Express

CS Fundamentals Express combines the best of Courses A-F into a single condensed course. We recommend Courses A-F for most classrooms, but if your school only offers one computer science course or you want to have your students work through multiple courses in a single year, the Express course is a better choice.

Lesson 1: Graph Paper Programming
In this lesson, you will program your friend to draw pictures!

Lesson 2: Programming with Angry Birds
Programming | Algorithms | Maze | Sequencing
Learn about sequences and algorithms with Angry Birds.

Lesson 3: Relay Programming
Unplugged | Relay Programming | Algorithms
Remember at the beginning of the course when you made drawings with code? In this lesson, you will be working with a team to do something very similar!

Lesson 4: Debugging with Scrat
Debugging | Bug | Maze
Find problems in puzzles and practice your debugging skills.

Lesson 5: Collecting Treasure with Laurel
Collector | Program | Programming
Write algorithms to help Laurel the Adventurer collect lots of gems!

Lesson 6: Creating Art with Code
Artist | Sequencing
Create beautiful images by programming the Artist.

Lesson 7: Getting Loopy
Unplugged | Loops
In this lesson, we'll have a dance party using repeat loops!

Lesson 8: Loops with Rey and BB-8
Loops | Maze
Help BB-8 through mazes using loops!

Lesson 9: Sticker Art with Loops
Loop | Artist
In this lesson, loops make it easy to make even cooler images with Artist!
Lesson 10: Digital Sharing
Lesson 11: Looking Ahead with Minecraft
  Conditional | Minecraft
  Avoid the lava! Here you will learn about conditionals in the world of Minecraft.
Lesson 12: Nested Loops in Maze
  Nested Loops | Loops | Bee | Maze
  Loops inside loops inside loops. What does this mean? This lesson will teach you what happens when you place a loop inside another loop.
Lesson 13: Snowflakes with Anna and Elsa
  Loop | Nested Loop | Frozen
  Anna and Elsa have excellent ice-skating skills, but need your help to create patterns in the ice. Use nested loops to create something super COOL.
Lesson 14: While Loops with the Farmer
  While Loops | Loops | Farmer
  Loops are so useful in coding. This lesson will teach you about a new kind of loop: while loops!
Lesson 15: Until Loops in Maze
  Conditional | Loop | Maze | Angry Bird | Zombie
  You can do some amazing things when you use `until` loops!
Lesson 16: Conditionals with Cards
  Conditionals | Unplugged
  It's time to play a game where you earn points only under certain conditions!
Lesson 17: If/Else with Bee
  Conditional | Bee | Maze
  Now that you understand conditionals, it's time to program Bee to use them when collecting honey and nectar.
Lesson 18: Harvesting with Conditionals
  Conditional | Loop | Harvester
  It's not always clear when to use each conditional. This lesson will help you get practice deciding what to do.
Lesson 19: Songwriting
  Unplugged | Function
  Even rockstars need programming skills. This lesson will teach you about functions using lyrics from songs.
Lesson 20: Functions in Minecraft
  Functions | Minecraft
  Can you figure out how to use functions for the most efficient code?
Lesson 21: Functions with Harvester
  Function | Harvester
  Functions will save you lots of work as you help the farmer with her harvest!
Lesson 22: Functions with Artist

Function | Artist
Make complex drawings more easily with functions!

Lesson 23: Screen Out the Mean

Common Sense Education | Cyberbullying | Unplugged
In this lesson, you'll learn what to do if something online makes you feel angry, sad, or scared.

Lesson 24: Determine the Concept

Bee
We aren't giving away any secrets! This lesson could use any of the skills you've learned so far.

Lesson 25: The Big Event

Unplugged | Events
Play a fun game to learn about events.

Lesson 26: Build a Flappy Game

Flappy | Event
Build you own Flappy Bird game however you like, then share it with your friends!

Lesson 27: The Power of Words

Common Sense Education | Cyberbullying
Bullying is never okay. This lesson will teach you about what is and isn't okay to say online.

Lesson 28: Envelope Variables

Unplugged | Variable
Envelopes and variables have something in common: both can hold valuable things. Here you will learn what variables are and the awesome things they can do.

Lesson 29: Variables with Artist

Variable | Artist
Don't forget to bring creativity to class! In these puzzles you will be making fantastic drawings using variables.

Lesson 30: Changing Variables with Bee

Variables | Bee
This bee loves variables!

Lesson 31: Changing Variables with Artist

Variable | Artist
In this lesson, you'll make drawings using variables that change as the program runs.

Lesson 32: For Loop Fun

Unplugged | For Loops
You're going to have loads of fun learning about `for` loops!

Lesson 33: For Loops with Bee

For Loop | Bee
Buzz buzz. In these puzzles you will be guiding a bee to nectar and honey using `for` loops!
Lesson 34: For Loops with Artist
For Loop | Artist
Get ready to make your next masterpiece. Here you will be using `for` loops to make some jaw-dropping pictures.

Lesson 35: Learning Sprite Lab
Event | Sprite Lab
Learn how to create and edit sprites.

Lesson 36: Alien Dance Party
Event | Sprite Lab
Practice making games to share with your friends and family.

Lesson 37: Pet Giraffe
Event | Sprite Lab
Make your own pet giraffe!

Lesson 38: Explore Project Ideas
Project | Define | Prepare | Try | Revise | Reflect
Time to get some inspiration! These puzzles will show you a handful of pre-built games and illustrations to help develop your plan for your BIG project.

Lesson 39: The Design Process
Project
Projects this big take time and plenty planning. Here, you will learn about the design process that you'll use to build your own creation.

Lesson 40: Build Your Project
Project
Finally you can start building your project!

Lesson 41: Revise Your Project
Project
Rome wasn't built in a day and your project shouldn't be, either. Take time to edit and revise your project to make it the best it can be.

Lesson 42: Present Your Project
Project
Time to show your work! Here you will be presenting your awesome project to your peers.

Lesson 43: Binary Bracelets
Unplugged | Binary
Create your very own binary bracelet and learn how computers remember information!

Lesson 44: Binary Images
Binary | Unplugged
Learn how computers store pictures using simple ideas like on and off.
Lesson 45: Binary Images with Artist

Binary | Artist
In this lesson, you will learn how to make images using on and off.

Lesson 46: The Internet

Unplugged | Internet
Ever wondered how information travels across the internet? It's not magic! This lesson will teach you the basics of how the internet works.

Lesson 47: Crowdsourcing

Unplugged | Crowdsourcing
This lesson will teach you about crowdsourcing, the process of building a project with a team.
If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 1: Graph Paper Programming

Overview
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. The class will start by having students use symbols to instruct each other to color squares on graph paper in an effort to reproduce an existing picture. If there’s time, the lesson can conclude with images that the students create themselves.

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.

Agenda
- Warm Up (10 min)
  - Introduction to Graph Paper Programming
- Main Activity (30 min)
  - Practice Together
  - The Students’ Turn
- Wrap Up (15 min)
  - Journaling / Flash Chat
  - Optional Assessment (10 min)
  - Graph Paper Programming - Assessment
- Extended Learning

View on Code Studio

Objectives
Students will be able to:
- Reframe a sequence of steps as an encoded program
- Explain constraints of translating problems from human language to machine language

Preparation
- Watch the Graph Paper Programming - Teacher Video.
- Watch the Graph Paper Programming - Lesson in Action Video.
- Print out one Graph Paper Programming - Worksheet for each group.
- Print one Graph Paper Programming - Assessment for each student.
- Supply each group with several drawing grids, paper, and pens/pencils.
- Make sure every student has a Think Spot Journal - Reflection Journal.

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

For the Teacher
- Graph Paper Programming - Teacher Video
- Graph Paper Programming - Lesson in Action Video
- Graph Paper Programming - Worksheet Answer Key
- Graph Paper Programming - Assessment Answer Key
For the Students

- **Graph Paper Programming** - Unplugged Video (download)
- **Graph Paper Programming** - Worksheet
- **Graph Paper Programming** - Assessment
- **Think Spot Journal** - Reflection Journal

**Vocabulary**

- **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.
Teaching Guide

Warm Up (10 min)

Introduction to Graph Paper Programming

In this 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: 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)

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?

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

Main Activity (30 min)

Practice Together

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: 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

Say: 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 can get a turn.

Display: 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.
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 the students a more complicated image, like the one below.

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.
Discussion: 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:

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 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.

See a sample solution below:
The Students’ Turn

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!

Wrap Up (15 min)

Journaling / Flash Chat

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

Journal Prompts:
What was today’s lesson about?
How did you feel during today’s lesson?
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?

Optional Assessment (10 min)

Graph Paper Programming - Assessment

- 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.

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?

Standards Alignment

CSTA K-12 Computer Science Standards
- AP - Algorithms & Programming

This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0). If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 2: Programming with Angry Birds

Overview

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

Purpose

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

Agenda

Warm Up (4 min)
- Introduction
Bridging Activity - Programming (10 min)
- Unplugged Activity Using Paper Blocks
- Preview of Online Puzzles as a Class
Previewing Online Puzzles as a Class (3 min)
Main Activity (30 min)
- CSF Express Course - Website
Wrap Up (5 - 10 min)
- Journaling
Extended Learning

Objectives

Students will be able to:
- Translate movements into a series of commands.
- Identify and locate bugs in a program.

Preparation

- Play through the CSF Express Course 2018 - Website in stage 2 to find any potential problem areas for your class.
- (Optional) Pick a couple of puzzles to do as a group with your class.
- Make sure every student has aThink Spot Journal - Reflection Journal

Links

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

For the Teacher
- CSF Express Course 2018 - Website
- CS Fundamentals Main Activity Tips - Lesson Recommendations

For the Students
- Graph Paper Programming - Worksheet
- Unplugged Blocks for Maze/Farmer/Bee - Grades 2-5 - Manipulatives (PDF)
- Think Spot Journal - Reflection Journal

Vocabulary

- Algorithm - A list of steps to finish a task.
- Bug - Part of a program that does not work correctly.
• **Debugging** - Finding and fixing problems in an algorithm or program.
• **Sequencing** - Putting commands in correct order so computers can read the commands.
Teaching Guide

Warm Up (4 min)

Introduction
Students will either be learning a lot of new concepts or reviewing a lot of basic concepts. Based on your class's experience, you can cover the following vocabulary or move on to a bridging activity. We recommend using the words in sentences if the definitions aren't explicitly covered.

Bridging Activity - Programming (10 min)

This activity will help bring the unplugged concepts from "Graph Paper Programming" into the online world that the students are moving into. Choose one of the following to do with your class:

Unplugged Activity Using Paper Blocks
Similar to "Graph Paper Programming", have the students in your class pair up. Pass out multiple fill 1 and move ___ blocks from the Unplugged Blocks for Maze/Farmer/Bee - Grades 2-5 - Manipulatives (PDF) to each pair. Have each pair of students draw a design on a four by four graph from Graph Paper Programming - Worksheet. Next, have the students work together to write the program needed to draw this design using the paper Blockly blocks. The students will need to write up, down, right, or left on the move ___ block. Make sure the students know that the program goes from top to bottom and the blocks need to touch!

Preview of Online Puzzles as a Class
Pull up a puzzle from CSF Express Course 2018 - Website. We recommend puzzle 5. Break up the students into groups of three or four. Have them "program" Red, the angry bird, to get to the pig using arrows from "Graph Paper Programming."

The class will not need to use the last arrow.
Once all the groups have an answer, discuss the path as a class.

Previewing Online Puzzles as a Class (3 min)
Students should now be ready to see a real puzzle in action!

Model: Pull up Lesson 2, Puzzle 5 to do in front of the class. This will be the same puzzle that they just saw in the bridging activity. While working through this puzzle with the class, remind students that making mistakes is okay and remind them that the only way to be successful is to be persistent. Remind them that finding the solution might be harder with limitations, but that you believe in them.

Discuss: Does anyone remember how to solve this puzzle?
As the teacher, you should decide if you will have the students remind you how to solve it from their seats, or come to the
computer to drag the actual blocks in one-by-one.

**Transition:** Now that students have seen an online puzzle in practice, they should be ready to start solving puzzles of their own. Continue to the lab or bring out their classroom machines.

## Main Activity (30 min)

### CSF Express Course - Website

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

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

**Discuss:** After providing students with end-of-class warnings, grab everyone’s attention and get them to reflect on the experiences that they just had.

- Did anyone feel frustrated during any of the puzzles?
- Did anyone notice the need to be persistent?

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

## Wrap Up (5 - 10 min)

### Journaling

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

**Journal Prompts:**

- What was today’s lesson about?
- How did you feel during today’s lesson?
- Draw an activity you like to do that you struggled with the first time. Draw or describe how you got better.

## Extended Learning

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

### Create Your Own

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

CSTA K-12 Computer Science Standards

► AP - Algorithms & Programming

If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 3: Relay Programming

Overview

This activity will begin with a short lesson on debugging and persistence, then will quickly move to a race against the clock as students break into teams and work together to write a program one instruction at a time.

Purpose

Teamwork is very important in computer science. Teams write and debug code with each other, instead of working as individuals. In this lesson, students will learn to work together while being as efficient as possible.

This activity also provides a sense of urgency that will teach students to balance their time carefully and avoid mistakes without falling too far behind. This experience can be stressful (which is expected!) Make sure you provide students with the tools to deal with potential frustration.

Agenda

- Warm Up (15 min)
  - Where did I go wrong?
- Main Activity (20 min)
  - Relay Programming
- Wrap Up (15 min)
  - Journaling
  - Extended Learning

Objectives

Students will be able to:
- Define ideas using code and symbols.
- Verify work done by teammates.
- Identify signs of frustration

Preparation

- Watch the Relay Programming - Teacher Video.
- Locate a wide open space for this activity, such as the gym or outdoor field.
- Print out one Relay Programming Activity Packet - Activity Packet for each group.
- Supply each group with plenty of paper and pens/pencils.
- Print one Relay Programming - Assessment for each student.
- Make sure every student has a Think Spot Journal - Reflection Journal.

Links

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

For the Teacher
- Relay Programming - Teacher Debugging - Image
- Relay Programming - Unplugged Video (download)
- Relay Programming - Teacher Video

For the Students
- Relay Programming Debugging Packet
- Relay Programming Activity Packet
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.
- **Frustrated** - Feeling annoyed or angry because something is not the way you want it.
- **Persistence** - Trying again and again, even when something is very hard.
- **Program** - An algorithm that has been coded into something that can be run by a machine.
Teaching Guide

Warm Up (15 min)

Recall that in "Graph Paper Programming" we guided our teammate's Automatic Realization Machine (ARM) using arrows. This warm up will bring back these ideas, which will be needed in the main activity.

Where did I go wrong?

**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. Once they've 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.

**Display:** Show your students the provided Relay Programming - Teacher Debugging - Image.

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. Your program has a bug, can they help you fix it?

Take a moment to walk them through the rules: - Start at the star - Follow the instructions step-by-step - End when all of the right squares are filled in

Optional: Follow along by filling in a blank grid. Express frustration when the picture doesn't turn out the way that you wanted it to.

**Think:** 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!

**Distribute:** To make sure that students understand the idea of finding and fixing errors (debugging) pass out the Relay Programming Debugging Packet 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 min)
Relay Programming

With Graph Paper Programming in mind, it's time to split up into teams and prepare to run the activity as a relay!

**Set-Up:** Prepare the Relay Programming Activity Packet - Activity Packet by printing out one copy for each team of 4-5 students. Cut or fold each page along the center dotted line.

Go over the rules of the game with your class:
- Divide students into groups of 3-5.
- Have each group queue up relay-style.
- Place an identical image at the other side of the room/gym/field from each team.
- Have the first student in line dash over to the image, review it, and write down the first symbol in the program to reproduce that image.
- The first student then runs back and tags the next person in line, then goes to the back of the queue.
- The next person in line dashes to the image, reviews the image, reviews the program that has already been written, then either debugs the program by crossing out an incorrect symbol, or adds a new one. That student then dashes back to tag the next person, and the process continues until one group has finished their program.

First group to finish with a program that matches the image is the winner! Play through this several times, with images of increasing difficulty.

Go through the game as many times as you can before time runs out or your students begin feeling exhausted.

**Transition:** Once the game is over, circle everyone up to share lessons learned.

**Discuss:** What did we learn today?
- What if each person on a team were allowed to do five arrows at a time?
  - How important would it be to debug our own work and the work of the programmer before us?
  - How about with 10 arrows?
  - 10,000? Would it be more or less important?
- Do you think a program is better or worse when more than one person has worked on it?
- Do you think people make more or fewer mistakes when they're in a hurry?
- If you find a mistake, do you have to throw out the entire program and start over?

**Wrap Up (15 min)**

**Journaling**

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

**Journal Prompts:**

---

**Clarifications**

Here are some clarifications that need to be shared from time to time:
- Only one person from each group can be at the image at one time.
- It is okay to discuss algorithms with the rest of the group in line, even up to the point of planning who is going to write what when they get to the image.
- When a student debugs a program by crossing out an incorrect instruction (or a grouping of incorrect instructions) this counts as their entire turn. The next player will need to figure out how to correct the removed item.

---

**Content Corner**

For more on persistence and frustration, try reading Stevie and the Big Project to your students. It will help them spot moments of frustration. It will also help give them the tools to deal with it.

If you do not read the book, 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
• What was today's lesson about?
• How did you feel during today's lesson?
• How did teamwork play a role in the success of writing today's program?
• Did you start to get frustrated at any point? What did you do about it?

Extended Learning

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

Pass the paper

• If you don’t have the time or room for a relay, you can have students pass the paper around their desk grouping, each writing one arrow before they move the paper along.

Fill It, Move It

• As the teacher, draw an image with as many filled squares as children in each group.
• Have the students write as many arrows in the program as it takes to get to a filled-in square (including actually filling that square in) before passing to the next person.

Debugging Together

Draw an image on the board. Have each student create a program for the image. Ask students to trade with their elbow partner and debug each other’s code.

• Circle the first incorrect step, then pass it back.
• Give the students another chance to review and debug their own work.
• Ask for a volunteer to share their program.

Ask the class:

• How many students had the same program?
• Anyone have something different?

Standards Alignment

CSTA K-12 Computer Science Standards

➤ AP - Algorithms & Programming

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

If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 4: Debugging with Scrat

Debugging | Bug | Maze

Overview
Debugging is an essential element of learning to program. In this lesson, students will encounter puzzles that have been solved incorrectly. They will need to step through the existing code to identify errors, including incorrect loops, missing blocks, extra blocks, and blocks that are out of order.

Purpose
Students in your class might become frustrated with this lesson because of the essence of debugging. Debugging is a concept that is very important to computer programming. Computer scientists have to get really good at facing the bugs in their own programs. Debugging forces the students to recognize problems and overcome them while building critical thinking and problem solving skills.

Agenda
- Warm Up (15 min)
  - Introduction
  - Vocabulary
- Main Activity (30 min)
  - CSF Express Course - Website
- Wrap Up (5 - 10 min)
  - Journaling
  - Extended Learning

Objectives
Students will be able to:
- Predict where a program will fail.
- Modify an existing program to solve errors.
- Reflect on the debugging process in an age-appropriate way.

Preparation
- Play through the CSF Express Course 2018 - Website in stage 4 to find any potential problem areas for your class.
- (Optional) Pick a couple of puzzles to do as a group with your class.
- Review CS Fundamentals Main Activity Tips - Lesson Recommendations.
- Make sure every student has a Think Spot Journal - Reflection Journal.
- Review Debugging Recipe - Student Handout with the class.

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

For the Teacher
- CS Fundamentals Main Activity Tips - Lesson Recommendations

For the Students
- Debugging Recipe - Student Handout
- Think Spot Journal - Reflection Journal

Vocabulary
- Bug - Part of a program that does not work correctly.
- **Debugging** - Finding and fixing problems in an algorithm or program.
Teaching Guide

Warm Up (15 min)

Introduction
Ask students to think about problems they have to solve in everyday life.

- How do you fix something that isn't working?
- Do you follow a specific series of steps?
- The puzzles in this unit have already been solved for you (yay!), but they don't seem to be working (boo!)
- We call the problems in these programs "bugs," and it will be your job to "debug" them.

Vocabulary
This lesson has three new and important vocabulary words:

- **Bug** - Say it with me - Buhh-g. Something that is going wrong. An error.
- **Debugging** - Say it with me: Dee-bug-ing. To find and fix errors.
- **Persistence** - Say it with me: Purr-siss-tense. Not giving up. Persistence works best when you try things many different ways, many different times.

Say:
Debugging is a process. First, you must recognize that there is an error in your program. You then work through the program step by step to find the error. Try the first step, did it work? Then the second, how about now? If you make sure that everything is working line by line, then when you get to the place that your code isn't doing what it's supposed to, you know that you've found a bug. Once you've discovered your bug, you can work to fix (or "debug") it!

If you think it will build excitement in the class you can introduce the character of today's puzzles, Scrat from Ice Age. If students aren't familiar with Scrat, show some videos of the quirky squirrel running into trouble.

Main Activity (30 min)

CSF Express Course - Website
Before letting the students start on the computer, remind them of the advantages of Pair Programming - Student Video and asking their peers for help. Sit students in pairs and recommend they ask at least two peers for help before they come to a teacher.

As mentioned in the purpose of this lesson, make sure the students are aware that they will face frustrating puzzles. Tell them it is okay to feel frustrated, but it is important to work through the problem and ask for help. As the students work through the puzzles, walk around to make sure no student is feeling so stuck that they aren't willing to continue anymore.

