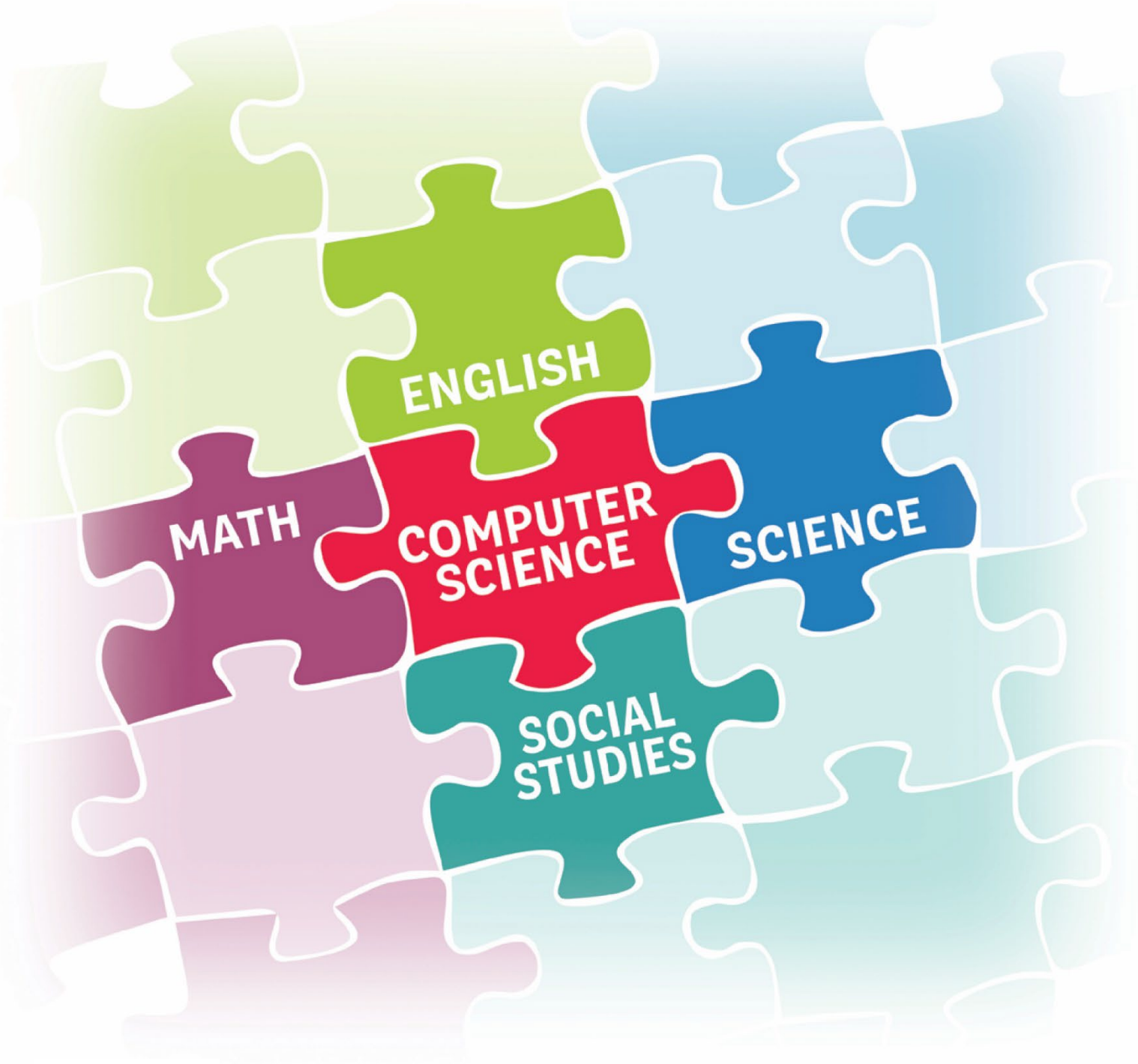


2023

Elementary Integration Guide

SIXTH GRADE

MISSISSIPPI STATE UNIVERSITY™
CENTER FOR CYBER EDUCATION

Acknowledgements

The following people assisted in the development of this integration guide:

Development Team

Melissa Banks, Director of Digital Learning, Mississippi Department of Education
Lizzie Brandon, Teacher, Houston High School
Lerenda Dixon, Teacher, McComb School District
Carrie Doubleday, Teacher, South Panola School District
Mary Dunaway, Teacher, Rankin County School District
Dr. Anthony Golding, Administrator, Tupelo Public School District
Shelly Hollis, Director, Center for Cyber Education
Courtney Hopper, Teacher, South Panola School District
Laura Sheffield, Instructional Coach, Tupelo Public School District
Amanda Taylor, Project Manager, Center for Cyber Education
Ann Thomas, Teacher, Lafayette County School District
Shelley Thompson, Project Manager, Center for Cyber Education
Dr. Louella Mack-Webster, STEM Program Supervisor, Mississippi Department of Education
Dr. Haley Wilson, Curriculum/Technology Specialist, Lafayette County School District

Funding for the development of this guide was provided by:

Introduction

In March 2021, The Mississippi Computer Science and Cyber Education Equality Act (House Bill 633) was passed requiring all districts to offer computer science content and courses by the 2024-2025 school year. The bill allows for a phased-in approach as listed below:

2022-2023:	All middle schools offer at least one (1) course in computer science, and 50% of elementary schools offer a minimum of one (1) hour of instruction in computer science each week at each grade level.
2023-2024:	All elementary schools offer a minimum of one (1) hour of instruction in computer science each week at each grade level, and 50% of high schools offer at least one (1) course in computer science.
2024-2025:	All schools will offer instruction in computer science.

To make the integration of computer science content as seamless as possible for elementary teachers, a task force of elementary teachers, principals, the Mississippi Department of Education, and the Mississippi State University Center for Cyber Education was formed to write an integration guide for each grade level, kindergarten through fifth grade. These guides were released on the CS4MS.org website for the 2022-2023 school year.

A team was constructed to look at the needs for having a sixth-grade integration guide that would benefit each school and/or district while addressing the plethora of options for scheduling sixth grade students. This integration guide contains a breakdown of content by integrated subjects only. The guide is divided into six subject areas: English Language Arts, Mathematics, Science, Social Studies, Social and Emotional Learning, and Digital Citizenship. In addition to a lesson overview and links to required resources, each lesson plan maps to a Mississippi Computer Science Standard and another subject area standard. A suggestion on how to break the lesson into smaller segments, to be covered throughout the week, is also provided in the "Time needed" section.

There are several resources available in each integration guide. Some may require the creation of accounts, but all resources referenced are free. A list of sites used is provided for technology departments to whitelist or unblock. All resources may be used on any internet-capable device, including Chromebooks and tablets.

Table of Contents by Integrated Subjects

30 – 31	Pizza Party Data	Digital Literacy Unplugged	DA.2.2	6.SP.5
32 – 33	Robotic Shape Trace	Robotics Coding	Ap.2.3a, AP.2.4a, AP.2.8a	6.G.1
34 – 35	Number Line Animation	Robotics Coding	Ap.2.3a, Ap.2.4a, AP.2.8a	6.NS.9c, 6.NS.9d
36 – 37	Using Algorithms to Find Answers	Coding Unplugged	Ap.2.1, AP.2.1a	6. EE.6

Science

Page(s)	Lesson Title	Topic(s)	CS Standard	Core Subject Standard
39 – 40	Living vs. Non-Living	Coding	CS.2.2a, AP.2.7a, AP.2.7b, AP.2.10a, AP.2.4a, AP.2.5a, AP.2.8a, AP.2.9a, AP.2.9b, AP.2.9c	L.6.1.1
41 – 42	Organelle Robotics	Robotics Coding	AP.2.4a, AP.5.a, AP.2.8a, AP.2.9a, AP.2.9b, AP.2.9c, AP.2.10b	L.6.1.3
43 – 44	Scratch Relationships	Coding	AP.2.4a, AP.2.5a, AP.2.6a, AP.2.9a, AP.2.9b, AP.2.9c	L.6.3.4
45 – 46	Energy In an Ecosystem: Part 1	Digital Citizenship	AP.2.4a, AP.2.5a, AP.2.6a, AP.2.9a, AP.2.9b, AP.2.9c	L.6.3.5
47 – 48	Energy In an Ecosystem: Part 2	Coding	AP.2.4a, AP.2.5a, AP.2.6a, AP.2.9a, AP.2.9b, AP.2.9c	L.6.3.5
49 – 50	Dichotomous Key Robotics	Robotics Coding	AP.2.4a, AP.2.5a, AP.2.8a, AP.2.10b	L.6.4.2
51 – 52	Coding Kingdoms	Coding	AP.2.4a, AP.2.5a	L.6.4.2

