutes

Overview

Grade Level: 3rd Grade - 4th Grade

In this **context-setting** lesson, students will write *conditional (if/else) statements* to state the rules of simple card games.

Purpose

One of the best parts of teaching conditionals is that students already understand the concept from their everyday lives.

This lesson merges computer science into the real world by building off of their ability to tell if a condition is true or false. Students will learn to use if statements to declare when a certain command should be run, as well as if / else statements to declare when a command should be run and what to run otherwise. Students may not recognize the word conditionals, but most students will understand the idea of using "if" to make sure that some action only occurs when it is supposed to.

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

▶ **AP** - Algorithms & Programming

Agenda

Warm Up (15 minutes)
- **Reflect**
- **Vocabulary**
- **Introduction**

Main Activity (25 minutes)
- **Conditionals with Cards**
- **Optional Assessment**

Wrap Up (10 minutes)
- **Reflection**

Extended Learning

Cross-Curricular Opportunity

Objectives

Students will be able to:

- Define circumstances when certain parts of a program should run and when they shouldn't.
- Determine whether a conditional is met based on criteria.
- Traverse a program and predict the outcome, given a set of input.

Preparation

- Gather decks of cards or something similar.
- Print or display the ***sample programs**.
- Print one ***assessment** for each student.

Links

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

For the teachers

- **CSF - Course D - Slides** - Slides (**Download**) ▾ Make a Copy
- **Conditionals with Cards** - Lesson in Action Video
- **Conditionals with Cards** - Assessment Video
- **Conditionals with Cards** - Assessment Answer Key ▾ Make a Copy
- **Conditionals with Cards Sample Program** - Teacher Prep Guide ▾ Make a Copy

For the students

- **Conditionals with Cards** - Unplugged Video (**Download**)
- **Conditionals with Cards** - Assessment ▾ Make a Copy

For all

- **Conditionals with Cards** (**Download**) ▾ Make a Copy

Vocabulary

- **Conditionals** - Statements that only run under certain conditions.
- **pseudocode** - A plain language description of the steps in an algorithm

Teaching Guide

Warm Up (15 minutes)

Reflect

🗉 **Display**: Show “Reflect” slide

Reflect: *We use the word “if” to make sure that some action only occurs when it is supposed to.*

Finish this statement: If I do well in my classes, then….

Vocabulary

🗉 **Display**: Show “Vocabulary” slide

- **Conditionals** - Statements that only run under certain conditions.
- **Pseudocode** - A plain language description of the steps in an algorithm.

Introduction

🗉 **Display**: Show “Reward” slide

- We can start this lesson off right away
  - Let the class know that if they can be completely quiet for thirty seconds, you will do something like:
    - Sing an opera song
    - Give five more minutes of recess
    - Do a handstand
  - Start counting right away.
  - If the students succeed, point out that they succeeded, so they get the reward.

- Otherwise, point out that they were not completely quiet for a full thirty seconds, so they do not get the reward.

🗉 **Display**: Show “If Statement” slide

Say: *What was the condition of the reward?*

- *The condition was IF you were quiet for 30 seconds*
  - *If you were quiet, the condition would be true, and you would get the reward.*
  - *If you weren't quiet, the condition would be false, so the reward would not apply.*
- *Can we come up with another conditional?*
  - *If you can guess my age correctly, the class can give you applause.*
  - *If I know an answer, I can raise my hand.*
  - *What examples can you come up with?*

🗉 **Display**: Show “Else Statement” slide

- *Sometimes, we want to have an extra condition, in case the "IF" statement is not true.*
  - *This extra condition is called an "ELSE" statement*

🗉 **Display**: Show “Else Statement (example)” slide

- *When the "IF" condition isn't met, we can look at the "ELSE" for what to do*
  - *Example: **IF** I draw a king from this deck of cards, everybody claps.
  - Or **ELSE**, everyone says "Awwwwwwe."*

🗉 **Display**: Show “Let's Try It” slide

- *Let's try it.*(Draw a card and see if your class reacts appropriately.)
  - Ask the class to analyze what just happened.
    - *What was the IF?*
    - *What was the ELSE?*
    - *Which condition was met?*

🗉 **Display**: Show “Believe it or Not” slide

- *Believe it or not, we have even one more option.*
  - **If** I draw a 7, clap.
  - **Else if** I draw something less than 7, you say "YAY".
  - **Else** say "Awwwwe".

