Table of Contents

Welcome to getCoding

Coding is so much more than a language. It's a tool for creating video games, music, and art. A tool for changing the world.

We believe that all students should be equipped with the coding skills they need to be creators and active citizens–not just consumers–in the new digital world.

Our coding curriculum gives you the tools you need to empower your students to be digital developers, artists, and citizens. Our curriculum is broken up into 3 stages:

1. preCoder - An introduction to coding concepts for students who do not yet have the language skills to read and write. (grades K-2)
2. coder - A slowly paced deep dive into core coding concepts through puzzle-based lessons. (grades 3-8)

3. developer - An exploration of the many applications of coding concepts through project- and product-based lessons. (grades 3-8)

Ideally students would start with the preCoder stage and move to the coder and developer stages in 3rd grade. However, teachers should feel free to grab units from preCoder, coder, and developer as they see fit.

Let's getCoding!

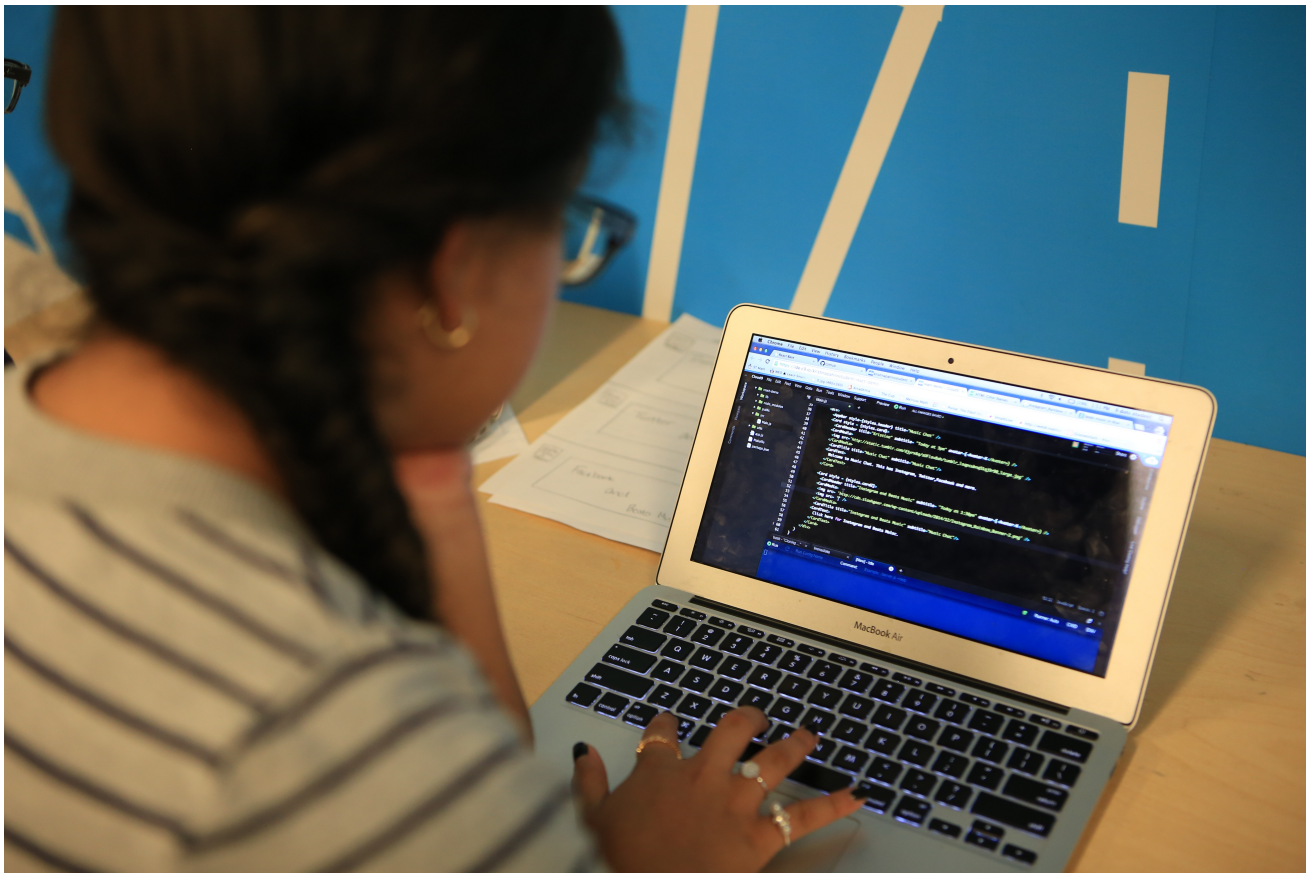
preCoder

The preCoder series targets learners who cannot yet read and write. The series introduces and reinforces coding concepts through fun and engaging play-based curricula. Lessons include unplugged activities, worksheets, and online games.

Curriculum

Coming Soon!

Coder



The coder series features game-based curricula designed to introduce and reinforce coding concepts in a fun and engaging way for students. These units include unplugged activities where students can explore coding in a physical space, worksheets to help organize student thinking, and online games to practice coding skills.

Curriculum

- Level 1 (Coming Soon!)
- Level 2
- Level 3

Coder Level 2

Students are introduced to core coding concepts: sequences, calling functions, passing arguments, creating loops, and using conditionals. New concepts are introduced in a steady progression from an icon-based language to a block-based language and finally to typed JavaScript. This course includes lesson plans (unplugged and online), worksheets, and a custom-designed coding environment.

Lessons

- sequencing pixels
- write some code
- write read repeat
- pixel bot online
- code.org maze
- dance dance js

- pixel bot js
- winter is coming - gather food
- space ranger and magic words
- coding arguments

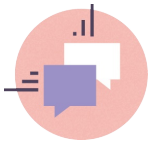
Download lesson plans

Download

Workbook

Download

Sequencing Pixels
Unplugged



OVERVIEW

In this lesson, students play with the order of commands in unplugged Pixel Bot exercises. The lesson explores a foundational concept in computer science—sequence.

OBJECTIVES

- Know that computers run code in a sequence.
- Read, write, and execute code in a sequence.

AGENDA

Length: 45 minutes

- Whet students' appetites for coding. Hold a discussion about how code shapes our world. Consider autonomous cars, music, online communities. Watch "A Day in the Life of a Software Engineer" (http://tinyurl.com/q966xd5). (10 minutes)

- Guide students to see the importance of sequence. Use Lesson 1 | Warm-up Worksheet. Students first develop ideas about how the arrow and paint elements work in Pixel Bot. Once settled on the meaning of each element, students read the two programs in the worksheet. The programs have slightly different sequences. Students predict what Pixel Bot will paint for each program and compare the results. The class reflects on the idea of sequence. (15 minutes)

- Demonstrate how a computer executes a program. Draw a Pixel Bot program and grid on the whiteboard (3 or 4 lines of code, 3 x 3 grid). Work with students to read,

step through, and number the program. While reading the program, trace and paint with the Magnetic Pixel Bot. Repeat a few times with new programs. (10 minutes)

- Students practice reading and executing code. Use Worksheet 1 | Page 1 & Page 2. Students read and step through code while tracing and painting with the pixel bot. (10 minutes)

CONTENT KNOWLEDGE

- Sequence - The idea that statements must be performed in the order they are written.

MATERIALS

1. Lesson 1 | Warm-up Worksheet
2. Worksheet 1 | Page 1 & Page 2
3. Small pixel bot cutout for each student
4. Magnetic pixel bot
5. Scratch paper grids
6. Pencils
7. Whiteboard
8. Queued up video: http://tinyurl.com/q966xd5

WELCOME TO CODING

Length: 10 minutes

Teacher Actions	Student Actions

1 Lead a discussion:
 	
`What does it mean to be a coder ?`
 	
`Where do we use code?`
 	
Chart responses on the whiteboard.
	1 Coders:
	
	`Work on computers`
	`Hack things`
	`Create video games`
	`Make websites`
	`Work with data`
	
	Code runs in:
	
	`Phones`
	`Computers`
	`Traffic lights`
	`Spaceships`
	`Game consoles`
	`Movies and tv shows`

2 Offer other examples:
 	
`Autonomous cars`
`Streetlights`
`Music`
`Flight simulators`
`And on and on...`	

3 Watch video: A day in the life of a software engineer: [http://tinyurl.com/q966xd5] (http://tinyurl.com/q966xd	
5	

PREDICT PIXEL BOT ICONS

Length: 15 minutes

Teacher Actions	Student Actions
1 Distribute Lesson 1 | Warm-up Worksheet.	

2 Introduce reading code with care:
`Imagine being a computer when y ou read code.`
`As computers, we read carefully . We pay attention to every det ail. Every line of code.`	

3 Have students interpret the coding elements in the warm-up:
`What do you think these element s do?`
	3 Possible Responses:
	`The bot rotates in place.`
	`The bot jumps.`
	`The bot moves until it hits the edge.`
	`The bot shoots lightning.`
	Correct Responses:
	`The bot moves one square at a t`

```
ime.
The bot moves up, right, left,
and down.
The bot paints.
```

4 Individual Work: Have students fill out the worksheet.

4 Trace the movement of the Bot. Paint with the Bot.

5 While students code, draw the worksheet's programs and grids on the whiteboard.

6 Reflect on sequence:

```
How did you arrive at your answ
er?
```

```
What is the difference between
the two programs?
```

```
How does the order of the icons
 matter?
```

6 I arrived at my answer by:

```
Reading one element at a time.
Moving the bot after each eleme
nt.
Following the correct sequence.
```

The two programs:

```
Have elements in different orde
rs.
```

Order matters because:

```
The bot paints a different squa
re.
```

7 Solve the two warm-up problems together on the whiteboard.

7 Call out lines of code and bot actions.

8 Work as a class to define each element.	8 Work as a class to figure out what each element makes the bot do.

DEMONSTRATE HOW TO READ AND STEP THROUGH PROGRAMS

Length: 10 minutes

Teacher Actions	Student Actions
1 Draw a blank 3x3 grid on the whiteboard.	
2 Write a short (3 or 4 line) program on the whiteboard.	
3 Explain sequence: `When a computer executes code, it steps through the code in th e correct order. This is called sequencing.`	
4 Read the first line of code together: `What is the first line of code?` Number the first line of code.	4 Students call out the first line of code.

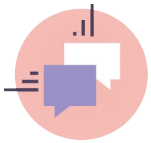
5 Move the Pixel Bot after reading each line. Trace its path or shade in a square.	5 Students call out where Pixel Bot should move.
6 Continue reading and stepping one line at a time.	6 Students continue helping to read the code.
7 Read and step through three new examples with the class. Design problems on the fly, making them interesting and complex enough.	

READ PIXEL BOT ICONS

Length: 5 minutes

Teacher Actions	Student Actions
1 Distribute Worksheet 1: Page 1 & Page 2.	
2 Leave the worked example from the previous activity on the whiteboard.	
3 Individual Work: Ask students to individually fill out the worksheet.	3 Students read the code, trace the pathway of the pixel bot, and paint the correct blocks on the worksheet.

Write Some Code
Unplugged



OVERVIEW

Students learn the different components of reading and writing code by exploring the roles (writer, reader, navigator, and stepper) of Coders & Bots.