53 – 54	Scratching the Surface of Newton's Law	Coding	AP.2.4a, AP.2.5a	P.6.6.1
55 – 56	Energy and Motion	Coding	AP.2.4a, AP.2.5a	P.6.6.7
57 – 58	Solar System	Coding	AP.2.4a, AP.2.5a, AP.2.9a, AP.2.9b, AP.2.9c	E.6.8.4
59 – 60	Moon Phases	Robotics Coding	AP.2.4a, AP.2.5a	E.6.8.6
61 – 62	Cell Explorations	Coding	AP.2.3a, AP.2.8a, AP.2.10a	L.6.1.3

Social Studies

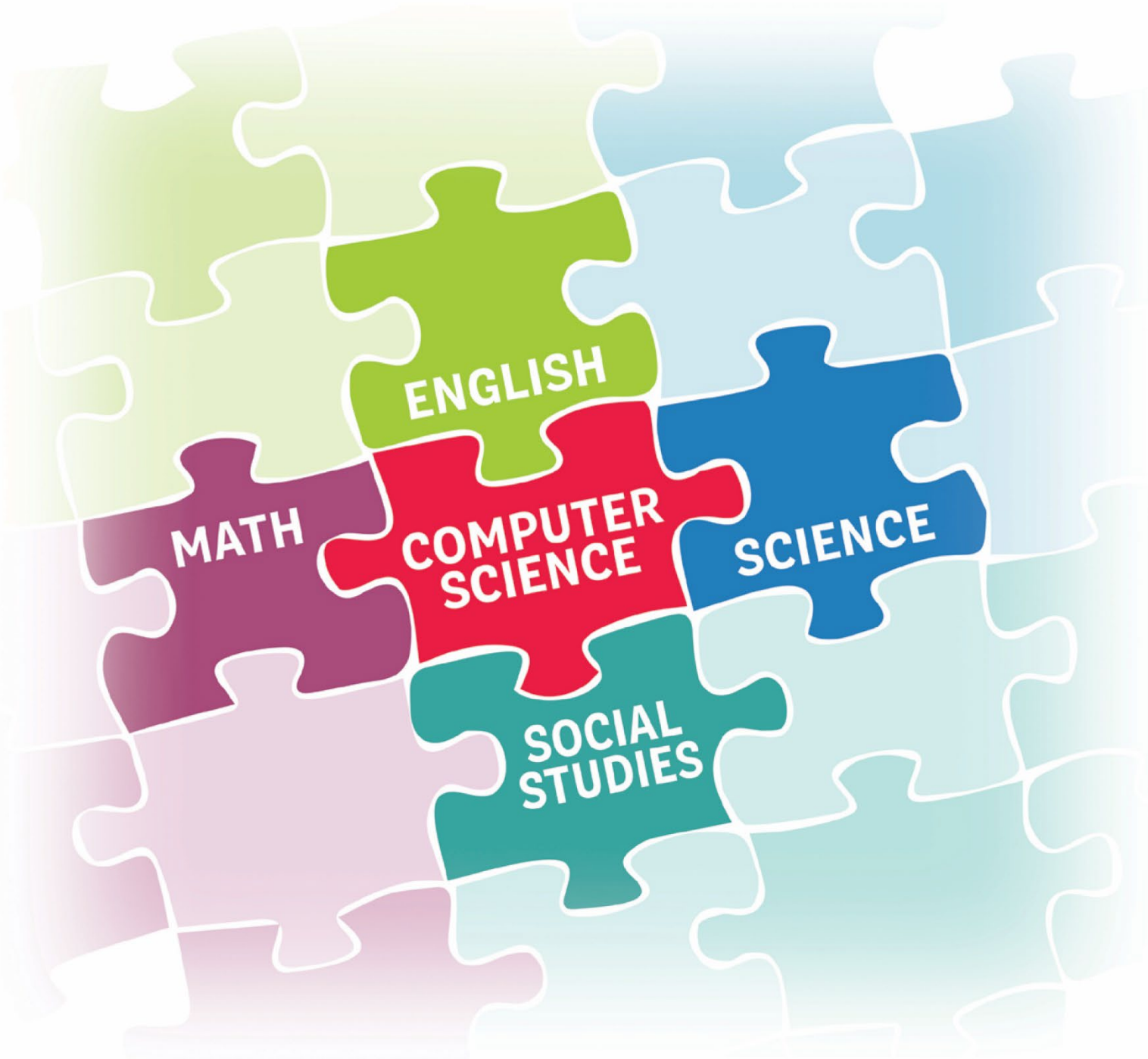
Page(s)	Lesson Title	Topic(s)	CS Standard	Core Subject Standard
64 – 65	Native American Star Quilts – Code.org Star Quilts Module: Lesson 1	Unplugged	AP.2.3a, AP.2.4a	H.6.1, H.6.1.1, H.6.1.2, H.6.1.3, 6.G.1
66 – 67	Getting Started with the Artist – Code.org Star Quilts Module: Lesson 2	Coding	Ap.2.3a, AP.2.4a	H.6.1, H.6.1.1, H.6.1.2, H.6.1.3, 6.G.1
68 – 69	Code Your Star Quilt – Code.org Star Quilts Module: Lesson 3	Coding	Ap.2.3a, AP.2.4a	H.6.1, H.6.1.1, H.6.1.2, H.6.1.3, 6.G.1
70	Designing for Accessibility	Unplugged	CS.2.1, CS.2.1a	G.6.7, G.6.7.3
71 – 72	Finding Credible News: How do we find credible information on the internet?	Digital Literacy	IC.2.4, IC.2.1a	CI.6.2, CI.6.2.1
73 – 74	Cybersecurity – Simple Encryption	Coding	NI.2.3a	H.6.1.1

Social and Emotional Learning

Digital Citizenship

Appendices

Resources	
Computing resources	• Code.org • Common Sense Digital Media • Scratch • Hour of Code • CS First - Google • Codesters • Tynker • VEX VR
CS4MS website materials	• 2018 Mississippi Computer Science Standards • CS4MS Website
Teacher/student accounts	• Code.org • Common Sense Digital Media • Scratch
For help with this guide	• Contact Mississippi State University's Center for Cyber Education: www.tinyurl.com/ccehelpdesk



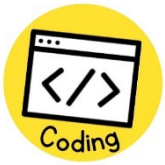
2023

Elementary Integration Guide

LANGUAGE ARTS

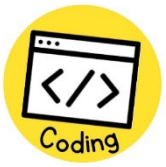
MISSISSIPPI STATE UNIVERSITY™
CENTER FOR CYBER EDUCATION

Code Words

Lesson overview:  	**Purpose:** • Students will learn about coding history and terminology. • Students will learn to code by creating an animated story online. **Lesson:** Warm Up: • Students will brainstorm the different ways 'code' is used (zip code, bar code, code of honor, dress code, cracking the code) • Hand out student magazines. Have students read independently or with a partner. Challenge them to use context to better understand the bolded words as they read and direct them to complete the quiz on the back page. Quiz answer key: 1. C; 2. B; 3. D; 4. A; 5. D; 6. B. • Review the central ideas in the magazine as a class. Prompt students to identify vocabulary words in the magazine that were new to them. Go over the meanings of the words as a class. (Extension: Have students find out five more facts about one of the people in the "Great Moments in Coding History" sidebar.). • Explain that computer coding is about solving problems by breaking a project into smaller tasks. Prompt students to name other things they do that depend on the same kind of logic and problem-solving involved in coding (examples: following a recipe to bake a cake, learning to play a musical instrument, making a craft project). Collect a list on the board. Activity: • Try the Adventure on the High Seas activity on the CS First site. Students will pick two "sprites," or characters, place them in a boat, and create dialogue to tell a story. • Prompt students to imagine a story inspired by a historical figure, current event, or fictional character they've studied in class. Is one of the characters an early explorer to the Americas? A scientist studying climate change? A fictional character (or two) from their favorite novel? Encourage them to get creative! • Tell students to spend a few minutes organizing their story plot on paper or in a digital document. Prompts: Are your characters traveling somewhere exciting? Are they searching the ocean for something? Are they lost? • Remind students to use realistic sounding dialogue. Based on their knowledge of the characters' historical background, interests, motivations, and personality, what might the conversation be about? What type of "voice" would each character have? How does their dialogue help move the story line forward? Wrap Up: • Have students share their animations with their classmates.
Lesson links/resources:	• https://www.scholastic.com/computerscience/index.html • https://www.scholastic.com/computerscience/pdf/Coding_Concepts.pdf • https://csfirst.withgoogle.com/c/cs-first/en/adventure-on-the-high-seas/overview.html • https://www.scholastic.com/computerscience/pdf/Student_Mag.pdf • https://csfirst.withgoogle.com/c/cs-first/en/adventure-on-the-high-seas/adventure-on-the-high-seas/high-seas-introduction.html

