CodeHS

Introduction to Programming in JavaScript with Arduino
1 year for High School (150-175 contact hours)

Course Overview and Goals

The Intro to Programming in JavaScript with Arduino course merges the Introduction to Computer Science in JavaScript and the Introduction to Physical Computing with Arduino courses. The JavaScript course will provide the prerequisite information needed before students apply programming concepts to their physical Arduino device.

Learning Environment: The course utilizes a blended classroom approach. The content is fully web-based, with students writing and running code in the browser. Teachers utilize tools and resources provided by CodeHS to leverage time in the classroom and give focused 1-on-1 attention to students. Each unit of the course is broken down into lessons. Lessons consist of video tutorials, short quizzes, example programs to explore, and written programming exercises, adding up to over 100 hours of hands-on programming practice in total. Each unit ends with a comprehensive unit test that assesses a student's mastery of the material from that unit.

Programming Environment: Students write and run JavaScript programs in the browser using the CodeHS editor. For the Arduino content, students write and run programs in the browser using the Tinkercad simulator and will download their programs to their Arduino devices using Arduino software for further testing.

More information: Browse the content of this course at https://codehs.com/course/9650

Prerequisites: The Intro to Programming in JavaScript with Arduino course is designed for complete beginners with no previous background in computer science. The course is highly visual, dynamic, and interactive, making it engaging for new coders.

Course Breakdown

Unit 1: Introduction to Programming in JavaScript with Karel the Dog (3 weeks/15 hours)

Browse the full content of this unit at https://codehs.com/library/course/9650/module/14557

Objectives / Topics Covered	• Commands • Defining vs. Calling Methods • Designing methods • Program entry points • Control flow • Looping • Conditionals • Classes • Commenting code • Preconditions and Postconditions • Top Down Design
Assignments / Labs	• 26 Karel programming exercises in total • Program-specific tasks for Karel the Dog

	○ Example Exercise: Pyramid of Karel Write a program to have Karel build a pyramid. There should be three balls on the first row, two in the second row, and one in the third row. ● Teach Karel new commands like `turnRight()` or `makePancakes()` ○ Example Exercise: Pancakes Karel is the waiter. He needs to deliver a stack of pancakes to the guests on the 2nd, 4th, and 6th avenue. Each stack of pancakes should have three pancakes. Create a method called `makePancakes()` to help Karel solve this problem. ● Solve large Karel problems by breaking them down into smaller, more manageable problems using Top Down Design ○ Example Exercise: The Two Towers In this program, Karel should build two towers of tennis balls. Each tower should be 3 tennis balls high. At the end, Karel should end up on top of the second tower, facing East. ● Using control structures and conditionals to solve general problems ○ Example Exercise: Random Hurdles Write a program that has Karel run to the other side of first street, jumping over all of the hurdles. However, the hurdles can be in random locations. The world is fourteen avenues long.

Unit 2: Karel Challenges (1.5 weeks, 7 hours)

Browse the full content of this unit at https://codehs.com/library/course/9650/module/14558

Objectives / Topics Covered	● Solving large and more complex problems using Karel
Assignments / Labs	● 5 Karel challenges to tie everything learned in the Karel module together ○ Example Exercise: Super Cleanup Karel Karel's world is a complete mess. There are tennis balls all over the place, and you need to clean them up. Karel will start in the bottom left corner of the world facing east, and should clean up all of the tennis balls in the world. This program should be general enough to work on any size world with tennis balls in any locations.

Unit 3: Javascript & Graphics (1 week/5 hours)

Browse the full content of this unit at https://codehs.com/library/course/9650/module/14559

Objectives / Topics Covered	● Variables ● User Input ● Arithmetic Expressions ● Graphics
Assignments / Labs	● 7 JavaScript & graphics programming exercises in total ● Using variables and getting user input using JavaScript ○ Example Exercise: Dinner Plans Prompt the user for their name, then ask them what time you should meet for dinner. Greet them by name and tell them you will meet them at the time

	they specified!

Unit 4: Graphics Challenges (1 week, 5 hours)

Browse the full content of this unit at https://codehs.com/library/course/9650/module/14560

Objectives / Topics Covered	● Pair Programming ● Solving large and more complex problems using graphics
Assignments / Labs	● 3 graphics challenges to tie everything learned in the JavaScript & Graphics module together ○ Example Exercise: Ghost Write a program to draw a ghost on the screen. You must do this by using the constant values given (this will allow us to easily alter the size or color of the ghost.)