OBJECTIVES

- Students will be able to read basic code.
- Students will be able to write basic code.
- Together as a class, students will be able to enact the Coders & Bots roles of Writer, Navigator, Reader, and Stepper to write and read code.

AGENDA

Length: 45 mintues

1. Pixel Bots: Practice Reading Code
2. Pixel Bots: Write Code
3. Pixel Bots: Write Code Together

VOCAB

- Computer - A device that can be instructed to do something.
- Program - A list of statements that a computer can perform.

MATERIALS

1. Lesson 2 | Warm-up Worksheet
2. Lesson 2 | Worksheet 1
3. Small pixel bot cutout for each student
4. Magnetic pixel bot
5. Scratch paper grids
6. Pencils
7. Whiteboard

PIXEL BOTS: PRACTICE READING CODE

Length: 15 minutes

Students practice reading basic block code sequences.

Prep: Draw on the whiteboard the grid and the lines of code from the example problem on the front page of Lesson 2 | Warm-up Worksheet

Teacher Actions	Student Actions
1 Remind students of the value of revisiting ideas explored in previous lessons. Revisiting past ideas helps to see them in a new light and makes sure they are not forgotten.	
2 Step through the example problem as a whole class, talking through the process of reading code.	
3 Hand out Lesson 2 | Warm-up Worksheet.	3 Students place their pixel bot at the starting square. Students read the code and move their pixel bot. Students do this a few times to get into a rhythm.

4 Individual Work: Ask students to place the movable pixel bot at the start block of the example problem just demonstrated on the board. Students should work individually, reading the code and moving their pixel bot along. Ask the students to get into a rhythm of reading and stepping.	
5 Individual Work: Ask students to flip the page and perform the same exercise with the new problem. Ask students to shade in the appropriate squares and trace the pathway of the pixel bot.	5 Students turn the page and attempt to solve the problem.
6 On the board, draw the empty grid and the code from the problem students have been working on. Read the code and step the pixel bot (tracing its pathway and shading in squares), narrating your thought process.	

PIXEL BOTS: WRITE CODE

Length: 10 minutes

Students write code to produce a simple pixel bot image.

Prep: Hand out Lesson 2 | Worksheet 1.

Teacher Actions	Student Actions
1 Talk to students about how they have already learned to read code that other people have written. Now they are going to explore writing code.	
2 Individual Work: Have students place their pixel bot at the start square. Ask students to write code that produces the provided pixel bot image. The students should think through their plan for their code, write their code, and move their pixel bot along.	2 Students place their pixel bot at the starting square, write code that creates the picture, and enact the pixel bot actions each step of the way.
3 As a whole class, solicit ideas from multiple students about how they designed their code to solve the problem. Hold off on presenting the correct answer until the next activity.	3 Students share their code, especially if they have solved the problem a different way.

PIXEL BOTS: WRITE CODE TOGETHER

Length: 20 minutes

Teacher shows student how to write code, emphasizing roles of the Writer, Navigator, Reader, and Stepper (see Coders and Bots Protocol). The students then enact these roles together as a whole class solving a new problem.

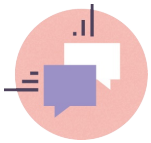
Prep: Draw the problem from Lesson 2 | Worksheet 1 on the whiteboard.

Teacher Actions	Student Actions
1 Code the solution to Lesson 2 | Worksheet 1 on the whiteboard. Voice your code writing process along the way: write and number a new line of code and only then move the pixel bot on the whiteboard. Use this as an opportunity to discuss how there are different ways to solve the same problem. After the code is written, introduce the idea that the computer reads code in sequence.	
2 After coding the solution, explain that you are switching gears into Bot mode. Read each line and step the pixel bot, checking to see if you coded the correct solution. Add a new problem on the board: a	2 Students take turns walking up to the board to write and navigate code.

checkerboard pattern. Divide the class in half and follow the process spelled out in the whole class section of the Coders and Bots Protocol. Students start as Coders. Assign half of the students to be writers and half to be navigators; one person from each team walks up to collaboratively add and enact a line of code. The two students then tag in a member of their team to walk up and write the next line of code. Allow the class to code a wrong solution.	
3 The whole class then switches roles to become Bots. Use this opportunity to emphasize the definition of a computer (see vocabulary section above). Following the same procedure as above, ask students to come up to the board to take turns reading and stepping through one line of code at a time before tagging in a new classmate.	3 Students take turns walking up to the board to read and step through the solution.
4 Have students switch back and forth between Coders and Bots until they code the solution on the checkerboard.	4 Students switch back and forth between Coders and Bots to complete the puzzle.

5 Summarize any conceptual difficulties. Introduce and define a Program (see vocabulary section above), and explain how it connects to the code the students just wrote.	
6 Ask students to write down on a piece of paper what each of the the four roles does, focusing on one role at a time. After each role definition, ask students to share their definition with the whole class. Pool the students' ideas into overall definitions of the roles.	6 Students provide descriptions of what each role entails.

Write.Read.Repeat.
Unplugged



OVERVIEW

Students continue to solve pixel bot problems by playing Coders & Bots to read and write code.

OBJECTIVES

- Students will perform the Coder & Bots roles of Writer, Navigator, Reader, and Stepper
- Students will collaborate in small groups to write and read code
- Students will become more proficient at writing code

AGENDA

Length: 45 minutes

1. Pixel Bots: Write Code Warm-up
2. Pixel Bots: Coders and Bots
3. Pixel Bots: Write Code Exit Ticket

VOCAB

- Programming Element - A command the computer understands.
- Action - An observable event that the code produces
- Error - The program fails to run because the computer cannot execute the code

MATERIALS

1. Lesson 3 | Warm-up Worksheet
2. Lesson 3 | Worksheet 1
3. Lesson 3 | Worksheet 2
4. Lesson 3 | Worksheet 3
5. Lesson 3 | Worksheet 4
6. Lesson 3 | Exit Ticket Worksheet
7. Scratch paper grids
8. Small pixel bot cutout for each student
9. Magnetic pixel bot
10. Scratch paper grids
11. Pencils
12. Whiteboard

CLASSROOM SETUP

Pods of four.

PIXEL BOTS: WRITE CODE WARM-UP

Length: 15 minutes

Students write code for a simple pixel bot image.

Prep: Hand out Lesson 3 | Warm-up.

Teacher Actions	Student Actions
1 Individual Work: Ask students to solve the problem on Lesson 3 | Warm-up Worksheet.	1 Students individually solve the problem on the worksheet.
2 As students attempt the problem, draw the problem on the whiteboard.	
3 Code the solution to the problem as a whole class. Call on students at random to provide each next line of code. Each student should walk up to the board and draw the next symbol. With each new symbol, you should play the role of Navigator, moving the turtle. Call out the roles of Writer and Navigator to make them explicit. On occasion, pick a Reader and Stepper to walk up to the board to test the code starting from line one.	3 If called on, students walk up to the board to write in (or read) the next line of code.

PIXEL BOTS: CODERS AND BOTS

Length: 30 minutes

Students work in groups to write and read code to produce two pixel bot images.

Prep: Consider having paper bags for each group. Each paper bags contains the four roles (Writer, Navigator, Reader, Stepper).

Teacher Actions	Student Actions
1 Teacher breaks students into groups of four and randomly assigns roles for the Programming Team and the Computer Team. (Consider handing each group a paper bag with the four roles inside – students reach into the bag and grab a role.) In this first round, groups either get Lesson 3 | Worksheet 1 or Lesson 3 | Worksheet 2 (alternate between groups). Bots should always help the Coders during the code writing phase of the activity.	1 Students enact their Coders and Bots roles. They write code to create the pixel bot image.
2 Follow the Coders and Bots Protocol by having the Bot Team switch to a new group when it comes time to check the code. Make sure that the Coders fold back their paper to hide the	2 The Bots switch to a new group to assess code. If the Bots find an error, report it to the Coders and tell them what happened and the line number the error happened on.

	provided pixel bot image before the Bots arrive (ensuring that the Bots are not biased in their reading). The Bots should test the code on an empty scratch paper grid. Ask students, "What do you think the computer does when it cannot understand the code or the code forces the turtle to break the rules (go outside the grid)?" Answer: An error! If the Bots notice this happening, they should report the error to the programming team and tell them what line the error happened on.		
3	After one round, pass out Lesson 3 | Worksheet 3 or Lesson 3 | Worksheet 4 to the groups (alternate again) and repeat the Coders and Bots Protocol.	3	Students repeat the above process with new coding challenges.
4	In whole class mode, ask students to share out any disagreements they had in the write and read process. Use this as an occasion to firm up any misconceptions. Also use this time to introduce and define Elements and Actions (see vocabulary section above).	4	Students summarize and describe the disagreements they could not resolve.

PIXEL BOTS: WRITE CODE EXIT TICKET

Length: 5 minutes

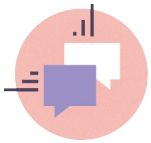
Time permitting, students individually fill out an exit ticket to check for understanding.

Prep: Hand out Lesson 3 | Exit Ticket Worksheet

Teacher Actions	Student Actions
1 Ask students to individually work on completing the Lesson 3 | Exit Ticket Worksheet.	1 Students work individually on the Lesson 3 | Exit Ticket Worksheet.

Pixel Bots Online

Plugged



OVERVIEW

Students review writing code offline. Then, students demonstrate their learning by writing programs to complete online pixel bot challenges.

OBJECTIVES

- Students will write write programs using an icon language.
- Students will continue to develop proficiency in writing and reading code.

AGENDA

Length: 45 mintues

1. Pixel Bots: Warm-up
2. Introduce Pixel Bots Online
3. Pixel Bots Online Practice
4. Pixel Bots: Exit Ticket

VOCAB

- Program - A list of statements that a computer can perform.

MATERIALS

1. Lesson 4 | Worksheet 1

2. Lesson 4 | Exit Ticket Worksheet
3. Scratch paper grids
4. Small pixel bot cutout for each student
5. Magnetic pixel bot
6. Scratch paper grids
7. Pencils
8. Whiteboard

PIXEL BOTS: WARM UP

Length: 10 minutes

Create a medium difficulty pixel bot image for the students on the whiteboard. Students solve the problem individually and then check the work of a peer.

Prep: Create a medium difficulty pixel bot image on the whiteboard.

Teacher Actions	Student Actions
1 Individual Work: Ask students to get a blank piece of paper, write line numbers, and write the code that creates the pixel bot image on the board.	1 Students work individually on coding a solution to the challenge.
2 Students discuss the coding solution with a peer.	2 When they finish, students check their work with a partner after both have finished.

INTRODUCE PIXEL BOT ONLINE

Length: 10 minutes

Show students how to solve a problem on pixel bot online. Also revisit `program` as a vocabulary word.

Prep: Browse to www.pixelbots.io and project onto the wall.

Teacher Actions	Student Actions
1 Now that students have practice creating code sequences on paper, they are ready to start writing programs.	1 Add student actions here and match numbers to teacher actions
2 Revisit the idea that a program is a sequence that a computer is able to understand.	
3 On a projector, show students www.pixelbots.io	
4 Use the problem you created in the warm-up to demonstrate how to solve the problem in the	

online interface. Show students a variety of features in the pixel bot interface: • How to add code • How to delete code • How to run program • How to reset after run • How to insert code	
5 Ask students, "What do you think we should do if we get stuck on a problem?"	5 Answer: Step through code starting at the beginning like you do as a Bot.