CS standards addressed:	**AP.2.2b** Students should use naming conventions to improve program readability. **AP.2.4a** Students should break down problems into subproblems, which can be further broken down to smaller parts. **AP.2.8a** Students will test programs by considering potential errors, such as what will happen if a user enters invalid input (e.g., negative numbers and zero instead of positive numbers). **AP.2.10** Document programs to make them easier to follow, test, and debug.
Time needed:	**Total time: 60 Mins** • Warm Up: 20 Mins • Activity: 30 Mins • Wrap Up: 10 Mins
Materials needed:	**Teacher:** • Computer • Projector/smartboard with sound **Students:** • Computer/tablet with internet access
Subject integrated:	ELA
Other standards addressed:	**RL.6.1** Cite textual evidence to support analysis of what the text says explicitly as well as inferences drawn from the text. **RL.6.4** Determine the meaning of words and phrases as they are used in a text, including figurative, connotative, and technical meanings. **SL.6.1c** Pose and respond to specific questions with elaboration and detail by making comments that contribute to the topic, text, or issue under discussion.
Vocabulary:	**Algorithm** – the steps you take to reach a goal or solve a problem. **Sequence** – doing things in a specific order. **Loop** – repeating a sequence. **Program** – using code to make a computer or other device perform a certain action. **Event** – an action that causes the program to begin or respond.
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at CS First

Illustrating with Coding

Lesson overview:  	**Purpose:** • The students will create coded animations for various scenes to illustrate a certain part of a story. Once each student, or group of students, complete their project, the entire story should be represented through the animations. **Lesson:** Warm Up: • Students will read a designated section of a story, provided by the teacher. Activity: • Students will plan out (sketch) their scene on paper. • Students will create a background to represent their part of the story. • Students will add Sprite(s) and animate them to represent their part of the story. • Students will add a text to their animation. Wrap Up: • Groups will share their scenes to the class to retell the entire story. The students can present their projects to the class, share their projects through Scratch, and/or post the link to a class learning management system.
Lesson links/resources:	• https://studio.code.org/s/csd3-2022/lessons/10 • https://studio.code.org/s/csd32022/lessons/10/levels/1?login_required=true
CS standards addressed:	**AP.2.2b** Students should use naming conventions to improve program readability. **AP.2.4a** Students should break down problems into subproblems, which can be further broken down to smaller parts. **AP.2.8a** Students will test programs by considering potential errors, such as what will happen if a user enters invalid input (e.g., negative numbers and zero instead of positive numbers). **AP.2.10** Document programs in order to make them easier to follow, test, and debug.
Time needed:	**Total time: 60 Mins** • Warm Up: 10 Mins • Activity: 35 Mins • Wrap Up: 15 Mins
Materials needed:	**Teacher:** • Computer • Projector/smartboard with sound **Students:** • Computer/tablet with internet access
Subject integrated:	ELA
Other standards addressed:	**RL.6.3** Describe how the plot of a literary text unfolds in a series of episodes as well as how the characters respond or change as the plot moves toward a resolution.

Vocabulary:	
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Code.org

Design the Device	
Lesson overview:  	**Purpose:** • Students will compare different forms of technology (laptops, phones, tablets) on the Open Classrooms sites - Design the Device. **Lesson:** Warm Up: • Assign students to work in pairs to analyze which features of a device of their choice that could be improved. Activity: • Student pairs will sketch out the design of their improved device and create a 1–2-minute advertisement or commercial that lists the benefits of the improved design. Wrap Up: • Pairs will present their designs and advertisements to the class.
Lesson links/resources:	• https://openclassrooms.com/en/courses/4555931-create-simple-prototypes-with-wireframes/4908316-design-for-the-device
CS standards addressed:	**CS.2.1** Recommend improvements to the design of computing devices based on an analysis of how users interact with the devices. **CS 2.1a** Students should make recommendations for existing devices (e.g. a laptop, phone, or tablet) or design their own components or interface (e.g. create their own controllers)
Time needed:	**Total time: 60 Mins** • Warm Up: 10 Mins • Activity: 30 Mins • Wrap Up: 20 Mins
Materials needed:	**Teacher:** • Computer • Projector/smartboard with sound **Students:** • Computer/tablet with internet access
Subject integrated:	ELA
Other standards addressed:	**RI.6.1** Cite textual evidence to support analysis of what the text says explicitly as well as inferences drawn from the text **RI.6.7** Integrate information presented in different media or formats as well as in words to develop a coherent understanding of a topic or issue **W.6.2** Write informative explanatory texts to examine a topic and convey ideas, concepts, and information through the selection, organization, and analysis of relevant content **W.6.2d** Use precise language and domain-specific vocabulary to inform about of explain the topic **W.6.9** Draw evidence from literary or informational texts to support analysis, reflection, and research
Vocabulary:	

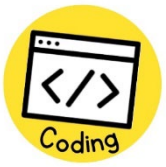
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Open Classrooms (when entering your DOB use DD/MM/YYYY).

Weather Design

Lesson overview: 	**Purpose:** • Students will learn how to collect data, interpret information presented in different media formats, and write an informational conclusion about how the weather various in different parts of the world. **Lesson:** Warm Up: • Students will collect weather data from a weather app or website for a week from specific areas of the globe. Activity: • Students will enter data into a spreadsheet (Microsoft Excel or Google Sheets) and create graphs to model the data. Wrap Up: • Students will then write reports to compare the data from the different areas to make inferences on how weather changes depending on time of year and location
Lesson links/resources:	• https://www.weatherbug.com/ • https://www.google.com/sheets/about/
CS standards addressed:	**CS.2.2** Design projects that combine hardware and software components to collect and exchange data **CS 2.2a** Students will design projects that use both hardware and software to collect and exchange data
Time needed:	**Total time: 60 Mins** • Warm Up: (Students research and record weather information for their location each day of the week) • Activity: 40 Mins • Wrap Up: 20 Mins
Materials needed:	**Teacher:** • Computer • Projector/smartboard with sound **Students:** • Computer/tablet with internet access
Subject integrated:	ELA
Other standards addressed:	**RI.6.7** Integrate information presented in different media or formats (e.g., visually quantitatively) as well as in words to develop a coherent understanding of a topic or issue **W.6.2** Write informative/explanatory texts to examine a topic and convey ideas, concepts, and information through the selection, organization, and analysis of relevant content. **W.6.2d** Use precise language and domain-specific vocabulary to inform about or explain the topic **W.6.2f** Provide a concluding statement or section that follows from the information or explanation presented. **W.6.6** Use technology, including the Internet, to produce and publish writing as well as to interact and collaborate with others; demonstrate sufficient command of keyboarding skills.

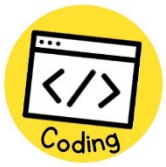
	W. 6.7 Conduct short research projects to answer a question, drawing on several sources and refocusing the inquiry when appropriate.
Vocabulary:	
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Open Classrooms This activity can be completed in Excel instead of Google Sheets.