🗉 **Display**: Show “If, Else-If, Else” slide

- *This is why we have the terms If, Else-If, and Else.*
  - ***If** is the first condition.*
  - ***Else-If** gets looked at only if the "If isn't true.*
  - ***Else** gets looked at only if nothing before it is true.*

Now let's play a game.

Main Activity (25 minutes)

Conditionals with Cards

Unplugged Activity **Conditionals with Cards - Video**

2 **Conditionals with Cards - Student Resources**

Conditionals with Cards Sample Program - Teacher Prep Guide

Directions:

- Create a few programs with your class that depend on things like a card's suit, color, or value to award or subtract points. You can write the program as an algorithm, pseudocode, or actual code.

Display: Show “Sample” slide

Here is a sample algorithm:

```
if (CARD is RED)
        Award YOUR team 1 point

Else
        Award OTHER team 1 point
```

Here is a sample of the same program in pseudocode:

```
If (card.color == RED){
        points.yours = points.yours + 1;
}

Else {
        points.other = points.other + 1;
}
```

- Decide how you want to split your class into teams.
- Each team should have a pile of cards (at least as many cards as team members) nearby.

Display: Show “Let's play a game!” slide

- Put one of your “Programs” up on the board for all to see.
- Have the teams take turns drawing cards and following the program to see how many points they score in each round.
- Play several times with several different programs to help the students really understand conditionals. (Three additional slides have been provided for this part of the activity if needed.)

Once the class has had some practice, you can encourage students to nest conditionals inside one another. Make sure they understand that if the card is red, YOUR team is awarded 1 point, and then *nothing else happens*, since the condition was met:

Display: Show "Let's nest conditionals” slide

```
If (CARD is RED)
        Award YOUR team 1 point

Else
        If (CARD is higher than 9)
                Award OTHER team 1 point
        Else
                Award YOUR team the same number of points on the card
```

Here is the same program in pseudocode:

```
If (card.color == RED ){
        points.yours = points.yours + 1;
}
Else {
        if (card.value > 9){
                points.other = points.other + 1;
        }
        Else {
                points.yours = points.yours + card.value;
        }
}
```

Optional Assessment (5 minutes)

Display: Show “Assessment” slide

Hand out ***Conditionals with Cards - Assessment** and allow students to complete the activity independently after the instructions have been well explained. This should feel familiar, thanks to the previous activities. If needed, refer to the **Assessment Video** resource as a guide.

Wrap Up (10 minutes)

Flash Chat: What did we learn?

Display: Show “Flash Chat” slide

Teaching Tip

Flash Chat questions are intended to spark big-picture thinking about how the lesson relates to the greater world and the students' greater future. Use your knowledge of your classroom to decide if you want to discuss these as a class, in groups, or with an elbow partner.

The two bolded questions below are the ones featured in the slides. Feel free to ask your students to answer the other Flash Chat questions as well.

- If you were going to code this up in Blocky, what would you need to add around your conditionals to let the code run more than one time? (A loop)
- What other things do you do during the day under certain conditions?

- **If you are supposed to do something when the value of a card is more than 5, and you draw a 5, do you meet that condition?**
- Notice that conditions are either "True" or "False." There is no assessment of a condition that evaluates to "Banana."
- **When you need to meet several combinations of conditions, we can use something called "nested conditionals."**
  - **What do you think that means?**
  - **Can you give an example of where we saw that during the game?**
- What part of that game did you like the best?

Reflection

Display: Show “Reflect” slide

Reflect:

- *What are some of the conditionals you used today?*
- *Can you come up with some more that you would use with a deck of cards?*

Extended Learning

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

True/False Tag

- Line students up as if to play **Red Light / Green Light**.
- Select one person to stand in front as the Caller.
- The Caller chooses a condition and asks everyone who meets that condition to take a step forward.
  - If you have a red belt, step forward.
  - If you are wearing sandals, take a step forward.
- Try switching it up by saying things like "If you are *not* blonde, step forward."