PIXEL BOT ONLINE PRACTICE

Length: 20 minutes

Students are given a set of pixel bot images to reproduce on the computer. Partners star the ones that are completed. Remind students that the images with lots of shaded squares will be difficult, but they should remember to read code from the beginning when things go wrong.

Prep: Distribute the Lesson 4 | Worksheet 1 and write www.pixelbots.io up on the board.

Teacher Actions	Student Actions
1 Explain the exercise: • The goal is to recreate each of the images from the worksheet by creating a program on the website. • Tell students that they will act as testers for the person sitting next to them: ◦ When they finish writing a program, students have their partner check to make sure the images match up. If they are a match, the tester puts a checkmark next to the image. ◦ The programmer should then explain how their code works to the tester. If the tester is	1 Students are faced away from their computers toward the teacher.

satisfied with the explanation, the tester checks the explain box. ◦ The programmer can now continue on to the next challenge.	
2 Students work on completing the challenges.	2 Students get on their computers and go to www.pixelbots.io. Students work individually on recreating each of the images with code. • When they finish a puzzle, students have their partner (student sitting next to them) check to make sure the images match up. If they are a match, the checker puts a checkmark next to the image and the programmer can continue on to the next challenge.
3 When students get stuck, ask them to imagine being on the Coder team from the group activities. They should try to play the roles of the writer and navigator. Then, ask students to imagine being a Bot to understand how the computer is reading the code.	3 Students raise their hands to provide answers.
4	

Discuss: Which was the hardest coding challenge? Why? Was there more than one way to solve the problem?	

PIXEL BOTS: EXIT TICKET

Length: 5 minutes

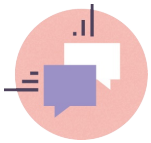
Students complete Exit Ticket.

Prep: Distribute Lesson 4 | Exit Ticket Worksheet.

Teacher Actions	Student Actions
1 Individual Work: Tell students they are going to work individually on coding the answer on the worksheet.	1 Students write their answers on the worksheet.

Code.org Maze

Plugged



OVERVIEW

Students will build on their understanding of algorithms learned in Pixelbots to solve problems with different commands in the Code.org environment.

OBJECTIVES

- Students will decompose problems with new elements
- Students will learn to use a block-based language

AGENDA

Length: 45 minutes

1. Solving problems with new code elements
2. Introduce code.org maze
3. Code.org Maze: Deliberate practice

VOCAB

Programming Language - A set of programming elements used to communicate to a computer.

MATERIALS

1. Scratch paper grids

2. Small pixel bot cutout for each student
3. Magnetic pixel bot
4. Scratch paper grids
5. Pencils
6. Whiteboard

CODE ELEMENT CHANGE-UP

Length: 15 minutes

Students are given different code elements to solve similar pixel bots problems.

Prep: Draw a medium difficulty coding challenge on the board.

Teacher Actions	Student Actions
1 Draw the normal icon elements students are accustomed to and ask students to use them to solve the puzzle.	1 Students use the elements to develop a solution.
2 Discuss student solutions.	2 Students raise their hands to offer solutions.
3 Introduce students to a different set of elements (degree turns – see the arrows below) by drawing them on the board. What do students think they mean? How much should they turn right? How much should they turn left? How far should they move forward? • ↑ - move one block forward • ↱ - turn right 90 degrees	3 Students raise their hands to discuss the new elements.

• - turn left 90 degrees	
4 Students solve the puzzle using the new code elements.	4 Students solve the problem using the new elements.
5 Discuss: What was the difference between the elements? How do the elements you have affect your solution?	5 Students raise their hand to discuss the difference when using new elements.

INTRO TO CODE.ORG MAZE

Length: 5 minutes

Introduce students to code.org by projecting the teacher screen and solving a maze puzzle as a class.

Prep:

1. Set up projector
2. Navigate to https://studio.code.org/s/course2/stage/3/puzzle/1.

Teacher Actions	Student Actions
1 Point out that code.org uses a different programming language. 1. Ask students to look at the available code elements (on the left side of the editor) and talk about what is different from pixelbots.io 2. A programming language is how a person can communicate with a computer. There are many programming languages. 3. Pixel bots used an icon language. Code.org uses a block based language.	1 Students offer answers about the differences between the elements in pixelbots.io and code.org
2 students the projected screen and show them how to use the code.org interface: • How to drag and drop blocks into the code editor • How to run the code • How to reset after a run	2 Students are off the computer and facing the teacher
3 As a class solve the first puzzle.	3 Students raise their hands to offer answers about what blocks to add to the code.

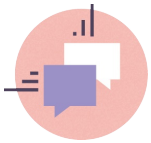
CODE.ORG MAZE: WRITE CODE PRACTICE

Length: 30 minutes

Students do the rest of the Code.org maze puzzles on their own.

Teacher Actions	Student Actions
1 Explain the exercise and facilitate students navigating their browsers to the correct place. 1. Students navigate to studio.code.org 2. Click on course 2 3. Click on the first Stage 3: Maze: Sequence	1 Students open their web browsers and navigate to the challenges 1. Students navigate to studio.code.org 2. Click on course 2 3. Click on the first Stage 3: Maze: Sequence
2 Tell students to work through as many of the exercises as they can.	2 Students work on completing the mazes.
3 When students get stuck, use the Writing, Reading, and Debugging protocols to support student learning. Ask students to imagine being on the Coder team from the group activities. They should try to play the roles of the writer and navigator. Then, ask students to imagine being a Bot to understand how the computer is reading the code.	

Dance Dance JS
Unplugged



OVERVIEW

Students will learn the JavaScript syntax for a function call and apply that knowledge and their knowledge of sequencing to write real JavaScript programs.

OBJECTIVES

- Students will learn the syntax for a function call.
- Students will continue to develop proficiency in writing and reading code.
- Students will develop pairs programming proficiency.
- Students will work in teams to solve problems through the use of the Coders and Bots Protocol.

AGENDA

Length: 45 minutes

1. Offline Pixel Bot Warm-up (5 minutes)
2. Pixel Bot JS Intro to Function Calls (15 minutes)
3. Dance Dance JS (25 minutes)

VOCAB

- Function call - A programming element that tells the computer to do something. In the beginning, most function calls will cause the computer to perform an action.
- Syntax - The way a programming element should be written.
- Programming language - A set of elements used to communicate with a computer.

MATERIALS

1. Lesson 6 | Warm-up Worksheet
2. Lesson 6 | Worksheet 1
3. Lesson 6 | Worksheet 2
4. Scratch paper grids
5. Small turtle cutout for each student
6. Magnetic turtle
7. Scratch paper grids
8. Pencils
9. Whiteboard

OFFLINE PIXEL BOT WARM-UP

Length: 5 minutes

Students warm up their code writing skills by performing a medium difficulty pixel bot exercise.

Prep: Hand out Lesson 6 | Warm-up Worksheet.

Teacher Actions	Student Actions
1 Ask students to solve the problem on the Lesson 6 | Warm-up Worksheet.	1 Students complete the warm-up.

INTRODUCE SYNTAX

Length: 15 minutes

Students perform the same exercise as in the warm-up, but the programming elements are replaced with real JavaScript syntax. Teacher emphasizes the importance of getting the syntax right.

Prep: Hand out Lesson 6 | Worksheet 1.

Teacher Actions	Student Actions
1 Point out to students that the programming elements changed but that they may still be able to solve the problem.	
2 Individual Work: Ask students to attempt the problem in Lesson 6 | Worksheet 1.	2 Students attempt to complete Lesson 6 | Worksheet 1.
3 Explain that these new programming elements are part of JavaScript.	
4 Explain that these particular programming elements are all function calls and that we know	

	they are function calls because they have an open and closed parenthesis after the name.	
5	Walk through the solution to the worksheet on the board. Highlight the importance of including the parenthesis.	
6	Ask students: What would happen if I did not include the parenthesis after the `up`? Answer: nothing, the computer would just skip that line.	

DANCE DANCE

Length: 25 minutes

Teacher performs dance based on dance program that only she/he can see. Then students try to figure out the code for the dance by playing Coders and Bots. If there is extra time, teams can write their own programs for dances.

Prep: Hand out Lesson 6 | Worksheet 2.

Teacher Actions	Student Actions
1 Explain how Dance Bot JS works (see activity directions below).	
2 Explain that you will be performing a dance and that the students will need to figure out what code programmed your dance. Luckily students will have their own dance bots to test the code on, because they will be playing Coders and Bots. All the roles will be the same for Coders and Bots as it was for the Pixel Bot exercise except that the stepper will be dancing.	
3	

	Explain that every time the Bots switch tables, she/he will perform the dance again.	
4	Ask students to pick their roles.	
5	Perform your dance.	
6	Let Coder and Bots progress, trying to figure out the code that guided the dance, until teams have produced the correct program.	6 Students start playing Coders and Bots to come up with the code for the dance. They can write the code on a blank Lesson 6 | Worksheet 2.
7	Individual Work: Tell students that they will now be creating their own dance program (5 minutes). Ask each student to write his/her own dance program.	7 Students individually write the code for their own dance using a blank Lesson 6 | Worksheet 2.
8	Tell the readers and writers at each table to perform their partner's (steppers & navigators) code.	8 Readers and writers dance their partner's code.
9		9

Tell the steppers and navigators at each table to perform their partner's code.	Steppers and navigators dance their partner's code.

DANCE BOT JS

Dance bot is a game in which a dance bot is programmed to perform a dance. The movement of the dance bot is very similar to players in dance dance revolution. Dance bot is a fun way for students to continue to explore the importance of sequencing and shows students how code and art can intersect.

ELEMENTS

- up() - move either foot up and return it
- left() - move left foot left and return it
- right() - move right foot right and return it
- down() - move either foot down and return it
- spinLeft() - spin a quarter turn left
- spinRight() - spin a quarter turn right
- wait() - don't do anything for a beat

RULES

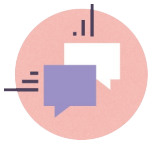
All commands should be executed on a beat. It is the job of the reader of the program to keep the beat.

The dance bot operates in a 9x9 grid (real or imaginary). It starts each move from the center of the grid. This means the dance bot should never move outside of the 9x9 grid - the dance is essentially performed in place.

EXAMPLE DANCE

```
up()
up()
left()
right()
up()
up()
left()
right()
spintLeft()
up()
up()
right()
left()
up()
up()
right()
left()
spinLeft()
back()
back()
left()
wait()
back()
back()
right()
wait()
spinLeft()
spinLeft()
left()
right()
```

Pixel bot js
Plugged



OVERVIEW

Students learn about how to type the special symbols required for JS and use Javascript on pixelbot.io to solve problems.

OBJECTIVES

1. Students will learn to identify syntax errors
2. Students will learn how to input special characters on the computer
3. Students will continue to develop proficiency in writing and reading code.

AGENDA

Length: 45 minutes

1. Unplugged Warm-up (5 minutes)
2. Start Coding
3. Pair Idea Exchange
4. Continue Coding

VOCAB

syntax error - An error from typing the JS language incorrectly

MATERIALS

1. Lesson 7 | Worksheet 1
2. Lesson 7 | Exit Ticket Worksheet
3. Scratch paper grids
4. Small turtle cutout for each student
5. Magnetic turtle
6. Scratch paper grids
7. Pencils
8. Whiteboard

WARM-UP

Length: 5 minutes

Students revisit JS in an unplugged challenge.