Scratch Story Board

Lesson overview:  	**Purpose:** • Students will apply what they have learned about a narrative story and how a particular sentence, chapter, or scene fits into the overall structure of the text and contributes to the development of the theme, setting, or plot. **Lesson:** Warm Up: • Student(s) will create a 2–3-minute story as the narrator. Activity: • Student(s) will create their narrative story in Scratch. • Students will create a background in Scratch that is appropriate for their story. • Students will add a minimum of two Sprites and eight sets of text to their storyboard in Scratch. Wrap Up: • Students will justify how the background (setting) of their story supports the story. • Students will share their stories with their peers.
Lesson links/resources:	• https://scratch.mit.edu/
CS standards addressed:	**AP.2.3** Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. **AP.2.3a** Students will design and develop programs that combine control structures
Time needed:	**Total time: 60 Mins** • Warm Up: 15 Mins • Activity: 35 Mins • Wrap Up: 10 Mins
Materials needed:	**Teacher:** • Computer • Projector/smartboard with sound **Students:** • Computer/tablet with internet access
Subject integrated:	ELA
Other standards addressed:	**RL.6.5** Analyze how a particular sentence, chapter, scene, or stanza fits into the overall structure of a text and contributes to the development of the theme, setting or plot. **RL.6.7** Compare and contrast the experience of reading a story, drama, or poem to listening to or viewing an audio, video, or live version of the text, including contrasting what they "see" and "hear" when reading the text to what they perceive when they listen or watch. **W.6.6** Use technology, including the Internet, to produce and publish writing as well as to interact and collaborate with others; demonstrate sufficient command of keyboarding skills.
Vocabulary:	

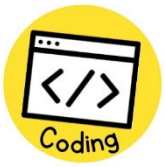
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Scratch

The World of Ozaria

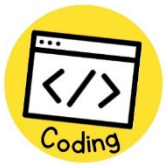
Lesson overview:  	**Purpose:** • Enter the world of Ozaria where you become a hero in an epic adventure. You must use the power of coding to defeat a darkness that has taken over the world! Along the way, you'll meet interesting characters and travel to different lands, practicing coding concepts like sequences, loops, debugging, and decomposition. In the end, you'll design a playable game that you can share with your friends! **Lesson:** Warm Up: • Introduction slides (provided) Activity: • Students will play through the suggested levels on the "Independent Practice Slide Wrap Up: • Use the check-in section as an opportunity for students to reflect on what they have learned in the lesson.
Lesson links/resources:	• https://docs.google.com/presentation/d/1KgFOg2tqbKEH8qNwIBdmK2QbHvTsxnW_Xo7LvjPsxwE/edit#slide=id.g8208a399ee_0_67 • https://hourofcode.com/codecombatozaria
CS standards addressed:	**CS.2.3** Systematically identify and fix problems with computing devices and their components. **AP 2.3** Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. **AP.2.7** Incorporate existing code, media, and libraries into original programs and give attribution.
Time needed:	**Total time: 60 Mins** • Warm Up: 15 Mins • Activity: 35 Mins • Wrap Up: 10 Mins
Materials needed:	**Teacher:** • Computer • Projector/smartboard with sound **Students:** • Computer/tablet with internet access
Subject integrated:	ELA
Other standards addressed:	**RI.6.2**: Determine a central idea of a text and how it is conveyed through particular details; provide a summary of the text distinct from personal opinions or judgments. **RI.6.7**: Integrate information presented in different media or formats (e.g., visually, quantitatively) as well as in words to develop a coherent understanding of a topic or issue. **RI.6.10**: By the end of the year, read and comprehend literary nonfiction in the grades 6-8 text complexity band proficiently, with scaffolding as needed at the high end of the range. **W.6.6**: Use technology, including the Internet, to produce and publish writing as well as to interact and collaborate with others; demonstrate sufficient

	command of keyboarding skills, **W.6.7**: Conduct short research projects to answer a question, drawing on several sources and refocusing the inquiry when appropriate.
Vocabulary:	
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at https://www.ozaria.com/

Coding in Python

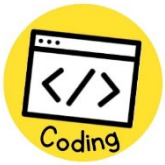
Lesson overview:  	**Purpose:** • In this lesson, students will focus on how events can be combined to create an interactive game. This lesson involves multiple events at once, some controlling the player sprite, another controlling the game environment, and another that awards the player points for doing the right action. Students will be able to use shapes, sprites variables, and events to create an object-drop game. **Lesson:** (Review the lesson before presenting to the students.) Follow the steps outlined in the lesson provided in the resources. Work as a whole group, small group, or independently and at a pace that is fitting for your group of students.
Lesson links/resources:	• http://guide.codesters.com/feed-the-fish • https://hourofcode.com/codestersfish
CS standards addressed:	**AP.2.2** Create clearly named variables that represent different data types and perform operations on their values. **AP.2.4** Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs **AP.2.5** Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs
Time needed:	**Total time: 60 Mins** (This lesson can be started and then students complete the various stages when time allows.)
Materials needed:	**Teacher:** • Computer • Projector/smartboard with sound **Students:** • Computer/tablet with internet access
Subject integrated:	ELA
Other standards addressed:	**RI.6.7** Integrate information presented in different media or formats (e.g., visually quantitatively) as well as in words to develop a coherent understanding of a topic or issue **W.6.6** Use technology, including the Internet, to produce and publish writing as well as to interact and collaborate with others; demonstrate sufficient command of keyboarding skills.
Vocabulary:	**sprite:** a character, shape, or text object that we add to our scene or game. **event:** a block of code that has a specific task and must receive a signal to run. **interval event:** a block of code that runs once every time a specified number of seconds has passed.
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Code.org

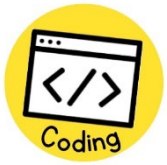
Recycle and Reuse: UN Sustainable Development Goals Project

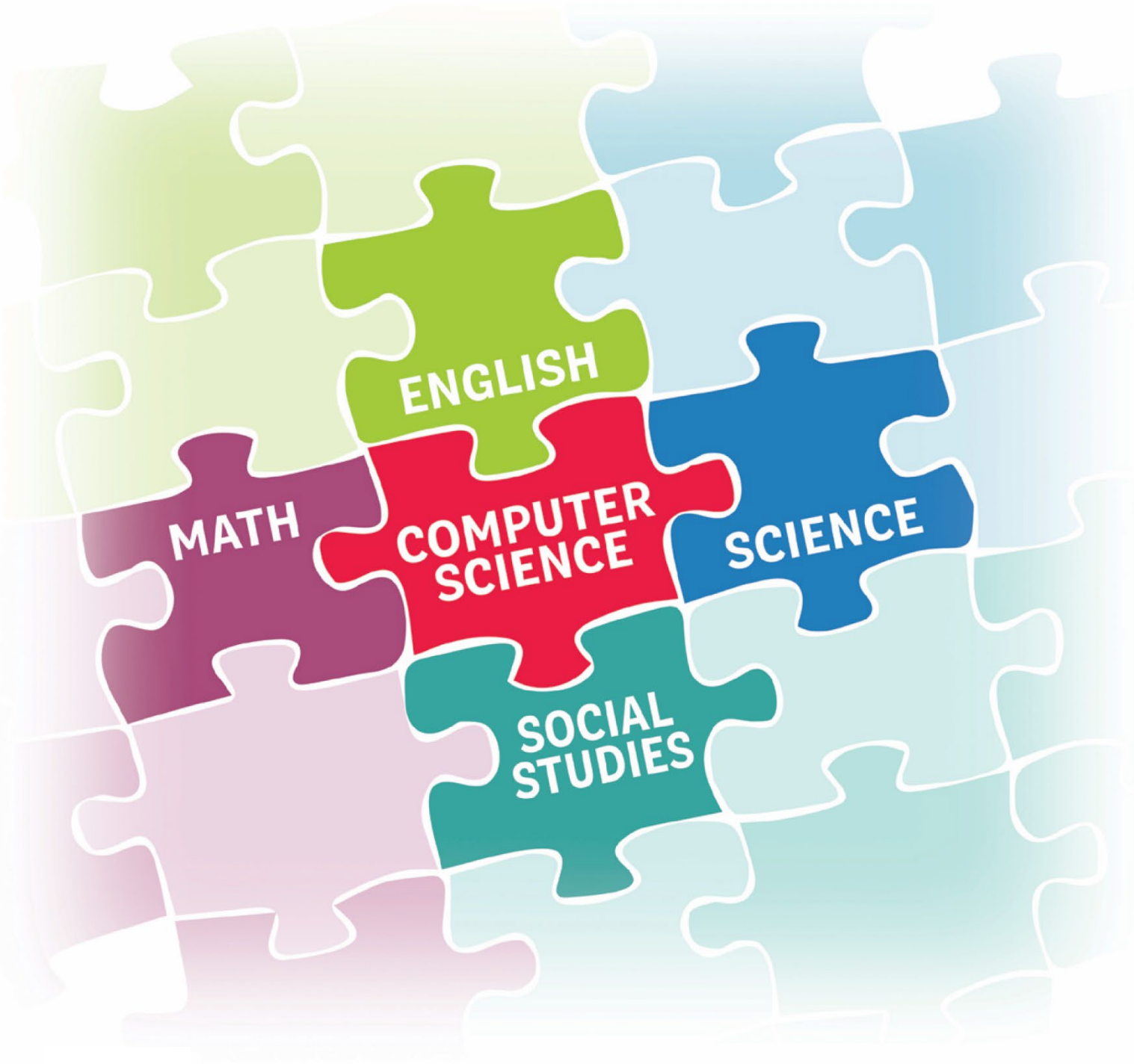
Lesson overview: 	**Purpose:** Students will apply coding concepts to create a Responsible Consumption and Production project and read about the Sustainable Development Goals. **Lesson:** Warm Up: • "17 Sustainable Development Goals" video • Ask students to read about *Goal 12: Responsible Consumption and Production* Activity: • Have students complete the "Responsible Consumption and Production" activity sheet. • Students will create a meaningful coding project about responsible consumption and production. Wrap Up: • Have students share their projects with the class, or classmates.
Lesson links/resources:	• https://www.tynker.com/hour-of-code/responsible-consumption-and-production-javascript-guide.pdf • https://www.tynker.com/hour-of-code/responsible-consumption-and-production-javascript
CS standards addressed:	**AP.2.4.** Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. **AP. 2.8** Systematically test and refine programs using a range of test cases.
Time needed:	**Total time: 60 Mins** (If students do not finish the complete assignment in class, they can work on the project at home.)
Materials needed:	**Teacher:** • Computer • Projector/smartboard with sound **Students:** • Computer/tablet with internet access
Subject integrated:	ELA
Other standards addressed:	**RI.6.4.** Determine the meaning of words and phrases as they are used in a text, including figurative, connotative, and technical meanings. **RI.6.7.** Integrate information presented in different media or formats (e.g. visually, quantitatively) as well as in words to develop a coherent understanding of a topic or issue. **SL.6.1.** Engage effectively in a range of collaborative discussions (one-on-one, in groups, and teacher-led) with diverse partners on grade 6 topics, texts, and issues, building on others' ideas and expressing their own clearly.
Vocabulary:	**Code** - the language that tells a computer what to do. **Sequence** -the order in which steps or events happen. **Function** - a set of known actions that the computer can perform. **Variable** - stores a value, such as a number or a string of text, at a named location.

