littleBits education

Code Kit Curricular Crosswalk
Standards Alignment

Overview

This chart provides an overview of the standards that can be met, or extended to meet, with the littleBits Code Kit and Expansion Pack lessons.

Alignments included in this document:

- Code Kit, 2017 CSTA
- Code Kit, Pre-2017 CSTA
- Code Kit, NGSS
- Code Kit Expansion Pack: Computer Science, 2017 CSTA
- Code Kit Expansion Pack: Computer Science, National Core Arts Standards
- Code Kit Expansion Pack: Technology, 2017 CSTA
- Code Kit Expansion Pack: Technology, ITEEA

Use the Code Master Workbook (for tutorials) and the Code Kit Invention Log (for inventions) as a way to assess whether your students have fulfilled these standards. See the Curriculum Guide for additional information on assessment strategies.

Code Kit: CSTA 2017

Elementary (Grade 3-5)

IDENTIFIER	CSTA STANDARD	CONCEPT	SUBCONCEPT	PRACTICE(S)	LESSONS THAT MEET THIS STANDARD
1B-CS-01	Describe how internal and external parts of computing devices function to form a system.	Computing Systems	Devices	Communicating	Hello World
1B-CS-03	Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies.	Computing Systems	Troubleshooting	Testing, Computational Problems	All Code Kit lessons, as needed
1B-AP-08	Compare and refine multiple algorithms for the same task and determine which is the most appropriate.	Algorithms & Programming	Algorithms	Testing, Computational Problems	Ultimate Shootout
1B-AP-09	Create programs that use variables to store and modify data.	Algorithms & Programming	Algorithms	Creating	Tutorials: Loops; Logic; Variables; Functions
1B-AP-10	Create programs that include sequences, events, loops, and conditionals.	Algorithms & Programming	Control	Creating	- Hello World, - Tutorials: Inputs and Outputs; Loops; Logic; Variables; Functions - Inventions: Ultimate Shootout, Hot Potato..of Doom!, Rockstar Guitar

littleBits education

1B-AP-13	Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.	Algorithms & Programming	Program Development	Inclusion, Creating	Ultimate Shootout Hot Potato...of Doom!, Rockstar Guitar
1B-AP-15	Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.	Algorithms & Programming	Program Development	Testing, Computational Problems	Ultimate Shootout, Hot Potato...of Doom!, Rockstar Guitar
1B-AP-16	Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.	Algorithms & Programming	Program Development	Collaborating	Hot Potato...of Doom!, Rockstar Guitar

Middle (Grade 6-8)

IDENTIFIER	CSTA STANDARD	CONCEPT	SUBCONCEPT	PRACTICE(S)	LESSONS THAT MEET THIS STANDARD
2-CS-03	Systematically identify and fix problems with computing devices and their components.	Computing Systems	Troubleshooting	Testing	All Code Kit lessons, as needed
2-AP-10	Use flowcharts and/or pseudocode to address complex problems as algorithms.	Algorithms & Programming	Algorithms	Abstraction	Ultimate Shootout, Hot Potato...of Doom!, Rockstar Guitar, Tug of War, Change the World Arcade Challenge
2-AP-11	Create clearly named variables that represent different data types and perform operations on their values.	Algorithms & Programming	Variables	Creating	Tutorials: Loops; Logic; Variables; Functions

2-AP-12	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	Algorithms & Programming	Control	Creating	Tutorials: Loops; Logic; Variables; Functions
2-AP-14	Create procedures with parameters to organize code and make it easier to reuse.	Algorithms & Programming	Modularity	Abstraction	Tug of War
2-AP-15	Seek and incorporate feedback from team members and users to refine a solution that meets user needs.	Algorithms & Programming	Program Development	Collaborating, Inclusion	Ultimate Shootout, Hot Potato...of Doom!, Rockstar Guitar, Tug of War, Change the World Arcade Challenge

High (Grade 9-12)

IDENTIFIER	CSTA STANDARD	CONCEPT	SUBCONCEPT	PRACTICE(S)	LESSONS THAT MEET THIS STANDARD
3A-AP-16	Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.	Algorithms & Programming	Control	Creating	Tug of War, Change the World Arcade Challenge
3A-AP-18	Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.	Algorithms & Programming	Modularity	Creating	Tug of War, Change the World Arcade Challenge

| 3A-AP-22 | Design and develop computational artifacts working in team roles using collaborative tools. | Algorithms & Programming | Program Development | Collaborating | Change the World Arcade Challenge |
| 3B-AP-22 | Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality). | Algorithms & Programming | Program Development | Creating | Tug of War, Change the World Arcade Challenge |