Unit 5: Intro to Arduino (1-2 weeks/5-10 hours)

Browse the full content of this unit at https://codehs.com/library/course/9650/module/14534

Objectives / Topics Covered	● Intro to physical computing ● Goal Setting ● Comments ● Pseudocode ● Analog vs. digital ● Variables ● Breadboards ● Potentiometers ● Debugging
Example Assignments / Labs	● 5 explorations ● 12 exercises total ● Example exercises: ○ Morse Code ■ Send a message using Morse code and your LED. ● Think of one letter you want to send. Translate the letter to Morse code. Make the LED blink to match the Morse code translation. Switch programs with a friend and translate each other's letter! ○ Opposite Blinking LEDs ■ Blink two LEDs opposite one another. One LED should be lit while the other is off. After 1 second, the lit LED should turn off and the unlit LED should turn on. After 1 second, they should switch again. This should continue until the program is manually ended ○ One Bright, One Fading ■ Turn a yellow LED on while a red LED fades. The yellow LED should turn on and the red LED should be set to a brightness of 250. The yellow LED should stay on for two seconds. Every half a second, the red LED should decrease brightness by 50. After two seconds, both LEDs should be off for a second. This should continue until the program is manually ended

Unit 6: JavaScript Control Structures (3 weeks/15 hours)

Browse the full content of this unit at https://codehs.com/library/course/9650/module/14561

Objectives / Topics Covered	• Booleans • For Loops • Conditionals • Nested Control Structures • While Loops

Assignments / Labs	• 22 control structures programming exercises in total
	• Using comparison and logical operators to control the flow of the program
	  ○ Example Exercise: Inventory
	  Write a program that keeps track of a simple inventory for a store. While there are still items left in the inventory, ask the user how many items they would like to buy. Then print out how many are left in inventory after the purchase. You should use a while loop for this problem.
	  Make sure you catch the case where the user tries to buy more items than there are in the inventory. In that case, you should print a message to the user saying that their request isn't possible.
	• Using for loops
	  ○ Example Exercise: All Dice Values
	  ■ Write a program that prints all possible dice rolls with 2 dice. To do so, you should use a double for loop. Hint: You can't use i for both for loops.
	• Drawing basic graphics using JavaScript
	  ○ Example Exercise: Caterpillar
	  This graphics program should draw a caterpillar. A caterpillar has NUM_CIRCLES circles. Every other circle is a different color, the even circles are red, and the odd circles are green (by even we mean when i is an even number). Use a for loop to draw the caterpillar, centered vertically in the screen. Also, be sure that the caterpillar is still drawn across the whole canvas even if the value of NUM_CIRCLES is changed.

Unit 7: Control Structures Challenges (1 week, 5 hours)

Browse the full content of this unit at https://codehs.com/library/course/9650/module/14562

Objectives / Topics Covered	• Solving large and more complex problems using control structures

Assignments / Labs	• 4 challenges using control structures to tie everything learned in the JavaScript Control Structures module together
	  ○ Example Exercise: Guessing Game
	  The computer picks a number between 1 and 100, and you have to guess it. The computer will tell you whether your guess was too high, too low, or correct. Your assignment is to generate a random number and let the user guess numbers until they guess the correct number. Make sure to let the user know what they should do at the beginning of the program!

Unit 8: Functions and Parameters (2 weeks/10 hours)

Browse the full content of this unit at https://codehs.com/library/course/9650/module/14563

Objectives / Topics Covered	● Functions with and without parameters ● Functions with and without return values ● Nested Control Structures ● Local variables and scope
Assignments / Labs	● 14 functions programming exercises in total ● Using various kinds of functions such as functions with and without parameters, and functions with and without return values ○ Example Exercise: Vertical Lines Write a function that draws vertical lines on the graphics canvas. If a line is vertical, then the x-values for the endpoints are the same. The parameters to your function should be the x location, and the length, and all of your lines should start at y position 0. ○ Example Exercise: Is it even? Write a function called isEven that returns a boolean of whether or not a value is even or odd. The isEven function should not print anything out or return a number. It should only take in a number and return a boolean. Once you've written this function, write a program that asks the user for integers and prints whether the number they entered is even or odd using your isEven function. You should let the user keep entering numbers until they enter the SENTINEL given.

Unit 9: Functions Challenges (1 week/5 hours)

Browse the full content of this unit at https://codehs.com/library/course/9650/module/14564

Objectives / Topics Covered	● Solving large and more complex problems using functions
Assignments / Labs	● 4 challenges using functions to tie everything learned in the Functions and Parameters module together ○ Example Exercise: Balloons You should use lines, circles, and random colors to draw a bunch of balloons. All the balloon strings should start two-thirds down the canvas. Each string line should travel upward to a random point and have a circle placed on top of the endpoint. Each balloon should be a random color and have a radius between `MIN_RADIUS` and `MAX_RADIUS`.

Unit 10: Program Control with Arduino (2-3 weeks/10-15 hours)

Browse the full content of this unit at https://codehs.com/library/course/9650/module/14535

Objectives / Topics Covered	● For loops ● While loops ● Variables ● If statements ● If/else statements ● Using buttons ● Using motors ● Operators (arithmetic, comparison, and logical) ● Using sensors (ultrasonic range finder, light sensor, temperature sensor) ● Functions and parameters

Example Assignments / Labs	● 5 explorations ● 10 exercises in total ● Example exercises: ○ Blinking Based on Potentiometer ■ Blink an LED at a speed based on the value of the potentiometer. If the potentiometer reads a value below 500, blink the LED on and off for a quarter second each. If the potentiometer reads a value above 500, blink the LED on and off for a half second each. ○ Servo Sweep with Reset ■ Slowly increase the position of the servo (increasing by 1 degree every 100ms is a good speed) until it reaches 180 degrees. If at any point Button A is pressed, set the servo back to 0 for 1 full second and then start the process over again. ○ Distance Warning Lights ■ No LEDs should be lit if an object is detected 15cm or farther from the ultrasonic range finder. If an object is detected closer than 15 cm, light only a green LED. If an object is detected closer than 10 cm, light only a yellow LED. If an object is detected closer than 5 cm, light only a red LED.

