littleBits™ education

CODE KIT CURRICULUM GUIDE

WHAT IS THIS GUIDE?

This guide is a starting point to get you thinking about the best way to use the littleBits Code Kit lessons and resources in your learning space. We'll walk you through:

- Identifying your learning pathway
- Choosing your lessons
- Assessing coding and the engineering design process

WHO IS IT FOR?

Any educator who wants to support learners in their quest to code! The educational resources were primarily designed for novice coders, but can be tailored for seasoned pros.

WHY DO I NEED IT?

We have compiled a suite of learning and support resources that aim to make teaching with the Code Kit an engaging and rewarding experience. This guide makes it easy to get started. You may follow the lessons to a T or use them as seeds of inspiration -- ultimately the journey and tools are yours to customize.

Educators: Start here!

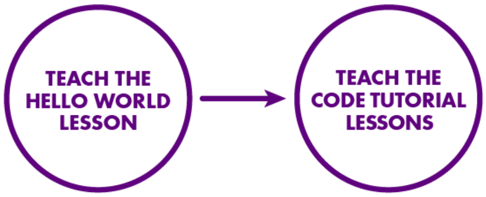
Before your students jump into the Code Kit, we recommend:

- Trying the Hello World activity in the Code Kit app
- Exploring the **CODE KIT EDUCATOR RESOURCES**
    - Your one-stop shop for orientation videos, customizable lessons and student handouts.

Learning Pathways

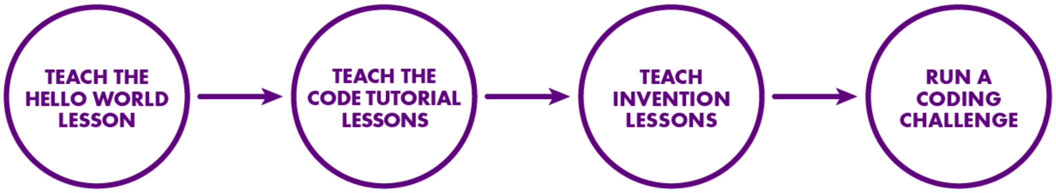
Based on your experience level, the level of your students, and the time that you have to work with the Code Kit, we recommend following one of the pathways below:

1. **WE'RE NEW TO CODING AND ONLY HAVE A FEW DAYS:**



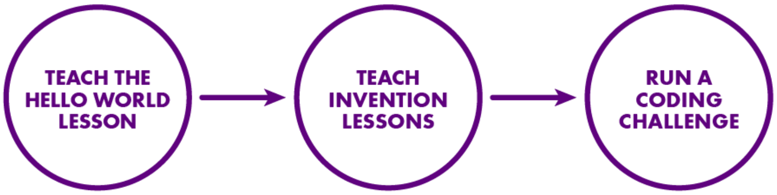
There are five Code Tutorial lessons; dive into as many as time permits! No craft materials required.

2. **WE'RE NEW TO CODING AND WANT A FULL CURRICULUM PLAN OR PROGRAM:**



Try one invention, or try them all! See the Unit Outline below for support in developing your plan.

3. **WE'RE COMPETENT WITH CODE AND WANT TO DIVE RIGHT INTO ENGINEERING AND CODING:**



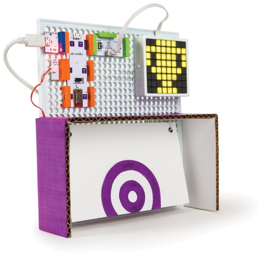
Skip the Code Tutorial lessons and dive straight into the app-supported Invention Lessons.

Unit Outline

Now that you've chosen a pathway, you can use this guide to adapt the lessons for your students and learning space. **Hello World is the recommended starting place for all learners (including educators!)**. The units that correspond to the four app-supported inventions (#3-6 below) can be taught in sequence, or you can choose to tackle just one. Try the "Change the World Arcade" Challenge when you are ready to put your coding and engineering design skills to the test!