Wrap Up (5 - 10 min)

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

Journal Prompts:

- What was today’s lesson about?
- How did you feel during today’s lesson?
- What kind of bugs did you find today?
• Draw a bug you encountered in one of the puzzles today. What did you do to "debug" the program?

Extended Learning

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

Planting bugs

Have students go back through previous levels, purposefully adding bugs to their solutions. They can then ask other students to debug their work. This can also be done with paper puzzles.

When other students are debugging, make sure that the criticisms are constructive. If this could be a problem for your class, go over respectful debugging before this activity by role playing with another student.

Standards Alignment

CSTA K-12 Computer Science Standards

▸ AP - Algorithms & Programming

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

If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 5: Collecting Treasure with Laurel

Overview
In this series of puzzles, students will continue to develop their understanding of algorithms and debugging. With a new character, Laurel the Adventurer, students will create sequential algorithms to get Laurel to pick up treasure as she walks along a path.

Purpose
In this lesson, students will be practicing their programming skills using a new character, Laurel the Adventurer. When someone starts programming they piece together instructions in a specific order using something that a machine can read. Through the use of programming, students will develop an understanding of how a computer navigates instructions and order. Using a new character with a different puzzle objective will help students widen their scope of experience with sequencing and algorithms in programming.

Agenda
- Warm Up (5 min)
  - Introduction
- Bridging Activity - Programming (10 min)
  - Previewing Online Puzzles as a Class
- Main Activity (30 min)
  - CSF Express Course - Website
  - Wrap Up (5 - 10 min)
  - Journaling

View on Code Studio

Objectives
Students will be able to:
- Order movement commands as sequential steps in a program.
- Represent an algorithm as a computer program.
- Develop problem solving and critical thinking skills by reviewing debugging practices.

Preparation
- Play through CSF Express Course 2018 - Website stage 5 to find potential problem areas for your class.
- Review CS Fundamentals Main Activity Tips - Lesson Recommendations.
- Make sure every student has a Think Spot Journal - Reflection Journal.

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

For the Teacher
- CSF Express Course 2018 - Website
- CS Fundamentals Main Activity Tips - Lesson Recommendations

For the Students
- Unplugged Blockly Blocks (Grades 2 - 5) - Manipulatives (download)
- Think Spot Journal - Reflection Journal

Vocabulary
- 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.

- **Programming** - The art of creating a program.
Teaching Guide

Warm Up (5 min)

Introduction
This lesson uses most of the same blocks from Course C, Stage 2 Programming in Maze and adds the ability to collect.
Tell the students that this block will allow Laurel the Adventurer to pick up the treasure that she is standing over. This new block will be discussed more in the bridging activity.

Bridging Activity - Programming (10 min)
This activity will help bring the unplugged concepts from Real-Life Algorithms: Paper Airplanes into the online world that the students are moving into.

Previewing Online Puzzles as a Class
Pull a puzzle from the corresponding online stage. We recommend puzzle 7. Have students discuss a pattern that they think will get Laurel the Adventurer to collect all the treasure. Ask the students to share. See how many other students had the same answer!

Main Activity (30 min)

CSF Express Course - Website
Laurel the Adventurer is looking to collect as much treasure as she can. Instruct the students to traverse the puzzle to collect whatever they can. Some levels will require you to only pick up one piece of treasure, but others will require you to pick up every piece of treasure. Pay attention to the instructions to know what to do!

Wrap Up (5 - 10 min)

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

Journal Prompts:
- What was today’s lesson about?
- How did you feel during today’s lesson?
- Draw a maze that you might solve with the blocks you used today.

Standards Alignment

CSTA K-12 Computer Science Standards
- AP - Algorithms & Programming

This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).
If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 6: Creating Art with Code

Overview

In this lesson, students will take control of the Artist to complete drawings on the screen. This Artist stage will allow students to create images of increasing complexity using new blocks like `move forward by 100 pixels` and `turn right by 90 degrees`.

Purpose

Building off of the students' previous experience with sequencing, this lesson will work to inspire more creativity with coding. The purpose of this lesson is to solidify knowledge on sequencing by introducing new blocks and goals. In this case, students learn more about pixels and angles using the new blocks, while still practicing their sequencing skills. Also, students will be able to visualize new goals such as coding the Artist to draw a square.

Agenda

- Warm Up (10 min)
  - Introduction
- Main Activity (30 min)
  - CSF Express Course - Website
- Wrap Up (10 - 15 min)
  - Journaling
- Extended Learning

Objectives

Students will be able to:
- Create a program to complete an image using sequential steps.
- Break complex shapes into simple parts.

Preparation

- Play through the CSF Express Course 2018 - Website in stage 6 to find any potential problem areas for your class.
- Review CS Fundamentals Main Activity Tips - Lesson Recommendations.
- (Optional) Obtain protractors for your class to visualize the angles they must use to complete the puzzles.
- Print one Turns & Angles - Student Handout for each student.

Links

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

For the Teacher
- CSF Express Course 2018 - Website
- CS Fundamentals Main Activity Tips - Lesson Recommendations

For the Students
- Artist Introduction - Student Video
- Turns & Angles - Student Video
- Turns & Angles - Student Handout
- Think Spot Journal - Reflection Journal
Teaching Guide

Warm Up (10 min)

Introduction
Show the students one or both of the following videos as an introduction to angles:

**Artist Introduction - Student Video** (1.5 minutes long)

**Turns & Angles - Student Video** (2 minutes long)

Use the **Turns & Angles - Student Handout** to show the students interior versus exterior angles for different shapes.

This document can be used as a hand out or you can choose to print it out as a poster for students to refer to.

**Ask:**
Discuss the square and triangle shapes from the document.

- How would you code a computer to draw that shape?
- What order do the instructions need to be in?

Tell the students that in these puzzles they will be moving a character who leaves a line everywhere he goes. The students will be writing code that gets the character to draw various shapes, including a square.

Main Activity (30 min)

**CSF Express Course - Website**

In this set of puzzles, the artist will no longer be constrained to 90 degree angles. Having physical protractors available can be help students better visualize the angles they need. Otherwise, the stage provides images of the angles as the student selects which angle to use. (Please note: Angle choices are limited to two inside of the dropdown menu, reducing the number of options students have to work through.)

Before sending the students to the computers to work on the puzzles, it might be beneficial to give a brief presentation of how to use the tools in this level. We recommend puzzle 5 as a good puzzle to show how to use the protractor online.

The eighth puzzle asks the students to draw a 6 sided polygon. This might be challenging for some students. We recommend getting the students to try a few times, ask a peer, then ask the teacher for help. Below is an image that might be helpful for the students.

![Image of a hexagon with various angle measurements]

Wrap Up (10 - 15 min)

**Journaling**

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

**Journal Prompts:**
What was today’s lesson about?
How did you feel during today’s lesson?
What are the interior angles that make up a square. What about for a triangle?
Sketch a simple shape on your paper and imagine the code used to draw it. Can you write that code out next to the shape?

**Extended Learning**

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

**The Copy Machine**

- Give students two pieces of paper
- On one sheet draw a simple image, using straight lines only.
- On the second sheet draw instructions for recreating that image commands to move straight and turn at various angles.
- Trade instruction sheets and attempt to recreate the image using only the provided instructions.

**Standards Alignment**

**CSTA K-12 Computer Science Standards**

- AP - Algorithms & Programming

If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 7: Getting Loopy

Unplugged | Loops

Overview

As we start to write longer and more interesting programs, our code often contains a lot of repetition. In this lesson, students will learn about how loops can be used to more easily communicate instructions that have a lot of repetition by looking at the repeated patterns of movement in 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.

Agenda

- Warm Up (5 min)
  - Repeat After Me
- Main Activity (15 min)
- Assessment (10 min)
- Wrap-Up (15 min)
  - Vocabulary
  - Flash Chat: What did we learn?
  - Journaling
- Extended Learning
  - So Moving
  - Connect It Back

Objectives

Students will be able to:
- Repeat actions initiated by the instructor.
- Translate a picture program into a real-world dance.
- Convert a series of multiple actions into a single loop.

Preparation

- Watch the Getting Loopy - Teacher Video.
- Print one Getting Loopy - Worksheet for the class.
- Print one Getting Loopy - Assessment for each student.
- Make sure every student has a Think Spot Journal - Reflection Journal.

Links

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

For the Teacher

- Getting Loopy - Unplugged Video (download)
- Getting Loopy - Teacher Video
- Getting Loopy - Assessment Answer Key

For the Students

- Getting Loopy - Worksheet
- Getting Loopy - Assessment
- Think Spot Journal - Reflection Journal

Vocabulary

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

- **Repeat** - Do something again
Lesson 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.

Teaching Guide

Warm Up (5 min)

Repeat After Me

**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 min)

**Dance Party**

**Say:** 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 Getting Loopy - 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:

Assessment (10 min)

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

Distribute: Hand out the Getting Loopy - 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 min)

Vocabulary
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?

**Journaling**

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

**Journal Prompts:**

- What was today’s lesson about?
- 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!

**Standards Alignment**

CSTA K-12 Computer Science Standards

- **AP** - Algorithms & Programming

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

If you are interested in licensing Code.org materials for commercial purposes, [contact us].
Lesson 8: Loops with Rey and BB-8

Loops | Maze

Overview
Building on the concept of repeating instructions from “Getting Loopy,” this stage will have students using loops to help BB-8 traverse a maze more efficiently than before.

Purpose
In this lesson, students will be learning more about loops and how to implement them in Blockly code. Using loops is an important skill in programming because manually repeating commands is tedious and inefficient. With the Code.org puzzles, students will learn to add instructions to existing loops, gather repeated code into loops, and recognize patterns that need to be looped. It should be noted that students will face puzzles with many different solutions. This will open up discussions on the various ways to solve puzzles with advantages and disadvantages to each approach.

Agenda
- Warm Up (5 min)
  - Introduction
- Bridging Activity - Loops (10 min)
  - Unplugged Activity Using Paper Blocks
  - Previewing Online Puzzles as a Class
- Main Activity (30 min)
  - CSF Express Course - Website
- Wrap Up (5 - 10 min)
  - Journaling
- Extended Learning

View on Code Studio

Objectives
Students will be able to:
- Identify the benefits of using a loop structure instead of manual repetition.
- Break down a long sequence of instructions into the largest repeatable sequence.
- Employ a combination of sequential and looped commands to reach the end of a maze.

Preparation
- Play through the CSF Express Course 2018 - Website in stage 8 to determine if there will be any problem areas for your class.
- Review CS Fundamentals Main Activity Tips - Lesson Recommendations.
- Make sure every student has a Think Spot Journal - Reflection Journal.

Links

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

For the Teacher
- CSF Express Course 2018 - Website
- CS Fundamentals Main Activity Tips - Lesson Recommendations

For the Students
- Unplugged Blockly Blocks (Grades 2 - 5) - Manipulatives (download)
- Think Spot Journal - Reflection Journal

Vocabulary
- **Loop** - The action of doing something over and over again.
- **Repeat** - Do something again
Teaching Guide

Warm Up (5 min)

Introduction
Review with students the "Getting Loopy" activity:

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

Quickly show the students a dance (with repeated steps) for the rest of the class to do. Ask the rest of the class to find the loops within the dance and point them out.

If you're comfortable, give an introduction to BB-8 from Star Wars. Many children may be familiar with the lovable robot, but an introduction will surely build excitement.

Bridging Activity - Loops (10 min)

This activity will help bring the unplugged concepts from "Getting Loopy" into the online world that the students are moving into. Choose one of the following to do with your class:

Unplugged Activity Using Paper Blocks
Revisit the dance from "Getting Loopy." This time, work with the class to "code" it out using Unplugged Blockly Blocks (Grades 2 - 5) - Manipulatives instead of writing the dance instructions on paper. Make sure the students know that the blocks need to go from top to bottom and they all need to touch!

Previewing Online Puzzles as a Class
Pull up the online puzzles and choose a puzzle to do in front of the class. We recommend puzzle 8 for its staircase pattern. Ask the students to write a program to solve the puzzle on paper. Have the students circle repeated chunks and label with the number of repeats, the same way they did in "Getting Loopy."

Main Activity (30 min)

CSF Express Course - Website
As students work through the puzzles, see if they can figure out how many blocks they use with a loop vs. not using a loop. Pair Programming - Student Video works really well with this set of puzzles because there are a few ways to fill the loops. Push for friendly discussion between pairs in instances of disagreement on how to solve the puzzle. Have the students ask each other questions like:

- How did you come up with that solution?
- What are some benefits of solving the puzzle that way?

We also recommend having paper on hand for students to write out their code and find any repetition to use in loops.

Wrap Up (5 - 10 min)

Journaling
Having students write about what they learned, why it’s useful, and how they feel about it can help solidify any knowledge they obtained today and build a review sheet for them to look to in the future.
Journal Prompts:
- What was today’s lesson about?
- How did you feel during today’s lesson?
- How did loops make your program easier to write?
- Think of something that repeats over and over again. What might the program for that look like?

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!

Standards Alignment

CSTA K-12 Computer Science Standards
- AP - Algorithms & Programming

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

If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 9: Sticker Art with Loops

Overview
Watch student faces light up as they make their own gorgeous designs using a small number of blocks and digital stickers! This lesson builds on the understanding of loops from previous lessons and gives students a chance to be truly creative. This activity is fantastic for producing artifacts for portfolios or parent/teacher conferences.

Purpose
This series highlights the power of loops with creative and personal designs.

Offered as a project-backed sequence, this progression will allow students to build on top of their own work and create amazing artifacts.

Agenda
- Warm Up (15 min)
  - Introduction
- Main Activity (30 min)
  - CSF Express Course - Website
- Wrap Up (15 min)
  - Journaling

View on Code Studio

Objectives
Students will be able to:
- Identify the benefits of using a loop structure instead of manual repetition.
- Differentiate between commands that need to be repeated in loops and commands that should be used on their own.

Preparation
- Play through the CSF Express Course 2018 - Website in lesson 9 to find any potential problem areas for your class.
- Review CS Fundamentals Main Activity Tips - Lesson Recommendations.
- Make sure every student has a Think Spot Journal - Reflection Journal.

Links

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

For the Teacher
- CSF Express Course 2018 - Website
- CS Fundamentals Main Activity Tips - Lesson Recommendations

For the Students
- Think Spot Journal - Reflection Journal

Vocabulary
- Loop - The action of doing something over and over again.
- Repeat - Do something again
Teaching Guide

Warm Up (15 min)

Introduction

Students should have had plenty of introduction to loops at this point. Based on what you think your class could benefit from, we recommend:

- Creating a new dance with loops just like in “Getting Loopy”
- As a class, playing through a puzzle from the last lesson, “Loops with Rey and BB-8”
- Reviewing how to use Artist by playing through a puzzle from “Programming in Artist”
- Previewing a puzzle from this lesson

All of these options will either review loops or the artist, which will help prepare your class for fun with the online puzzles!

Main Activity (30 min)

CSF Express Course - Website

Some students may discover where to add repeat loops by writing out the program without loops then circling sections of repetitions. If the students in your class seem like they could benefit from this, have them keep paper and pencils beside them at their machines. Students might also enjoy drawing some of the shapes and figures on paper before they program it online. (When drawing stamps, it can be easier to symbolize those with simple shapes like circles and squares.)

Wrap Up (15 min)

Journaling

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

Journal Prompts:

- What was today’s lesson about?
- How did you feel during today’s lesson?
- What was the coolest shape or figure you programmed today? Draw it out!
- What is another shape or figure you would like to program? Can you come up with the code to create it?

Standards Alignment

CSTA K-12 Computer Science Standards

► AP - Algorithms & Programming

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

If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 10: Digital Sharing

Overview
Loaned to Computer Science Fundamentals by the team over at Copyright and Creativity, this lesson exists to help students understand the challenges and benefits of respecting ownership and copyright, particularly in digital environments. Students should be encouraged to respect artists’ rights as an important part of being an ethical digital citizen.

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.

Agenda
Warm-Up (Optional) (15 min)
- Write a Character Sketch
Ethical Sharing (30 min)
- To Share or Not to Share?
  - Okay to Share
  - Not Okay to Share
Wrap-Up (10 min)
- Journaling / Flash Chat
Extended Learning

View on Code Studio

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 at Digital Sharing Ethics (Video) - Video
- 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.)
- 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 Teacher
- Digital Sharing Lesson Plan

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 min)

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

**Watch:** Have students watch the one-minute [Digital Sharing Ethics (Video) - Video](#). You may need to play it two or three times.

**Write a Character Sketch**
Ask students to 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 his/her name?
- Who are his/her friends?
- How long have they known each other?
- Who are the people in his/her family? What are they like?
- What’s his/her backstory?
- Where does he/she live?
- Where did he/she used to live?
- What exciting thing might have happened to them back in kindergarten, first grade, etc.?
- What is he/she looking forward to?
- What is he/she afraid of?

**NOTE:** 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.

**Ethical Sharing (30 min)**

**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 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.**

**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.
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.

Think: Have students think back to the art creation prompt where they first sat down. 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.

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?

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.

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

As we watch the video, decide if the music is OK to share or not.

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!

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 min)
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.

**Discussion**

**Goal**

Okay to Share

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.”]

**Not Okay to Share**

- 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.]

**Journaling / Flash Chat**

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

**Journal Prompts:**

- What was today's lesson about?
- How did you feel during today’s lesson?
- 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.
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.

Standards Alignment

CSTA K-12 Computer Science Standards

▶ AP - Algorithms & Programming

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

If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 11: Looking Ahead with Minecraft

Conditional | Minecraft

Overview
This lesson was originally created for the Hour of Code, alongside the Minecraft team. Students will get the chance to practice ideas that they have learned up to this point, as well as getting a sneak peek at conditionals!

Purpose
This set of puzzles will work to solidify and build on the knowledge of loops, and introduce conditionals. By pairing these two concepts together, students will be able to explore the potential for creating fun and innovative programs in a new and exciting environment.

Agenda
- Warm Up (15 min)
  - Introduction
- Main Activity (30 min)
  - CSF Express Course - Website
- Wrap Up (15 min)
  - Journaling
- Extended Learning

View on Code Studio

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.

Preparation
- Play through the CSF Express Course 2018 - Website associated with lesson 11 to find any potential problem areas for your class.
- Review CS Fundamentals Main Activity Tips - Lesson Recommendations.
- Make sure every student has a Think Spot Journal - Reflection Journal.

Links

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

For the Teacher
- CSF Express Course 2018 - Website
- CS Fundamentals Main Activity Tips - Lesson Recommendations

For the Students
- Think Spot Journal - Reflection Journal

Vocabulary
- Condition - A statement that a program checks to see if it is true or false. If true, an action is taken. Otherwise, the action is ignored.
- Conditionals - Statements that only run
under certain conditions.
Teaching Guide

Warm Up (15 min)

Introduction
Gather the class together and ask two volunteers to walk straight in some direction in the classroom. If they encounter a chair out of place, they must step over it. If they reach a wall, they must sit down.

Once all of the students are sitting down, ask how you would program a robot to respond to a wall or a chair. Remind students that you cannot simply say "Step over chair" unless you know there is a chair, and you will not always know there is a chair. It might be helpful to translate the task into instructions like:

- while there is a path ahead
  - walk forward
  - if there is a chair, step over it
- sit down

Tell students they will be using **conditionals** during this lesson. Give the definition of:

- **Condition**: A statement that a program checks to see if it is true or false. If true, an action is taken. Otherwise, the action is ignored.
- **Conditionals**: Statements that only run under certain conditions.

Open up a discussion of when you might use a conditional in your code.

Main Activity (30 min)

CSF Express Course - Website
Students are in for a real treat with this lesson. It's likely most of your students have heard of Minecraft, but give a brief introduction for those that may not know.

Minecraft is a game of cubes. You can play as Alex or Steve as you work through mazes. You'll need to avoid lava, pick up items, and explore in a world made up of cubes of things.

Demonstrate one of the puzzles to the class (we recommend puzzle 11.) Once all questions have been addressed, transition students to computers and let them start pair programming.

Wrap Up (15 min)

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

Journal Prompts:
- Draw a feeling face to show how you felt during today’s lesson.
- Draw something else you could have built in this minecraft world.
- Can you draw a scene where someone is using a conditional?

Extended Learning

More Minecraft
If you find that your class really enjoys the Minecraft environment, here are some links to other Minecraft games they can play online. These games will also teach basic coding skills.

Standards Alignment

CSTA K-12 Computer Science Standards

- AP - Algorithms & Programming

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

If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 12: Nested Loops in Maze

Nested Loops | Loops | Bee | Maze

Overview

In this online activity, students will have the opportunity to push their understanding of loops to a whole new level. Playing with the Bee and Plants vs Zombies, students will learn how to program a loop to be inside of another loop. They will also be encouraged to figure out how little changes in either loop will affect their program when they click Run.

Purpose

In this introduction to nested loops, students will go outside of their comfort zone to create more efficient solutions to puzzles.

In earlier puzzles, loops pushed students to recognize repetition. Here, students will learn to recognize patterns within repeated patterns to develop these nested loops. This stage starts off by encouraging students try to solve a puzzle where the code is irritating and complex to write out the long way. After a video introduces nested loops, students are shown an example and asked to predict what will happen when a loop is put inside of another loop. This progression leads into plenty of practice for students to solidify and build on their understanding of looping in programming.