Prep: Draw your own pixelbot challenge on the board.

Teacher Actions	Student Actions
1 Individual Work: Tell students to solve the pixelbot challenge in JavaScript.	1 Students write down their solutions on scratch paper.
2 Discuss solutions. Ask students, what would happen if I only added the open parenthesis after the `right` like `right(` ? Answer: The computer would throw a syntax error and the program would not run.	2 Students raise their hands to provide answers.

START CODING

Length: 35 minutes

Students learn how to input special characters on the keyboard and are given a series of images to recreate using pixelbot.

Prep: Distribute Lesson 7 | Worksheet 1

Teacher Actions	Student Actions
1 Instructions 1. The goal is to recreate the images on pixelbot.io 2. Ask students, "What are the special symbols you need to call a function?" Answer: parentheses 3. Show students where the parentheses are located on the computer and remind them that you need to hold down shift to use them. 4. How many elements (function calls) should be on each line? How do you go to the new line? Answer: There should be 1 element per line. 5. Show students where the return key is on the keyboard 6. Show student an example of a syntax error on pixelbot.io. Point out the	1 Students turn away from computers and face the teacher.

	red ‘x’ that appears in the editor when you add the error.	
3	When students get stuck, we suggest using the Read, Write, and Debug protocols to support students. Ask students to imagine being on the Coder team from the group activities. They should try to play the roles of the writer and navigator. Then, ask students to imagine being a Bot to understand how the computer reads the code.	

UNPLUGGED EXIT TICKET

Length: 5 minutes

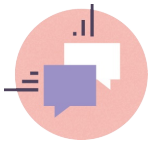
Students finish a JS worksheet to demonstrate learning

Prep: Distribute Lesson 7 | Exit Ticket Worksheet

Teacher Actions	Student Actions
1 Individual Work: Ask students to solve the exit ticket.	1 Students complete the exit ticket to the best of their abilities.

Gather Food - Winter is Coming

plugged



OVERVIEW

Students write JavaScript on getcoding.io to solve increasingly difficult challenges involving moving a squirrel to gather nuts.

OBJECTIVES

1. Students write basic JavaScript to solve simple navigation problems.

AGENDA

Length: 45 minutes

1. Warm-up
2. Help the Squirrel Gather Acorns
3. Pair Idea Exchange
4. Resume Helping the Squirrel Gather Acorns
5. Unplugged Exit Ticket

VOCAB

Code Editor - The place where coders assemble their program.

MATERIALS

1. Lesson 8 | Warm-up Worksheet

2. Lesson 8 | Exit Ticket
3. Laptops/Computers
4. Scratch paper grids
5. Small turtle cutout for each student
6. Magnetic turtle
7. Scratch paper grids
8. Pencils
9. Whiteboard

WARM-UP

Length: 10 minutes

Students practice writing basic javascript to create a simple Pixel Bot drawing.

Prep: Hand out Lesson 8 | Warm-up Worksheet

Teacher Actions	Student Actions
1 Individual Work: Ask students to write the code to produce the Pixel Bot image in the Lesson 8 | Warm-up Worksheet. Consider reminding students of the proper JavaScript syntax (see Elements on the worksheet).	1 Students individually fill out the Warm-up Worksheet.
2 Draw the Pixel Bot image on the whiteboard and code the solution with the students, randomly calling on one student at a time to provide each next line of code. (Note the problem can be solved in different ways. Students should follow the class' ongoing code which may differ from their own solution).	2 If called on, students call out the next line of code.

HELP THE SQUIRREL GATHER ACORNS

Length: 15 minutes

Students write code in JavaScript on the getcoding.io platform to help the squirrel gather acorns. Students are practicing simple sequences.

Prep: Have the getcoding.io platform open on your browser and projected on the wall. Students should also have their own computers.

Teacher Actions	Student Actions
1 Walk students through the getcoding.io platform. Show students how to: • open activities • use the Code Editor (where to type) • see elements • run code • step through the code one line at a time • change the speed	
2 Ask students to browse to getcoding.io and start moving through the challenges in the Calling Functions squirrel activity. Explain the goal of the activity: Move the squirrel to the nut and pick it up. Tell students	2 Students start solving the challenges in the Calling Functions squirrel activities.

	they are free to continue on to the second squirrel challenge when they are ready.	
3	When students get stuck, we suggest using the Read, Write, and Debug protocols to support students. Ask students to imagine being on the Coder team from the group activities. They should try to play the roles of the writer and navigator. Then, ask students to imagine being a Bot to understand how the computer reads the code.	

PAIR IDEA EXCHANGE

Length: 5 minutes

Talk with a peer about how the code is working.

Prep: None

Teacher Actions	Student Actions
1 Ask students to pause their progress and talk with a neighbor about their current problem. What is their plan? What have they tried? Is there anything standing in their way? Important: Ask students to offer questions instead of solutions.	1 Students pause their progress and talk with a neighbor about the problem they are currently trying to solve.

RESUME HELPING THE SQUIRREL GATHER ACORNS

Length: 10 minutes

Students continue writing code in JavaScript on the getcoding.io platform to help the squirrel gather acorns.

Prep: None

Teacher Actions	Student Actions
1 Ask students to resume coding individually in the Calling Functions squirrel activity.	1 Students resume coding.

UNPLUGGED EXIT TICKET

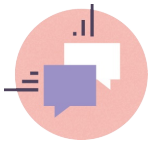
Length: 5 minutes

Students individually fill out an exit ticket focused on simple sequence.

Prep: Hand out Lesson 8 | Exit Ticket.

Teacher Actions	Student Actions
1 Individual Work: Ask students to fill out the Exit Ticket. Draw their attention to the Elements on the Exit Ticket worksheet because they differ ever so slightly from the squirrel elements.	1 Students fill out the exit ticket.

Space Ranger and Magic Words
Plugged



OVERVIEW

Students write JavaScript online in a series of increasingly difficult challenges involving maneuvering a squirrel to gather nuts.

OBJECTIVES

1. Students will become proficient at assembling JavaScript commands in sequence.

AGENDA

Length: 45 minutes

1. Warm-up
2. Help the Space Ranger
3. Pair Idea Exchange
4. Continue Coding

MATERIALS

1. Lesson 9 | Warm-up Worksheet
2. Laptops/Computers
3. Scratch paper grids
4. Small turtle cutout for each student
5. Magnetic turtle
6. Scratch paper grids
7. Pencils

8. Whiteboard

UNPLUGGED WARM-UP

Length: 10 minutes

Practice writing basic JavaScript to maneuver the squirrel to the nut.

Prep: Hand out the Lesson 9 | Warm-up Worksheet.

Teacher Actions	Student Actions
1 Individual Work: Ask students to write the code to maneuver the squirrel to the nut in the Lesson 9 | Warm-up Worksheet. Consider reminding students of the proper JavaScript syntax (see Elements on the worksheet).	1 Students individually fill out the Warm-up worksheet.
2 Draw the squirrel, nut, and grid on the whiteboard and code the solution with the students, randomly calling on one student at a time to provide each next line of code.	2 If called on, students provide the next line of code.

HELP THE SPACE RANGER

Length: 15 minutes

Students write code in JavaScript on the getCoding.io platform to help the Space Ranger gather parts. Students are practicing simple sequences with a limited set of elements.

Prep: Students should have their own computers.

Teacher Actions	Student Actions
1 Ask students to browse to getcoding.io and start moving through the challenges in Space Ranger and Space Ranger 2. Explain to the students that this challenge is a bit harder because they can no longer turn right and left. They can only turn left by using the 'rotate' command.	1 Students start solving the challenges in the Space Ranger activities.
2 When students get stuck, we suggest using the Read, Write, and Debug protocols to support students. Ask students to imagine being on the Coder team from the group activities. They should try to play the roles of the writer and navigator. Then, ask students to imagine being a Bot to understand how the computer reads the code.	

PAIR IDEA EXCHANGE

Length: 5 minutes

Talk with a peer about how the code is working.

Prep: None

Teacher Actions	Student Actions
1 Ask students to pause their progress and talk with a neighbor about the problem they are currently trying to solve. What is their plan? What have they tried? Is anything standing in their way? Important: Ask students to offer questions instead of solutions.	1 Students pause their progress and talk with a neighbor about the problem they are currently trying to solve.

CONTINUE CODING

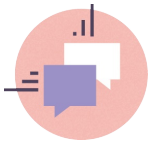
Length: 15 minutes

Students continue writing code in JavaScript on the getCoding.io platform for the two Space Ranger activities. When they finish the Space Ranger activities, students should move on to Magic Words. Talk about the rules to Magic Words: Whenever you call out a spell in your program, all of the gates with that spell will move (either up or down, depending on its last position).

Prep: None

Teacher Actions	Student Actions
1 Ask students to resume coding individually.	1 Students resume coding

Coding arguments
Unplugged



OVERVIEW

Introduce arguments by having kids do a very repetitive task.

OBJECTIVES

1. Students will be able to explain the advantage of using arguments
2. Students will be able to call functions with an argument

AGENDA

Length: 45 minutes

1. Warm-up - Large pixel bot grid
2. Arguments Analogies - Explore arguments using golf swing and drill bit analogies.
3. Pixel bot challenge - Solve pixel bot challenges with arguments

VOCAB

Argument - Specific value supplied to a function call

MATERIALS

1. Lesson 10 | Warm-up worksheet
2. Lesson 10 | Worksheet 1
3. Lesson 10 | Worksheet 2

4. Laptops/Computers
5. Scratch paper grids
6. Small turtle cutout for each student
7. Magnetic turtle
8. Scratch paper grids
9. Pencils
10. Whiteboard

WARM-UP

Length: 10 minutes

Students solve a puzzle in a large pixelbot grid.

Prep:

- Draw the Pixel Bot image from Lesson 10 | Warm-up worksheet on the whiteboard
- Distribute Lesson 10 | Warm-up worksheet

Teacher Actions	Student Actions
1 Individual work: Ask students to write code to create the image from Lesson 10 | Warm-up worksheet. Remind students that this exercise is using the icon language that they learned in Lesson 1.	1 Students individually fill out the Lesson 10 | Warm-up worksheet
2 Randomly call on one student at a time to provide each next line of code.	2 If called on, students provide the next line of code.
3 Discuss what made this particular picture difficult or frustrating to code? `Possible answer: It required a lot of code because of the size of the grid.`	3 Students raise their hands to provide an answer.

GOLF SWING AND DRILL BITS

Length: 20 minutes

Explore golf swing and drill bit analogies that help students arrive at concept of parameters/arguments.