Code Kit: CSTA Pre-2017

Elementary (Grade 3-5)

IDENTIFIER	CSTA STANDARD	FRAME CONCEPT	FRAMEWORK PRACTICE	LESSONS THAT MEET THIS STANDARD	WAYS TO ADD THIS STANDARD TO ANY LESSON
1B-A-2-1	Apply collaboration strategies to support problem solving within the design cycle of a program.	Algorithms & Programming	Collaborating	Invention Lessons: Ultimate Shootout; Hot Potato...of Doom!; Rockstar Guitar	
1B-A-5-3	Create a plan as part of the iterative design process, both independently and with diverse collaborative teams (e.g., storyboard, flowchart, pseudo-code, story map).	Algorithms & Programming	Creating Computational Artifacts	Invention Lessons: Ultimate Shootout; Hot Potato...of Doom!; Rockstar Guitar	
1B-A-5-4	Construct programs, in order to solve a problem or for creative expression, that include sequencing, events, loops, conditionals, parallelism, and	Algorithms & Programming	Creating Computational Artifacts	--All Tutorial Lessons: Inputs and Outputs; Loops; Logic; Variables; Functions	

	variables, using a block-based visual programming language or text-based language, both independently and collaboratively (e.g., pair programming).			--Invention Lessons: Ultimate Shootout; Hot Potato…of Doom!; Rockstar Guitar"	
1B-A-5-5	Use mathematical operations to change a value stored in a variable.	Algorithms & Programming	Creating Computational Artifacts		Ask students to track score or add a second player's score in their games.
1B-A-3-7	Construct and execute an algorithm (set of step-by-step instructions) which includes sequencing, loops, and conditionals to accomplish a task, both independently and collaboratively, with or without a computing device.	Algorithms & Programming	Recognizing and Defining Computational Problems	- Hello World - All Tutorial Lessons: Inputs and Outputs; Loops; Logic; Variables; Functions - Invention Lessons: Ultimate Shootout; Hot Potato…of Doom!; Rockstar Guitar	
1B-A-6-8	Analyze and debug (fix) an algorithm that includes sequencing, events, loops, conditionals, parallelism, and variables.	Algorithms & Programming	Testing and Refining	Invention Lessons: Ultimate Shootout; Hot Potato…of Doom!; Rockstar Guitar	Present students with an altered version of the invention code that you know doesn't work. Challenge students to fix it. The debugging checklist can be used as a reference.
1B-C-7-9	Model how a computer system works. [Clarification: Only includes basic elements of a computer system, such as input, output, processor, sensors, and storage.]	Computing Systems	Communicating about Computing		Ask students to describe how inputs, outputs, and the codeBit function in their circuit.

1B-C-6-11	Identify, using accurate terminology, simple hardware and software problems that may occur during use, and apply strategies for solving problems (e.g., reboot device, check for power, check network availability, close and reopen app).	Computing Systems	Testing & Refining	Hello World	Ask students, before they playtest their inventions, to predict what problems they may encounter with the hardware or software.
1B-D-4-14	Use numeric values to represent non-numeric ideas in the computer (binary, ASCII, pixel attributes such as RGB).	Data and Analysis	Developing and Using Abstractions	Tutorial Lessons: Loops; Logic; Variables; Functions	During the Remix phase, challenge students to add the [COLOR RGB] block to their code and set the values.

Middle (Grade 6-8)

IDENTIFIER	CSTA STANDARD	FRAME CONCEPT	FRAMEWORK PRACTICE	LESSONS THAT MEET THIS STANDARD	WAYS TO ADD THIS STANDARD TO ANY LESSON
2-A-2-1	Solicit and integrate peer feedback as appropriate to develop or refine a program.	Algorithms & Programming	Collaborating	All Invention Lessons: Ultimate Shootout; Hot Potato...of Doom!; Rockstar Guitar; Tug of War	Invite students to share their prototypes during the Remix phase, and then integrate the feedback into revisions.
2-A-7-4	Interpret the flow of execution of algorithms and predict their outcomes.	Algorithms & Programming	Communicating about Computing		Ask students to read the code before they run it for the first time, and predict what their circuit will do when they run it.
2-A-5-6	Develop programs, both independently and collaboratively, that include sequences with nested loops and multiple branches.	Algorithms & Programming	Creating Computational Artifacts	Tutorial Lessons: Loops; Logic; Variables; Functions	

2-A-5-7	Create variables that represent different types of data and manipulate their values.	Algorithms & Programming	Creating Computational Artifacts	Tutorial Lessons: Loops and Variables	
2-A-4-8	Define and use procedures that hide the complexity of a task and can be reused to solve similar tasks.	Algorithms & Programming	Developing and Using Abstractions	Invention Lesson: Tug of War	
2-A-6-10	Use an iterative design process to solve problems, both independently and collaboratively.	Algorithms & Programming	Testing and Refining	- All Invention Lessons: Ultimate Shootout; Hot Potato...of Doom!; Rockstar Guitar; Tug of War - Change the World Arcade Challenge	Have students use the Invention Log to guide the design process and record their explorations.