	Argument - value passed into a function. **Parameter** - an extra piece of information that is passed into a function. **Loop** - an action that repeats one or more commands over and over
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at tynker.com

Tree Planting: UN Sustainable Development Goals Project

Lesson overview:  	**Purpose:** Part of the UN Sustainable Development Goals project, this Python project will have you create your own tree-planting game. **Lesson:** Warm Up: • "17 Sustainable Development Goals" video • Ask students to read about *Goal 15: Life on Land* Activity: • Have students complete the "Life on Land" activity sheet. • Students will create a meaningful coding project about life on land. Wrap Up: • Have students share their projects with the class, or classmates.
Lesson links/resources:	• https://www.tynker.com/hour-of-code/life-on-land-python-guide.pdf • https://www.tynker.com/hour-of-code/life-on-land-python
CS standards addressed:	**AP.2.4.** Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs. **AP.2.8.** Systematically test and refine programs using a range of test cases.
Time needed:	**Total time: 60 Mins** If students do not finish the complete assignment in class, they can work on the project at home.
Materials needed:	**Teacher:** • Computer • Projector/smartboard with sound **Students:** • Computer/tablet with internet access
Subject integrated:	ELA
Other standards addressed:	**RI.6.4**. Determine the meaning of words and phrases as they are used in a text, including figurative, connotative, and technical meanings. **RI.6.7.** Integrate information presented in different media or formats (e.g. visually, quantitatively) as well as in words to develop a coherent understanding of a topic or issue. **SL.6.1.** Engage effectively in a range of collaborative discussions (one-on-one, in groups, and teacher-led) with diverse partners on grade 6 topics, texts, and issues, building on others' ideas and expressing their own clearly.
Vocabulary:	**Code** - the language that tells a computer what to do. **Sequence** -the order in which steps or events happen. **Function** - a set of known actions that the computer can perform. **Variable** - stores a value, such as a number or a string of text, at a named location. **Argument** - value passed into a function. **Parameter** - an extra piece of information that is passed into a function. **Loop** - an action that repeats one or more commands over and over
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at tynker.com

Time Travel	
Lesson overview: 	Learn basic coding concepts to correct mysterious mishaps throughout history! Travel back in time to save the future in this free Hour of Code lesson in Minecraft: Education Edition. Players will choose their own adventure and connect with great innovators and inventions in science, architecture, music, engineering, and more. Follow the steps below to get started!
Lesson links/resources:	• https://education.minecraft.net/en-us/lessons/hour-of-code-2021-timecraft • https://education.minecraft.net/en-us/resources/hour-code-2021
CS standards addressed:	**AP.2.1**. Use flowcharts and/or pseudocode to address complex problems as algorithms. **AP.2.3**. Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. **AP.2.4.** Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
Time needed:	**Total time: 60 Mins** If students do not finish the complete assignment in class, they can work on the project at home.
Materials needed:	**Teacher:** • Computer • Projector/smartboard with sound **Students:** • Computer/tablet with internet access
Subject integrated:	ELA
Other standards addressed:	**RI.6.4** Determine or clarify the meaning of unknown and multiple meaning words and phrases based on grade 6 reading and content, choosing flexibly from a range of strategies **RI.6.7.** Integrate information presented in different media or formats (e.g., visually, quantitatively) as well as in words to develop a coherent understanding of a topic or issue.
Vocabulary:	**Computer Science** – the study of using the power of computers to solve problems.
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Code.org



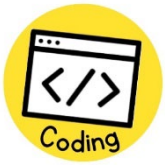
2023

Elementary Integration Guide

MATHEMATICS

MISSISSIPPI STATE UNIVERSITY™
CENTER FOR CYBER EDUCATION

Data Visualization

Lesson overview:  	**Purpose:** • This lesson builds off the concepts for bar charts and histograms that have already been introduced to students. Students will practice making conclusions from charts and learn to use the Data Visualizer in App Lab to create two different kinds of charts: a bar chart, and a histogram. This tool is designed to quickly connect students with real-world datasets and make it easy to create visualizations from data without learning how to navigate a more complex tool. They will also have access to several real-world datasets that they can use to create their charts. **Lesson:** Warm Up: This lesson is intended to enrich or reinforce existing lessons on bar charts and histograms. It is not meant to introduce new core content to students. Students should come into this lesson with the following prior knowledge and skills: • Review how to read a bar chart, understanding what the values on the x and y axes represent. • Know how to read a histogram, understanding that the range of values represent. Activity: • The lesson begins with a quick prompt to review the reasons charts are useful for looking at data. • Students then practice creating a bar chart in the Data Visualizer in App Lab. Then they learn how to make histograms for building charts in instances where bar charts may not be useful. • Students then have a chance to explore different real-world datasets and see how making charts can help communicate information about a topic they care about. Wrap Up: • Discuss how to determine when to use a bar chart and a histogram. • Discuss the consequences of using data that is not accurate.
Lesson links/resources:	• https://studio.code.org/s/explore-data-1-2021/lessons/1
CS standards addressed:	**DA.2.3a** Students will refine computational models by considering which data points are relevant, how data points relate to each other, and if the data is accurate.
Time needed:	**Total time: 45 Mins** • Warm Up: 10 Mins • Activity: 35 Mins • Wrap Up: 15 Mins
Materials needed:	**Teacher:** • Computer • Projector/smartboard with sound **Students:** • Computer/tablet with internet access

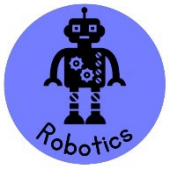
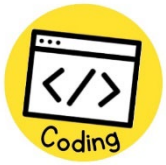
Subject integrated:	Math
Other standards addressed:	**6.SP.4** Display numerical data in plots on a number line, including dot plots, histograms, and box plots. **6.SP.5** Summarize numerical data sets in relation to their context, such as by: a. Reporting the number of observations. b. Describing the nature of the attribute under investigation, including how it was measured and its units of measurement. c. Giving quantitative measures of center (median and/or mean) and variability (interquartile range), as well as describing any overall pattern and any striking deviations from the overall pattern with reference to the context in which the data were gathered. d. Relating the choice of measures of center and variability to the shape of the data distribution and the context in which the data were gathered.
Vocabulary:	
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Code.org