Prep: Distribute Lesson 10 | Worksheet 1

Teacher Actions	Student Actions
1 Model a golf swing for students. Show how the same golf swing is used for different clubs. Show the Lesson 10 | Worksheet 1 golf diagram on the board and walk students through it.	
2 Model using a drill for students. Show how the same drill motion is used for different drill bits. Show the Lesson 10 | Worksheet 1 drill bit diagram on the board and walk students through it.	
3 Point students' attention to Lesson 10 | Worksheet 1. Ask students to find what is similar	3 Students look at Lesson 10 | Worksheet 1

about the two situations depicted.	
4 Individual work: Ask students to write down an answer.	4 Students individually write down their answers on the worksheet.
5 With a partner, students discuss their findings.	5 Students get with a partner and discuss their answers.
6 As a whole class, discuss the similarities between the two situations. Answer: The process is exactly the same (the golf swing never changes; the drill and the drill motion never change), but we can customize the output by changing the inputs (golf club, drill bit).	6 Students raise their hands to offer answers.

ARGUMENTS

Length: 5 minutes

Explain how to use arguments through observation.

Prep: None

Teacher Actions	Student Actions
1 Point students back to the problem on the whiteboard from Lesson 1 | Warm-up worksheet.	
2 Tell students that the elements can use an argument. An argument is extra information to customize the output of a function. The argument goes into the space to the right of the element `Example: → 5`	
3 Ask students what they think the argument will do in the case of the arrows?	3 Students raise their hand to provide an answer.

Answer: The number controls how many spaces to move in that direction.	
6 Ask students how adding a number next to the arrow icon relates to changing clubs in the golf swing? Answer: In both examples, the action is the same (swing the golf club, paint the square) but the input can be changed to customize the output.	6 Students raise their hand to provide an answer.
7 Solve the warm up problem while narrating the steps out loud.	7 Students observe as the teacher demonstrates how to solve the problem using arguments.

CODING WITH ARGUMENTS

Length: 10 minutes

Students use arguments to solve coding challenges.

Prep:

- Distribute Lesson 10 | Worksheet 2

Teacher Actions	Student Actions
1 Individual work: Students work on the problems on Lesson 10 | Worksheet 2. Remind students to use arguments to solve the problems more efficiently.	1 Students individually fill in their worksheet.

Coder Level 3

Students are introduced to core coding concepts: sequences, calling functions, passing arguments, creating loops, and using conditionals. All of the work in this unit is written in fully typed JavaScript. This course includes lesson plans: unplugged and online, worksheets, and a custom-designed coding environment.

Lessons

- sequencing pixels JS
- write some code
- write read repeat
- pixel bot online
- winter is coming - gather food
- space ranger and magic words
- coding arguments in JS

- practicing arguments
- fire ice and squirrels

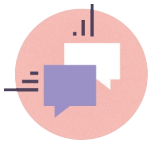
Download lesson plans

Download

Workbook

Download

Sequencing Pixels
Unplugged



OVERVIEW

Students program Pixel Bots to paint, focusing on sequence.

OBJECTIVES

- Students will learn that computers run code in a sequence.
- Students will learn how to read, write, and execute code in a sequence.

AGENDA

Length: 45 minutes

1. Welcome to coding (10 minutes)
2. Predict pixel bot JS (15 minutes)
3. Explain sequence (10 minutes)
4. Read pixel bot sequence (10 minutes)

VOCAB

- Sequence - The idea that statements must be performed in the order they are written.
- Function call - A programming element that tells the computer to do something. In the beginning, most function calls will cause the computer to perform an action.

MATERIALS

1. Lesson 1 | Warm-up Worksheet
2. Worksheet 1: Page 1 & Page 2
3. Small pixel bot cutout for each student
4. Magnetic pixel bot
5. Scratch paper grids
6. Pencils
7. Whiteboard

WELCOME TO CODING

Length: 10 minutes

Introduce students to the world of coding and get them excited about its endless possibilities.

Prep: Queue up video http://tinyurl.com/q966xd5

Teacher Actions	Student Actions
1 Lead a discussion about coding and what it means to be a coder. Suggested script: Starting with this class you are now coders. What do you think it means to be a coder? Where is code used in our world?	1 Students raise their hands to give responses to the questions.
2 Chart student responses on the board.	
3 Fill in additional interesting uses for code on the board, such as autonomous cars, streetlights, music, etc.	
4 Watch video: A day in the life of a software engineer.	

PREDICT PIXEL BOT JS

Length: 15 minutes

Students individually predict the outcome of sequences and then regroup to discuss findings.

Prep: Distribute Lesson 1 | Warm-up Worksheet

Teacher Actions	Student Actions
1 Tell students: Before we can write code, we need to learn how to read code	
2 Discuss the elements at the top of Lesson 1 | Warm-up Worksheet and ask students to speculate about what they mean. Answer: • up() - move up one square • down() - move down one square • right() - move to the right one square • left() - move to the left one square • paint() - paint the square that the pixel bot is on top of	2 Students raise their hands to give answers.

3 Individual Work: Tell students to read the elements on the worksheet and paint (color in) the correct square. While students are working on the worksheet, recreate the problems on the board.	3 Students work individually on their worksheet.
4 After they are finished, discuss the answers and how the students got to those answers. What is the difference between the two problems? Does the order of the elements matter?	4 Students raise their hands to give answers.
5 Students write in what each element means on their worksheets.	5 Students write in what each element means on their worksheet.

EXPLAIN SEQUENCE

Length: 10 minutes

Demonstrate how to read code by reading and stepping through three or four example programs.

Prep:

1. Draw a blank 3x3 grid on the whiteboard
2. Write a short (3 line) program on the whiteboard

Teacher Actions	Student Actions
1 Explain that when a computer executes code, it runs it in the order that it is written. This is called sequence.	
2 Explain that these programming elements are part of JavaScript. These particular programming elements are all function calls and that we know they are function calls because they have an open and closed parenthesis after the name.	
3	3 Students raise their hands to answer questions.

Point to the program on the whiteboard and ask students, "What is the first line of code?" After they answer, put a number 1 next to the corresponding line. Move the pixel bot according to the line of code just numbered.	
4 Continue reading and stepping one line at a time. Trace the path of the pixel bot as it moves and shade in the squares whenever it paints.	
5 Show students three new examples (design these problems on the fly, making them interesting and complex enough), reading and stepping together as a class.	5 Students follow along and offer answer for what each action does.

READ PIXEL BOT ICONS

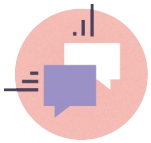
Length: 5 minutes

Students individually practice reading code.

Prep: Distribute Worksheet 1: Page 1 & Page 2

Teacher Actions	Student Actions
1 Individual Work: Leave the worked example from the previous activity on the whiteboard. Ask students to individually fill out the worksheet. Remind students to trace the path of the pixel bot and to shade in squares whenever the pixel bot paints.	1 Students read the code, trace the pathway of the pixel bot, and paint the correct blocks on the worksheet.

Write Some Code
Unplugged



OVERVIEW

Students learn the different components of reading and writing code by exploring the roles (writer, reader, navigator, and stepper) of Coders & Bots.

OBJECTIVES

- Students will be able to read basic code.
- Students will be able to write basic code.
- Together as a class, students will be able to enact the Coders & Bots roles of Writer, Navigator, Reader, and Stepper to write and read code.

AGENDA

Length: 45 mintues

1. Pixel Bots: Practice Reading Code
2. Pixel Bots: Write Code
3. Pixel Bots: Write Code Together

VOCAB

- Computer - A device that can be instructed to do something.
- Program - A list of statements that a computer can perform.

MATERIALS

1. Lesson 2 | Warm-up Worksheet
2. Lesson 2 | Worksheet 1
3. Small pixel bot cutout for each student
4. Magnetic pixel bot
5. Scratch paper grids
6. Pencils
7. Whiteboard

PIXEL BOTS: PRACTICE READING CODE

Length: 15 minutes

Students practice reading basic code sequences.

Prep: Draw on the whiteboard the grid and the lines of code from the example problem on the front page of Lesson 2 | Warm-up Worksheet.

Teacher Actions	Student Actions
1 Remind students of the value of revisiting ideas explored in previous lessons. Revisiting past ideas helps to see them in a new light and makes sure they are not forgotten.	
2 Step through the example problem as a whole class, talking through the process of reading code.	
3 Hand out Lesson 2 | Warm-up Worksheet.	3 Students place their pixel bot at the starting square. Students read the code and move their pixel bot. Students do this a few times to get into a rhythm.

4 Individual Work: Ask students to place the movable pixel bot at the start block of the example problem just demonstrated on the board. Students should work individually, reading the code and moving their pixel bot along. Ask the students to get into a rhythm of reading and stepping.	
5 Individual Work: Ask students to flip the page and perform the same exercise with the new problem. Ask students to shade in the appropriate squares and trace the pathway of the pixel bot.	5 Students turn the page and attempt to solve the problem.
6 On the board, draw the empty grid and the code from the problem students have been working on. Read the code and step the pixel bot (tracing its pathway and shading in squares), narrating your thought process.	

PIXEL BOTS: WRITE CODE

Length: 10 minutes

Students write code to produce a simple pixel bot image.

Prep: Hand out Lesson 2 | Worksheet 1.

Teacher Actions	Student Actions
1 Talk to students about how they have already learned to read code that other people have written. Now they are going to explore writing code.	
2 Individual Work: Have students place their pixel bot at the start square. Ask students to write code that produces the provided pixel bot image. The students should think through their plan for their code, write their code, and move their pixel bot along.	2 Students place their pixel bot at the starting square, write code that creates the picture, and enact the pixel bot actions each step of the way.
3 As a whole class, solicit ideas from multiple students about how they designed their code to solve the problem. Hold off on presenting the correct answer until the next activity.	3 Students share their code, especially if they have solved the problem a different way.

PIXEL BOTS: WRITE CODE TOGETHER

Length: 20 minutes

Teacher shows student how to write code, emphasizing roles of the Writer, Navigator, Reader, and Stepper (see Coders and Bots Protocol). The students then enact these roles together as a whole class solving a new problem.

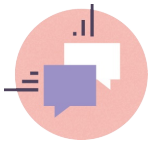
Prep: Draw the problem from Lesson 2 | Worksheet 1 on the whiteboard.

Teacher Actions	Student Actions
1 Code the solution to Lesson 2 | Worksheet 1 on the whiteboard. Voice your code writing process along the way: write and number a new line of code and only then move the pixel bot on the whiteboard. Use this as an opportunity to discuss how there are different ways to solve the same problem. After the code is written, introduce the idea that the computer reads code in sequence.	
2 After coding the solution, explain that you are switching gears into Bot mode. Read each line and step the pixel bot, checking to see if you coded the correct solution. Add a new problem on the board: a	2 Students take turns walking up to the board to write and navigate code.

checkerboard pattern. Divide the class in half and follow the process spelled out in the whole class section of the Coders and Bots Protocol. Students start as Coders. Assign half of the students to be writers and half to be navigators; one person from each team walks up to collaboratively add and enact a line of code. The two students then tag in a member of their team to walk up and write the next line of code. Allow the class to code a wrong solution.	
3 The whole class then switches roles to become Bots. Use this opportunity to emphasize the definition of a computer (see vocabulary section above). Following the same procedure as above, ask students to come up to the board to take turns reading and stepping through one line of code at a time before tagging in a new classmate.	3 Students take turns walking up to the board to read and step through the solution.
4 Have students switch back and forth between Coders and Bots until they code the solution on the checkerboard.	4 Students switch back and forth between Coders and Bots to complete the puzzle.