Nesting

- Break students up into pairs or small groups.
- Have them write if statements for playing cards on strips of paper, such as:
  - the suit is clubs
  - the color is red
- Have students create similar strips for outcomes.
  - Add one point
  - Subtract one point
- Once that's done, have students choose three of each type of strip and three playing cards, paying attention to the order selected.
- Using three pieces of paper, have students write three different programs using only the sets of strips that they selected, in any order.
  - Encourage students to put some if statements inside other if statements.
- Now, students should run through all three programs using the cards that they drew, in the same order for each program.
  - Did any two programs return the same answer?
  - Did any return something different?

Cross-Curricular Opportunity

Play the Deck with Conditionals (20-30 minutes)

Computer Science + English Language Arts + Math

Play the Deck with Conditionals is an optional activity aligned to Common Core ELA and Math standards, written by our teacher community. Students will use a deck of playing cards to review basic multiplication and division problems as you apply the computer science concept of conditionals.

Standards Addressed:

- **CCSS.ELA-LITERACY.SL.3.3:** Ask and answer questions about information from a speaker, offering appropriate elaboration and detail.
- **CCSS.MATH.CONTENT.3.OA.C.7:** Fluently multiply and divide within 100, using strategies such as the relationship between multiplication and division (e.g., knowing that 8 × 5 = 40, one knows 40 ÷ 5 = 8) or properties of operations. By the end of Grade 3, know from memory all products of two one-digit numbers.

This work 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 19: Songwriting

50 minutes

Overview

Grade Level: 4th Grade - 5th Grade

This **context-setting** lesson will help students understand why combining chunks of code into functions can be a helpful practice.

Purpose

The use of functions helps simplify code and develop the student's ability to organize their program. Students will quickly recognize that writing functions can make their long programs easier to read and easier to debug if something goes wrong.

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

- ▶ **AP** - Algorithms & Programming

Agenda

Warm Up (20 minutes)
- **Introduction**

Main Activity (20 minutes)
- **Songwriting**

Wrap Up (5 minutes)
- **Reflection**

Assessment (5 minutes)

Extended Learning

Cross-Curricular Opportunity

Objectives

Students will be able to:

- Describe how functions can make programs easier to write.
- Identify sections of a song to pull into a function.
- Locate repeating phrases inside song lyrics.

Preparation

- (Optional) Watch the ***Lesson in Action** Video.
- Print several ***worksheets** for each group.
- Print one ***assessment** for each student.
- Secure access to songs and lyrics for activity.
- 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 teachers

- **Functions Unplugged: Songwriting** - Lesson in Action Video
- **Functions Unplugged: Songwriting** - Assessment Answer Key
  Make a Copy
- **Songwriting** - Slides (**Download**)
  Make a Copy

For the students

- **Functions Unplugged: Songwriting** - Worksheet Make a Copy

- **Functions Unplugged: Songwriting** - Assessment Make a Copy
- **Songwriting with Functions** - Unplugged Video (**Download**)

Vocabulary

- **Function** - A piece of code that you can call over and over again.

Teaching Guide

Warm Up (20 minutes)

Introduction

Display: Show "Reflect" slide

Reflect: *What is the chorus to your favorite song?*

Vocabulary

Display: Show "Vocabulary" slide

This lesson has one new and important word:

- **Function** - A piece of code that you can call over and over again.

Sing a Song

Display: Show "Baby Shark" slide

- Let the class know that today is song day!
- We're going to learn a song together.
  - Start with a simple song, either written out or projected on the screen.
  - Point to the chorus and be sure that the class knows how it goes before you begin on the song.
  - Blast through the song, singing it with them in the beginning, then see what happens when you get to the part where it calls the chorus.

Teaching Tip

Baby Shark is being used here as an example only. If your students know this song, feel free to use it. Otherwise, choose an appropriate song that they might be more familiar with (either from music class or the radio.)

Key Gestures:

Baby shark: make a shark mouth with finger and thumb.

Mommy shark: make a mouth with both hands, opening at the wrist.

Daddy shark: make a big mouth with both of your arms.

Grandma shark: like mama shark, but with fingers curled in.

Grandpa shark: like papa shark, but with fingers curled in.

Hungry shark: one hand on head like a fin, rubbing tummy with the other hand.

Little fish: both hands together with fingers pointing forward, move from side to side.

Swim away: pretend to swim.

Safe at last: wipe forehead, and flick hand like removing sweat.