Pizza Party Data

Lesson overview:   	**Purpose:** • Create a bar chart based on a set of data. • Explain why a set of data must be cleaned before a computer can use it. • Identify and remove irrelevant data from a data set. **Lesson:** Warm up: • The teacher will have students analyze the data displayed in the three representations from a survey. • Journal prompt - Which one of these makes it easiest for a human to make a decision about which pet is the most popular? Which one makes it easiest for a computer to make a decision? • Go over vocabulary. Activity: • Students will look at the structuring data activity guide using it to create a bar chart for the raw data and have them answer the reflection questions at the bottom of the guide. • Students will discuss with a partner their answers to the reflect questions then discuss them as a class. • Teacher will demonstrate the Pizza Party Data App with the class. • Students will use the Pizza Party Data App to clean up the data from their interpretations so that the computer can organize the information properly. Wrap up: • Students will answer the following questions in their journals: Can you think of a time in the past when you had data collected about you, maybe by filling out a form? What do you think were some strategies this form used to help make sure it collected clean data?
Lesson links/resources:	https://studio.code.org/s/csd5-2022/lessons/10#activity-1082078
CS standards addressed:	**DA.2.2** Collect data using computational tools and transform the data to make it more useful and reliable.
Time needed:	**Total time: 45 Mins** • Warm Up: 5 Mins • Activity: 35 Mins • Wrap Up: 5 Mins
Materials needed:	Teacher: • Computer • Projector/smartboard with sound • Structuring data slides (code.org) • Code.org account Students: • Computer/tablet with internet access • Structuring Data 2021- Activity guide (code.org) • Code.org account
Subject integrated:	Math

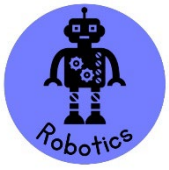
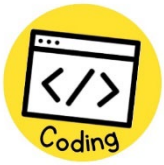
Other standards addressed:	**6.SP.5** Summarize numerical data sets in relation to their context
Vocabulary:	**Raw data** - The way information is first collected
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Code.org

Robotic Shape Trace

Lesson overview:  	**Purpose:** • Students will code a virtual robot to trace various shapes (i.e., triangles, quadrilaterals, and parallelograms). • Students will review the name/classification of shapes. **Lesson:** Warm up: • The teacher will review the characteristics of triangles and how to classify them by sides and angles. • The teacher will review the characteristics of quadrilaterals and parallelograms. Activity: This activity can be completed individually, or in small groups. • The student(s) will draw/select a shape from the VEX VR selection playground titled, "Shape Tracer." (This can be done individually or in small groups) • Students will list the characteristics and name of their shape on paper. • Students will code the virtual robot to trace the shape they selected. • Student(s) will calculate side lengths for their shape and find the area and perimeter. • Students who have the same shape, will record their time for the robot to trace their shape. The student(s) with the fastest robot (changing the velocity of the robot will affect the speed), will win the robot "competition." • Students will present their shape (characteristics, name, classification, etc.) to the class and demonstrate their robot tracing their shape. Wrap up: • Reflection and Sharing: Questions to discuss: what code command controls the speed of the robot? What angle degrees were needed to code the robot to trace triangles? Quadrilaterals? Why are those angle measurements the same or why are they different? Would there by a way of simplifying the total number of code lines – what is that called?
Lesson links/resources:	• VEX.code VR Activity • Virtual Robot Coding Activity (choose "select playground" in the top right corner, and select "Shape Trace" activity)
CS standards addressed:	**AP.2.3a** Students will design and develop programs that combine control structures. **AP.2.4a** Students should break down problems into subproblems, which can be further broken down to smaller parts. **AP.2.8a** Students will test programs by considering potential errors, such as what will happen if a user enters invalid input (e.g., negative numbers and zero instead of positive numbers).
Time needed:	**Total time: 60 Mins** • Warm Up: **10 Mins** • Activity: **30 Mins**

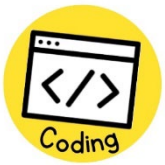
	• Wrap Up: **20 Mins**
Materials needed:	Teacher: • Computer • Projector/smartboard with sound Students: • Computer/tablet with internet access • VEX VR Coding Playground
Subject integrated:	Math
Other standards addressed:	**6.G.1** Find the area of right triangles, other triangles, special quadrilaterals, and polygons by composing into rectangles or decomposing into triangles and other shapes; apply these techniques in the context of solving real-world and mathematical problems.
Vocabulary:	
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at vr.vex.com

Number Line Animation

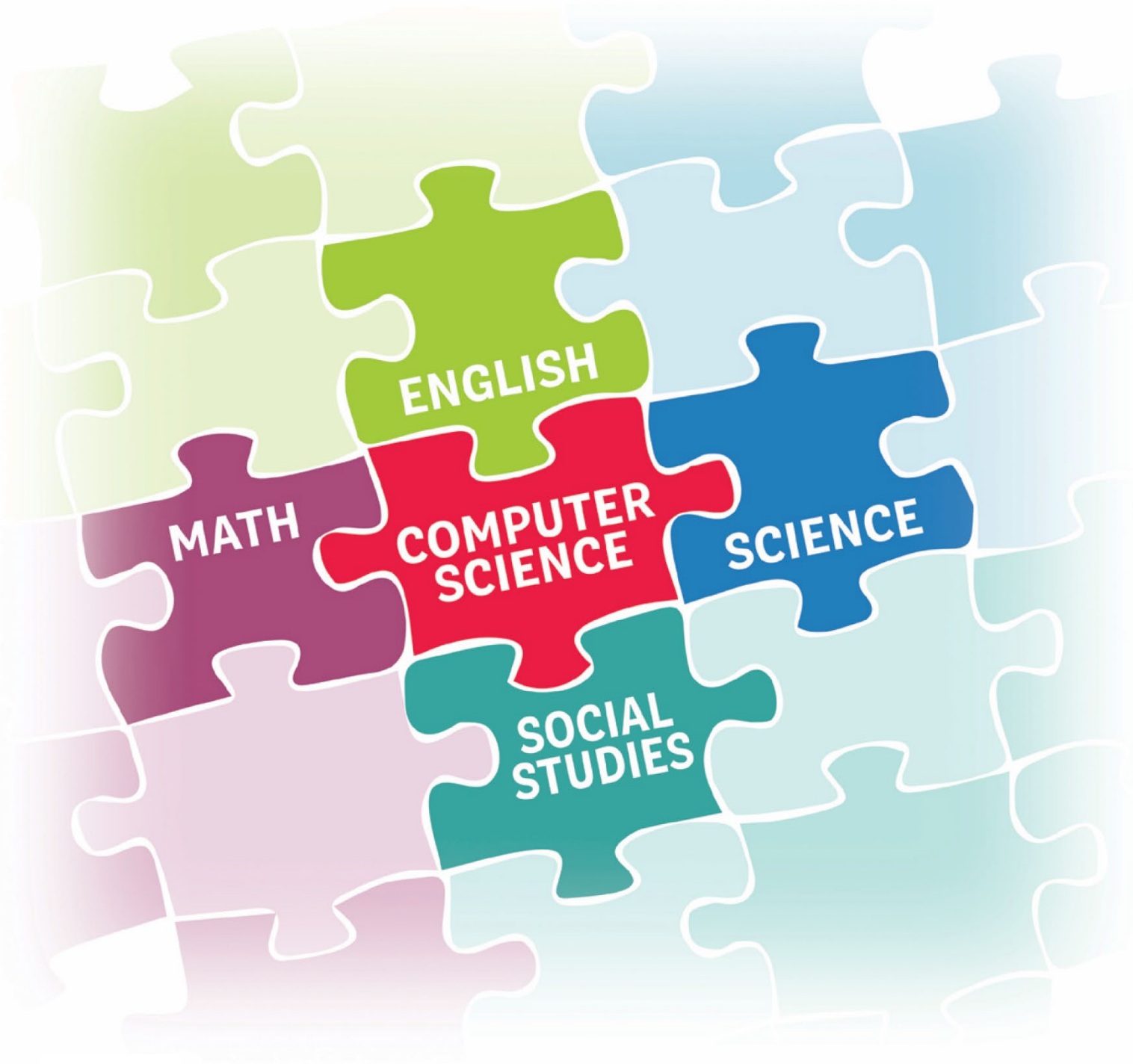
Lesson overview:   	**Purpose:** • Students will create an animation of a robot or Sprite moving on a number line to represent addition and subtraction of integers. • Students will create 3-5 addition or subtraction integer problems and code the robot or Sprite to move along the number line to represent the problem and solution. **Lesson using Scratch:** This activity can be completed individually, or small groups. Warm up: • The teacher will review the rules for adding and subtracting integers. Activity: • The teacher will facilitate how to insert the number line graphic into the Scratch stage. • The teacher can provide 3-5 adding or subtraction of integers problems or the students can create their own. • The student(s) will create and animate a Sprite to move along the number line to demonstrate the math problems. Wrap up: • The student(s) will share at least one animation of their math problems. **Lesson using a robot:** This activity can be completed individually, or small groups. Warm up: • The teacher will review the rules for adding and subtracting integers. Activity: • The teacher will provide a number line drawn on butcher paper or the floor. (The students can make their own number line if time allows) • The teacher will provide 3-5 adding or subtraction of integers problems. • The student(s) will program a robot to move along the number line to demonstrate each math problem. The students will write their program next to each math problem provided. • The students will create 1-2 math problems and share with the group next to them for programming and solving. Wrap up: • The student(s) will share at least one animation of their math problems.
Lesson links/resources:	• Adobe Stock Number Line • Scratch
CS standards addressed:	**AP.2.3a** Students will design and develop programs that combine control structures. **AP.2.4a** Students should break down problems into subproblems, which can be further broken down to smaller parts. **AP.2.8a** Students will test programs by considering potential errors, such as what will happen if a user enters invalid input (e.g., negative numbers and zero instead of positive numbers).
Time needed:	**Total time: 60 Mins** • Warm Up: **10 Mins**