Unit 11: Animation and Games (3 weeks/15 hours)

Browse the full content of this unit at https://codehs.com/library/course/9650/module/14565

Objectives / Topics Covered	● Timers ● Randomizing Games ● Mouse Events ● Keyboard Events
Assignments / Labs	● 15 animations programming exercises in total ● Using timers to add randomizations to graphical programs ○ Example Exercise: Paint Splatter Write a program that splatters paint on the screen every DELAY milliseconds. To splatter paint, pick a random color and draw CIRCLES_PER_SPLATTER circles of that color at random places on the screen. The radius of each circle should be a random value between MIN_RADIUS and MAX_RADIUS. Remember to use helper functions. ● Using mouse events for interactive programs ○ Example Exercise: Target Draw a target on the screen that moves to aim at where your mouse is located. A target consists of a horizontal line that goes from 0 to the window width and a vertical line that goes from 0 to the window height. The lines should cross paths where the mouse is. If you're feeling adventurous, you can extend this to draw a small red circle whenever you click. If you're feeling really adventurous, you can have a bouncing ball on the screen and see if you can remove it when it gets clicked. You can use remove(obj) to remove something from the screen and

	getElementAt(x, y) to get an object at the given position. It will return the object or will return null if there is no object there. ● Using keyboard events for interactive programs ○ Example Exercise: Basic Snake Write a basic version of the snake game. The way our game works is by first creating a green square at the center of the screen. The snake should be moving to the right. If you hit an arrow key, you should change the snake's direction.

Unit 12: Animations Challenges (1 week/5 hours)

Browse the full content of this unit at https://codehs.com/library/course/9650/module/14566

Objectives / Topics Covered	● Solving large and more complex problems using animation
Assignments / Labs	● 2 challenges using animation to tie everything learned in the Animation & Games module together ○ Example Exercise: Blinking Rectangles You should divide the canvas into an imaginary grid with `NUM_RECTANGLES_ACROSS` rectangles across, and `NUM_RECTANGLES_DOWN` rectangles down. Each time the user moves the mouse, a rectangle aligned with this grid should be drawn so that the mouse's location is within the rectangle. The rectangle should change color each time the mouse passes over it.

Unit 13: Project: Breakout (2 weeks/10 hours)

Browse the full content of this unit at https://codehs.com/library/course/9650/module/14567

Objectives / Topics Covered	● Basic graphics ● Mouse events ● Collision detection
Assignments / Labs	● Guided exercises to build a Breakout Game ● Breakout is made up of bricks at the top of the screen, a paddle that you control at the bottom of the screen, and a ball that bounces around. Your goal is to direct the paddle with your mouse to bounce the ball until all of the bricks have been hit and disappear.

Unit 14: Advanced Arduino (4-7 weeks/20-35 hours)

Note: This project can be combined with the final project in the next module

Browse the full content of this unit at https://codehs.com/library/course/9650/module/14536

Objectives / Topics Covered	● Challenges ● Explore a new sensor ● Build a step-by-step project ● Final project
Example Assignments / Labs	● Example exercises: ○ Explore a new sensor ■ Research a sensor we have not studied in this course and explore how it is used. In a group, develop and present a lesson to teach your peers about your chosen sensor,

	including exercises where they can practice using the sensor for themselves. ○ Build a step-by-step project ■ Find a ready-made project online. Follow the steps to recreate the project using your Arduino and any needed materials. ■ Create an updated set of directions complete with pictures and tips from your experience. ○ Final project ■ Use your Arduino to bring an idea to life using sensors and external components. ■ Present your project to peers, administration, and family!

Unit 15: Final Project (2-4 weeks/10-20 hours)

Note: This project can be combined with the final project in the previous Arduino module

Browse the full content of this unit at https://codehs.com/library/course/9650/module/14568

Objectives / Topics Covered	● Collaborative Programming ● Project Planning ● Pseudocode
Assignments / Labs	● Collaborative open-ended final project which encourages creativity ● Program Requirements: Your program: ○ must use JavaScript Graphics ○ must allow the user to interact with your project with either their mouse or keyboard ○ must use at least one timer ○ must break down the program into multiple functions ○ must utilize control structures where applicable

Optional Supplemental Materials (Remainder of school year)

These supplemental materials should be used following the Prerequisite Units mentioned:

Supplementary Units	**Prerequisite/Recommended Unit(s)**	**# of activities**
Extra Karel Practice	Programming with Karel and Karel Challenges	12
Extra Karel Puzzles	Programming with Karel and Karel Challenges	11
Functions and Parameters Practice	Functions & Parameters	8
Tic Tac Toe	Data Structures	4
Helicopter Game	Data Structures	24