5 Summarize any conceptual difficulties. Introduce and define a Program (see vocabulary section above), and explain how it connects to the code the students just wrote.	
6 Ask students to write down on a piece of paper what each of the the four roles does, focusing on one role at a time. After each role definition, ask students to share their definition with the whole class. Pool the students' ideas into overall definitions of the roles.	6 Students provide descriptions of what each role entails.

Write.Read.Repeat.
Unplugged



OVERVIEW

Students continue to solve pixel bot problems by playing Coders & Bots to read and write code.

OBJECTIVES

- Students will perform the Coder & Bots roles of Writer, Navigator, Reader, and Stepper
- Students will collaborate in small groups to write and read code
- Students will become more proficient at writing code

AGENDA

Length: 45 minutes

1. Pixel Bots: Write Code Warm-up
2. Pixel Bots: Coders and Bots
3. Pixel Bots: Write Code Exit Ticket

VOCAB

- Programming Element - A command the computer understands.
- Action - An observable event that the code produces
- Error - The program fails to run because the computer cannot execute the code

MATERIALS

1. Lesson 3 | Warm-up Worksheet
2. Lesson 3 | Worksheet 1
3. Lesson 3 | Worksheet 2
4. Lesson 3 | Worksheet 3
5. Lesson 3 | Worksheet 4
6. Lesson 3 | Exit Ticket Worksheet
7. Scratch paper grids
8. Small pixel bot cutout for each student
9. Magnetic pixel bot
10. Scratch paper grids
11. Pencils
12. Whiteboard

CLASSROOM SETUP

Pods of four.

PIXEL BOTS: WRITE CODE WARM-UP

Length: 15 minutes

Students write code for a simple pixel bot image.

Prep: Hand out Lesson 3 | Warm-up.

Teacher Actions	Student Actions
1 Individual Work: Ask students to solve the problem on Lesson 3 | Warm-up Worksheet.	1 Students individually solve the problem on the worksheet.
2 As students attempt the problem, draw the problem on the whiteboard.	
3 Code the solution to the problem as a whole class. Call on students at random to provide each next line of code. Each student should walk up to the board and write the next line of code. With each new line of code, you should play the role of Navigator, moving the turtle. Call out the roles of Writer and Navigator to make them explicit. On occasion, pick a Reader and Stepper to walk up to the board to test the code starting from line one.	3 If called on, students walk up to the board to write in (or read) the next line of code.

PIXEL BOTS: CODERS AND BOTS

Length: 30 minutes

Students work in groups to write and read code to produce two pixel bot images.

Prep: Consider having paper bags for each group. Each paper bags contains the four roles (Writer, Navigator, Reader, Stepper).

Teacher Actions	Student Actions
1 Teacher breaks students into groups of four and randomly assigns roles for the Programming Team and the Computer Team. (Consider handing each group a paper bag with the four roles inside – students reach into the bag and grab a role.) In this first round, groups either get Lesson 3 | Worksheet 1 or Lesson 3 | Worksheet 2 (alternate between groups). Bots should always help the Coders during the code writing phase of the activity.	1 Students enact their Coders and Bots roles. They write code to create the pixel bot image.
2 Follow the Coders and Bots Protocol by having the Bot Team switch to a new group when it comes time to check the code. Make sure that the Coders fold back their paper to hide the	2 The Bots switch to a new group to assess code. If the Bots find an error, report it to the Coders and tell them what happened and the line number the error happened on.

	provided pixel bot image before the Bots arrive (ensuring that the Bots are not biased in their reading). The Bots should test the code on an empty scratch paper grid. Ask students, "What do you think the computer does when it cannot understand the code or the code forces the turtle to break the rules (go outside the grid)?" Answer: An error! If the Bots notice this happening, they should report the error to the programming team and tell them what line the error happened on.		
3	After one round, pass out Lesson 3 | Worksheet 3 or Lesson 3 | Worksheet 4 to the groups (alternate again) and repeat the Coders and Bots Protocol.	3	Students repeat the above process with new coding challenges.
4	In whole class mode, ask students to share out any disagreements they had in the write and read process. Use this as an occasion to firm up any misconceptions. Also use this time to introduce and define Elements and Actions (see vocabulary section above).	4	Students summarize and describe the disagreements they could not resolve.

PIXEL BOTS: WRITE CODE EXIT TICKET

Length: 5 minutes

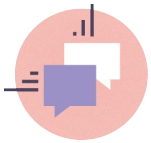
Time permitting, students individually fill out an exit ticket to check for understanding.

Prep: Hand out Lesson 3 | Exit Ticket Worksheet

Teacher Actions	Student Actions
1 Ask students to individually work on completing the Lesson 3 | Exit Ticket Worksheet.	1 Students work individually on the Lesson 3 | Exit Ticket Worksheet.

Pixel Bots Online

Plugged



OVERVIEW

Students review writing code offline. Then, students demonstrate their learning by writing programs to complete online pixel bot challenges.

OBJECTIVES

- Students will write write programs using JavaScript.
- Students will continue to develop proficiency in writing and reading code.

AGENDA

Length: 45 mintues

1. Pixel Bots: Warm-up
2. Introduce Pixel Bots Online
3. Pixel Bots Online Practice
4. Pixel Bots: Exit Ticket

VOCAB

- Program - A list of statements that a computer can perform.

MATERIALS

1. Lesson 4 | Worksheet 1

2. Lesson 4 | Exit Ticket Worksheet
3. Scratch paper grids
4. Small pixel bot cutout for each student
5. Magnetic pixel bot
6. Scratch paper grids
7. Pencils
8. Whiteboard

PIXEL BOTS: WARM UP

Length: 10 minutes

Create a medium difficulty pixel bot image for the students on the whiteboard. Students solve the problem individually and then check the work of a peer.

Prep: Create a medium difficulty pixel bot image on the whiteboard.

Teacher Actions	Student Actions
1 Individual Work: Ask students to get a blank piece of paper, write line numbers, and write the code that creates the pixel bot image on the board.	1 Students work individually on coding a solution to the challenge.
2 Students discuss the coding solution with a peer.	2 When they finish, students check their work with a partner after both have finished.

INTRODUCE PIXEL BOT ONLINE

Length: 10 minutes

Show students how to solve a problem on pixel bot online. Also revisit `program` as a vocabulary word.

Prep: Browse to www.pixelbots.io and project onto the wall.

Teacher Actions	Student Actions
1 Now that students have practice creating code sequences on paper, they are ready to start writing programs.	1 Add student actions here and match numbers to teacher actions
2 Revisit the idea that a program is a sequence that a computer is able to understand.	
3 On a projector, show students www.pixelbots.io	
4 Use the problem you created in the warm-up to demonstrate how to solve the problem in the	

online interface. Show students a variety of features in the pixel bot interface: • How to add code • How to delete code • How to run program • How to reset after run • How to insert code	
5 Ask students, "What do you think we should do if we get stuck on a problem?"	5 Answer: Step through code starting at the beginning like you do as a Bot.

PIXEL BOT ONLINE PRACTICE

Length: 20 minutes

Students are given a set of pixel bot images to reproduce on the computer. Partners star the ones that are completed. Remind students that the images with lots of shaded squares will be difficult, but they should remember to read code from the beginning when things go wrong.

Prep: Distribute the Lesson 4 | Worksheet 1 worksheet and write www.pixelbots.io up on the board.

Teacher Actions	Student Actions
1 Explain the exercise: • The goal is to recreate each of the images from the worksheet by creating a program on the website. • Tell students that they will act as testers for the person sitting next to them: ◦ When they finish writing a program, students have their partner check to make sure the images match up. If they are a match, the tester puts a checkmark next to the image. ◦ The programmer should then explain how their code works to the	1 Students are faced away from their computers toward the teacher.

tester. If the tester is satisfied with the explanation, the tester checks the explain box. ◦ The programmer can now continue on to the next challenge.	
2 Students work on completing the challenges.	2 Students get on their computers and go to www.pixelbots.io. Students work individually on recreating each of the images with code. • When they finish a puzzle, students have their partner (student sitting next to them) check to make sure the images match up. If they are a match, the checker puts a checkmark next to the image and the programmer can continue on to the next challenge.
3 When students get stuck, ask them to imagine being on the Coder team from the group activities. They should try to play the roles of the writer and navigator. Then, ask students to imagine being a Bot to understand how the computer is reading the code.	3 Students raise their hands to provide answers.

4 Discuss: Which was the hardest coding challenge? Why? Was there more than one way to solve the problem?

PIXEL BOTS: EXIT TICKET

Length: 5 minutes

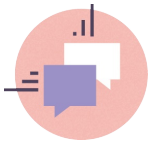
Students complete Exit Ticket.

Prep: Distribute Lesson 4 | Exit Ticket Worksheet.

Teacher Actions	Student Actions
1 Individual Work: Tell students they are going to work individually on coding the answer on the worksheet.	1 Students write their answers on the worksheet.

Gather Food - Winter is Coming

plugged



OVERVIEW

Students write JavaScript on getcoding.io to solve increasingly difficult challenges involving moving a squirrel to gather nuts.

OBJECTIVES

1. Students write basic JavaScript to solve simple navigation problems.

AGENDA

Length: 45 minutes

1. Warm-up
2. Help the Squirrel Gather Acorns
3. Pair Idea Exchange
4. Resume Helping the Squirrel Gather Acorns
5. Unplugged Exit Ticket

VOCAB

Code Editor - The place where coders assemble their program.

MATERIALS

1. Lesson 5 | Warm-up Worksheet

2. Lesson 5 | Exit Ticket
3. Laptops/Computers
4. Scratch paper grids
5. Small turtle cutout for each student
6. Magnetic turtle
7. Scratch paper grids
8. Pencils
9. Whiteboard

WARM-UP

Length: 10 minutes

Students practice writing basic javascript to create a simple Pixel Bot drawing.

Prep: Hand out Lesson 5 | Warm-up Worksheet

Teacher Actions	Student Actions
1 Individual Work: Ask students to write the code to produce the Pixel Bot image in the Lesson 5 | Warm-up Worksheet. Consider reminding students of the proper JavaScript syntax (see Elements on the worksheet).	1 Students individually fill out the Warm-up Worksheet.
2 Draw the Pixel Bot image on the whiteboard and code the solution with the students, randomly calling on one student at a time to provide each next line of code. (Note the problem can be solved in different ways. Students should follow the class' ongoing code which may differ from their own solution).	2 If called on, students call out the next line of code.

HELP THE SQUIRREL GATHER ACORNS

Length: 15 minutes

Students write code in JavaScript on the getcoding.io platform to help the squirrel gather acorns. Students are practicing simple sequences.

Prep: Have the getcoding.io platform open on your browser and projected on the wall. Students should also have their own computers.

Teacher Actions	Student Actions
1 Walk students through the getcoding.io platform. Show students how to: • open activities • use the Code Editor (where to type) • see elements • run code • step through the code one line at a time • change the speed	
2 Ask students to browse to getcoding.io and start moving through the challenges in the Calling Functions squirrel activity. Explain the goal of the activity: Move the squirrel to the nut and pick it up. Tell students	2 Students start solving the challenges in the Calling Functions squirrel activities.

	they are free to continue on to the second squirrel challenge when they are ready.	
3	When students get stuck, we suggest using the Read, Write, and Debug protocols to support students. Ask students to imagine being on the Coder team from the group activities. They should try to play the roles of the writer and navigator. Then, ask students to imagine being a Bot to understand how the computer reads the code.	