High (Grade 9-12)

IDENTIFIER	CSTA STANDARD	FRAME CONCEPT	FRAMEWORK PRACTICE	LESSONS THAT MEET THIS STANDARD	WAYS TO ADD THIS STANDARD TO ANY LESSON
3A-A-2-1	Design and develop a software artifact working in a team.	Algorithms & Programming	Collaborating	- Invention Lesson: Tug of War - Change the World Arcade Challenge	
3A-A-5-4	Design, develop, and implement a computing artifact that responds to an event (e.g., robot that responds to a sensor, mobile app that responds to a text message, LED monster that responds to a broadcast).	Algorithms & Programming	Creating Computational Artifacts	- Invention Lesson: Tug of War - Change the World Arcade Challenge	
3A-C-5-14	Create, extend, or modify existing programs to add new features and behaviors using different forms of inputs and outputs (e.g.,	Computing Systems	Communicating About Computing	- Invention Lesson: Tug of War - Change the World Arcade Challenge	

	inputs such as sensors, mouse clicks, data sets; outputs such as text, graphics, sounds).				
3B-A-7-3	Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality).	Algorithms & Programming	Communicating About Computing	- Invention Lesson: Tug of War - Change the World Arcade Challenge	

Code Kit: NGSS

Elementary (Grade 3-5)

IDENTIFIER	PERFORMANCE EXPECTATION	LESSONS THAT MEET THIS STANDARD
3-5-ETS1-2	Generate and compare multiple possible solutions to a problem based on how well each is likely to meet the criteria and constraints of the problem.	Invention Lessons: Ultimate Shootout; Hot Potato...of Doom!; Rockstar Guitar
3-5-ETS1-3	Plan and carry out fair tests in which variables are controlled and failure points are considered to identify aspects of a model or prototype that can be improved.	Invention Lessons: Ultimate Shootout; Hot Potato...of Doom!; Rockstar Guitar

Middle (Grade 6-8)

IDENTIFIER	PERFORMANCE EXPECTATION	LESSONS THAT MEET THIS STANDARD
MS-ETS1-1	Define the criteria and constraints of a design problem with sufficient precision to ensure a successful solution, taking into account relevant scientific principles and	Coding Challenge

	potential impacts on people and the natural environment that may limit possible solutions.	
MS-ETS1-2	Evaluate competing design solutions using a systematic process to determine how well they meet the criteria and constraints of the problem.	- All Invention Lessons: Ultimate Shootout; Hot Potato...of Doom!; Rockstar Guitar; Tug of War - Change the World Arcade Challenge
MS-ETS1-3	Analyze data from tests to determine similarities and differences among several design solutions to identify the best characteristics of each that can be combined into new solutions to better meet the criteria for success.	- All Invention Lessons: Ultimate Shootout; Hot Potato...of Doom!; Rockstar Guitar; Tug of War - Change the World Arcade Challenge

High (Grade 9-12)

IDENTIFIER	PERFORMANCE EXPECTATION	LESSONS THAT MEET THIS STANDARD
HS-ETS1-1	Analyze a major global challenge to specify qualitative and quantitative criteria and constraints for solutions that account for societal needs and wants.	Change the World Arcade Challenge
HS-ETS1-2	Design a solution to a complex real-world problem by breaking it down into smaller, more manageable problems that can be solved through engineering.	Change the World Arcade Challenge

Expansion Pack: Computer Science, CSTA 2017

Elementary (Grade 3-5)

IDENTIFIER	CSTA STANDARD	CONCEPT	SUBCONCEPT	PRACTICE(S)	LESSONS THAT MEET THIS STANDARD
1B-CS-02	Model how computer hardware and software work together as a system to accomplish tasks.	Computing Systems	Hardware & Software	Abstractions	Hungry for Functions; Digital Pet; Let's Get Random; Monster P-ART-Y
1B-CS-03	Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies.	Computing Systems	Troubleshooting	Testing, Computational Problems	Invent a Digital Assistant, ArtLab
1B-AP-09	Create programs that use variables to store and modify data.	Algorithms & Programming	Algorithms	Creating	Digital Pet; Invent a Digital Assistant; Let's Get Random; Monster P-ART-Y; ArtLab
1B-AP-10	Create programs that include sequences, events, loops, and conditionals.	Algorithms & Programming	Control	Creating	Hungry for Functions; Digital Pet; Invent a Digital Assistant; Let's Get Random; Monster P-ART-Y; ArtLab
1B-AP-11	Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions	Algorithms & Programming	Modularity	Collaborating	Digital Pet; Invent a Digital Assistant; ArtLab
1B-AP-12	Develop plans that describe a program's sequence of events, goals, and expected outcomes	Algorithms & Programming	Program Development	Inclusion; Collaborating	Digital Pet; Invent a Digital Assistant; ArtLab