Agenda

<table>
<thead>
<tr>
<th>Warm Up (10 min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Main Activity (30 min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSF Express Course - Website</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Wrap Up (15 min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Journaling</td>
</tr>
</tbody>
</table>

View on Code Studio

Objectives

Students will be able to:
- Break complex tasks into smaller repeatable sections.
- Recognize large repeated patterns as made from smaller repeated patterns.
- Identify the benefits of using a loop structure instead of manual repetition.

Preparation

- Play through CSF Express Course 2018 - Website in lesson 12 to find any potential problem areas for your class.
- Review CS Fundamentals Main Activity Tips - Lesson Recommendations.
- Make sure every student has a Think Spot Journal - Reflection Journal.

Links

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

For the Teacher
- CSF Express Course 2018 - Website
- CS Fundamentals Main Activity Tips - Lesson Recommendations

For the Students
- Think Spot Journal - Reflection Journal

Vocabulary

- Command - An instruction for the computer. Many commands put together make up algorithms and computer programs.
- Loop - The action of doing something over
and over again.

- **Repeat** - Do something again
Teaching Guide

Warm Up (10 min)

Introduction

Briefly review with the class what loops are and why we use them.

- What do loops do?
  - Loops repeat a set of commands. (see vocabulary on command if students don't recognize it)
- How do we use loops?
  - We use loops to create a pattern made of repeated actions.

Tell the class that they will now be doing something super cool: using loops inside loops. Ask the class to predict what kinds of things we would be using a loop inside of a loop for.

"If a loop repeats a pattern, then looping a loop would repeat a pattern of patterns!"

Students don't need to understand this right away, so feel free to move on to the online puzzles even if students still seem a little confused.

Main Activity (30 min)

CSF Express Course - Website

We highly recommend Pair Programming - Student Video in this lesson. This may not be an easy topic for the majority of your students. Working with a partner and discussing potential solutions to the puzzles might ease the students' minds.

Also, have paper and pencils nearby for students to write out their plan before coding. Some puzzles have a limit on the number of certain blocks you can use, so if students like to write out the long answer to find the repeats, paper can be useful.

Wrap Up (15 min)

Journaling

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

Journal Prompts:
- What was today's lesson about?
- How did you feel about today's lesson?
- What is a nested loop?
- Can you draw a puzzle that would use a nested loop? Try coding the solution to your own puzzle.

Standards Alignment

CSTA K-12 Computer Science Standards

- AP - Algorithms & Programming
If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 13: Snowflakes with Anna and Elsa

Loop | Nested Loop | Frozen

Overview

Now that students know how to layer their loops, they can create so many beautiful things. This lesson will take students through a series of exercises to help them create their own portfolio-ready images using Anna and Elsa's excellent ice-skating skills!

Purpose

In this series, students will get practice nesting loops while creating images that they will be excited to share.

Beginning with a handful of instructions, students will make their own decisions when it comes to creating designs for repetition. They will then spin those around a variety of ways to end up with a work of art that is truly unique.

Agenda

Warm Up (15)
  Introduction
Main Activity (30)
  CSF Express Course - Website
Wrap Up (15)
  Journaling

View on Code Studio

Objectives

Students will be able to:

- Break apart code into the largest repeatable sequences using both loops and nested loops.
- Recognize the difference between using a loop and a nested loop.
- Describe when a loop, nested loop, or no loop is needed.

Preparation

Play through the CSF Express Course 2018 - Website corresponding to this lesson to find and potential problem areas for your class.

Review CS Fundamentals Main Activity Tips - Lesson Recommendations.

Make sure every student has a Think Spot Journal - Reflection Journal.

Links

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

For the Teacher

- CSF Express Course 2018 - Website
- CS Fundamentals Main Activity Tips - Lesson Recommendations

For the Students

- Think Spot Journal - Reflection Journal

Vocabulary

- Loop - The action of doing something over and over again.
- **Repeat** - Do something again
Teaching Guide

Warm Up (15)

Introduction
Ask the class to discuss the last set of puzzles.
- What did they like/dislike?
- Which puzzles were hard? Why?
- Which puzzles were easy? Why?
- If you were to teach nested loops to a friend, what would you say to help them understand?

If there's time, give an introduction to the main characters of today's puzzles, Anna and Elsa from Frozen. Give the class the sister's back story if the class doesn't already know. To build excitement, tell the class they will be using nested loops to make some fantastic drawings with Anna and Elsa's ice skates!

Main Activity (30)

CSF Express Course - Website
This set of puzzles is set up as a progression. This means every puzzle builds a foundation for the next puzzle. Students will enjoy making more and more interesting designs by making small and simple changes to code they have already written.

Wrap Up (15)

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

Journal Prompts:
- What was today's lesson about?
- How did you feel during today's lesson?
- When do you use a loop? When do you use a nested loop?
- Thought exercise: Can you make everything a nested loop can with just a normal loop? Can you draw out an example?
  - Answer: Yes, you can, but it is a lot more difficult. Nested loops make programs simpler.

Standards Alignment

CSTA K-12 Computer Science Standards
- AP - Algorithms & Programming

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

If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 14: While Loops with the Farmer

While Loops | Loops | Farmer

Overview

By the time students reach this lesson, they should already have plenty of practice using repeat loops, so now it's time to mix things up.

**While loops** are loops that continue to repeat commands while a condition is met. While loops are used when the programmer doesn't know the exact number of times commands need to be repeated, but does know what condition needs to be true in order for the loop to continue repeating. For example, students will be working to fill holes and dig dirt in Farmer. They will not know the size of the holes or the height of the mountains of dirt, but the students will know they need to keep filling the holes and digging the dirt as long as the ground is not flat.

Purpose

As your students continue to deepen their knowledge of loops, they will come across problems where a command needs to be repeated, but it is unknown how many times it needs to be repeated. This is where while loops come in. In today’s lesson, students will develop a beginner’s understanding of condition-based loops and also expand their knowledge of loops in general.

Agenda

- **Warm Up (10 min)**
  - Introduction
- **Main Activity (30 min)**
  - CSF Express Course - Website
- **Wrap Up (15 min)**
  - Journaling
- **Extended Learning**
  - Bridging Activity (15 min)
  - Unplugged Activity Using Paper Blocks
  - Preview of Online Puzzles

View on Code Studio

Objectives

Students will be able to:

- Distinguish between loops that repeat a fixed number of times and loops that repeat as long as a condition is true.
- Use a while loop to create programs that can solve problems with unknown values.

Preparation

- Play through CSF Express Course 2018 - Website to find any potential problem areas for your class.
- Review CS Fundamentals Main Activity Tips - Lesson Recommendations.
- Make sure every student has aThink Spot Journal - Reflection Journal.

Links

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

For the Teacher

- Conditionals with Cards Sample Program - Teacher Prep Guide
- CSF Express Course 2018 - Website
- CS Fundamentals Main Activity Tips - Lesson Recommendations

For the Students

- Unplugged Blockly Blocks (Grades 2 - 5) - Manipulatives (download)
- Think Spot Journal - Reflection Journal

Vocabulary

- **Condition** - A statement that a program
checks to see if it is true or false. If true, an action is taken. Otherwise, the action is ignored.

- **Loop** - The action of doing something over and over again.
- **Repeat** - Do something again
- **While Loop** - A loop that continues to repeat while a condition is true.
Teaching Guide

Warm Up (10 min)

Introduction

Use "while" in a sentence in front of the students. Ask the students what the word "while" means. If you were to say "while there is a hole, fill it with dirt" what would they do? How long would they do that?

When you use a word like "while", you are relying on a condition to tell the computer how long the loop should run. A condition is a statement that is tested and found to be true or false. In the case above, the condition is if there is a hole. It's only possible for there to be a hole or for there not to be a hole, thus the statement is only ever true or false.

Tell the students they will be learning about a new kind of loop. Previously, students only used loops to repeat a command a certain number of times. Here, they won't always know how many times to repeat the command, however, they will know when to stop or when to keep going. While loops allow the programmer to repeat a command as long as a condition is still true. In the previous example, the condition is the existence of a hole.

If there's time, have the students discuss other times using a while loop would be useful. Examples include:

- Running toward a ball while it is in front of you.
- Filling a glass while it has space for more liquid.
- Walk forward while there is a path ahead.

Main Activity (30 min)

CSF Express Course - Website

While loops are not always a difficult concept for students to understand, but if you think your class might struggle with these puzzles, we recommend Pair Programming - Student Video. This will allow students to bounce ideas of each other before implementing the code. Pair programming works to increase confidence and understanding with topics like while loops.

Wrap Up (15 min)

Journaling

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

Journal Prompts:

- What was today's lesson about?
- How do you feel about today's lesson?
- What is the difference between a while loop and a normal repeat loop?
- Give an example of a puzzle where you would use a while loop, but not use a repeat loop. Can you give an example of a puzzle where you would use a repeat loop, but not a while loop?

Extended Learning

While Simon Says

Go out to a large playing field and have the students stand in a line facing you. Make sure every student can see you. Declare a couple of while loops such as:
While my right hand is up, you can walk toward me
While I cover my eyes, you can skip toward me
While my head is turned to the right, you have to walk backwards

The first student to get to you wins. If there's time, let other students be “Simon” at the front of the class.

Bridging Activity (15 min)

This activity will help bring the unplugged concepts from "Conditionals With Cards" into the online world that the students are moving into. It will also help bridge the concept of conditionals to a new type of loop—while loops. Choose one of the following to do with your class:

Unplugged Activity Using Paper Blocks

Print and cut out 2-3 while / do blocks and blank action blocks from Unplugged Blockly Blocks (Grades 2 - 5) - Manipulatives and pull out a deck of cards. Ask the class to come up with a couple of conditionals to use with the deck of cards like they did in "Conditionals with Cards." This time, instead of using the conditional in an if / else statement, they will be using it in a while loop.

When the conditionals have been decided on as a class, fill in the blank part of the while block with the various card groups that the kids came up with. Examples include "Even Numbered", "Red card", or "Diamonds". Fill in the action blocks with the actions the students came up with. Make sure the students know the action blocks need to be directly inside the while block. Line the blocks up sequentially so students can see how it runs as a program.

Below is an example.

Now shuffle the deck of cards and play "Conditionals with Cards" again. Flip through the deck card-by-card, reacting to cards if a conditional has been made for it. To maintain authenticity for this while loop experience, make sure you hold up the selected card for a good long while to give students time to react over several beats. Feel free to jump back to previous cards to get the point across that once you have left a while loop, you continue moving forward and do not return to the previous loop just because the conditions match again.

Preview of Online Puzzles

Pull up a puzzle from CSF Express Course - Website, we recommend Puzzle 6.

- Ask the class what the farmer should do when she gets to the pile of dirt.
  - The farmer should use a conditional in a while loop to check if there is a pile.
- Use the while there is a pile / do block. Ask the class what the farmer should do if there is no pile. When should the farmer stop?
  - The farmer should remove 1 if there is a pile. The while loop will continue until there is no pile, so then your code is finished!

Fill in the rest of the code and press Run. Discuss with the class why this worked.

Standards Alignment

CSTA K-12 Computer Science Standards
If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 15: Until Loops in Maze

Conditional | Loop | Maze | Angry Bird | Zombie

Overview

In this lesson, students will learn about until loops. Students will build programs that have the main character repeat actions until they reach their desired stopping point.

Purpose

This set of puzzles will work to solidify and build on the knowledge of loops by adding the until conditional. By pairing these concepts together, students will be able to explore the potential for creating complex and innovative programs.

Agenda

- Warm Up (10 min)
  - Introduction
- Bridging Activity (15 min)
  - Unplugged Activity Using Paper Blocks
  - Preview of Online Puzzles
- Main Activity (30 min)
  - CSF Express Course - Website
- Wrap Up (15 min)
  - Journaling
  - Extended Learning

View on Code Studio

Objectives

Students will be able to:
- Build programs with the understanding of multiple strategies to implement conditionals.
- Translate spoken language conditional statements and loops into a program.

Preparation

- Play through CSF Express Course 2018 - Website stage 15 to find any potential problem areas for your class.
- Review CS Fundamentals Main Activity Tips - Lesson Recommendations.
- Make sure every student has a Think Spot Journal - Reflection Journal.

Links

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

For the Teacher

- CSF Express Course 2018 - Website
- CS Fundamentals Main Activity Tips - Lesson Recommendations

For the Students

- Think Spot Journal - Reflection Journal

Vocabulary

- Condition - A statement that a program checks to see if it is true or false. If true, an action is taken. Otherwise, the action is ignored.
- Conditionals - Statements that only run under certain conditions.
• **Loop** - The action of doing something over and over again.
• **Repeat** - Do something again
• **Until** - A command that tells you to do something only up to the point that something becomes true.
Teaching Guide

Warm Up (10 min)

Introduction
In this lesson, students will be creating loops that only run until a condition is true. Help the students understand how this works by leading them in group activities and having them do an action until some condition is true. For example: Have students touch their nose until you tell them to stop.

Bridging Activity (15 min)

This activity will help bring the unplugged concepts from "Conditionals With Cards" into the online world that the students are continuing to move into. It will also help bridge the concept of conditionals to repeat until loops. Choose one of the following to do with your class:

Unplugged Activity Using Paper Blocks
Print and cut out 2-3 repeat until loops and some blank action blocks from Unplugged Blockly Blocks (Grades 2 - 5) - Manipulatives. Pull out a deck of cards. Ask the class to come up with a couple of conditionals to use with the deck of cards like they did in "Conditionals with Cards." When the conditionals have been decided on as a class, fill in the blank part of the repeat until loop with the various card groups that the kids came up with. Examples include "Even Numbered", "Red card", or "Diamonds". Fill in the action blocks with the actions the students came up with. Make sure the students know the action blocks need to be directly inside the repeat until block. Line the blocks up sequentially so students can see how it runs as a program. Below is an example.

Now shuffle the deck of cards and play "Conditionals with Cards" again. Flip through the deck card-by-card, reacting to cards if a conditional has been made for it. Make sure students know that once you have left the until loop, you continue moving forward and do not return to the previous loop just because the conditions match again.

Preview of Online Puzzles
Pull up a puzzle from Course D Online Puzzles - Website, we recommend Puzzle 4.

- Ask the class what the bird should repeat to get to the pig.
  - The bird should repeat move forward, turn right, move forward, and then turn left.
- Ask the class what they can use to repeat this code.
  - The bird should repeat this pattern until it reaches the pig.

Fill in the rest of the code using the repeat until loop and press Run. Discuss with the class why this worked.

Main Activity (30 min)

CSF Express Course - Website
Bringing together concepts is not easy, but this set of lessons is meant to help students see the endless possibilities of coding when using conditions. If students struggle at all with understanding the similarities or differences between while loops and until loops, have them try to think of how they would use similar statements in their real lives.

Wrap Up (15 min)

Journaling

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

Journal Prompts:

- What was today's lesson about?
- How do you feel about today's lesson?
- What's the difference between an until loop and a while loop?

Extended Learning

Until Simon Says

Go out to a large playing field and have the students stand in a line facing you. Make sure every student can see you. Declare a couple of "until loops" such as:

- Until my right hand is raised up, you can walk toward me
- Until I say "eggplant", you can walk backwards in my direction
- Until I turn my head to the right, you have to walk like a crab

The first student to get to you wins. If there's time, let other students be "Simon" at the front of the class.

Standards Alignment

CSTA K-12 Computer Science Standards

- AP - Algorithm & Programming

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

If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 16: Conditionals with Cards

Conditionals | Unplugged

Overview

This lesson demonstrates how conditionals can be used to tailor a program to specific information. We don’t always have all of the information we need when writing a program. Sometimes you will want to do something different in one situation than in another, even if you don’t know what situation will be true when your code runs. That is where conditionals come in. Conditionals allow a computer to make a decision, based on the information that is true any time your code is run.

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 do 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.

Agenda

Warm Up (20 min)
  Vocabulary
  Introduction
Main Activity (20 min)
  Conditionals with Cards Sample Program - Teacher Prep Guide
Wrap Up (15 min)
  Flash Chat: What did we learn?
  Journaling
Assessment (5 min)
  Conditionals with Cards - Assessment
Extended Learning

View on Code Studio

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

- Watch the Conditionals with Cards - Teacher Video.
- Watch the Conditionals with Cards - Lesson in Action Video.
- Gather decks of cards or something similar.
- One Conditionals with Cards Sample Program - Teacher Prep Guide for the class to look at.
- Print one Conditionals with Cards - Assessment for each student.
- Make sure every student has a Think Spot Journal - Reflection Journal.

Links

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

For the Teacher

- Conditionals with Cards - Unplugged Video (download)
- Conditionals with Cards - Teacher Video
- Conditionals with Cards - Lesson in Action Video
- Conditionals with Cards Sample
Program - Teacher Prep Guide
- Conditionals with Cards - Assessment
- Conditionals with Cards - Assessment Video
- Conditionals with Cards - Assessment Answer Key

For the Students
- Think Spot Journal - Reflection Journal

Vocabulary
- Conditionals - Statements that only run under certain conditions.
Teaching Guide

Warm Up (20 min)

Vocabulary
This lesson has one new and important word:

**Conditionals** - Say it with me: Con-di-shun-uls

Statements that only run under certain conditions.

Introduction

- 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.
- Ask the class "What was the condition of the reward?"
  - The condition was IF you were quiet for 30 seconds
    - If you were, the condition would be true, and you would get the reward.
    - If you weren't, 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?
- Sometimes, we want to have an extra condition, in case the "IF" statement is not true.
  - This extra condition is called an "ELSE" statement
  - 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 "Awwwwwe."
    - 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?
- Believe it or not, we have even one more option.
  - What if I wanted you to clap if I draw a 7, or else if I draw something less than seven you say "YAY," or else you say "Awwwwwe"?
    - 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 (20 min)

**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.

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.
- 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.

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:

```
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;
    }
}
```

Wrap Up (15 min)

**Flash Chat: What did we learn?**

- 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)
Lesson 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.

- 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?

Journaling

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

Journal Prompts:
- What was today’s lesson about?
- How do you feel about today’s lesson?
- What is a conditional? How did you use a conditional today?
- 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?

Assessment (5 min)

Conditionals with Cards - Assessment

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. Here’s a Conditionals with Cards - Assessment Video to watch as a guide.

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?

## Standards Alignment

**CSTA K-12 Computer Science Standards**

- **AP - Algorithms & Programming**

If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 17: If/Else with Bee

Conditional | Bee | Maze

Overview

Up until this point students have been writing code that executes exactly the same way each time it is run - reliable, but not very flexible. In this lesson, your class will begin to code with conditionals, allowing them to write code that functions differently depending on the specific conditions the program encounters.

Purpose

After being introduced to conditionals in "Conditionals with Cards," students will now practice using them in their programs. The if / else blocks will allow for a more flexible program. The bee will only collect nectar if there is a flower or make honey if there is a honeycomb. Students will also practice and recognize a connection between if / else blocks and while loops in this set of puzzles.

Agenda

- Warm Up (10 min)
- Introduction
- Bridging Activity - Conditionals (15 min)
- Unplugged Activity Using Paper Blocks
- Preview of Online Puzzles
- Main Activity (30 min)
- CSF Express Course - Website
- Wrap Up (15 min)
- Journaling
- Extended Learning

Objectives

Students will be able to:
- Translate spoken language conditional statements into a program.
- Solve puzzles using a combination of looped sequences and conditionals.

Preparation

- Play through CSF Express Course 2018 - Website stage 17 to find any potential problem areas for your class.
- Review CS Fundamentals Main Activity Tips - Lesson Recommendations.
- Make sure every student has a Think Spot Journal - Reflection Journal.

Links

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

For the Teacher
- CSF Express Course 2018 - Website
- CS Fundamentals Main Activity Tips - Lesson Recommendations

For the Students
- Unplugged Blockly Blocks (Grades 2 - 5) - Manipulatives (download)
- Think Spot Journal - Reflection Journal

Vocabulary

- Conditionals - Statements that only run under certain conditions.
Teaching Guide

Warm Up (10 min)

Introduction

Review the "Conditionals with Cards" activity with your students.

- What is a conditional statement?
- When is a conditional useful?
- What are some of the conditions you used in the last activity?

Now we’re going to use conditionals with the Code.org bee to help him deal with some mysterious clouds. We don't know if his flowers have nectar or not, so we'll need to use conditionals to make sure that we collect nectar if it's there, but that we don't try to collect nectar from a flower that doesn't have any.

Bridging Activity - Conditionals (15 min)

This activity will help bring the unplugged concepts from "Conditionals With Cards" into the online world that the students are moving into. Choose one of the following to do with your class:

Unplugged Activity Using Paper Blocks

Print and cut out 2-3 if / else and blank action blocks from Unplugged Blockly Blocks (Grades 2 - 5) - Manipulatives and pull out a deck of cards. Ask the class to come up with a couple of conditionals to use with the deck of cards like they did in "Conditionals with Cards." When the conditionals have been decided on as a class, fill in the blank part of the if block with the various card values that the kids came up with. Examples include "King of Hearts", "Even Numbered", or "Diamonds". Fill in the action blocks with the actions the students came up with. Make sure the students know the action blocks need to be directly under the if or else block. Below is an example.

Now shuffle the deck of cards and play "Conditionals with Cards" again. Flip through the deck card-by-card, reacting to cards if a conditional has been made for it.