	• Activity: **40 Mins** • Wrap Up: **10 Mins**
Materials needed:	Teacher: • Computer • Projector/smartboard with sound Students: • Computer/tablet with internet access
Subject integrated:	Math
Other standards addressed:	**6.NS.9c** Understand subtraction of integers as adding the additive inverse, p – q = p + (–q). Show that the distance between two integers on the number line is the absolute value of their difference and apply this principle in real-world contexts. **6.NS.9d** Apply properties of operations as strategies to add and subtract integers.
Vocabulary:	
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Scratch

Using Algorithms to Find Answers

Lesson overview:   	**Purpose:** • Students will read about how programmers use bits, bytes, and algorithms to design programs to solve problems. • Students will use an algorithm to solve a problem when designing using App Lab by Code.org • (Unplugged Option - Students will use an algorithm to build a paper airplane that flies the longest distance.) **Lesson:** Warm Up: • Discuss what students already know about algorithms? • Give commands for a simple algorithm for students to complete. (Examples are available in lesson resources). • Discuss what happens when the algorithm has a bug. Activity: • Read the NewsELA article The Math Behind Bits and Bytes and answer quiz questions. • Use App Lab by Code.org to design an app using an algorithm. Wrap Up: • Have students write, discuss, or reflect on why are algorithms important for computer programming?
Lesson links/resources:	• The Math Behind the Bits and Bytes • Unplugged Algorithm Practice Option: Code.org Real Life Algorithms Paper Airplane Worksheet • 7 Examples of Algorithms in Everyday Life for Students
CS standards addressed:	**AP.2.1** Use flowcharts and/or pseudocode to address complex problems as algorithms. **AP.2.1a** Students will use pseudocode and/or flowcharts to organize and sequence an algorithm that addresses a complex problem, even though they may not actually program the solutions.
Time needed:	**Total time: 60 Mins** • Warm Up: **10 Mins** • Activity: **40 Mins** • Wrap Up: **10 Mins**
Materials needed:	Teacher: • Computer • Projector/smartboard with sound Students: • Computer/tablet with internet access
Subject integrated:	ELA
Other standards addressed:	**6.EE.6** Use variables to represent numbers and write expressions when solving a real-world or mathematical problem; understand that a variable can represent an unknown number, or depending on the purpose at hand, any number in a specified set.
Vocabulary:	**Algorithm** – the steps you take to reach a goal or solve a problem. **Bit** - the smallest piece of information in a computer. **Byte -** a unit of digital information that consists of 8 bits.

Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at newsela



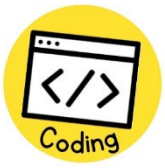
2023

Elementary Integration Guide

SCIENCE

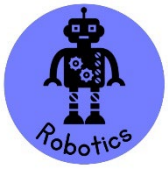
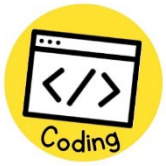
MISSISSIPPI STATE UNIVERSITY™
CENTER FOR CYBER EDUCATION

Living vs. Non-Living

Lesson overview:  	**Purpose:** • Living vs. Non-Living Clicker Game: Students will create an interactive clicker game that will help students learn about living and non-living things. • The purpose in the game will be for the students to create a clicker game where they will differentiate between living (biotic) and non-living (abiotic) things. {Students should include examples such as viruses and bacteria.} **Lesson:** Warm-Up: • As a class, watch the Scratch tutorial video to learn how to program a clicker game using Scratch. Activity: • Students will work in groups of 2 – 4 to create a clicker game, using Scratch, that will earn point by selecting the correct answer and deducting points based on incorrect answers. • Students must utilize various sprites, have at least one sound, have at least on background, and must assign point values to correct/incorrect answers. Wrap Up: • Have students share their projects with other peers to try. • Discuss the different ways students programmed their games. • Discuss ways to help students debug their code if needed.
Lesson links/resources:	• How to Make a Clicker Game in Scratch Tutorial • scratch.mit.edu
CS standards addressed:	**CS.2.2a** Students will design projects that use both hardware and software to collect and exchange data. **AP.2.7a** Students should use portions of code, algorithms, and/or digital media in their own programs and websites. **AP.2.7b** Students should test and refine programs using a range of test cases. **AP.2.10a** Students should provide documentation for end users that explains their artifacts and how they function. **AP.2.4a** Students should break down problems into subproblems, which can be further broken down to smaller parts. **AP.2.5a** Students will create procedures and/or functions that are used multiple times within a program to repeat groups of instructions. **AP.2.8a** Students will test programs by considering potential error, such as what will happen if a user enters invalid input (e.g. negative numbers and zero instead of positive number). **AP.2.9a** Students will work collaboratively in groups. **AP.2.9b** Students should assume predefined roles within their teams and manage the project workflow using structured timelines. **AP.2.9c** Students should give attribution to the original creators to acknowledge their contributions.
Time needed:	**Total time: 60 Mins** • Warm Up: **10 Mins** • Activity: **40 Mins** • Wrap Up: **10 Mins** (This activity can be started in about 10 minutes. Students can complete

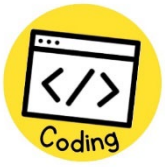
	their work as time allows.)
Materials needed:	Teacher: • Computer • Projector/smartboard with sound Students: • Computer/tablet with internet access
Subject integrated:	Science
Other standards addressed:	**L.6.1.1** Use argument supported by evidence in order to distinguish between living and non-living things, including viruses and bacteria.
Vocabulary:	
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Scratch

Organelle Robotics

Lesson overview:   	**Purpose:** • Students will program a robot, or another student, to move around a grid to answer review questions for organelle review. **Lesson:** Warm-Up: • Whole group discussion on how to give commands, or program, a robot to move around a square grid. (Commands may include forward, backwards, turn left, turn right) Activity: • The teacher will create a grid 10 x 10 grid. This grid can be taped out on a table or the floor for a codable robot or student, or it can be completed using the Organelle Robotics Activity sheet. • There should be a clearly defined start square at the top left corner of the grid. • The teacher will print pictures or words of various organelles to place into the grid. (Example grid is provided in the lesson resources.) • The teacher will call out review questions, and the students will write a program for the "robot" to follow to get to the correct answer. Wrap-up: • Student should swap their program with a classmate or group to test. • Discuss why students' answers may vary (the code for the robot to get from start to the correct answer can be written in many different ways) Example: Teacher asks: Which organelle is considered the powerhouse of the cell? The student may answer: Forward, forward, turn right, forward, forward, forward, forward, forward (it would land on mitochondria).
Lesson links/resources:	• Organelle Robotics Activity • Printable Arrows
CS standards addressed:	**AP.2.4a** Students should break down problems into subproblems, which can be further broken down to smaller parts. **AP.2.5a** Students will create procedures and/or functions that are used multiple times within a program to repeat groups of instructions. **AP.2.8a** Students will test programs by considering potential error, such as what will happen if a user enters invalid input (e.g., negative numbers and zero instead of positive number). **AP.2.9a** Students will work collaboratively in groups. **AP.2.9b** Students should assume predefined roles within their teams and manage the project workflow using structured timelines. **AP.2.9c** Students should give attribution to the original creators to acknowledge their contributions. **AP.2.10b** Students should incorporate comments in their product (comments in the code).
Time needed:	**Total time: 60 Mins** • Warm Up: **10 Mins** • Activity: **40 Mins** • Wrap Up: **10 Mins**
Materials needed:	**Teacher:**