1B-AP-13	Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.	Algorithms & Programming	Program Development	Inclusion; Creating	Invent a Digital Assistant; ArtLab
1B-AP-14	Observe intellectual property rights and give appropriate attribution when creating or remixing programs	Algorithms & Programming	Program Development	Creating; Communicating	Digital Pet; Invent a Digital Assistant; ArtLab
1B-AP-15	Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.	Algorithms & Programming	Program Development	Testing, Computational Problems	Digital Pet
1B-AP-16	Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.	Algorithms & Programming	Program Development	Collaborating	Invent a Digital Assistant; ArtLab
1B-AP-17	Describe choices made during program development using code comments, presentations, and demonstrations.	Algorithms & Programming	Program Development	Communicating	Digital Pet; Invent a Digital Assistant; ArtLab
1B-IC-18	Discuss computing technologies that have changed the world, and express how those technologies influence, and are influenced by, cultural practices.	Impacts of Computing	Culture	Inclusion	Invent a Digital Assistant
1B-IC-19	Brainstorm ways to improve the accessibility and usability of technology products for the	Impacts of Computing	Culture	Collaborating	Invent a Digital Assistant

littleBits education

	diverse needs and wants of users.				
1B-IC-20	Seek diverse perspectives for the purpose of improving computational artifacts.	Impacts of Computing	Social Interactions	Inclusion	Invent a Digital Assistant; ArtLab

Expansion Pack: Computer Science, National Core Arts

Elementary (Grade 3-5)

IDENTIFIER	NCA STANDARD	TOPIC	PRACTICE(S)	LESSONS THAT MEET THIS STANDARD
VA:Cr1.1.3a	Elaborate on an imaginative idea.	Visual Arts	Creating	Invent a Digital Assistant; ArtLab
VA:Cr1.1.4a	Brainstorm multiple approaches to a creative art or design problem	Visual Arts	Creating	Invent a Digital Assistant; ArtLab
VA:Cr1.1.5a	Combine ideas to generate an innovative idea for art-making.	Visual Arts	Creating	Invent a Digital Assistant; ArtLab
VA:Cr1.2.3a	Apply knowledge of available resources, tools, and technologies to investigate personal ideas through the art-making process.	Visual Arts	Creating	Invent a Digital Assistant; ArtLab
VA:Cr1.2.4a	Collaboratively set goals and create artwork that is meaningful and has purpose to the makers.	Visual Arts	Creating	Invent a Digital Assistant; ArtLab
VA:Cr1.2.5a	Identify and demonstrate diverse methods of artistic investigation to choose an approach for beginning a work of art.	Visual Arts	Creating	Invent a Digital Assistant; ArtLab

littleBits™ education

VA:Cr3.1.4a	Revise artwork in progress on the basis of insights gained through peer discussion.	Visual Arts	Creating	ArtLab
VA:Re.7.2.3a	Determine messages communicated by an image.	Visual Arts	Responding	ArtLab
VA:Re.7.1.4a	Compare responses to a work of art before and after working in similar media.	Visual Arts	Responding	ArtLab
VA:Re.7.1.5a	Compare one's own interpretation of a work of art with the interpretation of others.	Visual Arts	Responding	ArtLab

Expansion Pack: Technology, CSTA 2017

Middle (Grade 6-8)

IDENTIFIER	CSTA STANDARD	CONCEPT	SUBCONCEPT	PRACTICE(S)	LESSONS THAT MEET THIS STANDARD
2-CS-02	Design projects that combine hardware and software components to collect and exchange data.	Computing Systems	Creating	Creating	Space Communication, Bit Messenger, A-MAZE-ing Racer, Supercharge Your Ride
2-CS-03	Systematically identify and fix problems with computing devices and their components.	Computing Systems	Troubleshooting	Testing	Bit Messenger, Supercharge Your Ride
2-AP-10	Use flowcharts and/or pseudocode to address complex problems as algorithms.	Algorithms & Programming	Algorithms	Abstraction	Bit Messenger, Supercharge Your Ride