Chorus:

doo doo, doo doo doo doo

Song:

Baby Shark

Chorus

Baby Shark

Chorus

Baby Shark

Chorus

Baby Shark

Mommy Shark

Chorus

Mommy Shark

Chorus

Mommy Shark

Chorus

Mommy Shark

Daddy Shark

Chorus

Daddy Shark

Chorus

Daddy Shark

Chorus

Daddy Shark

(Continuing on, if time permits, with "Grandma shark", "Grandpa shark", "Hungry shark", "Little fish", "Swim away", "Swim faster", "Safe at last".)

- It's quite likely that the majority of the class will sing the lyrics for the chorus when you point to that bit.
  - Stop the song once that happens, and explicitly highlight what just happened.
    - You defined the chorus.
    - You called the chorus.
    - They sang the chorus.

Display: Show “Discuss” slide

- Ask the class why they suppose you only wrote the chorus once at the top of the paper instead of writing it over and over in each place where it is supposed to be sung.
  - What are other benefits of only writing the chorus once when you sing it many times?

Teaching Tip

Call out the difference between a function (“doo doo”) and a loop (“baby shark doo doo” x 3) - or ask your students if they can tell you the difference between them.

To hit this point home, you can look up the lyrics for some popular songs on the Internet. Show the students that the standard for repeating lyrics is to define the chorus at the top and call it from within the body of the song.

Display: Show “Connection” slide Now, imagine that this song is a computer program. Defining a title (like "chorus") for a little piece of code that you use over and over again is called creating a function.

This is helpful to computer scientists for some of the same reasons that it is helpful to songwriters.

- It saves time not having to write all the code over and over in the program.
- If you make a mistake, you only have to change it one place.
- The program feels less complicated with the repeating pieces defined just once at the top.

We are going to play with songs a little more, to try to really understand how often this technique is used!

Main Activity (20 minutes)

Songwriting

A fantastic way to compare functions to something we see in our everyday lives is to look at songs. Songs often have certain groups of lyrics that repeat over and over. We call such a group a "chorus."

Unplugged Activity **Songwriting - Functions Video**

2 **Songwriting - Student Resources**

Teaching Tip

It's most exciting for students to do this lesson with popular music from the radio, but if you're having a hard time finding appropriate songs where the lyrics repeat exactly, here are a few timeless options:

- **You Are My Sunshine**
- **Boom, Boom, Ain't it Great**
- **How Much Is That Doggie in the Window**
- **I Love Trash**

Display: Show “Songwriting” slide

Directions:

- Divide into groups of 4, 5, or 6.
- Give each group several copies of the ***Songwriting Worksheet**.
- Play a short song for the class that contains a clear chorus that does not change from verse to verse.
- Challenge the class to identify (and write down) the chorus.

- Compare results from each group.

Did everyone get the same thing? Sing your choruses together to find out! Play this game over and over until the class has little trouble identifying the choruses.

- It is often easier just to have the class listen to (or watch) the song, then vote on what the chorus is by singing it together, rather than writing the whole thing down. If you choose this method, consider having the class do a written chorus for the final song selection to be sure that the visual learners get proper reinforcement.

Wrap Up (5 minutes)

Reflection

Display: Show "Discussion" slide

Teaching Tip

Flash Chat questions are intended to spark big-picture thinking about how the lesson relates to the greater world and the students' greater future. Use your knowledge of your classroom to decide if you want to discuss these as a class, in groups, or with an elbow partner.

Flash Chat: What did we learn?

- Would you rather write lyrics over and over again or define a chorus?
- Do you think it's possible to make multiple choruses for the same song?
- Does it make sense to make a new chorus for every time it's needed in a song?

Display: Show "Reflect" slide

Reflect: *Can you think of another activity (besides songwriting) where you might want to call a special group of instructions several times?*

Assessment (5 minutes)

Display "Songwriting Assessment" slide.

Hand out the ***Songwriting Assessment worksheet** and allow students to complete the activity independently after the instructions have been well explained. This should feel familiar, thanks to the previous activities.

Extended Learning

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

Functional Suncatchers Visit the **CS Fundamentals Unplugged Table** or click on the link for **Functional Suncatchers**. This activity does take a few supplies from the craft store, but it helps students to see the value of calling multiple functions.

Create Your Song

- Start by creating a chorus together, then repeat it between verses of a song that you develop around it.
- Make a change to the chorus, and ponder how much easier it is to change in just one place.
- Change the chorus again, making it much longer than it was originally.

- Add a second chorus and alternate between them in your verses.