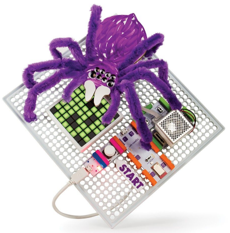
1. Hello World
    a. Your very first introduction to the Code Kit! This in-app instruction covers littleBits basics and how to use the software to communicate with your codeBit. Even for learners that have littleBits experience, this is the recommended starting point.
    b. Lesson: One 50-minute class period recommended
    c. Level: Beginner

2. Code Tutorials
    a. Short activities and fun videos included in the app provide an overview of the core coding concepts that students will encounter in the inventions, and how they relate to the real world. The five tutorial groups are how to's in:
        i. Inputs and Outputs: Control images, sound, and motion
        ii. Loops: Make animated images
        iii. Logic: Add choices to a game
        iv. Variables: Create and use image, number, and coordinates variables
        v. Functions: Make custom blocks to level-up a game
    b. Lessons: One topic per 50-minute class period; five topics in total (older or more advanced students could cover two topics per 50-minute period).
    c. Level: Beginner
    d. For a more detailed overview of the specific learning goals for each tutorial, see **APPENDIX 1**.



3. Ultimate Shootout
    a. Students will use their knowledge of loops, conditional logic and variables to create a 2-player score tracker game that they code and build themselves.

b. Note: This will be your students’ first foray into combining their knowledge of coding concepts with physical design engineering through the littleBits Invention Cycle. We recommend using this unit as a launchpad into the other inventions.
c. Lesson: Three 50-minute class periods recommended
d. Level: Beginner
e. Helpful tutorials: 1.0, 2.0, 4.0, 4.1



4. Hot Potato…*of Doom*!
    a. Students will use loops, conditional logic and variables to create a new spin on the game of Hot Potato.
    b. Note: This lesson provides a choice between a beginner or intermediate-level exploration into understanding the invention’s code.
    c. Lesson: Three 50-minute class periods recommended
    d. Level: Beginner - Intermediate
    e. Helpful tutorials: 1.0, 1.1, 2.0, 2.1, 3.1, 4.0

5. Rockstar Guitar
    a. Students will design a musical instrument that makes it easy for anyone, even for those who cannot read music, to play simple songs.
    b. Note: This lesson starts with a challenge. Experience with the beginner level-lessons (Ultimate Shootout and/or Hot Potato...*of Doom*!) is recommended before tackling this one.
    c. Lesson: Four 50-minute class periods recommended
    d. Level: Intermediate
    e. Helpful tutorials: 1.0, 1.1, 1.2, 2.0, 3.0, 4.0

6. Tug of War
    a. Students will create and remix the Tug of War game to explore how functions are used to structure code.
    b. Note: This lesson provides a deeper exploration into debugging and gamification remixes. An optional review of functions is provided in the introduction.
    c. Lesson: Four 50-minute class periods recommended
    d. Level: Advanced
    e. Helpful tutorials: 1.0, 2.1, 2.2, 4.1, 4.2, 5.0

7. Change the World Arcade
    a. Students design and prototype a game that will help make life easier for people in their community. At the end, each group will present their prototypes at a "Change the World Arcade" that will be open to school or community members.
    b. Note: This unit moves students through an open-ended design prompt. It is recommended that students complete a few of the app-supported inventions before trying this challenge.
    c. Lesson: Six 50-minute class periods recommended
    d. Level: Advanced

Assessment

How will you ask students to document their learnings? Are you offering multiple opportunities and ways to show what they know? Assessment tools provided in **CODE KIT EDUCATOR RESOURCES** are outlined below. Feel free to expand/add your own! CSTA and NGSS alignment for the Code Kit learning content can be found in the **CURRICULAR CROSSWALK**.

EMBEDDED ASSESSMENT

Track student learning dynamically over time with these resources.

CODE MASTER WORKBOOK	Print out a workbook for each student. As your class moves through the tutorial lessons, have them record their notes and reflections for each coding concept. This workbook can serve as a reference tool as they move into the Invention Units.
INVENTION LOG	Students will document their design and thought processes in the Invention Log, which follows the littleBits Invention Cycle. Use the Invention Log checklist at the back of the Log to ask questions about students' current task and ensure that they are on the right track.

SELF & PEER ASSESSMENT

Students evaluate themselves and receive feedback from peers as a measure of growth.

INVENTION LOG	The checklist at the back of the Invention Log can be used as a self-assessment tool as students move from phase to phase.
FEEDBACK CHART	Encourage students or groups to document their "Glow, Grow, Question, and Idea" as other groups present their work. The presenting group will use the feedback provided in the charts to improve their inventions. Charts can also be used as a way to assess how well students are understanding the process and their ability to think critically and constructively about the work they are evaluating.

SUMMATIVE ASSESSMENT

Evaluate learnings at the end of the instructional unit.