littleBits education

2-AP-11	Create clearly named variables that represent different data types and perform operations on their values.	Algorithms & Programming	Variables	Creating	Blnary Communicator, Bit Messenger, Supercharge Your Ride
2-AP-12	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	Algorithms & Programming	Control	Creating	A-MAZE-ing Racer
2-AP-13	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	Algorithms & Programming	Computational Problems	Recognizing, Defining	Bit Messenger, Supercharge Your Ride
2-AP-14	Create procedures with parameters to organize code and make it easier to reuse.	Algorithms & Programming	Modularity	Abstraction	A-MAZE-ing Racer
2-AP-15	Seek and incorporate feedback from team members and users to refine a solution that meets user needs.	Algorithms & Programming	Program Development	Collaborating, Inclusion	Bit Messenger, Supercharge Your Ride
2-AP-16	Incorporate existing code, media, and libraries into original programs, and give attribution.	Algorithms & Programming	Attribution	Abstraction, Creating, Communicating	Bit Messenger, Supercharge Your Ride

2-AP-17	Systematically test and refine programs using a range of test cases.	Algorithms & Programming	Computational Artifacts	Testing	Supercharge Your Ride
2-AP-18	Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.	Algorithms & Programming	Project Management	Collaborating	Bit Messenger, Supercharge Your Ride
2-AP-19	Document programs in order to make them easier to follow, test, and debug.	Algorithms & Programming	Documentation	Communicating	Bit Messenger, Supercharge Your Ride
1B-IC-20	Seek diverse perspectives for the purpose of improving computational artifacts.	Impacts of Computing	Social Interactions	Inclusion	Smart Car

littleBits education

Expansion Pack: Technology, ITEEA

Middle (Grade 6-8)

IDENTIFIER	ITEEA STANDARD	LESSONS THAT MEET THIS STANDARD
1.F	New products and systems can be developed to solve problems or to help do things that could not be done without the help of technology.	Space Communication
1.H	Technology is closely linked to creativity, which has resulted in innovation.	Space Communication
2.P	Technological systems can be connected to one another.	Supercharge Your Ride
3.P	Technological systems often interact with one another.	Supercharge Your Ride
4.D	The use of technology affects humans in various ways, including their safety, comfort,	Smart Car
4.E	Technology, by itself, is neither good nor bad, but decisions about the use of products and systems can result in desirable or undesirable consequences	Smart Car
4.F	The development and use of technology poses ethical issues.	Smart Car
8.E	Design is a creative planning process that leads to useful products and systems.	Bit Messenger, Supercharge Your Ride
8.F	There is no perfect design.	Bit Messenger, Supercharge Your Ride
8.G	Requirements for design are made up of criteria and constraints.	Bit Messenger, Supercharge Your Ride
9.F	Design involves a set of steps, which can be performed in different sequences and repeated as needed.	Bit Messenger, Supercharge Your Ride
9.G	Brainstorming is a group problem-solving design process in which each person in the group presents his or her ideas in an open forum.	Bit Messenger, Supercharge Your Ride
9.H	Modeling, testing, evaluating, and modifying are used to transform ideas into practical solutions.	Bit Messenger, Supercharge Your Ride

10.F	Troubleshooting is a problem-solving method used to identify the cause of a malfunction in a technological system.	Bit Messenger, Supercharge Your Ride
10.H	Some technological problems are best solved through experimentation.	Bit Messenger, Supercharge Your Ride
11.I	Specify criteria and constraints for the design.	Bit Messenger, Supercharge Your Ride
11.J	Make two-dimensional and three-dimensional representations of the designed solution.	Bit Messenger, Supercharge Your Ride
11.K	Test and evaluate the design in relation to pre-established requirements, such as criteria and constraints, and refine as needed.	Bit Messenger, Supercharge Your Ride
11.L	Make a product or system and document the solution.	Bit Messenger, Supercharge Your Ride
12.J	Use computers and calculators in various applications.	Bit Messenger, Supercharge Your Ride
17.H	Information and communication systems allow information to be transferred from human to human, human to machine, and machine to human.	Binary Communicator
17.I	Communication systems are made up of a source, encoder, transmitter, receiver, decoder, and destination.	Binary Communicator, Bit Messenger
17.J	The design of a message is influenced by such factors as intended audience, medium, purpose, and the nature of the message.	Binary Communicator, Bit Messenger
17.K	The use of symbols, measurements, and drawings promotes a clear communication by providing a common language to express ideas.	Space Communication
18.G	Transportation vehicles are made up of subsystems, such as structural propulsion, suspension, guidance, control, and support, that must function together for a system to work effectively.	Supercharge Your Ride