Preview of Online Puzzles

Pull up a puzzle from Course D, we recommend puzzle 9.

- Ask the class what the bee should do when it gets to the cloud.
  - The bee should use a conditional to check for a flower or a honeycomb.
- Use the if at flower / else block. Ask the class what the bee should do if there's a flower. If there's not a flower, there will be a honeycomb. What should the bee do then?
  - The bee should get nectar if there is a flower and make honey if there is a honeycomb.

Fill in the rest of the code and press Run. Discuss with the class why this worked.

Main Activity (30 min)
CSF Express Course - Website

These puzzles might sprout some questions, so have the students work in pairs or implement the “Ask three before you ask me” rule (have the students ask three other peers for help before they go to the teacher.) This will spark discussions that will develop each student's understanding.

Wrap Up (15 min)

Journaling

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

Journal Prompts:

- What was today’s lesson about?
- How did today’s lesson make you feel?
- What conditionals did you use in your code today?
- What are some other conditionals a bee might use? Examples include:
  - if there is a tree in front of me, buzz out of the way
  - if my wing is hurt, rest on the ground
  - if I see another bee, say “Hello!”

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:
  - If the suit is clubs
  - If 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?

Standards Alignment
If you are interested in licensing Code.org materials for commercial purposes, **contact us**.
Lesson 18: Harvesting with Conditionals

Conditional | Loop | Harvester

Overview

Students will practice while loops, until loops, and if/else statements. All of these blocks use conditionals. By practicing all three, students will learn to write complex and flexible code.

Purpose

Practicing the use of conditionals in different scenarios helps to develop a student's understanding of what conditionals can do. In the previous lesson, students only used conditionals to move around a maze. In this lesson, students will use conditionals to help the farmer know when to harvest crops. New patterns will emerge and students will use creativity and logical thinking to determine the conditions where code should be run and repeated.

Agenda

- Warm Up (5 min)
  - Introduction
- Main Activity (30 min)
  - CSF Express Course - Website
- Wrap Up (15 min)
  - Journaling

View on Code Studio

Objectives

Students will be able to:

- Nest conditionals to analyze multiple value conditions using if, else if, else logic.
- Pair a loop and conditional statement together.

Preparation

- Play through CSF Express Course 2018 - Website stage 18 to find any potential problem areas for your class.
- Review CS Fundamentals Main Activity Tips - Lesson Recommendations.
- Make sure every student has a Think Spot Journal - Reflection Journal.

Links

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

For the Teacher

- CSF Express Course 2018 - Website
- CS Fundamentals Main Activity Tips - Lesson Recommendations

For the Students

- Think Spot Journal - Reflection Journal

Vocabulary

- Condition - A statement that a program checks to see if it is true or false. If true, an action is taken. Otherwise, the action is ignored.
- Conditionals - Statements that only run under certain conditions.
- **Loop** - The action of doing something over and over again.
- **Repeat** - Do something again
- **While Loop** - A loop that continues to repeat while a condition is true.
Teaching Guide

Warm Up (5 min)

Introduction
Students shouldn't need as much of an introduction to concepts today because they have had practice with them in the previous lesson. Instead, you can share the story of the harvester.

The harvester is trying to pick crops like pumpkins, lettuce, and corn. However, the farmer has forgotten where she planted these crops, so she needs to check each plant before harvesting.

Main Activity (30 min)

CSF Express Course - Website
Students will continue to work with if / else statements, while loops, and until loops. These puzzles are a bit more challenging, though, so encourage students to stick with them until they can describe what needs to happen for each program.

Wrap Up (15 min)

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

Journal Prompts:
- What was today's lesson about?
- How do you feel about today's lesson?
- How can you see conditionals being useful in programs?
- What if people only spoke in if/else statements? What would be some advantages and disadvantages of this?

Standards Alignment

CSTA K-12 Computer Science Standards
- AP - Algorithms & Programming

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

If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 19: Songwriting

Unplugged | Function

Overview

One of the most magnificent structures in the computer science world is the function. Functions (sometimes called procedures) are mini programs that you can use over and over inside of your bigger program. This lesson will help students intuitively understand why combining chunks of code into functions can be such 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.

Agenda

- Warm Up (20 min)
  - Vocabulary
  - Sing a Song
- Main Activity (20 min)
  - Functions Unplugged: Songwriting - Worksheet
- Wrap Up (5 min)
  - Flash Chat: What did we learn?
  - Journaling
- Assessment (5 min)
  - Functions Unplugged: Songwriting - Assessment

Extended Learning

View on Code Studio

Objectives

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

Preparation

- Watch the Songwriting and Songwriting with Parameters (Functions) - Teacher Video.
- Watch the Functions Unplugged: Songwriting - Lesson in Action Video.
- Print several Functions Unplugged: Songwriting - Worksheet for each group.
- Print one Functions Unplugged: Songwriting - Assessment for each student.
- Access to the internet, or pre-downloaded songs and lyrics for activity.
- Make sure every student has a Think Spot Journal - Reflection Journal.

Links

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

For the Teacher

- Songwriting with Functions - Unplugged Video (download)
- Songwriting and Songwriting with Parameters (Functions) - Teacher Video
- Functions Unplugged: Songwriting - Lesson in Action Video
For the Students

- **Think Spot Journal** - Reflection Journal

**Vocabulary**

- **Function** - A named group of programming instructions. Functions are reusable abstractions that reduce the complexity of writing and maintaining programs.
Teaching Guide

Warm Up (20 min)

Vocabulary
This lesson has one new and important word:

- **Function** - Say it with me: Func-shun

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

Sing a Song

- 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.

チョーク:

Little bunny Foo Foo
Hopping through the forest
Scooping up the field mice
And bopping ’em on the head
Down came the Fairy
And she said
  “Little bunny Foo Foo
I don’t wanna see you
Scooping up the field mice
And bopping ’em on the head”

Song:

Chorus
I'll give you 3 chances.
Then I'll turn you into a goon!
The next day... 

Chorus
I'll give you 2 chances.
Then I'll turn you into a goon!
The next day... 

Chorus
I'll give you 1 chance.
Then I'll turn you into a goon!
The next day... 

Chorus
"I gave you two chances.
Now I'll turn you into a goon!"
(POOF!)
And the moral of the story is:
Hare today, goon tomorrow!

- 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.
- 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?

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 min)

Functions Unplugged: Songwriting - Worksheet

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."

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 min)

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?

Journaling

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

Journal Prompts:
• What was today’s lesson about?
• How do you feel about today’s lesson?
• What is a function and how do you use it?
• Can you think of another activity where you might want to call a special group of instructions several times?

Assessment (5 min)

Functions Unplugged: Songwriting - Assessment

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.

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.

Standards Alignment

CSTA K-12 Computer Science Standards

• AP - Algorithms & Programming
If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 20: Functions in Minecraft

Overview

Students will begin to understand how functions can be helpful in this fun and interactive Minecraft adventure!

Purpose

Students will discover the versatility of programming by practicing functions in different environments. Here, students will recognize reusable patterns and be able to incorporate named blocks to call predefined functions.

Agenda

- Warm Up (10 min)
  - Introduction
- Bridging Activity - Functions (15 min)
  - Unplugged Activity Using Some Blockly
  - Preview of Online Puzzles
- Main Activity (30 min)
  - CSF Express Course - Website
- Wrap Up (15 min)
  - Journaling

Objectives

Students will be able to:
- Use functions to simplify complex programs.
- Use pre-determined functions to complete commonly repeated tasks.

Preparation

- Play through CSF Express Course 2018 - Website stage 20 to find any potential problem areas for your class.
- Review CS Fundamentals Main Activity Tips - Lesson Recommendations.
- Make sure every student has a Think Spot Journal - Reflection Journal.

Links

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

For the Teacher

- CSF Express Course 2018 - Website
- CS Fundamentals Main Activity Tips - Lesson Recommendations

For the Students

- Think Spot Journal - Reflection Journal

Vocabulary

- Function - A named group of programming instructions. Functions are reusable abstractions that reduce the complexity of writing and maintaining programs.
Teaching Guide

Warm Up (10 min)

Introduction
Help the class understand that functions are simply a chunk of code that has a name. Once defined, you can use that name over and over in your program to tell the computer to run the chunk of code that you assigned to it.

Bridging Activity - Functions (15 min)

This activity will help bring the unplugged concepts from "Functions Unplugged: Songwriting" into the online world that the students are moving into. Choose one of the following to do with your class:

Unplugged Activity Using Some Blockly
Pick a song to play that the students enjoy and print out the lyrics. You can use the same song from "Functions Unplugged: Songwriting." Break your class into groups or pairs. Pass out the printed out lyrics (including the repeated chorus) and the basic function blocks from Unplugged Blockly Blocks (Grades 2 - 5) - Manipulatives to each group or pair of students. See lesson tip for details.

Ask the students to cross out any part of the song that can be made into a function (the chorus is a good example) and put it into the function blocks provided. Students should fill in the function declaration with a function name and the words of the repeated lyrics. Once the function declaration is done, ask the students to fill in the function calls and place them on top of the crossed out lyrics.

Once every group or pair is done, ask the class where they put their functions and why. Did everyone make the same function? How often is the function repeated?

Preview of Online Puzzles

Pull up a puzzle from Course E Online Puzzles - Website. We recommend puzzle 9 of this lesson. As a class, work through the puzzle without using functions. Once you have gotten the solution, display it on a white board or overhead. Ask the class to point to the repeated code. Ask the class how they would simplify the program. Why can you not just use a loop?

On the white board or overhead, rewrite the program without the repeated code, but leaving one line space. In that/those line space(s), call a function. Off to the side, declare the function like the left example block in the lesson tip. Ask the class what they think the code will do now.

Open up a discussion with the class on why functions could be useful in programming. Invite students to discuss the difference between functions and loops.

Main Activity (30 min)

CSF Express Course - Website
We recommend providing paper and pencils for students to write (or draw) out ideas. Also, if students are having trouble
recognizing patterns, have them work with a partner on the harder puzzles.

Wrap Up (15 min)

Journaling

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

- What was today's lesson about?
- How do you feel about today's lesson?
- What did your functions do in the programs you wrote today? How did that help you?
- When should you use a function instead of a loop?

Standards Alignment

CSTA K-12 Computer Science Standards

- AP - Algorithms & Programming

If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 21: Functions with Harvester

Function | Harvester

Overview
Students have practiced creating impressive designs in Artist and navigating mazes in Bee, but today they will use functions to harvest crops in Harvester. This lesson will push students to use functions in the new ways by combining them with while loops and if / else statements.

Purpose
This lesson is meant to further push students to use functions in more creative ways. By also using conditionals and loops, students will learn there are many ways to approach a problem, but some are more efficient than others. These puzzles are intended to increase problem solving and critical thinking skills.

Agenda
Warm Up (10 min)
  Introduction
Main Activity (30 min)
  CSF Express Course - Website
Wrap Up (15 min)
  Journaling

View on Code Studio

Objectives
Students will be able to:
- Recognize when a function could help to simplify a program.
- Use pre-determined functions to complete commonly repeated tasks.

Preparation
- Play through CSF Express Course 2018 - Website stage 21 to find any potential problem areas for your class.
- Review CS Fundamentals Main Activity Tips - Lesson Recommendations.
- Make sure every student has a Think Spot Journal - Reflection Journal.

Links

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

For the Teacher
- CSF Express Course 2018 - Website
- CS Fundamentals Main Activity Tips - Lesson Recommendations

For the Students
- Think Spot Journal - Reflection Journal

Vocabulary
- Function - A named group of programming instructions. Functions are reusable abstractions that reduce the complexity of writing and maintaining programs.
Teaching Guide

Warm Up (10 min)

Introduction
At this point, your students should already be introduced to functions. Take this time to have them discuss the advantages and disadvantages of using functions in a program. Either have them pair share or discuss as a class. Try using examples of hard or easy puzzles in either Artist or Bee.

Ask the class:
- When would you use a function?
- Why does a function help to simplify your program?
- Do you think functions make programming easier or harder? Why?

Main Activity (30 min)

CSF Express Course - Website
Some puzzles will have a function pre-declared for the students to fill in. It may be helpful for the students to write the entire program without a function first, then determine where a function would be useful in the program.

It's important to make sure that every student is completing each puzzle with a dark green dot. If some of your students are struggling to simplify code and use functions, set up teams of expert students within your class to go around and answer questions.

Don't forget to provide pencils and paper to help students sketch out possible solutions.

Wrap Up (15 min)

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

Journal Prompts:
- What was today’s lesson about?
- How did you feel about today’s lesson?
- What makes you realize a function could help your program?
- How do while loops and if / else statements help your program?

Standards Alignment

CSTA K-12 Computer Science Standards
- AP - Algorithms & Programming

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

If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 22: Functions with Artist

Function | Artist

Overview

Students will be introduced to using functions on Code.org. Magnificent images will be created and modified with functions in Artist. For more complicated patterns, students will learn about nesting functions by calling one function from inside another.

Purpose

One of the most important components to this lesson is providing students with a space to create something they are proud of. These puzzles progress to more and more complex images, but each new puzzle only builds off the previous puzzle. At the end of this lesson, students will feel confident with themselves and proud of their hard work.

Agenda

- Warm Up (15 min)
  - Introduction
- Main Activity (30 min)
  - CSF Express Course - Website
- Wrap Up (15 min)
  - Journaling
- Extended Learning

View on Code Studio

Objectives

Students will be able to:

- Categorize and generalize code into useful functions.
- Recognize when a function could help to simplify a program.

Preparation

- Play through puzzles in CSF Express Course 2018 - Website stage 22 to find any potential problem areas for your class.
- Review CS Fundamentals Main Activity Tips - Lesson Recommendations.
- Make sure every student has a Think Spot Journal - Reflection Journal.

Links

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

For the Teacher

- CSF Express Course 2018 - Website
- CS Fundamentals Main Activity Tips - Lesson Recommendations

For the Students

- Unplugged Blockly Blocks (Grades 2 - 5) - Manipulatives (download)
- Think Spot Journal - Reflection Journal

Vocabulary

- **Function** - A named group of programming instructions. Functions are reusable abstractions that reduce the complexity of writing and maintaining programs.
Teaching Guide

Warm Up (15 min)

Introduction
Ask the class to think back to “Functions Unplugged: Songwriting” and recall what a function is. Open a discussion about when to use a function when writing a song.

Tell the class that there are two main components to using functions.

1. **The Declaration**: Function declarations are what create a function. In a function declaration, you fill in the function with code and give the function a name. You must declare a function before you can use it.

2. **The Call**: Function calls are what makes the program run the code in the function. To call a function, you place the name of the function in your program. Make sure your function is properly defined before calling it in your program.

The class can use songwriting as an example to understand these two components. In the unplugged activity, the function containing the lyrics to the chorus was named “chorus”. When we first made this function, we circled the lyrics that would go in the function. Once we named the function, we could read through the lyrics and replace the repeated chorus lyrics with a function call to “chorus”.

Continue the conversation until students have a basic understanding of functions being declared and called. If students don't get to this point, make sure to do one of the bridging activities before moving into the Code.org puzzles.

Main Activity (30 min)

CSF Express Course - Website
Students may benefit from writing code without functions then creating functions from the repeated code. If students don't enjoy doing this in the Code.org workspace, we recommend providing paper and pencils for students to write (or draw) out their ideas.

Wrap Up (15 min)

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

Journal Prompts:
- What was today’s lesson about?
- How do you feel about today’s lesson?
- What are some differences between functions and loops?
- Sketch out a drawing you made today. Can you write the code needed to create this?
- Draw a picture you would like to create with code. Try writing or drafting the code that would make that drawing.

Extended Learning

Draw by Functions
Break the class into groups of 2-3 students. Have each group write a function that draws some kind of shape and a program that uses that function. Depending on the creativity or focus the groups, students might need to be assigned a shape to create. Once every group is done, have the groups switch programs. On a separate piece of paper, each group should draw
what the program creates. The groups should then return the programs and drawings to the original group. Did every group get the drawing they expected? If not, what went wrong? Have the class go through the debugging process and try again.

Standards Alignment

CSTA K-12 Computer Science Standards

► AP - Algorithms & Programming

If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 23: Screen Out the Mean

Common Sense Education | Cyberbullying | Unplugged

Overview

This lesson helps children to recognize that it is essential to tell a trusted adult if something online makes them feel angry, sad, or scared.

Students learn that other people can sometimes act like bullies when they are online. They will explore what cyberbullying means and what they can do when they encounter it. After reading a scenario about mean online behavior, students discuss what cyberbullying is, how it can make people feel, and how to respond. Finally, they use their knowledge to create a simple tip sheet on cyberbullying in their journal.

Purpose

Students may not ever have the misfortune of experiencing cyberbullying, but we want to make sure that the students are prepared for and knowledgeable about it, in case they ever witness it during an online situation. Students will learn how to identify cyberbullying and what steps they should take to make it stop. This may become helpful in later puzzles when students have the opportunity to share their work. If someone negatively responds to a student's work, this lesson will provide them with the tools that they need to handle the situation.

Agenda

- **Warm Up (5 min)**
  - Introduction
- **Main Activity (35 min)**
  - What Is Cyberbullying?
  - What to Do About Cyberbullying
- **Wrap Up (15 min)**
  - Flash Chat: What did we learn?
  - Journaling
- **Assessment (5 - 10 min)**
  - Screen Out the Mean - Teacher Prep Guide

View on Code Studio

Objectives

Students will be able to:
- Analyze online behaviors that could be considered cyberbullying.
- Explain how to deal with a cyberbullying situation.
- Recognize the importance of engaging a trusted adult if the student experienced cyberbullying.

Preparation

- Review Screen Out the Mean - Teacher Prep Guide from Common Sense Education's website.
- Print out a worksheet from the link above (page 6) for each student.
- Print out an assessment from the link at the top (page 7) for each student.
- Make sure every student has aThink Spot Journal - Reflection Journal.
- Print or display the Online Safety Poster - Student Handout for the class to see.
- Review CSF Digital Citizenship - Resource List for more online safety content.

Links

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

For the Teacher
- Screen Out the Mean - Teacher Prep Guide
- Common Sense Education - Website
- CSF Digital Citizenship - Resource List
For the Students

- **Online Safety Poster** - Student Handout
- **Think Spot Journal** - Reflection Journal

**Vocabulary**

- **Cyberbullying** - Doing something on the internet, usually again and again, to make another personal feel angry, sad, or scared.
- **Online** - Connected to the internet.
Teaching Guide

Warm Up (5 min)

Introduction

Encourage:
Encourage students to share what they know about bullying.

Ask:
- What kinds of things count as bullying?
  - Students should understand that bullying is behavior that is purposefully mean or scary to someone else. For example, making fun of how someone looks, telling lies about them, or threatening to do something bad to them.
- How does bullying make other people feel?
  - Hurt, angry, upset, scared
- What is the best thing to do when you feel bullied, or when you see someone else being bullied?
  - Students should know to always tell a trusted adult when they experience or witness bullying.

Explain:
Students will be learning about a kind of bullying that can take place when they use the internet.

Main Activity (35 min)

What Is Cyberbullying?

Define:
- Online: Connected to the internet
- Cyberbullying: Doing something on the internet, usually again and again, to make another person feel angry, sad, or scared

Discuss:
Some kids do not go online very much at all, either because of their family’s rules or because they do not like it very much. Other kids do go online to do different things.

Ask:
- What do you do online, or what do you think you might like to do?
  - Students may mention activities like sending messages to friends and playing games.

Share:
Most of the time when students go online it is to do fun or interesting things. But sometimes people can be mean to each other online and this is called cyberbullying.

Ask:
- Did you ever see someone make someone else feel bad online?
  - Answers will vary. Remind students to tell what happened, but not to use real names.

Explain:
Tell students that they will be learning more about how cyberbullying occurs, and what to do when it happens to them or to someone they know.

What to Do About Cyberbullying
Discuss:
Read aloud these two scenarios and discuss them briefly with the class.

- Kyle keeps getting instant messages from someone saying means things about him. The person who is sending the messages doesn’t use a real name, but Kyle can tell the messages are coming from someone who also makes fun of him at school in gym class.
- Sasha is a new girl at school, and she’s making a lot of friends. Then Sasha finds out that another girl sent around an email that had a picture of a cow with Sasha’s name on it.

Next, pass out the Screen Out the Mean - Teacher Prep Guide worksheet from page 6. Read aloud the story at the top and ask students to work in pairs or groups to finish the worksheet.

Ask the class to discuss to Jada’s story. Tell the class there are specific steps to handling a cyberbully.

- Jada should STOP using the computer.
- Jada should TELL an adult she trusts what happened.
- Jada should not go back online or return to the pony website until an adult says it is OK.
- If Jada and Michael are good friends, Jada may want to tell Michael how his actions made her feel after she gets help from an adult.
- If Michael continues cyberbullying her, she should play with other kids who don’t cyberbully others.

In general, there are four steps students should take if they or someone they know are experiencing cyberbullying.

1. Stop using the computer until it is safe.
2. Tell an adult you trust.
3. Go online only when a trusted adult says it is okay.
4. Play online only with kids who you know and are nice.