PAIR IDEA EXCHANGE

Length: 5 minutes

Talk with a peer about how the code is working.

Prep: None

Teacher Actions	Student Actions
1 Ask students to pause their progress and talk with a neighbor about their current problem. What is their plan? What have they tried? Is there anything standing in their way? Important: Ask students to offer questions instead of solutions.	1 Students pause their progress and talk with a neighbor about the problem they are currently trying to solve.

RESUME HELPING THE SQUIRREL GATHER ACORNS

Length: 10 minutes

Students continue writing code in JavaScript on the getcoding.io platform to help the squirrel gather acorns.

Prep: None

Teacher Actions	Student Actions
1 Ask students to resume coding individually in the Calling Functions squirrel activity.	1 Students resume coding.

UNPLUGGED EXIT TICKET

Length: 5 minutes

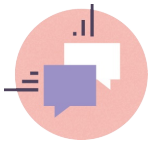
Students individually fill out an exit ticket focused on simple sequence.

Prep: Hand out Lesson 5 | Exit Ticket.

Teacher Actions	Student Actions
1 Individual Work: Ask students to fill out the Exit Ticket. Draw their attention to the Elements on the Exit Ticket worksheet because they differ ever so slightly from the squirrel elements.	1 Students fill out the exit ticket.

Space Ranger and Magic Words

Plugged



OVERVIEW

Students write JavaScript online in a series of increasingly difficult challenges involving maneuvering a squirrel to gather nuts.

OBJECTIVES

1. Students will become proficient at assembling JavaScript commands in sequence.

AGENDA

Length: 45 minutes

1. Warm-up
2. Help the Space Ranger
3. Pair Idea Exchange
4. Continue Coding

MATERIALS

1. Lesson 6 | Warm-up Worksheet
2. Laptops/Computers
3. Scratch paper grids
4. Small turtle cutout for each student
5. Magnetic turtle
6. Scratch paper grids
7. Pencils

8. Whiteboard

UNPLUGGED WARM-UP

Length: 10 minutes

Practice writing basic JavaScript to maneuver the squirrel to the nut.

Prep: Hand out the Lesson 6 | Warm-up Worksheet.

Teacher Actions	Student Actions
1 Individual Work: Ask students to write the code to maneuver the squirrel to the nut in the Lesson 6 | Warm-up Worksheet. Consider reminding students of the proper JavaScript syntax (see Elements on the worksheet).	1 Students individually fill out the Warm-up worksheet.
2 Draw the squirrel, nut, and grid on the whiteboard and code the solution with the students, randomly calling on one student at a time to provide each next line of code.	2 If called on, students provide the next line of code.

HELP THE SPACE RANGER

Length: 15 minutes

Students write code in JavaScript on the getcoding.io platform to help the Space Ranger gather parts. Students are practicing simple sequences with a limited set of elements.

Prep: Students should have their own computers.

Teacher Actions	Student Actions
1 Ask students to browse to getcoding.io and start moving through the challenges in Space Ranger and Space Ranger 2. Explain to the students that this challenge is a bit harder because they can no longer turn right and left. They can only turn left by using the 'rotate' command.	1 Students start solving the challenges in the Space Ranger activities.
2 When students get stuck, we suggest using the Read, Write, and Debug protocols to support students. Ask students to imagine being on the Coder team from the group activities. They should try to play the roles of the writer and navigator. Then, ask students to imagine being a Bot to understand how the computer reads the code.	

PAIR IDEA EXCHANGE

Length: 5 minutes

Talk with a peer about how the code is working.

Prep: None

Teacher Actions	Student Actions
1 Ask students to pause their progress and talk with a neighbor about the problem they are currently trying to solve. What is their plan? What have they tried? Is anything standing in their way? Important: Ask students to offer questions instead of solutions.	1 Students pause their progress and talk with a neighbor about the problem they are currently trying to solve.

CONTINUE CODING

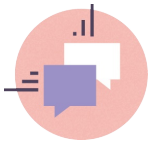
Length: 15 minutes

Students continue writing code in JavaScript on the getCoding.io platform for the two Space Ranger activities. When they finish the Space Ranger activities, students should move on to Magic Words. Talk about the rules to Magic Words: Whenever you call out a spell in your program, all of the gates with that spell will move (either up or down, depending on its last position).

Prep: None

Teacher Actions	Student Actions
1 Ask students to resume coding individually.	1 Students resume coding

Coding arguments
Unplugged



OVERVIEW

Introduce arguments by having kids do a repetitive task.

OBJECTIVES

1. Students will be able to explain the advantage of using arguments
2. Students will be able to call functions with an argument

AGENDA

Length: 45 minutes

1. Warm-up - Large pixel bot grid
2. Arguments Analogies - Explore arguments using golf swing and drill bit analogies.
3. Pixel bot challenge - Solve pixel bot challenges with arguments

VOCAB

Argument - Specific value supplied to a function call

MATERIALS

1. Lesson 7 | Warm-up worksheet
2. Lesson 7 | Worksheet 1-1
3. Lesson 7 | Worksheet 1-2

4. Lesson 7 | Worksheet 1-3
5. Lesson 7 | Worksheet 2
6. Scratch paper grids
7. Small pixel bot cutout for each student
8. Magnetic pixel bot
9. Scratch paper grids
10. Pencils
11. Whiteboard

WARM-UP

Length: 10 minutes

Students solve a puzzle in a large pixelbot grid.

Prep:

- Draw the Pixel Bot image from Lesson 7 | Warm-up worksheet on the whiteboard
- Distribute Lesson 7 | Warm-up worksheet

Teacher Actions	Student Actions
1 Individual work: Ask students to write code to create the image from Lesson 7 | Warm-up worksheet.	1 Students individually fill out the Lesson 7 | Warm-up worksheet
2 Randomly call on one student at a time to provide each next line of code.	2 If called on, students provide the next line of code.
3 Discuss what made this particular picture difficult or frustrating to code. `Possible answer: It required a lot of code because of the size of the grid.`	3 Students raise their hands to provide an answer.

GOLF SWING AND DRILL BITS

Length: 20 minutes

Explore golf swing and drill bit analogies that help students arrive at concept of parameters/arguments.

Prep: Distribute Lesson 7 | Worksheet 1-1

Teacher Actions	Student Actions
1 Individual work: Ask students to fill out Lesson 7 | Worksheet 1-1.	1 Students fill out Lesson 7 | Worksheet 1-1.
2 As a whole class, pool students' ideas.	2 Students share ideas.
3 Individual work: Hand out Lesson 7 | Worksheet 1-2 and ask students to give it a try.	3 Students fill out Lesson 7 | Worksheet 1-2.
4 Discuss how to write the proper syntax for the golf and drill bit programs. Write a few examples of the syntax on the board and ask students to	4 Students predict the teacher's code.

predict how far the ball would go or how big the hole would be.	
5 Individual work: Hand out the next Lesson 7 | Worksheet 1-3 Cont'd and ask students to map these ideas over to pixel bot.	5 Students fill out next part of Lesson 7 | Worksheet 1-3.
6 Discuss students' ideas for Question 6. Answer: The process is exactly the same (the golf swing never changes; the drill and the drill motion never change), but we can customize the output by changing the inputs (golf club, drill bit).	6 Students raise their hands to provide answers

ARGUMENTS

Length: 5 minutes

Explain how to use arguments through observation.

Prep: None

Teacher Actions	Student Actions
1 Point students back to the problem on the whiteboard from Lesson 1 | Warm-up worksheet.	
2 Tell students that the elements can use an argument. An argument is extra information to customize the output of a function. The argument goes in between the parenthesis that follow the name of the function. Example: `up( 5 )`	
3 Ask students to say once again what the argument will do in the case of the movements? Answer: The number controls how many spaces to move in that direction.	3 Students raise their hand to provide an answer.

4 Add `paint('blue')`	
5 Ask students what they think the argument will do in the case of the paint icon? Answer: the argument controls what color the turtle will paint	5 Students raise their hand to provide an answer.
6 Ask students how changing the color next to the icon relates to changing clubs in the golf swing? Answer: In both examples, the action is the same (swing the golf club, paint the square) but the input can be changed to customize the output.	6 Students raise their hand to provide an answer.
7 Solve the warm up problem while narrating the steps out loud.	7 Students observe as the teacher demonstrates how to solve the problem using arguments.

CODING WITH ARGUMENTS

Length: 10 minutes

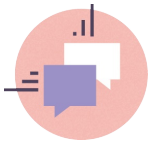
Students use arguments to solve coding challenges.

Prep:

- Distribute Lesson 7 | Worksheet 2

Teacher Actions	Student Actions
1 Individual work: Students work on the problems on Lesson 7 | Worksheet 2. Remind students to use arguments to solve the problems more efficiently.	1 Students individually fill in the problems on their worksheet.

Practicing arguments
Plugged



OVERVIEW

Students practice utilizing function calls with arguments offline in coders and bots, and then on pixelbots.io.

OBJECTIVES

1. Students will be able to use arguments to solve programming challenges.
2. Students will become proficient at assembling JavaScript commands in sequence.

AGENDA

Length: 45 minutes

1. Warm-up with large pixel bot exercise (5 minutes)
2. Coders and Bots with arguments (30 minutes)
3. Individual work (10 minutes)

MATERIALS

1. Lesson 8 | Warm-up Worksheet
2. Lesson 8 | Worksheet 1
3. Lesson 8 | Worksheet 2
4. Lesson 8 | Worksheet 3
5. Laptops/Computers
6. Scratch paper grids
7. Small turtle cutout for each student

8. Magnetic turtle
9. Scratch paper grids
10. Pencils
11. Whiteboard

ACTIVITY TITLE

Length: 5 minutes

Students warm up by completing exercise on Lesson 8 | Warm-up Worksheet

Prep: Distribute Lesson 8 | Warm-up Worksheet

Teacher Actions	Student Actions
1 Individual Work: Ask students to write the code to produce the Pixel Bot image in the Lesson 8 | Warm-up Worksheet. Consider reminding students of the proper JavaScript syntax (see Elements on the worksheet).	1 Students individually fill out the Warm-up Worksheet.
2 Draw the Pixel Bot image on the whiteboard and code the solution with the students, randomly calling on one student at a time to provide each next line of code. (Note the problem can be solved in different ways. Students should follow the class' ongoing code which may differ from their own solution).	2 If called on, students call out the next line of code.

CODERS AND BOTS

Length: 30 minutes

Students explore more complex coding problems in groups using the coder and bots protocol.

Prep: Distribute Lesson 8 | Worksheet 1

Teacher Actions	Student Actions
1 Teacher breaks students into groups of four and randomly assigns roles for the Programming Team and the Computer Team. (Consider handing each group a paper bag with the four roles inside – students reach into the bag and grab a role.) In this first round, groups either get Lesson 8 | Worksheet 1. Bots should always help the Coders during the code writing phase of the activity.	1 Students enact their Coders and Bots roles. They write code to create the pixel bot image.
2 Explain that this problem can be solved more than one way. The goal is for students to develop code that solves the problem using the FEWEST lines of code.	