Songwriting a Program

- What if we acted out songs instead of singing them? All of a sudden, our chorus would be a function of repeated actions, rather than words.
- Use the concepts of the arrows from the Graph Paper Programming lesson and create a program with lots of repeating instructions.
  - Circle those repeating actions so that the class can see where they are.
  - Define a function called "Chorus" above the program.
  - Cross out everywhere the repeating actions appear in the program and write "Chorus" instead.
- Repeat until the class can go through this process with little direction.

Cross-Curricular Opportunity

Fossil Fact Rap (45-60 minutes)

Computer Science + English Language Arts + Science

Fossil Fact Rap is an optional activity aligned to Common Core ELA and Next Generation Science Standards, written by our teacher community. Students will expand their knowledge of functions to write a rap using academic terminology and facts related to fossils and rock layers. They will then learn how to “call a function” using the provided “defined function,” or rap song’s hook.

Standards Addressed:

- **CCSS.ELA-LITERACY.W.4.2.D:** Use precise language and domain-specific vocabulary to inform about or explain the topic.
- **CCSS.ELA-LITERACY.W.4.7:** Conduct short research projects that build knowledge through investigation of different aspects of a topic.
- **CCSS.ELA-LITERACY.SL.4.5:** Add audio recordings and visual displays to presentations when appropriate to enhance the development of main ideas or themes.
- **NGSS.4-ESS1-1:** Identify evidence from patterns in rock formations and fossils in rock layers to support an explanation for changes in a landscape over time.

This work 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 20: Blank Space Stories

55 minutes

Overview

Grade Level: 5th Grade - 6th Grade

In this **context-setting** lesson, students will use fill-in-the-blank stories (similar to Mad Libs®) as a context for understanding how computers take and store input from a user, then use it later as a program runs.

Purpose

This lesson provides a shared context that will be helpful for understanding how variables work when students return to programming.

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

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

Agenda

Warm Up (15 minutes)
- **Favorite Meal**
- **Party Games**

Main Activity (30 minutes)

Wrap Up (10 minutes)
- **Introducing Variables**
- **Reflection**

Cross-Curricular Opportunity

Objectives

Students will be able to:

- Assign a variable a value
- Call a variable multiple times in a program.
- Declare a variable
- Determine the relationship between how a variable is defined, stored, and retrieved when we press "Run" on a program.

Preparation

Obtain 6 envelopes (also suitable: paper or index cards, folded in half).

- Label the front of each envelope with large letters: object, liquid, number, adjective, body part, number
- Print copies of the student ***worksheet** (or post the digital version to your LMS).

Links

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

For the teachers

- **Blank Space Stories** - Answer Key ▾ Make a Copy
- **Blank Space Stories** - Slides (**Download**) ▾ Make a Copy

For the students

- **Blank Space Stories** - Activity Guide ▾ Make a Copy

Vocabulary

- **Variable** - A label for a piece of information used in a program.

Teaching Guide

Warm Up (15 minutes)

Display: Show "How are things stored?" slide

Remarks

> One way to store things is with labeled containers.

Display: Show "Blank Space Stories" slide

Ask students to think about how we organize and store things. Do students ever keep things in boxes? Are those boxes labeled? Why? If students do not have this experience, ask them to imagine helping you pack up the classroom for the summer. What if instead of objects, you needed to store information?

Inform students that today we will be playing a word game, similar to Mad Libs®, where you will be prompted for words that will then be placed into a silly story.

Favorite Meal

Display: Show "Blank Space Story: Favorite Meal" slide

Do this: Show the first blank space story (Favorite Meal) on the board using the lesson slides.

Prompt: *How do you think this blank space story works? What do you notice about the labels for each blank space?*

It's important that students understand one person typically provides the words designated by each label without seeing the story first. Students may notice that each space has a unique label.

Do This: Prepare the first three envelopes for demonstration. If using the slides, advance the animation to show the three purple boxes.

Remarks

> In a moment, I'll have volunteers to help us fill in the blanks. We're going to use envelopes to keep track of the information they provide. Each envelope has a label that goes with one of the blank spaces. If I hand you an envelope, please write a word that goes with that label and place it inside, then hand it back to me.

Do This: Facilitate handing out and collecting the envelopes marked "object", "liquid", and "number". Once collected, keep the envelopes closed and begin reading the story. Only when you reach each blank, open the envelope and read the value stored.