Wrap Up (15 min)

Flash Chat: What did we learn?
Ask:
- What is cyberbullying. How does it make people feel?
  - Students should recognize that cyberbullying is any kind of online behavior that makes people feel sad, scared, angry or upset.
- What four things can you do to help stop cyberbullying?
  - S. Stop using the computer until it is safe.
  - T. Tell an adult you trust.
  - O. Go Online only when a trusted adult says it is okay.
  - P. Play online only with kids who are nice.
- What is the most important thing to do if someone starts cyberbullying you?
  - Telling a trusted adult is the most important response whenever someone makes them feel sad, scared, or angry online.
Journaling

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

Journal Prompts:

- What was today's lesson about?
- How did you feel during today's lesson?
- Write down the names of some trusted adults you can go to if you ever feel bullied.
- What are the four steps you should take if you or someone you know is being cyberbullied.

Assessment (5 - 10 min)

Screen Out the Mean - Teacher Prep Guide

Pass out an assessment to each student. Allow students a few minutes to complete it then review the answers (page 9 of the link above) with the class. If there's time, allow for a discussion about the questions.

If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 24: Determine the Concept

Bee

Overview
This series brings together concepts from previous lessons and gives students a chance to think critically about how they would solve each problem, but without telling them which concept to apply. Students will review basic algorithms, debugging, repeat loops, conditionals, while loops, and functions.

Purpose
It's important for students to remember that computer science provides plenty of opportunities to be creative. Every topic can be combined with another to make something bigger and better.

In this lesson, students will use previously learned concepts together, allowing for a "big picture" view of programming projects. This lesson will also bridge any gaps in understanding of when to use certain programming tools over others.

Agenda
- Warm Up (10 min)
  - Introduction
- Main Activity (30 min)
  - CSF Express Course - Website
- Wrap Up (15 min)
  - Journaling

Objectives
Students will be able to:
- Recognize which programming concept to use to solve a given problem.
- Describe the different ways one could solve a given problem.

Preparation
- Play through the CSF Express Course 2018 - Website stage 24 to find any potential problem areas for your class.
- Review CS Fundamentals Main Activity Tips - Lesson Recommendations.
- Make sure every student has a Think Spot Journal - Reflection Journal.

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

For the Teacher
- CSF Express Course 2018 - Website
- CS Fundamentals Main Activity Tips - Lesson Recommendations

For the Students
- Think Spot Journal - Reflection Journal
Teaching Guide

Warm Up (10 min)

Introduction
This lesson is meant to review previous concepts by mixing them together.

Take this time to ask students to discuss their favorite parts of the course so far.

- What did they like learning about? Why?
- What did they not like learning about? Why not?
  - If it was too difficult, would they be interested in doing more activities to better understand those concepts?
- What was something really cool they got to make?
- What is something they hope to make with code someday? What kinds of programming tools might they use to make that?

This discussion should help the students get into the "bigger picture" framework that this lesson is trying to teach. Ask the students how they would pair certain programming tools together like:

- Loops and conditionals
- Functions and events

Main Activity (30 min)

CSF Express Course - Website
Because the concepts in this set of lessons are not explicitly given, students may face more frustration than usual. Make sure to walk around and do a "wellness check" on each student. If one student is struggling in some manner, pair them with another student in order to stimulate discussion between the two.

Wrap Up (15 min)

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

Journal Prompts:
- What was today’s lesson about?
- How do you feel about today’s lesson?
- What is your favorite thing you have learned in your coding class so far? Why? How have you used that in programs lately?

Standards Alignment

CSTA K-12 Computer Science Standards
- AP - Algorithms & Programming

This curriculum is available under a Creative Commons License (CC BY-NC-SA 4.0).
If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 25: The Big Event

Unplugged | Events

Overview

Students will soon learn that events are a great way to add flexibility to a pre-written algorithm. Sometimes you want your program to be able to respond to the user exactly when the user wants it to. Events can make your program more interesting and interactive.

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.

Agenda

Warm Up (15 min)
  Vocabulary
  A Series of Events
Main Activity (15 min)
  The Big Event (Courses C-F) - Worksheet
Wrap Up (10 min)
  Flash Chat: What did we learn?
Assessment (10 min)
  The Big Event - Assessment
Extended Learning

View on Code Studio

Objectives

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

Preparation

- Watch the The Big Event - Teacher Video.
- Print one The Big Event (Courses C-F) Worksheet and Event Controller.
- Print one The Big Event - Assessment for each student.
- Make sure every student has a Think Spot Journal - Reflection Journal.

Links

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

For the Teacher
  • The Big Event - Unplugged Video (download)
  • The Big Event - Teacher Video
  • The Big Event (Courses C-F) - Worksheet
  • The Big Event - Assessment
  • The Big Event - Assessment Answer Key

For the Students
  • Think Spot Journal - Reflection Journal

Vocabulary

  • Event - An action that causes something to
happen.
Teaching Guide

Warm Up (15 min)

Vocabulary
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

- Prep your class to answer a question:
  - "I'm going to ask you a question. I want you to raise your hand if you want me to call on you for the answer."
- Ask a simple question that most of your students should be able to answer, such as:
  - How many thumbs do I have?
  - What is bigger, a bird or a horse?
- Call on a student who has their hand raised and let them give their answer.
- Upon finishing that display, ask the class how you knew that the student wanted you to call on them.
  - Your class will likely mention the raising of the hand.
- Explain to everyone that when students raise their hand, it is an "event" that causes you to know that they want to be called on.
- Ask the class if they can think of any other events that give signals.
  - You may need to remind them that you're not talking about an event like a birthday party or a field trip.
  - If they have trouble, you can remind them that an event is an action that causes something to happen.
    - What about an alarm clock going off? What does that make happen?
    - What about pressing "Start" on the microwave? What does that do?
    - What about pressing the power button on your tv remote?
- Today, we're going to practice changing programs by introducing events.

Main Activity (15 min)

The Big Event (Courses C-F) - Worksheet

- Do you remember guiding Red from Angry Birds to the pig in the Maze puzzles?
  - In that exercise, you knew in advance exactly where you wanted Red to go, so you could make a program that took Red 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.

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!

Directions:
Project the Event Controller onto your classroom screen.

Decide with your class what each button does. We suggest:
- Pink Button -> Say “Wooooo!”
- Teal Button -> “Yeah!”
- Purple Dial -> “Boom!”
- Green Button -> Clap
- Orange Dial -> Stomp

Practice tapping the buttons on the overhead 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.
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.

Wrap Up (10 min)

Flash Chat: What did we learn?
- Why do we need to be able to handle events in a program?
- What are some other kinds of events that you can think of?

Assessment (10 min)

The Big Event - Assessment
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.

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!

Standards Alignment

CSTA K-12 Computer Science Standards
  ▶ AP - Algorithms & Programming

If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 26: Build a Flappy Game

Flappy | Event

Overview

In this special stage, students get to build their own Flappy Bird game by using event handlers to detect mouse clicks and object collisions. At the end of the level, students will be able to customize their game by changing the visuals or rules.

Purpose

Events are very common in computer programs. In this lesson, students will further develop their understanding of events by making a Flappy Bird game. Students will learn to make their character move across the screen, make noises, and react to obstacles based on user-initiated events.

Agenda

Warm Up (10 min)
  Introduction
Bridging Activity - Events (10 min)
  Unplugged Activity Using Paper Blocks
  Preview of Online Puzzles as a Class
Main Activity (30 min)
  Express Course Online Puzzles - Website
Wrap Up (10 - 15 min)
  Journaling
Extended Learning

View on Code Studio

Objectives

Students will be able to:
- Match blocks with the appropriate event handler.
- Create a game using event handlers.
- Share a creative artifact with other students.

Preparation

- Play through CSF Express Course 2018 - Website in lesson 26 to find any potential problem areas for your class.
- Review CS Fundamentals Main Activity Tips - Lesson Recommendations.
- Make sure every student has a Think Spot Journal - Reflection Journal.

Links

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

For the Teacher
- Express Course Online Puzzles - Website
- CS Fundamentals Main Activity Tips - Lesson Recommendations

For the Students
- Think Spot Journal - Reflection Journal

Vocabulary

- Event - An action that causes something to happen.
Lesson Tip

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

Warm Up (10 min)

Introduction

- Review "The Big Event" activity with students:
  - What did we "program" the button click events to do?
  - Now we're going to add events to our coding. Specifically, we're going to create an event for clicking the mouse and one for when the bird hits an object like the ground or an obstacle. When have you seen a character touch another object as an event in games?

Teacher Tip

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

Bridging Activity - Events (10 min)

This activity will help bring the unplugged concepts from "The Big Event" into the online world that the students are moving into. Choose one of the following to do with your class:

Unplugged Activity Using Paper Blocks

Using the remote from the The Big Event (Courses C-F) - Worksheet and Unplugged Blockly Blocks (Grades 2 - 5) - Manipulatives, gather your class to reprise the activity from the previous lesson. Ask the class "when the teal button is pushed, what do we do?" then fill in one of the when event blocks and one of the blue action blocks accordingly. Make sure that the students understand that the when blocks need to be on top of the blue block and they need to touch in order for the program to run.

Preview of Online Puzzles as a Class

Pull a lesson from the corresponding online stage, we recommend puzzle 2. Ask the students what should happen when the Flappy Bird runs into something like the ground or an obstacle. Explain that Flappy in this game will move forward with a click of the mouse and the game will end if Flappy runs into anything.

Complete the puzzle with the class and allow time for a quick discussion on what was and wasn't an event. For every event, ask the students what the action corresponding to this event is.

Main Activity (30 min)

Express Course Online Puzzles - Website

In the final stage of this lesson students are able to tweak their game to make it unique - encourage them to see how different they can make each game within the constraints provided. If the class doesn't use Pair Programming - Student Video, then tell students to go around and look at other student's games. Otherwise, have students discuss and try out different ways to set up their game with their partner.

Teacher Tip

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

Wrap Up (10 - 15 min)
Journaling

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

**Journal Prompts:**
- What was today's lesson about?
- How did you feel about today's lesson?
- What did you do to make your game unique?
- Draw out a game you want to make in the future.

Extended Learning

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

**Look Under the Hood**

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

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

Standards Alignment

**CSTA K-12 Computer Science Standards**

- AP - Algorithms & Programming

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

If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 27: The Power of Words

Common Sense Education | Cyberbullying

Overview

Students consider that while they are enjoying their favorite websites they may encounter messages from other kids that can make them feel angry, hurt, sad, or fearful. They explore ways to handle cyberbullying and how to respond in the face of upsetting language online.

Students discuss all the ways they use technology for communication and explore the similarities and differences between in-person and online communication. Students then brainstorm ways to respond to cyberbullying.

Purpose

This lesson will provide students with the tools that they need to handle cyberbullying if they are ever in the situation of having someone negatively responds to their online postings.

Students may not ever have the misfortune of experiencing cyberbullying, but they should understand what it is so that they can spot it online. Students will learn how to identify cyberbullying and what steps they should take to make it stop. This may become helpful in later puzzles when students have the opportunity to share their work.

Agenda

- Warm Up (5 min)
  - Introduction
- Main Activity (35 min)
  - What's the Problem?
  - Crossing the Line
  - Talk and Take Action
- Wrap Up (15 min)
  - Flash Chat: What did we learn today?
  - Journaling
- Assessment (10 min)

View on Code Studio

Objectives

Students will be able to:

- Empathize with those who have received mean and hurtful messages.
- Judge what it means to cross the line from harmless to harmful communication online.
- Generate solutions for dealing with cyberbullying.

Preparation

- Preview the Common Sense Education - Power of Words - Teacher Prep Guide and prepare to show it to your class.
- Print out the Words Can Hurt Handout from Common Sense Education - Power of Words - Teacher Prep Guide (page 7) for each group of four.
- Print out the Talk and Take Action Handout from Common Sense Education - Power of Words - Teacher Prep Guide (page 6) for each student.
- Print out the assessment on page 8-9 of Common Sense Education - Power of Words - Teacher Prep Guide.
- Obtain colored pencils and a string the length of the classroom.

Links

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

For the Teacher

- Common Sense Education - Power of Words - Teacher Prep Guide
- The Power of Words - Lesson Video
- **Common Sense Education** - Website
- **CSF Digital Citizenship** - Resource List

**For the Students**
- **Feeling Faces** - Emotion Images
- **Think Spot Journal** - Reflection Journal

**Vocabulary**

- **Cyberbully** - Using technology tools to deliberately upset someone else.
Teaching Guide

Warm Up (5 min)

Introduction


Invite the students to suggest emotions that match each face's expression. With every suggestion, write the emotion next to the feeling face. Answers will vary.

Tell students that not everyone will react to a particular situation in the same way, but just because a reaction is different from our own, doesn't mean we should discount others' feelings.

Explain to students they are going to watch a video about how words, whether typed or spoken, can impact how someone else feels.

Show students The Power of Words - Lesson Video.

Ask:
- Who has heard of the saying, “Sticks and stones may break my bones, but words will never hurt me”?
- What did Guts mean in his text that sometimes words can hurt?
  - Words are powerful. Sometimes it is hard to ignore what someone is saying when it's a mean name. Names can make you feel sad or hurt.

Remind students to keep Leg's question in the back of their mind during this lesson How do you treat others online?

Main Activity (35 min)

What’s the Problem?

Organize students into groups of four and have each group pick a person to record their ideas.

Distribute the Words Can Hurt Student Handout. Have the groups of students read the scenario about Rani and Aruna receiving mean messages through a children's game website.

Have each group answer the questions, then have them share their responses with the class. Look for responses that show empathy for Rani and Aruna and acknowledge that the messages sent to them were mean and hurtful. Ask the students to read the 'Use Common Sense!' section on the Words Can Hurt Student Handout.

Invite students to share their own stories.

Ask:
- Have you seen mean messages sent to you or others online? Tell us about it, but do not use real names.

Divide students into pairs.

 Invite one partner to write the phrase “You're weird” on a piece of paper, then hand it to their partner. Tell them that they just received this text.

Ask:
- What are the reasons the person might have texted “You're weird”?
  - They're continuing an inside joke; the first person did something silly at an earlier time; a group of kids is teasing the kid; the person who sent the text really does think the person is weird but is afraid to say it to his or her face.
- How did the partner feel about being called weird?
  - Possibly like the other person was kidding around, but maybe that the person was teasing or being hurtful.
Tell one person from each pair to say to the other person, "You're weird," with a smile on their face.

Ask:
- Why might you feel differently if you could see the person?
  - People give non-verbal cues through facial expressions and body language.

Crossing the Line

Place the piece of string across the length of the classroom. Ask students to stand on one side of the line. Then ask them to imagine that they are online and somebody has sent them a message, which you will read to them. Tell the students to stay where they are if they think the message is okay; to cross over the line if they think the message is not okay; or to stand on the line if they think the message is in between.

Read:
- You are my friend.
- You are an idiot.
- I'm having a party and you're not invited.
- I like your new haircut.
- You are ugly.
- Thanks for the advice. Next time, will you tell me in person rather than through text?
- Did you finish your homework?
- Why is it taking you so long to finish it?
- You are such a freak.

Review with the students that kids like to go online and use cell phones to email, chat, watch videos, send messages, play games, and do homework. But sometimes the language can get mean or scary. Messages that make people feel bad cross the line. Sometimes that meanness is unintentional, but when people use tools such as the internet and cell phones to deliberately upset someone else over and over, that's cyberbullying.

Talk and Take Action

Have students return to their seats.

Discuss how easy it is to feel angry or upset when somebody sends you a mean or scary message online.

Define:
- **Cyberbullying**: Using technology tools such as the internet and cell phones to deliberately upset someone else.

Explain that cyberbullies deliberately try to make you feel that way, just like real-life bullies.

Discuss:
- Cooling down can be a good first step when you receive a mean message online. Taking a deep breath, counting backwards from 10, or pausing to think about what you will do next can give you time to think of the BEST way to handle the situation.
- Finding help or telling a trusted adult or friend can be a good way to take action. You shouldn't deal with the cyberbullying situation alone. The person you tell should be someone who wants to hear what you have to say and will help you work on a solution. Adults can be especially good because they often have the power to influence the situation or they can give you advice about what to do.
- Ignoring the person who is cyberbullying you can be very effective. Those who bully often like attention.
- Whatever you do, remember to keep a copy of your communication with the individual who is cyberbullying you. If you delete the communication, there is no proof of how the bully treated you if you need to show a trusted adult.

Distribute the **Talk and Take Action Student Handout** to each student. Encourage them to depict a cyberbullying scenario and possible solution. They can use pencils and paper to make the comics.

Wrap Up (15 min)
Flash Chat: What did we learn today?

You can use these questions to assess your students’ understanding of the lesson objectives. You may want to ask students to reflect in writing on these questions in their Think Spot Journal - Reflection Journal.

Ask:

- Why is it a bad idea to send mean or scary messages online?
  - Because they can make the person who gets the message upset, angry, or scared.
- Why might there be more misunderstandings between people when they send online messages as opposed to a face-to-face discussion?
  - Online messages can be more confusing or scarier than face-to-face messages because there are no face-to-face cues to help you understand people’s intentions.
- What can kids do when they get cyberbullying messages?
  1. Stay calm and take a deep breath
  2. Tell a friend or trusted adult who can help develop a plan to handle the situation
  3. Ignore the bully
  4. Keep a copy of the communication with the bully.

Journaling

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

Journal Prompts:

- What was today’s lesson about?
- How do you feel about today’s lesson?
- What is cyberbullying?
- Who are some people you can go to if you are ever bullied online or in person?

Assessment (10 min)

Print out the assessment from Common Sense Education - Power of Words - Teacher Prep Guide (page 8-9) and distribute it to the class. Give students enough time to complete the assessment, but make sure there is enough time to go over answers.

Standards Alignment

CSTA K-12 Computer Science Standards

- NI - Networks & the Internet

If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 28: Envelope Variables

Overview
Variables are used as placeholders for values such as numbers or words. Variables allow for a lot of freedom in programming. Instead of having to type out a phrase many times or remember an obscure number, computer scientists can use variables to reference them. This lesson helps to explain what variables are and how we can use them in many different ways. The idea of variables isn't an easy concept to grasp, so we recommend allowing plenty of time for discussion at the end of the lesson.

Purpose
Variables are very helpful in programming. Students will be introduced to this topic using envelopes to represent variables that have been given names. The value of the variable will be written on a card inside of an envelope. This lesson helps students understand how names can be a placeholder for values in the physical world, so that programming with variables will seem less confusing in the virtual world.

Agenda
- Warm Up (10 min)
  - Vocabulary
  - Introduction
- Main Activity (20 min)
  - Envelope Variables - Worksheet
- Wrap Up (10 min)
  - Flash Chat: What did we learn?
  - Journaling
- Assessment (10 min)
  - Envelope Variables - Assessment

Extended Learning

View on Code Studio

Objectives
Students will be able to:
- Identify variables and determine their values.
- Define and call variables in the context of real-life activities.
- Create situations which require the use of variables.

Preparation
- Watch the Envelope Variables - Teacher Video.
- Obtain 6 or more blank envelopes for warm up plus some for the main activity.
- Print one Envelope Variables - Worksheet per student.
- Print one Envelope Variables - Assessment for each student.
- Provide students with envelopes, paper, pens & pencils.
- Make sure every student has a Think Spot Journal - Reflection Journal.

Links

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

For the Teacher
- Variables in Envelopes - Unplugged Video (download)
- Envelope Variables - Teacher Video
- Envelope Variables - Worksheet
- Envelope Variables - Worksheet Answer Key
- Envelope Variables - Assessment
- Envelope Variables - Assessment Answer
Key

For the Students
- Think Spot Journal - Reflection Journal

Vocabulary
- Variable - A placeholder for a piece of information that can change.
Teaching Guide

Warm Up (10 min)

Vocabulary
This lesson has one important word:

- **Variable** - Say it with me: Vayr-ee-ah-buhl
A placeholder for a piece of information that can change.

Introduction
Call four volunteers to the front of the room and line them up. Let the students know that you are going to write a poem for each of them.

On the board (or under your document camera) write the sentence for your first student (suppose it's Bill):

"My student Bill, standing proud
is a fine example for the crowd"

Encourage the students to clap at your abilities and thank Bill for volunteering. Allow Bill to sit down (or go to the back of the line) as you erase the board, then call the next volunteer (we'll say that she's called Annie).

"My student Annie, standing proud
is a fine example for the crowd"

Again, accepting applause, erase the board and invite the next volunteer.

"My student Jenny, standing proud
is a fine example for the crowd"

As you call the final volunteer, inquire as to whether everyone in the class would like a poem written about each of them. Maybe the everyone in the whole school? Goodness, that's going to take a while! Pose the question to your students:

"How could I do this more quickly?"

Your students will likely pick up on the fact that only one word is changing, and that word is simply a person's name. Help them see the location by circling Jenny's name on the board and writing "firstName" next to it.

"It would take a long time to write a poem for everyone in the school if I couldn't start until I knew who I was writing it about, wouldn't it?"

- How long do you think it would take to make a video game if they couldn't start until they knew your username?
- How expensive would video games be if they had to be created separately for each person?
- How do you think we can get around that?

By this time, it's quite likely that your class will come up with the idea of having a placeholder. With that, they're most of the way into understanding where this lesson goes.

- What would we call that placeholder?
  - We need to call it something that makes sense. We wouldn't want to call it "age" if it was a placeholder for their name, right?

Now, let's add some more volunteers. Give them each a piece of paper to write their name on, and have them tuck it inside individual envelopes labeled firstName.

This time, put the poem on the board with a blank space labeled "firstName" where the student's name will go.

