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

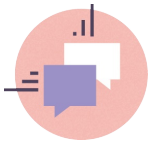
Download lesson plans

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

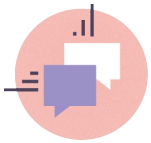
Length: 5 minutes

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.

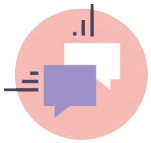
Prep: Draw the problem from Lesson 2 | Worksheet 1 on the whiteboard.

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

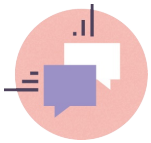
Time permitting, students individually fill out an exit ticket to check for understanding.

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

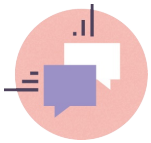
Students complete Exit Ticket.

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

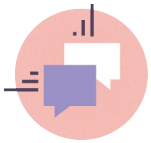
Students individually fill out an exit ticket focused on simple sequence.

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

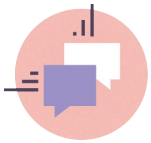
Length: 15 minutes

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

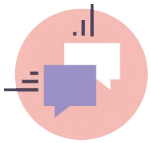
Students use arguments to solve coding challenges.

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

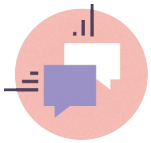
Students practice using arguments to create images on pixelbots.io

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