3 Follow the Coders and Bots Protocol by having the Bot Team switch to a new group when it comes time to check the code. Make sure that the Coders fold back their paper to hide the provided pixel bot image before the Bots arrive (ensuring that the Bots are not biased in their reading). The Bots should test the code on an empty scratch paper grid. If the Bots notice an error, they should report the error to the programming team and tell them what line the error happened on.	3 The Bots switch to a new group to assess code. If the Bots find an error, report it to the Coders and tell them what happened and the line number the error happened on.
4 Discuss as a whole class how many lines it took to complete the image. Then, the group with the fewest lines shares writes their code on the whiteboard. In their groups, students have 2 minutes to discuss the optimal solution. How was this program able to use less lines of code?	4 Students discuss the code and develop new strategies for the next round.
5 After one round, pass out Lesson 8 | Worksheet 2 to the groups and repeat the Coders and Bots Protocol.	5 Students repeat the above process with new coding challenges.
6	6

In whole class mode, ask students to share out any disagreements they had in the write and read process. Use this as an occasion to firm up any misconceptions.	Students summarize and describe the disagreements they could not resolve.
7 What strategies did the groups come up with to write the least lines of code? Answer: The best strategy is to make sure that each move takes the pixel bot to a square that needs to be painted.	7 Students raise their hands to offer solutions.

ONLINE ARGUMENTS

Length: 10 minutes

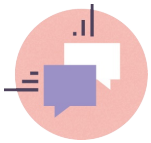
Students practice using arguments to create images on pixelbots.io

Prep: Distribute Lesson 8 | Worksheet 3

Teacher Actions	Student Actions
1 Individual work: Students work on the problems on Lesson 8 | Worksheet 3 on pixelbots.io. Remind students to use arguments to solve the problems more efficiently.	1 Students get on their computer and navigate to pixelbots.io to complete the challenges on Lesson 8 | Worksheet 3

Fire Ice and Squirrels

Plugged



OVERVIEW

Students write JavaScript online in a series of increasingly difficult challenges involving maneuvering a squirrel to gather acorns, and a wizard using magic words.

OBJECTIVES

1. Students will become proficient at assembling JavaScript commands in sequence.
2. Students will become proficient at using arguments
3. Students will be able to define a string
4. Students will be able to pass a string as an argument

AGENDA

Length: 45 minutes

1. Warm-up
2. Help the squirrel
3. Introduce strings
4. Fire and ice

VOCAB

- String - A sequence of characters or words.

MATERIALS

1. Lesson 9 | Unplugged Warm-up Worksheet
2. Laptops/Computers
3. Scratch paper grids
4. Small turtle cutout for each student
5. Magnetic turtle
6. Scratch paper grids
7. Pencils
8. Whiteboard

UNPLUGGED WARM-UP

Length: 10 minutes

Practice writing function calls with arguments to maneuver the pixel bot.

Prep: Hand out the Lesson 9 | Unplugged Warm-up Worksheet.

Teacher Actions	Student Actions
1 Individual Work: Ask students to write the code to maneuver the squirrel to the nut in the Lesson 9 | Unplugged Warm-up Worksheet. Consider reminding students of the proper JavaScript syntax (see Elements on the worksheet).	1 Students individually fill out the Warm-up worksheet.
2 Draw the example on the whiteboard and code the solution with the students, randomly calling on one student at a time to provide each next line of code.	2 If called on, students provide the next line of code.

HELP THE SQUIRREL

Length: 15 minutes

Students write code in JavaScript on the getCoding.io platform to help the squirrel gather the acorns. Students are practicing simple sequences with arguments.

Prep: Students should have their own computers.

Teacher Actions	Student Actions
1 Ask students to browse to getcoding.io and start moving through the challenges in Passing Arguments: Squirrel Climber. Explain to the students that now the squirrel movements can take one argument - the number of squares to move.	1 Students start solving the challenges in the Space Ranger activities.
2 When students get stuck, we suggest using the Read, Write, and Debug protocols to support students. Ask students to imagine being on the Coder team from the group activities. They should try to play the roles of the writer and navigator. Then, ask students to imagine being a Bot to understand how the computer reads the code.	

INTRODUCING STRINGS

Length: 5 minutes

Gather students and introduce the concept of strings.

Prep:

- Set up projector to show Passing Arguments: Magic Words on getcoding.io
- Write examples of all of the function calls that students have used with arguments on the whiteboard

```
From pixelbots.io:
up(2)
down(3)
left(2)
right(4)

From getcoding.io:
move(4)
```

Teacher Actions	Student Actions
1 Show students the function calls on the whiteboard. Ask students what pattern they notice about the arguments they have been using. Answer: All of the arguments up until this point have been numbers.	1 Students raise their hands to offer an answer.

2 Show students Passing Arguments: Magic Words. Ask students what kind of argument is needed to cast a spell in the game. Answer: A word	2 Students raise their hands to offer an answer.
3 Explain that in programming text is a different type of data called a string. To let the computer know that a value is a string, it needs to be put inside quotation marks. Ex. `cast('fire')`	
4 As a class do the first puzzle of Passing Arguments: Magic Words. Make sure to draw attention to the quotation marks that go around the string.	

MAGIC WORDS

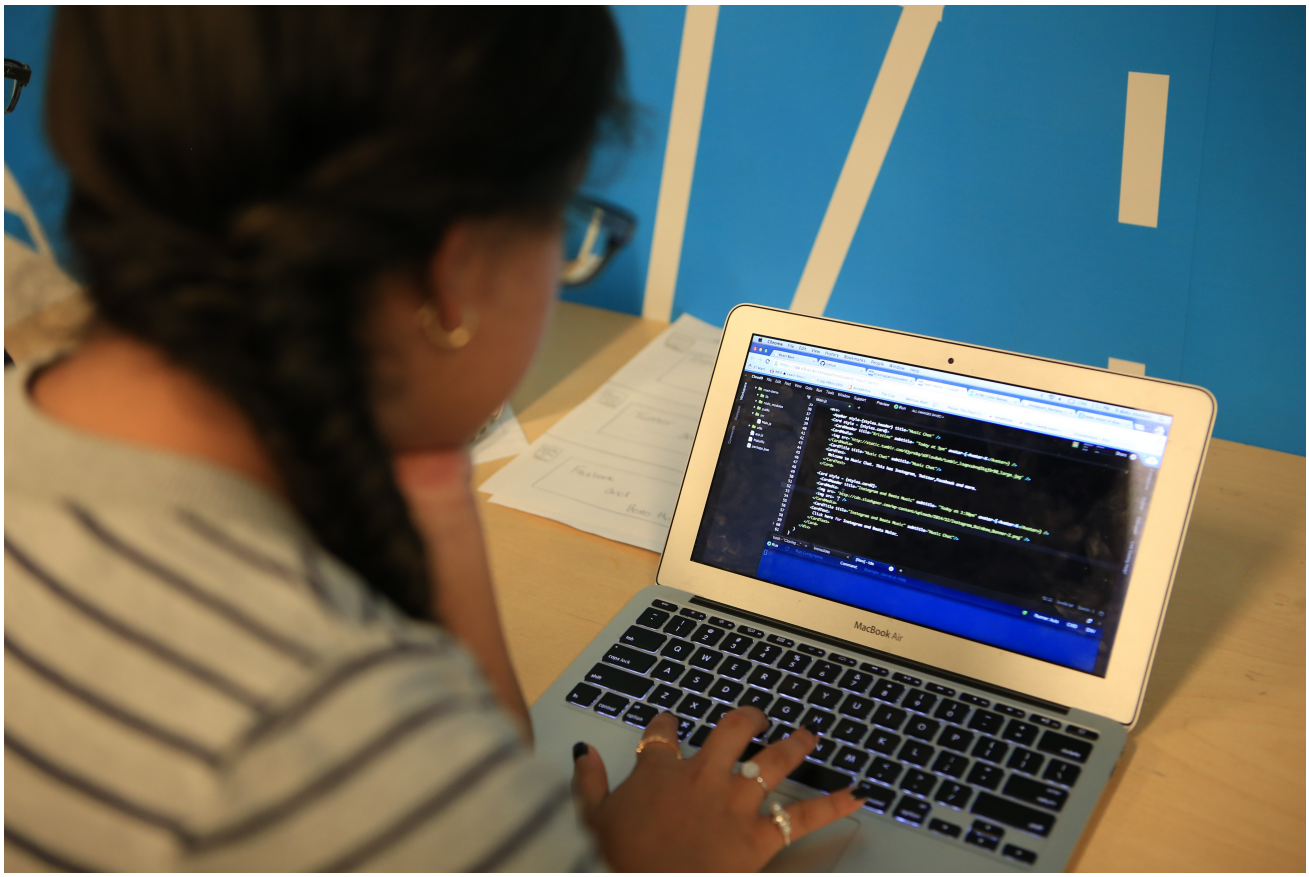
Length: 15 minutes

Students continue writing code in JavaScript on the getCoding.io platform for the Passing Arguments: Magic Words. Remind them that in this version, in addition to opening the gates, whenever there is a fire or ice wall, they need to cast a spell of the opposite element.

Prep: None

Teacher Actions	Student Actions
1 Ask students to resume coding individually.	1 Students resume coding

Developer



The developer series features project and product based curricula designed to introduce and reinforce applied learning in projects motivated by the product design cycle. The lessons in these units are couched in the context of an overarching product that the student is working on developing. Students will create games, apps, and collaborative worlds.

Curriculum

- Level 1
- Level 2 (Coming Soon!)

Overview: Escape the Maze

In this 27-lesson course, students master computational thinking skills and the fundamentals of coding to build an interactive maze game with multiple obstacles, enemies, and levels. These lessons incorporate unplugged and online activities that build connections between coding concepts and apply them to projects. Along with hard skills in coding, students learn the processes and mindsets of a computer programmer. Students learn to approach failure as an opportunity, collaborate with peers on writing and reviewing code, and tap into their own creativity without reservations. At the end of the course students apply their knowledge of the software design process to plan an unplugged event - a class-wide Arcade Day - in which they share their final maze projects and celebrate their development as a coder.

Purpose

1. Empower students to use computational practices to analyze problems, build solutions, and be creative.
2. Empower students to apply computational practices to understand and change the world.

Big Goals

1. Students will utilize computational thinking to read their world and will utilize their coding skills to write their world.
2. Students will experience failure as a learning opportunity to master core concepts and the process of creation.
3. Students will collaboratively build software and assess its effectiveness for users.

Big Ideas

1. Software is built by composing simple, common coding building blocks.
2. Reusable and robust software is readable, modular, and testable.
3. Impactful software is built iteratively and collaboratively through stages of planning, enacting, and assessing.
4. When code breaks, we fearlessly, creatively, and systematically debug it.

Essential Questions

1. How can we use coding building blocks to design something new or break something down?
2. When your code breaks or when you get stuck, what can you do?
3. How do you write code that other programmers can use?
4. How do you adapt your software to meet the needs of your users?

Developer Level 1

Students begin applying core coding concepts into an Escape the Maze project. In this level, students are introduced to the Scratch block-based language and the offline maze grid. These platforms help student experience and understand new coding concepts such as sequencing, creating loops, and using conditionals.

Lessons

- Lesson 1: I am a coder
- Lesson 2: Getting Started on Scratch
- Lesson 3: Maze Scavenger Hunt
- Lesson 4: Dance Off