- Have the first student in line (likely the last student from the previous example) pull their name from the envelope and that's what you'll write in the space.
- When you erase the board, only erase the portion with the last student's name in it.
Call the next student to show their variable.
Repeat as many times as is entertaining

Now it's time for the main activity.

Main Activity (20 min)

Envelope Variables - Worksheet
Once the students understand how the envelopes relate to the sentences, pass out the activity worksheet and let them prepare some variables of their own.

Directions:
- Divide students into groups of 2-4.
- Have students design (draw) a robot.
- After 10-15 minutes, request that the students fill their envelopes with important details about their robot such as its name, height, and purpose.
- Collect each group's envelopes, then bring them to the front of the room to share.
- Write on the board, "My robot's name is robotName, it is numUnitsTall tall, and it's purpose is purpose."
- Use the envelopes to fill the appropriate variable in the sentence, then ask each group to stand when they hear the sentence that describes their creation.

Wrap Up (10 min)

Flash Chat: What did we learn?
- What did we learn today?
- Can you think of anywhere that you have seen variables before?
- There is at least one variable at the top of most homework hand outs? Can you think of what it could be?
- Why do you think that professionals do not put spaces in variable names?
- What would happen if there was a variable "eye" a variable "color" and a variable "eye color"?
- Variables can be used to store numbers, too.
  - Suppose I have envelopes labeled num1 and num2, then I write num1+num2?
  - What happens if the "num1" envelope contains the number 4 and "num2" contains the number 5?

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

Journal Prompts:
- What was today's lesson about?
- How do you feel about today's lesson?
- What is a variable?
- Why do you think variables are important in programming?

Assessment (10 min)

Envelope Variables - Assessment
Allow students enough time to finish this assessment. If you are willing to spare more time, go over the answers as a class.

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

**What's in the box?**

- Draw boxes on a piece of paper with simple mathematical operators between them.
- For instance [] + [] = []
- Have similar size squares with numbers between 1 & 20.
- Ask one student to come create a true equation, using the numbers provided.
- Once the student has finished (and the class verifies the equation) exchange one of the numbers with another one, then remove a second number entirely.
- Tell the students that there is a hidden number in the empty box that makes that equation true again.
- What number is in the box?
- Play this game over and over again until you can remove the number from any location and the students can figure out what it is supposed to be.

**Standards Alignment**

**CSTA K-12 Computer Science Standards**

- AP - Algorithms & Programming

If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 29: Variables with Artist

Variable | Artist

Overview
In this lesson, students will explore the creation of repetitive designs using variables in the Artist environment. Students will learn how variables can be used to make code easier to write and easier to read, even when the values don’t change at runtime.

Purpose
This stage teaches the most basic use for variables, as a constant that reoccurs frequently in a program.

Agenda
- Warm Up (15 min)
  - Introduction
- Bridging Activity - Variables (15 min)
  - Unplugged Activity Using Variables as Constant
  - Preview of Online Puzzles as a Class
- Main Activity (30 min)
  - CSF Express Course - Website
- Wrap Up (15 min)
  - Journaling

Objectives
Students will be able to:
- Assign values to existing variables.
- Utilize variables in place of repetitive values inside of a program.

Preparation
- Play through the CSF Express Course 2018 - Website associated with this lesson to find any potential problem areas for your class.
- Review CS Fundamentals Main Activity Tips - Lesson Recommendations.
- Make sure every student has a Think Spot Journal - Reflection Journal.

Links

For the Teacher
- CSF Express Course 2018 - Website
- CS Fundamentals Main Activity Tips - Lesson Recommendations

For the Students
- Variables - Student Video (download)
- Unplugged Blockly Blocks (Grades 2 - 5) - Manipulatives (download)
- Think Spot Journal - Reflection Journal

Vocabulary
- Constant - A variable used throughout a program that never changes value
- Variable - A placeholder for a piece of
information that can change.
Teaching Guide

Warm Up (15 min)

Introduction

This is the first online lesson dealing with variables, so it might be a worthwhile exercise to review the "Envelope Variables" unplugged activity from last time, as well as the vocabulary that was introduced in that lesson.

Discussion:

✦ What is a variable? (A placeholder for a piece of information that can change.)
✦ When can a variable be helpful? (When you don't know what information is going to be used in a certain place until runtime, or when you have lots of places that one piece of information will be used, but that information might change someday.)

Ask the class when they could see a variable being helpful in programming. When would they NOT want to use a variable?

If the class seems interested, continue the discussion. Otherwise, move on to one of the bridging activities.

Bridging Activity - Variables (15 min)

This activity will help bring the unplugged concepts from "Envelope Variables" into the online world that the students are moving into.

Unplugged Activity Using Variables as Constant

Discuss: Remember our robot article? It used each variable only once...but what if I wanted to mention the robot's name several times?

Display: Write a paragraph on the board that refers back to the name of a specific robot several times. Ask the students what happens when you need to make the article about a different robot.

Think/Pair: Ask students to work together to see if they can come up with an idea to make changing the article for each robot easier.

Share: Work with students until you eventually get to the place where you have defined a variable called robotName somewhere before the paragraph, then set robotName equal to the robot that you are writing the article about. Replace all specific naming instances for the previous robot with the variable robotName.

Preview of Online Puzzles as a Class

Demo: Display a puzzle for the class. We recommend the 6th puzzle. Go over the code with the students to make sure they understand what's happening before they help you convert the code to use variables. Can they think of something that might happen that would make them really glad that they used variables instead of hardcoded numbers?

Transition: Now it's time for your students to move to their own machines and get started!

Main Activity (30 min)

CSF Express Course - Website

Notice that this stage first covers the idea of a variable as a constant (a variable that you use in many places, but it does not change.) This might be something that students find helpful as they're creating their own projects.

Watch out for puzzle #6. It is the first time that students will be expected to set a variable on their own. This can be tricky if they don't have a true grasp on the concept. If they're having trouble, send them back to the prediction level (#5) and have
them explain to their partners why the answer ended up what it was. Once both partners are convinced, let them continue back to puzzle #6.

**Wrap Up (15 min)**

**Journaling**

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

**Journal Prompts:**
- What was today’s lesson about?
- How did you feel during today’s lesson?
- What is a variable? Why is it helpful in programming?
- How well do you think you understand variables? (Answer on a scale from 1-5 or with an emoticon.) If you're having troubles, can you put into words what you don't understand?

**Standards Alignment**

**CSTA K-12 Computer Science Standards**

- AP - Algorithms & Programming

---

If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 30: Changing Variables with Bee

Variables | Bee

Overview

This lesson will help illustrate how variables can make programs more powerful by allowing values to change while the code is running.

Purpose

You don't always know what a value is going to be before you begin your program. Sometimes, values change while your code is running. This lesson will illustrate how code with changing values can be helpful.

Agenda

- Warm Up (15 min)
  - Introduction
- Main Activity (30 min)
  - CSF Express Course - Website
- Wrap Up (15 min)
  - Journaling

View on Code Studio

Objectives

Students will be able to:
- Identify areas where they can use variables to modify quantities during runtime.
- Examine code to find places where variables can be substituted for specific values.

Preparation

- Play through the CSF Express Course 2018 - Website associated with this lesson to find any potential problem areas for your class.
- Review CS Fundamentals Main Activity Tips - Lesson Recommendations.
- Make sure every student has aThink Spot Journal - Reflection Journal.

Links

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

For the Teacher

- CSF Express Course 2018 - Website
- CS Fundamentals Main Activity Tips - Lesson Recommendations

For the Students

- Think Spot Journal - Reflection Journal

Vocabulary

- Variable - A placeholder for a piece of information that can change.
Introduction
This series is a little different than what students have done in the past. Now, instead of simply assigning a value to a variable and running your code, you'll need to help students see how a variable can be modified during program runtime.

Display: Show students the play area from one of the later puzzles.

There are several things to unpack here, so you might need to give your students a chance to look at it critically before you expect them to do anything with it.

Think/Pair: Suppose all of the flowers/honeycomb in this picture had the same amount of nectar/honey. How would you solve this puzzle?

Share: Let students share their ideas until you decide on one that the entire class is good with.

Discuss: Now, imagine that we didn't want to have to write a separate chunk of code for each set of flowers and honeycomb. How could we use a variable to have our loop do this for us?
(Eventually, you'll want to get to the place where you initialize a variable to the original value, then change it each time through the loop so that it's ready for the next time.)

Main Activity (30 min)

CSF Express Course - Website

This set of puzzles takes some serious computational thinking skills. If you find that students are getting stuck, help them break down the puzzles into the individual pieces:

- What would it look like if the flowers/honeycomb all had the same amount of nectar/honey?
- What would it look like without the functions?
- Now how can you use a variable to get the quantities the way you want them?
- Now can you build it back up to use a function?

Hint: Puzzle 7 becomes much easier if students utilize the while path ahead loop instead of a variable.

Wrap Up (15 min)

Journaling

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

Journal Prompts:

- What was today's lesson about?
- How did you feel during today's lesson?
- What are some ways you have used variables so far?
- What else do you think you can do with variables?

Standards Alignment

CSTA K-12 Computer Science Standards

- AP - Algorithms & Programming

If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 31: Changing Variables with Artist

Variable | Artist

Overview
In this lesson, students will explore the creation of repetitive designs using variables in the Artist environment. Students will learn how variables can be used to make code easier to write and easier to read. After guided puzzles, students will end in a freestyle level to show what they have learned and create their own designs.

Purpose
Variables are essentially placeholders for values that might be unknown at the time that you run your program or for values that can change during the execution of a program. These are vital to creating dynamic code because they allow your program to change and grow based on any number of potential modifications. This stage reinforces the use of variables, using the most basic capabilities of setting and using them.

Agenda

- Warm Up (5 min)
  - Introduction
- Main Activity (20 min)
  - CSF Express Course - Website
- Wrap Up (15 min)
  - Journaling

Objectives
Students will be able to:
- Assign values to existing variables.
- Utilize variables in place of repetitive values inside of a program.
- Use variables to change values inside of a loop.

Preparation
- Play through the CSF Express Course 2018 - Website associated with this lesson to find any potential problem areas for your class.
- Review CS Fundamentals Main Activity Tips - Lesson Recommendations.
- Make sure every student has a Think Spot Journal - Reflection Journal.

Links

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

For the Teacher
- CSF Express Course 2018 - Website
- CS Fundamentals Main Activity Tips - Lesson Recommendations

For the Students
- Variables - Student Video (download)
- Unplugged Blockly Blocks (Grades 2 - 5) - Manipulatives (download)
- Think Spot Journal - Reflection Journal

Vocabulary
- Variable - A placeholder for a piece of
information that can change.
Teaching Guide

Warm Up (5 min)

Introduction

It might be helpful to remind students of “Variables that Change in Bee,” since variables will be used in a similar way here.

- How can we change the value of a variable inside of a loop?
- Do we have to change the value of a variable only by one each time?

Main Activity (20 min)

CSF Express Course - Website

The latter half of this series is made up of freeplay puzzles. Students will have the opportunity to play with variables, shape, and color to create something unique.

Wrap Up (15 min)

Journaling

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

Journal Prompts:

- What was today’s lesson about?
- How did you feel during today’s lesson?
- Have you tried mixing multiple variables into one program? What might that look like? When would it be helpful?

Standards Alignment

CSTA K-12 Computer Science Standards

- AP - Algorithms & Programming

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

If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 32: For Loop Fun

Unplugged | For Loops

Overview

We know that loops allow us to do things over and over again, but now we’re going to learn how to use loops that have extra structures built right in. These new structures will allow students to create code that is more powerful and dynamic.

Purpose

At this point, students have become masters of loops. Today, they will learn about another loop commonly used in programming. The for loop repeats commands a certain number of times, but also keeps track of the values it is iterating over. For example, a for loop that begins at 4, ends with 8, and has a step value of 1 will repeat 4 times, but the values 4, 5, 6, and 7 will also be captured for use elsewhere. Using this structure with variables can create some pretty fantastic programs. Today, students will simply be learning the basics of a for loop before diving into programming with them next time!

Agenda

- Warm Up (20 min)
  - Vocabulary
  - For One and All
- Main Activity (20 min)
  - For Loop Fun - Worksheet
- Wrap Up (15 min)
  - Flash Chat: What did we learn?
  - Journaling
- Assessment (5 min)
  - For Loop Fun - Assessment
- Extended Learning

View on Code Studio

Objectives

Students will be able to:

- Determine starting value, stopping value, and stepping value for a ‘for’ loop.
- Illustrate the counter values hit each time through a for loop during runtime.

Preparation

- Watch the For Loop Fun - Teacher Video.
- Watch the For Loop Fun - Lesson in Action Video.
- Print one For Loop Fun - Worksheet per group.
- Print one For Loop Fun - Assessment for each student.
- Make sure every student has a Think Spot Journal - Reflection Journal.

Links

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

For the Teacher

- For Loop Fun - Unplugged Video (download)
- For Loop Fun - Assessment Key - Teacher Key (download)
- For Loop Fun - Teacher Video
- For Loop Fun - Lesson in Action Video
- For Loop Fun - Worksheet
- For Loop Fun - Worksheet Answer Key
- For Loop Fun - Assessment

For the Students
Vocabulary

- **For Loop** - Loops that have a predetermined beginning, end, and increment (step interval).
Vocabulary
This lesson has one new and important word:

- **For Loop** - Say it with me: For-Loop

Loops that have a predetermined start, stop, and step value.

For One and All
- Point out that there are certain loops that happen very frequently, for example, loops where you need to keep track of how many times you have been through
- Sometimes, you don’t want to start with one
- Sometimes, you don’t want to count by ones
- for loops give you a powerful way to keep a counter that starts when you want, ends when you want, and increases by whatever size step that you want

Here, you can jump right into a sample of the game (example in English)
Main Activity (20 min)

For Loop Fun - Worksheet

Sometimes we want to repeat things a certain number of times, but we want to keep track of values as we do. This is where a for loop comes in handy. When you use a for loop, you know right from the start what your beginning value is, what your ending value is, and how much the value changes each time through the loop.
It may be difficult for young students to understand this written in pseudocode, but it may be helpful to have you explain it out loud (and perhaps with a diagram) what they will be using as the content of a for loop.

- Divide students into pairs
- To start the round, each student rolls three times:
  - One die to determine the starting value of X
  - Three dice to determine the stopping value for X
  - One die to determine the stepping value of X each time through
- Use one of the provided number lines to trace the for loop that they’ve made
  - Start at the starting value of X
  - Count down the number line, circling the numbers at the rolled interval
  - Stop when you get to the predetermined stopping value
- Add all of the circled values to your score, then let the other player take a turn
- Best 2 out of 3 wins

Wrap Up (15 min)

Flash Chat: What did we learn?

- What would your interval need to be if you wanted to count from 4 to 13 by threes?
- What kinds of things do you think you could do with a for loop?
- Can you reproduce a normal loop using a for loop?
- What would you need to do?

Journaling

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

Journal Prompts:

- What was today’s lesson about?
- How do you feel about today’s lesson?
- What is a for loop?
- Why would you use a for loop instead of a repeat loop or a while loop?

Assessment (5 min)

For Loop Fun - Assessment

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.

Extended Learning

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

- Try this activity again, but this time have the start number be selected using three dice, and the stop number with only one. Make sure to have a negative increment!

Hop Scotch

- Using chalk, draw a hop scotch diagram outside on the blacktop
  - Number the squares from bottom to top
  - Have students give each other a start square, stop square, and how many at a time they need to jump
  - When the jumper is done, have them write down the loop they just performed
  - Start adding additional activities to be done at each square, this will add complexity to the written portion, as well

Standards Alignment

CSTA K-12 Computer Science Standards

- AP - Algorithms & Programming

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

If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 33: For Loops with Bee

For Loop | Bee

Overview

Featuring Bee, this lesson focuses on for loops and using an incrementing variable to solve more complicated puzzles. Students will begin by reviewing loops from previous lessons, then they'll walk through an introduction to for loops so they can more effectively solve complicated problems.

Purpose

Today's concept, for loops, are a very important topic in computer science. Not only are they widely used, the process of learning for loops enhances the learning of other important concepts (such as variables and parameters.) Students will have plenty of practice critically thinking through problems by determining the starting, ending, and stepping values for each for loop. This concept uses plenty of math as well, so feel free to pair it with a math lesson for an even deeper learning experience.

Agenda

Warm Up (15 min)
   Introduction
Bridging Activity - For Loops (15 min)
   Unplugged Activity Using Paper Blocks
   Previewing Online Puzzles as a Class
Main Activity (30 min)
   CSF Express Course - Website
Wrap Up (15 min)
   Journaling

View on Code Studio

Objectives

Students will be able to:
• Determine starting value, stopping value, and stepping value for a ‘for’ loop.
• Recognize when to use a ‘for’ loop and when to use other loops such as `repeat` and `while` loops.

Preparation

- Play through the CSF Express Course 2018 - Website associated with this lesson to find any potential problem areas for your class.
- Review CS Fundamentals Main Activity Tips - Lesson Recommendations.
- Make sure every student has a Think Spot Journal - Reflection Journal.

Links

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

For the Teacher
• CSF Express Course 2018 - Website
• CS Fundamentals Main Activity Tips - Lesson Recommendations

For the Students
• Unplugged Blockly Blocks (Grades 2 - 5) - Manipulatives (download)
• Think Spot Journal - Reflection Journal

Vocabulary

• For Loop - Loops that have a predetermined beginning, end, and increment (step interval).
Teaching Guide

Warm Up (15 min)

Introduction
Remind students of the work they did in “For Loop Fun”. Open a discussion about what they learned, why they think it might be useful, and if they had any fun. Here are some discussion starters.

- What did you learn in “For Loop Fun”?
- What are the three main components of a for loop?
  - starting value, step interval, ending value
- Why do you think a for loop might be helpful in programming?
  - Many students might not know an answer to this. Let them hypothesize, but don’t dwell on this question for too long.
- Did you have fun learning about for loops? Why or why not?
- Are you excited to use for loops in online puzzles?

Bridging Activity - For Loops (15 min)

This activity will help bring the unplugged concepts from “For Loop Fun” into the online world that the students are moving into. Choose one of the following to do with your class:

Unplugged Activity Using Paper Blocks
Split up the students of your class into pairs. Ideally have the pairs be the same from when your class did “For Loop Fun”. Print out a for loop from Unplugged Blockly Blocks (Grades 2 - 5) - Manipulatives for each pair of students. Pass out one die to each pair. Have the partners take turns rolling the die to obtain the following values:

- One roll for the starting value
- Four rolls for the ending value
- One roll for the step value

Have each pair fill in the for loop with the appropriate values in the correct spot.

![Diagram of for loop with variables](image)

Using a basic number line, like the one used in "For Loop Fun", have the students mark the beginning, ending, and middle values that this for loop will touch. When everyone is done, see who got the most points by totaling the starting, middle, and ending numbers of each pair.

Previewing Online Puzzles as a Class
Display a puzzle from the Course F Online Puzzles - Website associated with this lesson. We recommend puzzle #4 because it displays a potential solution and asks the user to evaluate it.

Using a number line similar to the ones used in "For Loop Fun", mark the start and ending values of the given for loop (if you aren’t using puzzle #4, you will need to come up with a potential solution first). With the class’s help, circle the values
between the start and end that the for loop will touch. If you are working on puzzle #4, ask the class what they think the answer is to the question, given what they found with the number line.

Main Activity (30 min)

CSF Express Course - Website

Some students may have a hard time differentiating between repeat loops and for loops. We recommend having scratch paper out for students to make guesses on values like the start, stop, and step. Implementing pair programming amongst the class might also be helpful for your students.

Wrap Up (15 min)

Journaling

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

Journal Prompts:
- What was today’s lesson about?
- How did you feel during today’s lesson?
- How is a for loop different from a repeat loop?
- Why do you think for loops could be useful?

Standards Alignment

CSTA K-12 Computer Science Standards

[ AP - Algorithms & Programming ]

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

If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 34: For Loops with Artist

Overview
In this lesson, students continue to practice for loops, but this time with Artist. Students will complete puzzles combining the ideas of variables, loops, and for loops to create complex designs. At the end, they will have a chance to create their own art in a freestyle level.

Purpose
Creativity and critical thinking come together beautifully in this lesson. Students will continue their practice with for loops and variables while they create jaw-dropping images. This lesson inspires a creative mind while teaching core concepts to computer science.

Agenda
- Warm Up (15 min)
  - Introduction
- Main Activity (30 min)
  - CSF Express Course - Website
- Wrap Up (15 min)
  - Journaling

Objectives
Students will be able to:
- Use ‘for’ loops to change loop several times with different values.
- Recognize when to use a ‘for’ loop and when to use other loops such as ‘repeat’ and ‘while’ loops.

Preparation
- Play through the CSF Express Course 2018 - Website associated with this lesson to find and potential problem areas for your class.
- Review CS Fundamentals Main Activity Tips - Lesson Recommendations.
- Make sure every student has a Think Spot Journal - Reflection Journal.

Links

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

For the Teacher
- CSF Express Course 2018 - Website
- CS Fundamentals Main Activity Tips - Lesson Recommendations

For the Students
- Think Spot Journal - Reflection Journal

Vocabulary
- For Loop - Loops that have a predetermined beginning, end, and increment (step interval).
Teaching Guide

Warm Up (15 min)

Introduction

On a board displayed to the entire class, draw (or display via projector) one of the final projects from the Course F Online Puzzles - Website associated with this lesson. We recommend one of the following:

![Drawing Examples]

Ask the class how a computer might draw the drawing you displayed.