Party Games

Display: Show "Blank Space Story: Party Games" slide

Do This:

- Prepare the envelopes marked “adjective”, “object”, “body part”, and “number”.
- Repeat the game above with the second story. This time you can begin with the labels before showing the story to build more suspense. The animations in the lesson slides are already set up to help facilitate this.

(Optional) Role Play Extension

A connection can be made between the volunteers and the idea of an app having a “user”. In this analogy, your role as the teacher has been like the computer. You will request and store the information from the "users" and then work out how to put the story together. To involve more students and have them act out this model, assign roles:

Computers: Pass out and collect envelopes. Open each envelope as the story is being read to check for the values being stored.

Users: Submit the words needed to complete the story.

Reader: Read the finished story out loud.

Debrief

Display: Show “Reflect” slide

Reflect: *How is writing a blank space story different from writing a normal story?*

Vocabulary

Display: Show “Vocabulary” slide

The blank spaces in these stories work like containers for information. Each container also has a label.

In computer science, we call this a variable.

- **Variable** - A label for a piece of information used in a program.

Main Activity (30 minutes)

Create Your Own Story (20 mins)

Display: Show “Unplugged Variables in Envelopes” video

Remarks

> In today’s activity, you’ll work with a group to create your own blank space story.

Unplugged Activity **Video: Unplugged Variables in Envelopes**

2 **Blank Space Stories - Student Resources**

Display: Show “Create Your Own Story” slide

Do This: Pass out the ***student worksheet** and go over the instructions.

Teaching Tip

Writing a blank space story can be challenging. It can be helpful to give students suggestions of themes to inspire and focus their writing. Consider having students work on this task in groups so they can collaborate.

Topic suggestions:

- imaginary class field trip
- unusual holiday
- movie review
- morning routine
- a day at the beach
- how to fly a kite

Share Your Stories (10 mins)

Display: Show "Share Your Stories" slide

Do This: Pair up two groups and have them take turns sharing their stories.

If you have enough time and materials, give students their own envelopes to label and exchange. This will help students build the right mental model for how this exchange works when they transition back to programming.

Teaching Tip

Save students' stories!

In a couple of lessons, students will apply what they've been learning to create another Sprite Lab project. There are several options for this project, but one of them is to create a digital blank space story.

For example, **here** is a Sprite Lab version of the *Party Games* story from this lesson.

Wrap Up (10 minutes)

Introducing Variables

Display: Show "Blank Space Stories Wrap-Up" slide

Do this: Summarize the rules of blank space stories once more with the class.

Display: Show next slide

Remarks

> *These apps are all asking the user to provide some information. How does this relate to today's activity?*

Display: Show "Variables" slide

Remarks

> *Information is stored in the computer's memory with variables. Each variable has a label that tells us what kind of information to expect. When the computer needs that information later, it looks for the label of the variable.*

Display: Show next "Variables" slide

Remarks

> *Whether you're storing objects in a box, words in an envelope, or information in a variable, it's important to use good labels, so you don't get confused later.*
>
> *When a user types into a prompt, it's like the computer stores the information in a container. The variable label tells us what kind of information to expect.*

Display: Show next slide

Reflection

Display: Show "Reflect" slide

Reflect: *How do you think blank space stories relate to learning computer science?*

Cross-Curricular Opportunity

Food Chain Variables (45-60 minutes)

Computer Science + English Language Arts + Math + Science

Food Chain Variables is an optional activity aligned to Common Core ELA, Common Core Math and Next Generation Science Standards, written by our teacher community. Students will identify various values for the variables found in food chains (Example: plants, animals, decomposer, etc.) as they document these values on notecards and show the movement of matter through an environment. Students will also learn about the 10% Rule and further describe energy flow with a writing prompt.

Standards Addressed:

- **CCSS.ELA-LITERACY.W.5.2.B:** Develop the topic with facts, definitions, concrete details, quotations, or other information and examples related to the topic.
- **CCSS.MATH.CONTENT.5.NF.B.7.C:** Solve real world problems involving division of unit fractions by non-zero whole numbers and division of whole numbers by unit fractions, e.g., by using visual fraction models and equations to represent the problem.
- **NGSS.5-LS2-1:** Develop a model to describe the movement of matter among plants, animals, decomposers, and the environment.

This work 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**.