FINAL PRESENTATION	Students will share their final product with their peers or community. Students should be able to explain their process, as well as understand how the electronics and code work together within the context of the challenge. Possible formats include poster or slide presentations. Consider a science fair or arcade day setup for sharing amongst multiple groups or a larger audience.
INVENTION LOG	Use the Invention Log to review students' entries into their Invention Log and assess their understanding of the challenge and the invention process as a whole. Be sure students have completed the self-assessment checklist; you can use this tool to add your own notes and evaluation of the Log.

WHAT DOES "SUCCESS" LOOK LIKE?

Examples of Code Kit invention builds from students can be found on the **LITTLEBITS INVENTION PLATFORM**; filter by "Code Kit."

Keep in mind that success with the Code Kit is more than just physical builds and final products; it includes dispositions as well. Consider how you will encourage and/or evaluate growth mindsets and curiosity in your learning environment.

According to our littleBits Lead Educators, a successful student is:

"Somebody who sticks with the challenge of programming until they have finished. They don't get overly frustrated and give up. They understand that coding, like most things, is a series of iterations and that rarely is a first attempt the best they can do."

– Chris Kunkel,
Math Teacher & STEM Coordinator at Hoboken Charter School

"A student who will take the preloaded inventions as inspiration but then continue remixing and designing new and unique ideas with code, not just settle with playing with the code provided."

– Kelly Knight,
STEAM Coordinator at Riverside Presbyterian Day School

APPENDIX 1. TUTORIAL OVERVIEW

Code Tutorials are presented in a series of five plug-and-play lessons featuring:

1. In-app interactive tutorials, merging visual programming with Bits, that will teach your students the basics of inputs & outputs, loops, logic, variables, and functions
2. Fun videos that introduce each of these code concepts
3. A Code Master Workbook to use for assessment and reference as students move into the Invention Units

CATEGORY	LEARNING GOALS
INPUTS & OUTPUTS	**Students will learn how to use inputs and outputs to control image, sound, and motion.**
1.0 YOUR FIRST GAME CONTROLLER	Students will make a simple controller. They will learn how to use all the input Bits in their kit to control an image on the LED matrix Bit.
1.1 ARCADE SOUND EFFECTS	Students will make a sound effect. They will learn how to control sound using a pressure sensor and the speaker.
1.2 EXTREME SOUND MIXING	Students will make a sound mixer. They will learn how to convert the signal from the slide dimmer Bit into sound on the speaker Bit.
LOOPS	**Students will learn how to use loops to make animations.**
2.0 ANIMATE YOUR OWN GAME CHARACTER	Students will make a simple animation. They will learn how use a repeat loop and a forever loop, draw a character on the matrix screen, then add frames to animate it.
2.1 SHHH! IT'S A SECRET MESSAGE	Students will make a secret message appear when they press a button. They will learn how to use a [REPEAT WHILE] and [REPEAT UNTIL] loop to hide a scrolling text message.
2.2 SUPER SPECIAL EFFECTS	Students will make a special effect. They will learn how to use a for loop with variable and math blocks to animate pixels shooting across the screen.

LOGIC	**Students will learn how to use conditional logic to program rules into their games.**
3.0 CHOOSE YOUR OWN EMOJI	Students will make a multiple-choice game using emojis. They will learn how to draw emojis, use IF ,DO statements with a single input as well as math to add multiple choices to a game.
3.1 RETRIEVE THE MYSTICAL SWORD	Students will make a lock combination that retrieves a sword. They will learn how to use multiple input Bits, the IF ,DO, ELSE statements, as well as the AND conditional statements to create different lock combinations.
VARIABLES	**Students will learn how to create and use image, number, and coordinates variables.**
4.0 BIRTHDAY COUNTDOWN	Students will make a countdown using a birthday melody. They will learn how to create a number variable to control how fast each note in the birthday song plays.
4.1 JUMPY MONSTER SPRITE	Students will make a game sprite that jumps and makes a sound effect when a button is pressed. They will learn how to create and use an image variable to create an animated character sprite.
4.2 TWO PLAYER RACE	Students will make a two-player race game. They will learn how to create and use multiple variables to track the position of a player on the x and y axis of the matrix screen. They will also learn how to use variables with logic conditional statements to make a multiplayer game.
FUNCTIONS	**Students will learn how to create their own unique code block with functions to level-up their games.**
5.0 EPIC DANCE MOVES	Students will make a character dance using multiple animations. They will learn how to create functions for each of the custom moves, then reuse them multiple times to create the final animation.