After a few predictions have been said, reply with for loops of course!

Tell the students they will soon be learning how to create these fine drawings using for loops and variables.

Main Activity (30 min)

CSF Express Course - Website

These puzzles are super fun, but it may be helpful for students to have protractors and scratch paper to see these designs made in the physical form. If that isn’t an option in your class, try to get the students to trace on the computer screen with their fingers.

Wrap Up (15 min)

Journaling

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

Journal Prompts:
- What was today’s lesson about?
- How did you feel during today’s lesson?
- Draw one of the designs you made today. What was the code needed to create it?
- What are some designs you would like to create? How do you think for loops or variables could help create those?

Standards Alignment

CSTA K-12 Computer Science Standards

- AP - Algorithms & Programming
If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 35: Learning Sprite Lab

Event | Sprite Lab

Overview
In this lesson, students will learn about the two concepts at the heart of Sprite Lab: sprites and behaviors. Sprites are characters or objects on the screen that students can move, change, and manipulate. Behaviors are actions that sprites will take continuously until they are stopped.

Purpose
This lesson is designed to introduce students to the core vocabulary of Sprite Lab, and allow them to apply concepts they learned in other environments to this tool. By creating a fish tank, students will begin to form an understanding of the programming model of this tool, and explore ways they can use it to express themselves.

Agenda
- Warm Up (10 min)
  - Introduction
  - Review of "Personal and Private Information"
- Main Activity (20 min)
  - CSF Express Course - Website
- Wrap Up (15 min)
  - Journaling

Objectives
Students will be able to:
- Define “sprite” as a character or object on the screen that can be moved and changed.
- Create a new sprite and choose its appearance.
- Create a new sprite and choose its appearance.

Preparation
- Play through the CSF Express Course 2018 - Website associated with this lesson to find any potential problem areas for your class.
- Review CS Fundamentals Main Activity Tips - Lesson Recommendations.
- Make sure every student has a Think Spot Journal - Reflection Journal.

Links

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

For the Teacher
- Sprite Lab Pilot Doc
- CSF Express Course 2018 - Website
- CS Fundamentals Main Activity Tips - Lesson Recommendations

For the Students
- Think Spot Journal - Reflection Journal

Vocabulary
- Behavior - An action that a sprite performs continuously until it’s told to stop.
- **Sprite** - A graphic character on the screen with properties that describe its location, movement, and look.
Teaching Guide

Warm Up (10 min)

Introduction

Today students will learn how to work with sprites in Sprite Lab.

Display: Pull up a previous puzzle from Code.org, ideally one containing a "main character" like Scrat from Ice Age or one of the Angry Birds.

Discuss: Let the students know that this character on the screen is a "sprite." It is a graphic that is controlled by a program. In this lesson, students will have the opportunity to choose their own sprites to control.

Display: Begin by showing Puzzle 1 to your students.

Think/Pair: Ask them to predict what will happen when the code is run, and to discuss with their neighbors. Run the code, and discuss the outcome.

Now is a great time to introduce the lesson vocabulary.

Review of "Personal and Private Information"

Remind students of information that is safe to share online and information that is strictly private.

<table>
<thead>
<tr>
<th>SAFE - Personal Information</th>
<th>UNSAFE - Private Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Your favorite food</td>
<td>Mother’s maiden name</td>
</tr>
<tr>
<td>Your opinion</td>
<td>Social Security number</td>
</tr>
<tr>
<td>(though it should be done</td>
<td>Your date of birth</td>
</tr>
<tr>
<td>respectfully)</td>
<td>Parents’ credit card information</td>
</tr>
<tr>
<td>First name</td>
<td>Phone number</td>
</tr>
<tr>
<td>(with permission)</td>
<td></td>
</tr>
</tbody>
</table>

Discuss other examples of the two categories above.

Main Activity (20 min)

Goal: Today, students will be creating their own Fish Tank. They’ll begin by learning how to put some sprites on the screen, then they will make them move. Finally, they’ll customize their fish tank to add whatever creatures and objects they want.

CSF Express Course - Website

Transition: Move students to their machines. Encourage students to follow the instructions for each puzzle. Help them realize that this is a creative activity, intended to help them learn Sprite Lab. It is not an assessment activity of any sort.

Reminder: If puzzles are sharable, remind the students to only share their work with their close friends or family.

For more information watch or show the class Pause and Think Online - Video.
Wrap Up (15 min)

Journaling

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

Journal Prompts:
- What was today’s lesson about?
- How do you feel about today’s lesson?
- How did it feel to make a scene that was more creative?
- Was it difficult to finish a lesson where there was no clear “right” and “wrong”?

Standards Alignment

CSTA K-12 Computer Science Standards
- AP - Algorithms & Programming

Teaching Tip

Encourage students with questions/challenges to start by asking their partner. Unanswered questions can be escalated to a nearby group, who might already know the solution. Have students describe the problem that they’re seeing:
- What is it supposed to do?
- What does it do?
- What does that tell you?

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

If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 36: Alien Dance Party

Event | Sprite Lab

Overview
This lesson features Sprite Lab, a platform where students can create their own alien dance party with interactions between characters and user input. Students will work with events to create game controls.

Purpose
Students will use events to make characters move around the screen, make noises, and change backgrounds based on user input. This lesson offers a great introduction to events in programming and even gives a chance to show creativity! At the end of the puzzle sequence, students will be presented with the opportunity to share their projects.

Agenda
- Warm Up (15 min)
  - Introduction
  - Review of "Personal and Private Information"
- Main Activity (30 min)
  - CSF Express Course - Website
- Wrap Up (15 min)
  - Journaling

View on Code Studio

Objectives
Students will be able to:
- Identify actions that correlate to input events.
- Create an animated, interactive game using sequence and events.

Preparation
- Play through the CSF Express Course 2018 - Website associated with this lesson to find any potential problem areas for your class.
- Review CS Fundamentals Main Activity Tips - Lesson Recommendations.
- Make sure every student has a Think Spot Journal - Reflection Journal.

Links

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

For the Teacher
- Sprite Lab Pilot Doc [Make a Copy]
- CSF Express Course 2018 - Website
- CS Fundamentals Main Activity Tips - Lesson Recommendations

For the Students
- Think Spot Journal - Reflection Journal

Vocabulary
- Event - An action that causes something to happen.
Teaching Guide

Warm Up (15 min)

Introduction

Today students will visit events in programming.

Demo: Ask the students to raise their hands in the air.

What you did was declare an event. When you say "raise your hands in the air" the students responded by raising their hands. In coding, you would declare this by saying something like "when I say 'raise your hands,' you raise your hands".

You can also think of cities as declaring events. There are laws that say "when there is a green light, cars move through the intersection".

Discuss: Ask the students why they think this is an event.

Today, students will play in Play Lab, but the events they will be working on will be more like the video games they are used to playing. Events will take the form of actions, such as clicking the screen or two characters running into each other.

Display: Begin by showing Puzzle 1 to your students.

Think/Pair: Ask them to predict what will happen when the code is run, and to discuss with their neighbors. Run the code, and discuss the outcome.

Review of "Personal and Private Information"

Remind students of information that is safe to share online and information that is strictly private.

<table>
<thead>
<tr>
<th>SAFE - Personal Information</th>
<th>UNSAFE - Private Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Your favorite food</td>
<td>Mother's maiden name</td>
</tr>
<tr>
<td>Your opinion</td>
<td>Social Security number</td>
</tr>
<tr>
<td>(though it should be done</td>
<td>Your date of birth</td>
</tr>
<tr>
<td>respectfully)</td>
<td>Parents' credit card</td>
</tr>
<tr>
<td>First name</td>
<td>information</td>
</tr>
<tr>
<td>(with permission)</td>
<td>Phone number</td>
</tr>
</tbody>
</table>

Discuss other examples of the two categories above.

Main Activity (30 min)

Goal: Today, students will be creating their own alien dance party! They’ll begin by reviewing how to put sprites on the screen, then they will assign them behaviors and learn to change those behaviors when an event is initiated.

CSF Express Course - Website

Transition: Move students to their machines. Encourage students to follow the instructions for each puzzle. Help them realize that this is a creative activity, intended to help them learn Sprite Lab. It is not an assessment activity of any sort.

Reminder: If puzzles are sharable, remind the students to only share their work with their close friends or family. For more information watch or show the class Pause and Think Online - Video.
Wrap Up (15 min)

Journaling

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

Journal Prompts:
- What was today’s lesson about?
- How do you feel about today’s lesson?
- How did it feel to have control over what your characters were able to do?
- Did you change the program in any way to make it feel more like your own?

Standards Alignment

CSTA K-12 Computer Science Standards
- AP - Algorithms & Programming

If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 37: Pet Giraffe

Overview

Here, students will use Sprite Lab to play with sprites and their properties. Students will use events, behaviors, and custom code to create their very own pet giraffe that gets hungry, playful, and even filthy!

Purpose

Students will use events to make characters move around the screen, change size, and change colors based on user input. This lesson offers a great introduction to events in programming and even gives a chance to show creativity!

Agenda

- Warm Up (15 min)
  - Introduction
  - Review of “Personal and Private Information”
- Main Activity (30 min)
  - CSF Express Course - Website
- Wrap Up (15 min)
  - Journaling

View on Code Studio

Objectives

Students will be able to:
- Identify actions that correlate to input events.
- Create an animated, interactive game using sequence and events.

Preparation

- Play through the CSF Express Course 2018 - Website associated with this lesson to find any potential problem areas for your class.
- Review CS Fundamentals Main Activity Tips - Lesson Recommendations.
- Make sure every student has a Think Spot Journal - Reflection Journal.

Links

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

For the Teacher
- CSF Express Course 2018 - Website
- CS Fundamentals Main Activity Tips - Lesson Recommendations

For the Students
- Think Spot Journal - Reflection Journal
Teaching Guide

Warm Up (15 min)

Introduction
Today students will revisit events in programming.

Review: Ask students questions about the Alien Dance Party.
- Do you remember what an event is?
- Can you name any of the events that you used to make the aliens dance? What do they do?
  - when clicked
  - when sprite1 touches sprite2
  - when arrow pressed

Display: Begin by showing Puzzle 1 to your students.

Think/Pair: Ask them to predict what will happen when the code is run, and to discuss with their neighbors. Run the code, and discuss the outcome.

Review of "Personal and Private Information"
Remind students of information that is safe to share online and information that is strictly private.

<table>
<thead>
<tr>
<th>SAFE - Personal Information</th>
<th>UNSAFE - Private Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Your favorite food</td>
<td>Mother’s maiden name</td>
</tr>
<tr>
<td>Your opinion</td>
<td>Social Security number</td>
</tr>
<tr>
<td>(though it should be done respectfully)</td>
<td>Your date of birth</td>
</tr>
<tr>
<td>First name</td>
<td>Parents' credit card information</td>
</tr>
<tr>
<td>(with permission)</td>
<td>Phone number</td>
</tr>
</tbody>
</table>

Discuss other examples of the two categories above.

Main Activity (30 min)

Goal: Today, students will be creating their own pet giraffe! They’ll begin by reviewing how to put sprites on the screen, then they will assign events that cause actions and behaviors upon interaction.

CSF Express Course - Website

Transition: Move students to their machines. Encourage students to follow the instructions for each puzzle. Help them realize that this is a creative activity, intended to help them learn Sprite Lab. It is not an assessment activity of any sort.

Reminder: If puzzles are sharable, remind the students to only share their work with their close friends or family. For more information watch or show the class Pause and Think Online - Video.
Wrap Up (15 min)

Journaling

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

Journal Prompts:
- What was today's lesson about?
- How do you feel about today's lesson?
- What other options would you like to be able to have your pet do?

Standards Alignment

CSTA K-12 Computer Science Standards

▶ AP - Algorithms & Programming

Teaching Tip

Encourage students with questions/challenges to start by asking their partner. Unanswered questions can be escalated to a nearby group, who might already know the solution. Have students describe the problem that they're seeing:
- What is it supposed to do?
- What does it do?
- What does that tell you?

If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 38: Explore Project Ideas

Project | Define | Prepare | Try | Revise | Reflect

Overview

The next five lessons provide an opportunity for students to put their coding skills to use in a capstone project. This project will help individuals gain experience with coding and produce an exemplar to share with peers and loved ones. This is intended to be a multi-lesson or multi-week project where students spend time brainstorming, learning about the design process, building, and then presenting their final work.

In the "Explore" stage, students will play around with pre-built Artist and Sprite Lab programs for inspiration. Next, students will learn about the design process and how to implement it in their own projects. They will then be given the space to create their own project in Artist, Sprite Lab, or any other interface that you are comfortable providing. (This is likely the longest stage of the project.) Students will then revise their code after testing and peer review. Finally, students will be able to present their finished work to their classmates.

Purpose

Exploring project ideas is meant to inspire students with realistic and entertaining ideas for their culminating projects.

Agenda

Day 1 - Explore Project Ideas (45 min)
   Example Projects
Day 2 - The Design Process (45 min)
   Define and Prepare
Day 3 - Build Your Project (45 min)
   Try
Day 4 (Recommended for 5th Grade) - Revise Your Project (45 min)
   Reflect and Try Again
Day 5 & 6 - Present Your Project (45 min each)
   Presentations
   Extension Activity

View on Code Studio

Objectives

Students will be able to:
- Learn to plan in advance for an ongoing assignment.
- Be able to explain how system limitations can affect project design.
- Describe how compromise can help keep a project on track and inspire creativity.

Preparation

- Play through the online Course F Project Examples to get an idea of the strengths, weaknesses, and limitations of the tool.
- Decide whether or not you will have your students do the section on Revisions (recommended for Course F).
- Print one copy of the Design Process - Teacher Prep Guide for each student.
- Modify the CS Fundamentals Final Project - Rubric to fit your class goals and print out a copy for each student.
- Modify the Final Project Design - Worksheet to fit your class and print one packet for each student.

Links

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

For the Teacher

- Course F Project Examples
- Design Process - Teacher Prep Guide
  Make a Copy
- Final Project Design - Worksheet
  Make a Copy

Preparation

- Play through the online Course F Project Examples to get an idea of the strengths, weaknesses, and limitations of the tool.
- Decide whether or not you will have your students do the section on Revisions (recommended for Course F).
- Print one copy of the Design Process - Teacher Prep Guide for each student.
- Modify the CS Fundamentals Final Project - Rubric to fit your class goals and print out a copy for each student.
- Modify the Final Project Design - Worksheet to fit your class and print one packet for each student.

Links

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

For the Teacher

- Course F Project Examples
- Design Process - Teacher Prep Guide
  Make a Copy
- Final Project Design - Worksheet
  Make a Copy
Vocabulary

- **Define** - Figure out the details of the problems that you are trying to solve
- **Prepare** - Research, plan, and acquire materials for the activity you are about to do
- **Reflect** - Carefully think back on something with the intention of improving the outcome in the future
- **Try** - Attempt to do something
Teaching Guide

Day 1 - Explore Project Ideas (45 min)

Example Projects

Goal: This part of the process is an exploration. Students will sit down with a stage full of example projects to remix and learn. Not only will this give students an idea of what is possible, it will also help them see the limitations of the tool.

Give students a day to play with and remix the projects found in Course F Project Examples. Have them use their journals (or notebook paper) to keep track of thoughts and ideas as they go.

This activity should be done in the same pairs/groups that will be working on projects together over the next several lessons.

Make sure your class understands that they will be spending the next several weeks working with projects of their own, so they should pay close attention to how these programs were written, as well as the concepts that they use.

Day 2 - The Design Process (45 min)

Define and Prepare

Goal: Students will come up with a project and plan their strategy for programming that project in a single day. Students should have a project sketch and a description by the time the day is done.

Preparing Students for the Process:

The most important responsibility you have in kicking off this segment is to help your class understand the scope of this project. Students should be clear about the various expectations over the coming weeks so that they can prepare for their review and presentations appropriately.

To help your class manage this multi-stage undertaking, they should be given both the Final Project Design - Worksheet and the CS Fundamentals Final Project - Rubric on the first day of planning. Students will then be able to follow the rubric each step of the way to predict what their project grade will be in the end.

The Final Project Design Worksheet will provide a place for students to capture relevant thoughts and processes as they go, so they are more prepared for their reviews and presentations in the end.

As the teacher, you should download a copy of the documents and decide which elements are important to you. Be sure to edit or remove anything that you do not intend to draw student focus.

Define and Prepare:

Now that the class has their Final Project Design Worksheet in hand, they should start filling out the questions under Day 1.

Students will likely need to refer back to their notes from playing with the example projects, especially if they don’t have access to online Artist or Play Lab project levels while they plan.

Students should focus on defining and planning their project during Day 1, and not cross over into building until their ideas have been written up and/or drawn out.

If students get stuck, help them work through ideas by asking questions and recalling examples, rather than offering solutions.

Day 3 - Build Your Project (45 min)
Try

Goal: Students will use this day to build an initial version of their project.

Equipped with their Final Project Design Worksheet, students should head to the computers to start bringing their projects to life.

This process will come complete with plenty of trial and error. Projects are likely to become truncated versions of the original scope (if not morphed altogether). Remind students that this kind of compromise is common in software design, but they need to be sure to document the reasons for the changes in their product.

Don’t let the class forget to fill out their Final Project Design Worksheets as they go. It might be helpful to suggest that pairs/groups take a worksheet break to begin discussing these questions about halfway through their lab time. Alternatively, the navigator can keep their eyes open for pertinent answers while the driver codes.

Be sure that each team member has their own Final Project Design Worksheet, as there are questions about each student's own individual thoughts and behaviors that need to get captured along the way.

Day 4 (Recommended for 5th Grade) - Revise Your Project (45 min)

Reflect and Try Again

Goal: Students will work with another group to give and receive feedback in an effort to make each other's projects stronger.

Reflect:

For reflections, have each group pair up with another group to try each other's projects. After about 10 minutes, have the groups discuss the questions in the Final Project Design Worksheet.

Encourage students to ask the questions on the FPDW and write down feedback provided by their reviewing teams so that they can refer back to it later. This portion should take approximately 15 more minutes.

Try Again:

With their new reflections in hand, students can head back to their machines to make a handful of edits. With just 10 minutes left, they will likely have to select only the most important feedback to incorporate.

Day 5 & 6 - Present Your Project (45 min each)

Presentations

Goal: Students will create and present their projects in an approved manner (written, oral, or using multimedia).

Create:

Ideally, you will have class time available to give students to work on their presentations. This will allow them to incorporate rich multimedia components, like Google Slides. For other presentation ideas, visit 72 Creative Ways for Your Students to Show What They Know - Website.

Encourage students to include all of the information from Section J of the Final Project Design Worksheet into their presentation, as well as two or more questions from Section K.
Present

Students should showcase their apps first, then they can discuss the questions that they covered in their presentations.

It can be very helpful to have students sign up for a specific order in which to give their presentations, so that they are able to enjoy the demonstrations of their classmates without worrying about whether they will be called on next.

Extension Activity

If your students are already comfortable with coding concepts, try having them create their projects in another platform, like Scratch or Alice.

Standards Alignment

CSTA K-12 Computer Science Standards

- AP - Algorithms & Programming

If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 39: The Design Process

Project

Overview

Over the course of five lessons, students will be building up to building a project of their own design using either Sprite Lab or Artist as their programming environment. In this portion of the project, students will learn about the design process and how to implement it in their own projects. The lesson guide overviewing all five stages of the process can be found in the beginning of the project process, here.

Purpose

Students may be ready to jump straight into building their projects, but this lesson will help shape their ideas into plans. This structure will keep the dreamers grounded and illuminate a path for those feeling left in the dark.

Agenda

Day 2 - The Design Process (45 min)
Define and Prepare

View on Code Studio

Objectives

Students will be able to:
- Shape ideas into reasonable goals and plans.
- Recognize any potential obstacles such as time constraints or bugs.
Teaching Guide

Day 2 - The Design Process (45 min)

Define and Prepare
Students will come up with a project and plan their strategy for programming that project in a single day. Students should have a project sketch and a description by the time the day is done.

Preparing Students for the Process:
The most important responsibility you have in kicking off this segment is to help your class understand the scope of this project. Students should be clear about the various expectations over the coming weeks so that they can prepare for their presentations appropriately.

To help your class manage this multi-stage undertaking, they should be given both the Final Project Design - Worksheet and the CS Fundamentals Final Project - Rubric on the first day of planning. Students will then be able to follow the rubric each step of the way to predict what their project grade will be in the end.

The Final Project Design - Worksheet will provide a place for students to capture relevant thoughts and processes as they go, so they are more prepared for their presentations in the end.

As the teacher, you should decide which elements of these documents are important to you and be sure to edit or remove anything that you do not intend to draw student focus.

Define and Prepare:
Now that the class has their Final Project Design - Worksheet in hand, they should start filling out the questions under Day 1.

Students will likely need to refer back to their notes from playing with the example projects, especially if they don't have access to online Artist or Play Lab project levels while they plan.

Students should focus on defining and planning their project during Day 1, and not cross over into building until their ideas have been written up and/or drawn out.

If students get stuck, help them work through ideas by asking questions and recalling examples, rather than offering solutions.

Standards Alignment

CSTA K-12 Computer Science Standards

- AP - Algorithms & Programming

If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 40: Build Your Project

Project Overview

Over the course of five lessons, students will be building up to building a project of their own design using either Sprite Lab or Artist as their programming environment. Now the students will be given their own space to create their project with either Artist or Sprite Lab. This will be the longest portion of the project. The lesson guide overviewing all five stages of the process can be found in the beginning of the project process, here.

Purpose

This lesson provides students with ample time to build and revise their projects. The trial and error inevitably involved in this lesson will teach problem solving and persistence.

Agenda

Day 3 - Build Your Project (45 min)

Try

View on Code Studio

Objectives

Students will be able to:

- Use the planned design as a blueprint for creation.
- Overcome obstacles such as time constraints or bugs.
Teaching Guide

Day 3 - Build Your Project (45 min)

Try

Students will use this day to build an initial version of their project.

Equipped with their Final Project Design - Worksheet, students should head to the computers to start bringing their projects to life.

This process will come complete with plenty of trial and error. Projects are likely to become truncated versions of the original scope (if not morphed altogether). Remind students that this kind of compromise is common in software design, but they need to be sure to document the reasons for the changes in their product.

Don’t let the class forget to fill out their Final Project Design - Worksheet as they go. It might be helpful to suggest that pairs/groups take a worksheet break to begin discussing these questions about halfway through their lab time. Alternatively, the navigator can keep their eyes open for pertinent answers while the driver codes.

Be sure that each team member has their own Final Project Design Worksheet, as there are questions about each student’s own individual thoughts and behaviors that need to get captured along the way.

Standards Alignment

CSTA K-12 Computer Science Standards

- AP - Algorithms & Programming

If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 41: Revise Your Project

Project

Overview

Over the course of five lessons, students will be building up to building a project of their own design using either Sprite Lab or Artist as their programming environment. Now that the projects are built, students are given the opportunity to get feedback from peers and revise their projects. The lesson guide overviewing all five stages of the process can be found in the beginning of the project process, here.

Purpose

This lesson helps students take a step back and view their project from a new perspective. Here, students will be able to decide if they have reached their goals. If they haven't, this lesson gives them time and space to complete the project.

Agenda

Day 4 - Revise Your Project (45 min)
  Reflect and Try Again

View on Code Studio

Objectives

Students will be able to:
- Determine if the criteria set in a rubric has been met with their current project.
- Draft and implement plans to resolve any issues in their code.
Lesson Tip:
Teachers should avoid assigning the final bit of project work as homework unless they are certain that students both live within a close proximity to one another and have internet access at home.

Teaching Guide

Day 4 - Revise Your Project (45 min)

Reflect and Try Again

Goal: Students will work with another group to give and receive feedback in an effort to make each other’s projects stronger.

Reflect:
For reflections, have each group pair up with another group to try each other’s projects. After about 10 minutes, have the groups discuss the questions in the Final Project Design Worksheet.

Encourage students to ask the questions on the FPDW and write down feedback provided by their reviewing teams so that they can refer back to it later. This portion should take approximately 15 more minutes.

Try Again:
With their new reflections in hand, students can head back to their machines to make a handful of edits. With just 10 minutes left, they will likely have to select only the most important feedback to incorporate.

Standards Alignment

CSTA K-12 Computer Science Standards

- AP - Algorithms & Programming

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

If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 42: Present Your Project

Project

Overview

Over the course of five lessons, students will be building up to building a project of their own design using either Sprite Lab or Artist as their programming environment. Finally, students will be able to present their finished work to their peers or share with their loved ones with a special link. The lesson guide overviewing all five stages of the process can be found in the beginning of the project process, here.

Purpose

At this point, students have worked very hard on their projects, so this lesson is meant to offer a space for the students to share their projects. This lesson will build a supportive community where students will build their own confidence and feel connected to their hardworking peers.

Agenda

Day 5 & 6 - Present Your Project (45 min each)
  Presentations

View on Code Studio

Objectives

Students will be able to:

- Clearly indicate where each criteria point from the rubric is satisfied in the code for the finished culminating project.
- Articulate the design process and how it helped shape the finished culminating project.
Teaching Guide

Day 5 & 6 - Present Your Project (45 min each)

Presentations

Students will create and present their projects in an approved manner (written, oral, or using multimedia).

Create:

Ideally, you will have class time available to give students to work on their presentations. This will allow them to incorporate rich multimedia components, like Google Slides. For other presentation ideas, visit 72 Creative Ways for Your Students to Show What They Know - Website.

Encourage students to include all of the information from Section J of the Final Project Design Worksheet into their presentation, as well as two or more questions from Section K.

Present:

Students should showcase their apps first, then they can discuss the questions that they covered in their presentations.

It can be very helpful to have students sign up for a specific order in which to give their presentations, so that they are able to enjoy the demonstrations of their classmates without worrying about whether they will be called on next.

Standards Alignment

CSTA K-12 Computer Science Standards

► AP - Algorithms & Programming

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

If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 43: Binary Bracelets

Unplugged | Binary

Overview

Binary is extremely important in the world of computers. The majority of computers today store all sorts of information in binary form. This lesson helps demonstrate how it is possible to take something from real life and translate it into a series of ons and offs.

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.

Agenda

- Warm Up (15 min)
  - Vocabulary
  - Off and On
- Main Activity (20 min)
  - Binary Bracelets - Worksheet
- Wrap Up (5 min)
  - Flash Chat: What did we learn?
  - Journaling
- Assessment (15 min)
  - Binary Bracelets - Assessment
- Extended Learning

View on Code Studio

Objectives

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

Preparation

- Watch the Binary Bracelets - Teacher Video.
- Watch the Binary Bracelets - Lesson in Action Video.
- Gather markers for the bracelets. Other decorations like beads and pipe cleaners are optional.
- Print one Binary Bracelets - Worksheet per student.
- Print one Binary Bracelets - Assessment per student.
- Make sure every student has a Think Spot Journal - Reflection Journal.
- (Optional) Write a short message on the board in binary.
- Prepare to show the Bits Versus Bytes - Student Video.

Links

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

For the Teacher
- Binary Bracelets - Unplugged Video (download)
- Binary Bracelets - Teacher Video
- Binary Bracelets - Lesson in Action Video
Vocabulary

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

This lesson has one new and important word:

Binary - Say it with me: Bye-nair-ee

A way of representing information using only two options

Off and On

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

- 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.
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.
- How do you suppose we can convert real-life things that we want to store in a computer into binary?
  - Let's start with letters.
  - Use the Binary Bracelets - Worksheet to show how a computer might represent capital letters.
    - 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.
    - Watch the Bits Versus Bytes - Student Video (~1 minute)
  - 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 min)

**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.

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.
- 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.

<table>
<thead>
<tr>
<th>A</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>O</td>
</tr>
<tr>
<td>C</td>
<td>P</td>
</tr>
<tr>
<td>D</td>
<td>Q</td>
</tr>
<tr>
<td>E</td>
<td>R</td>
</tr>
<tr>
<td>F</td>
<td>S</td>
</tr>
<tr>
<td>G</td>
<td>T</td>
</tr>
<tr>
<td>H</td>
<td>U</td>
</tr>
<tr>
<td>I</td>
<td>V</td>
</tr>
<tr>
<td>J</td>
<td>W</td>
</tr>
<tr>
<td>K</td>
<td>X</td>
</tr>
<tr>
<td>L</td>
<td>Y</td>
</tr>
<tr>
<td>M</td>
<td>Z</td>
</tr>
</tbody>
</table>

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

Lesson 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.
Wrap Up (5 min)

Flash Chat: What did we learn?
- 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?

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

Journal Prompts:
- What was today’s lesson about?
- How did you feel during today’s lesson?
- Use the activity worksheet to write out the rest of your name or your favorite word in binary.
- Imagine a world where we spoke in binary, saying “on” or “off”, but nothing else. Draw two characters trying to talk to each other in binary.

Assessment (15 min)

Binary Bracelets - Assessment
- 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.

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.

Standards Alignment

CSTA K-12 Computer Science Standards
- AP - Algorithms & Programming

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

If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 44: Binary Images

Binary | Unplugged

Overview

Though many people think of binary as strictly zeros and ones, students will be introduced to the idea that information can be represented in a variety of binary options. This lesson takes that concept one step further as it illustrates how a computer can store even more complex information (such as images and colors) in binary, as well.

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 code and share it with peers to view as images. 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 stored.

Agenda

- Warm Up (10 min)
  - Vocabulary
  - Introduction to Binary
- Main Activity (20 min)
  - Binary Images - Worksheet
- Wrap Up (10 min)
  - Flash Chat: What did we learn?
  - Journaling
- Assessment (10 min)
  - Binary Image - Assessment
- Extended Learning

View on Code Studio

Objectives

Students will be able to:

- Identify methods for encoding images into binary.
- Relate images to a peer using binary encoding.
- Reproduce an image, based on binary code.

Preparation

- Watch the Binary Images - Teacher Video.
- Print one Binary Images - Worksheet per pair.
- Print one Binary Image - Assessment per student.
- Gather groupings of items that can show opposites for students to use when coming up with their own binary encodings (Optional).
- Make sure every student has a Think Spot Journal - Reflection Journal.

Links

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

For the Teacher

- Binary Images - Unplugged Video (download)
- Binary Images - Teacher Video
- Binary Images - Worksheet
- Binary Image - Assessment
- Binary Images - Assessment Answer Key
- Binary Baubles - Thinkersmith Lesson
For the Students
- Think Spot Journal - Reflection Journal

Vocabulary
- **Binary** - A way of representing information using only two options.
- **Binary Alphabet** - The two options used in your binary code.
Teaching Guide

Warm Up (10 min)

Vocabulary
This lesson has two new terms:

- **Binary** - Say it with me: Bi-na-re-ee
  A way of representing information using only two options.
- **Binary Alphabet** - Say it with me: Bi-na-re-ee Al-fa-bet
  The two options used in your binary code.

Introduction to Binary
What if we had a picture like this, where there's only two color options for each square, black or white.

![Binary Image](https://example.com/binary.png)

How might we encode this so that someone else could recreate the picture without seeing it?

- Some students might think back to the Graph Paper Programming lesson. While there could be a lot of similarities, let them know that this is different enough that they should not use that lesson to guide this one.

You may hear suggestions like: "Say 'black, white, white, black'."

- "That's a great suggestion! Now I'm going to break you up into pairs. Work with your teammate to decide on a binary alphabet."

Decide whether you want your pairs to share their encodings with the other groups ahead of time, and tell them if they will be creating a key, or keeping their methods secret.

- '"Now, let's encode some images, just like a computer would!"

Main Activity (20 min)

Binary Images - Worksheet
Now it's the students' turn!

Activity Directions:
1. Divide students into pairs.
2. Have them choose an image with their partner.
3. Encourage them to figure out what their binary alphabet is going to be.
4. Have them encode their image using their new binary alphabet.
5. Instruct students to trade encodings with another team and see if they can figure out which picture the other worked on.
6. Choose a Level
   - Easy: Let the other team know what your encoding method was
   - Tough: Have the other team guess your encoding method.

Wrap Up (10 min)
Flash Chat: What did we learn?

- What did we learn today?
- What kind of binary alphabet did you create?
- Can you think of how you could encode an image using only your fingers?
- Do you think you could create a binary alphabet out of sounds?

Journaling

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

Journal Prompts:

- What was today's lesson about?
- How do you feel about today's lesson?
- What is a binary alphabet?
- What kind of information can you share using binary?

Assessment (10 min)

Binary Image - Assessment

Pass out this assessment for students to do individually. Try to save time at the end to go over answers.

Extended Learning

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

Storing Color Images

- If your class really gets the idea behind storing binary images, they may want to know how to do color images.
  - First, you'll need to discuss how color works using binary (as in Binary Baubles - Thinkersmith Lesson, page 21).
  - Then, introduce some images that use combinations of those colors
- Encourage your students to come up with ways to code these color images.

Hexadecimal

- Take the idea of color one step further to introduce hexadecimal color codes.

Standards Alignment

CSTA K-12 Computer Science Standards

- AP - Algorithms & Programming

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

If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 45: Binary Images with Artist

Binary | Artist

Overview
This series of online lessons will have students learning to make images using on and off.

Purpose
This will help reinforce the fact that computers can do a multitude of things with 0s and 1s.

Agenda
- Warm Up (15)
- Main Activity (30)
  - CSF Express Course - Website
- Wrap Up (15)
  - Journaling

View on Code Studio

Objectives
Students will be able to:
- Create pictures using unique combinations of on and off
- Identify repeated sequences and break long codes up into smaller chunks that can be looped
- Utilize loops and binary code to recreate provided images

Preparation
- Play through the CSF Express Course 2018 - Website corresponding to this course to find any potential problem areas for your class.
- Review CS Fundamentals Main Activity Tips - Lesson Recommendations.
- Make sure every student has aThink Spot Journal - Reflection Journal.

Links

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

For the Teacher
- CSF Express Course 2018 - Website

Vocabulary
- Binary - A way of representing information using only two options.
Teaching Guide

Warm Up (15)

To begin, it can be helpful to review the previous lesson, specifically different ways of using binary to indicate how to create an image on a grid. This stage will translate the unplugged activity into a simple, independent online lesson.

Main Activity (30)

CSF Express Course - Website

Not all of these images are intuitive. Encourage students to click “Run” to see what happens, even if their code isn't "finished" yet.

Wrap Up (15)

Journaling

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

Journal Prompts:
- What was today’s lesson about?
- How did you feel during today’s lesson?
- Did you like drawing on the 8x8 grid or the 16x16 grid better? Why?
- Computers also store sounds using binary. Use your imagination to write down a possible way for that to work.

Standards Alignment

CSTA K-12 Computer Science Standards

- AP - Algorithms & Programming
Lesson 46: The Internet

Overview

Even though many people use the internet daily, not very many know how it works. In this lesson, students will pretend to flow through the internet, all the while learning about connections, URLs, IP Addresses, and the DNS.

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.

Agenda

- Warm Up (20 min)
  - Vocabulary
  - Getting the Message
- Main Activity (20 min)
  - The Internet
- Wrap Up (15 min)
  - Flash Chat: What did we learn?
  - Journaling
- Assessment (5 min)
  - Internet - Assessment

View on Code Studio

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 Think Spot Journal - Reflection Journal.

Links

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

For the Teacher

- The Internet - Unplugged Video (download)
- Internet - Teacher Video
- IP Address Cards and Delivery Type Cards - Manipulatives
- Internet - Assessment
- Internet - Assessment Answer Key

For the Students

- Think Spot Journal - Reflection Journal

Vocabulary
- **DNS** - The service that translates URLs to IP addresses.
- **DSL/Cable** - A method of sending information using telephone or television cables.
- **Fiber Optic Cable** - A connection that uses light to transmit information.
- **Internet** - A group of computers and servers that are connected to each other.
- **IP Address** - A number assigned to any item that is connected to the Internet.
- **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 min)

Vocabulary
This lesson has several new and important words:

- **IP Address** - Say it with me: I-P Add-ress
  A number assigned to any item that is connected to the Internet
- **DNS (Domain Name Service)** - Say it with me: D-N-S
  The service that translates URLs to IP addresses
- **URL (Universal Resource Locator)** - Say it with me: U-R-L
  An easy-to-remember address for calling a web page (like www.code.org)
- **Internet** - Say it with me: In-ter-net
  A group of computers and servers that are connected to each other
- **Servers** - Say it with me: Ser-vers
  Computers that exist only to provide things to others
- **Fiber Optic Cable** - Say it with me: Fye-ber Op-tic Cay-bl
  A connection that uses light to transmit information
- **Wi-Fi** - Say it with me: Wye-Fye
  A wireless method of sending information using radio waves
- **DSL/Cable** - Say it with me: D-S-L / Cay-bl
  A method of sending information using telephone or television cables
- **Packets** - Say it with me: Pack-ets
  Small chunks of information that have been carefully formed from larger chunks of information

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 send required thousands of envelopes!
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.

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.

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.)

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.)

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!

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:

<table>
<thead>
<tr>
<th>#</th>
<th>URL</th>
<th>IP Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>code.org</td>
<td>54.243.71.82</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Lesson 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.
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 min)**

**The Internet**

**Directions:**
- 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)
- Each group will draw an IP Address Cards and Delivery Type Cards - Manipulatives 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!

**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.

**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.
Lesson 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.

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.

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 (in English):

![Sample of Classroom Group Layout During Game Play](image)

Wrap Up (15 min)

Flash Chat: What did we learn?
- 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?

Journaling

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

**Journal Prompts:**
- What was today’s lesson about?
- How do you feel about today’s lesson?
- What’s something you learned about the internet today?
- Why is learning about the internet important?

### Assessment (5 min)

**Internet - Assessment**

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.

### Standards Alignment

**CSTA K-12 Computer Science Standards**

- **NI** - Networks & the Internet

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

If you are interested in licensing Code.org materials for commercial purposes, contact us.
Lesson 47: Crowdsourcing

Unplugged | Crowdsourcing

Overview

In computer science, we face some big, daunting problems. Challenges such as finding large prime numbers or sequencing DNA are almost impossible to do as an individual. Adding the power of others makes these tasks manageable. This lesson will show your students how helpful teamwork can be in the industry of computer science.

Purpose

It’s very rare that one computer scientist works completely alone on a project. Even when that does happen, there is always benefit in numbers. Today, students will learn what it means to crowdsource a project. This activity builds teamwork and creates an efficient environment for students to solve problems.

Agenda

- Warm Up (20 min)
  - Vocabulary
  - Introduction
- Main Activity (20 min)
  - Crowdsourcing - Worksheet
- Wrap Up (15 min)
  - Flash Chat: What did we learn?
  - Journaling
- Extended Learning

View on Code Studio

Objectives

Students will be able to:
- Identify a large task that needs to be done.
- Rearrange a large task into several smaller tasks.
- Build a complete solution from several smaller solutions.

Preparation

- Watch the Crowdsourcing - Teacher Video.
- Review Crowdsourcing - Worksheet.
- Obtain a jar of lots of something (pennies, buttons, slips of paper, etc) and a deck of cards.
- Make sure every student has a Think Spot Journal - Reflection Journal.

Links

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

For the Teacher

- Crowdsourcing - Unplugged Video (download)
- Crowdsourcing - Teacher Video
- Crowdsourcing - Worksheet

For the Students

- Think Spot Journal - Reflection Journal

Vocabulary

- Crowdsourcing - Getting help from a large group of people to finish something faster.
Warm Up (20 min)

Vocabulary
This lesson has one new and important word:
Crowdsourcing - Say it with me: Crowd-sore-sing
Getting help from a large group of people to finish something faster.

Introduction
- Show your students your jar full of something.
  - "Look at this jar. I have a lot of buttons in here, and I need to tell the principal how many there are before the end of class."
  - "Can you think of a way I could get these counted quickly?"
- Your students may guide you toward seeking help, but if they don’t, you can suggest it, too.
- Pour all of the buttons (or pennies, etc.) into a pile on the floor.
- Invite all of the students to come up and grab a small number (ten is good, but you can do more if your students can handle it).
- Once they’ve counted out their ten, have them report to you, drop their buttons back in the jar, and go again until the pile is gone.
- Comment on how fast the task went.
  - Have the class reflect on how long it might have taken or how hard it may have felt to do alone.

Main Activity (20 min)
Crowdsourcing - Worksheet
Sometimes you have a big job that needs to get done, but it feels like it will take forever. Crowdsourcing is a way of using teamwork to make the job go much faster! In this game, we’ll use crowdsourcing to sort decks of playing cards.

Directions:
1. Divide into groups of 4, 5, or 6.
2. Grab your deck of playing cards and dump it into a bag, bucket, or even a loose pocket that you can make with the bottom of your shirt.
3. Shake the cards until they’re all mixed up.
4. Dump the cards out onto a table or desk where the whole group can see them.
5. Decide how to break up the task of sorting the deck so that every person has something to do and no one is doing too much.
6. Time yourself sorting the cards. Can you figure out a way to do it faster?
7. Repeat the game over and over until you think you have found the fastest way of crowdsourcing the card sorting activity.
Wrap Up (15 min)

Flash Chat: What did we learn?
- Have you ever tried to sort a pile of cards by yourself?
- Do you think it was easier or harder to have help?
- What other things do you have to do in life that could be easier with help?

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

Journal Prompts:
- What was today’s lesson about?
- How do you feel about today’s lesson?
- What are the benefits of crowdsourcing?
- What kind of things do you want to make with computer science? How do you see crowdsourcing being beneficial in those projects?

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

Reverse Crowdsourcing
Often we think of crowdsourcing as pulling things apart to make them more simple. You can also make big, beautiful things with the same technique.

Have your students each grab three cards and build one segment of a card house. Each student can go one after another to build a grand card tower.

Try with two, or even three students adding their chunk at a time.
- Does crowdsourcing always make a task easier?

Crowdsourcing in the Round
- You can crowdsource all at the same time or you can do it one person at a time. Try having the whole class sort the same deck of cards, one student at a time.
  - Shuffle the cards and place them in a pile in the center of the room.
  - Have each student approach the pile and choose four cards.
    - Have four piles for the students to sort their cards into
      - Spades
      - Clubs
      - Hearts
      - Diamonds
    - Once all cards have been put in their four piles, have the following four students sort the individual piles.
    - The last person will put all four piles together.
- This version may not save a lot of time, but it still divides the work and lets each individual have more free time!

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