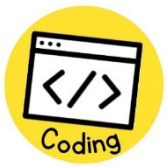
	• Robot Grid • Questions for Organelle Review **Students:** • Pencil and Paper • Robot/Student Robot
Subject integrated:	Science
Other standards addressed:	**L.6.1.3** Develop and use models to explain how specific cellular components (cell wall, cell membrane, nucleus, chloroplast, vacuole, and mitochondria) function together to support the life of prokaryotic and eukaryotic organisms to include plants, animals, fungi, protists, and bacteria (not to include biochemical function of cells or cell part).
Vocabulary:	
Notes:	

Scratch Relationships

Lesson overview:  	**Purpose:** • Students will work in groups of 2-4 to create a scratch program that will explain the following relationships: predation, competition, cooperation, and symbiotic relationships. **Lesson:** Warm-Up: • Review how to access and use Scratch • Review predation, competition, cooperation, and symbiotic relationships Activity: • Students will set the backdrop to at least one ecosystem located. • Students will assign appropriate sprites to each relationship scenario. (Ex: Predator - Shark / Prey - Fish) {The sprites should be appropriate to the ecosystem. No sharks in the desert.} • Students will create a script for each scenario to the sprites by using the "say" option under the **Looks** tab. (Ex: Shark will say "I am the predator. I eat fish." Fish will say, "I am the prey. Sharks eat me for food. The shark will benefit from our relationship, where I do not benefit." Warm-up: • Once groups have completed their scenarios, teacher/students can conduct a gallery walk to view each groups project and provide positive constructive feedback.
Lesson links/resources:	• scratch.mit.edu • csfirst.withgoogle.com • Introduction to Interactive Presentation • Build Your Own Presentation • https://scratch.mit.edu/projects/145540445
CS standards addressed:	**AP.2.4a** Students should break down problems into subproblems, which can be further broken down to smaller parts. **AP.2.5a** Students will create procedures and/or functions that are used multiple times within a program to repeat groups of instructions. **AP.2.6a** Students should begin to seek diverse perspectives throughout the design process to improve their computational artifacts. **AP.2.9a** Students will work collaboratively in groups. **AP.2.9b** Students should assume predefined roles within their teams and manage the project workflow using structured timelines. **AP.2.9c** Students should give attribution to the original creators to acknowledge their contributions.
Time needed:	**Total time: 60 Mins** • Warm Up: **10 Mins** • Activity: **40 Mins** • Wrap Up: **10 Mins** (This activity can be started in about 15 minutes. Students can complete their work as time allows.)
Materials needed:	**Teacher:** • Computer • Projector/smartboard with sound

	Students: • Computer/tablet with internet access
Subject integrated:	Science
Other standards addressed:	**L.6.3.4** Investigate organism interactions in a competitive or mutually beneficial relationships (predation, competition, cooperation, or symbiotic relationships).
Vocabulary:	
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Scratch and csfirst.withgoogle.com

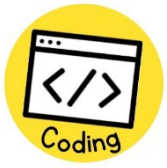
Energy in an Ecosystem: Part 1

Lesson overview:  	**Purpose:** • Students will work in groups to create a program that depicts energy flow through an ecosystem from producers to consumers to decomposers. **Lesson:** Warm Up: • As a whole group, watch the Sprite Lab: Introducing Sprite Lab Video Activity: • Students will log-on to code.org and open up a new Sprite Lab that they will rename to "Energy in an Ecosystem". • Students will utilize the **world** tool to set an appropriate background representing the ecosystem. (There is a larger selection of backgrounds if you click "set background to" and click the down arrow. At the bottom of the background menu students will see "more". They can click that and have access to more backgrounds. • Students will utilize the **sprites** tool to create the plants, animals, etc. (There is a larger selection of sprites if you click "Make new..." and click the down arrow. At the bottom of the sprite menu students will see "costumes" they will click that. Then they will click "New Costume" and select something from the costume's library, draw their own, or upload and image. • Students will utilize tools such as events, behaviors, loops, variables, text, etc. to generate an informative sprite lab that explains energy flow through an ecosystem and each sprite's relevance. Wrap Up: • Once students have completed their lab, have groups share their lab and provide/receive positive constructive feedback.
Lesson links/resources:	• https://studio.code.org/projects/spritelab/HkijY54cDjSNLegP4y9m8QqMj7ffyT-mU6DmlagkWXw/view • https://code.org/educate/spritelab
CS standards addressed:	**AP.2.4a** Students should break down problems into subproblems, which can be further broken down to smaller parts. **AP.2.5a** Students will create procedures and/or functions that are used multiple times within a program to repeat groups of instructions. **AP.2.6a** Students should begin to seek diverse perspectives throughout the design process to improve their computational artifacts. **AP.2.9a** Students will work collaboratively in groups. **AP.2.9b** Students should assume predefined roles within their teams and manage the project workflow using structured timelines. **AP.2.9c** Students should give attribution to the original creators to acknowledge their contributions.
Time needed:	**Total time: 60 Mins** • Warm Up: **10 Mins** • Activity: **40 Mins** • Wrap Up: **10 Mins** (This activity can be started in about 15 minutes. Students can complete their work as time allows.) This activity is a total of 120 minutes. The next lesson will be Energy in an Ecosystem: Part 2

Materials needed:	**Teacher:** • Computer • Projector/smartboard with sound **Students:** • Computer/tablet with internet access
Subject integrated:	Science
Other standards addressed:	**L.6.3.5** Develop and use food chains, webs, and pyramids to analyze how energy is transferred through an ecosystem from producers (autotrophs) to consumers (heterotrophs, including humans) to decomposers.
Vocabulary:	
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Code.org

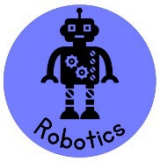
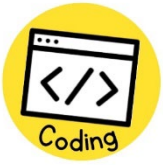
Energy in an Ecosystem: Part 2

(Continuation from "Energy in an Ecosystem: Part 1)

Lesson overview:  	**Purpose:** • Students will work in groups to create a program that depicts energy flow through an ecosystem from producers to consumers to decomposers. **Lesson:** Warm Up: • As a whole group, watch the Sprite Lab: Introducing Sprite Lab Video Activity: • Students will log-on to code.org and open up a new Sprite Lab that they will rename to "Energy in an Ecosystem". • Students will utilize the **world** tool to set an appropriate background representing the ecosystem. (There is a larger selection of backgrounds if you click "set background to" and click the down arrow. At the bottom of the background menu students will see "more". They can click that and have access to more backgrounds. • Students will utilize the **sprites** tool to create the plants, animals, etc. (There is a larger selection of sprites if you click "Make new..." and click the down arrow. At the bottom of the sprite menu students will see "costumes" they will click that. Then they will click "New Costume" and select something from the costume's library, draw their own, or upload and image. • Students will utilize tools such as events, behaviors, loops, variables, text, etc. to generate an informative sprite lab that explains energy flow through an ecosystem and each sprite's relevance. Wrap Up: • Once students have completed their lab, have groups share their lab and provide/receive positive constructive feedback.
Lesson links/resources:	• https://studio.code.org/projects/spritelab/HkijY54cDjSNLegP4y9m8QqMj7ffyT-mU6DmlagkWXw/view • https://code.org/educate/spritelab
CS standards addressed:	**AP.2.4a** Students should break down problems into subproblems, which can be further broken down to smaller parts. **AP.2.5a** Students will create procedures and/or functions that are used multiple times within a program to repeat groups of instructions. **AP.2.6a** Students should begin to seek diverse perspectives throughout the design process to improve their computational artifacts. **AP.2.9a** Students will work collaboratively in groups. **AP.2.9b** Students should assume predefined roles within their teams and manage the project workflow using structured timelines. **AP.2.9c** Students should give attribution to the original creators to acknowledge their contributions.
Time needed:	**Total time: 60 Mins** • Warm Up: **10 Mins** • Activity: **40 Mins** • Wrap Up: **10 Mins** (This activity can be started in about 15 minutes. Students can complete their work as time allows.)
Materials needed:	**Teacher:**

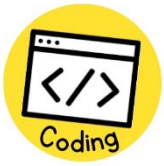
	• Computer • Projector/smartboard with sound **Students:** • Computer/tablet with internet access
Subject integrated:	Science
Other standards addressed:	**L.6.3.5** Develop and use food chains, webs, and pyramids to analyze how energy is transferred through an ecosystem from producers (autotrophs) to consumers (heterotrophs, including humans) to decomposers.
Vocabulary:	
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Code.org

Dichotomous Key Robotics

Lesson overview:  	**Purpose:** • Students will use review various specimens' and their groups. • Student will program a robot to move about the dichotomous key to group each specimen. **Lesson:** Warm-Up: • Watch the tutorial video as a whole group and discuss the assignment expectations. (The dichotomous key and activity are based on the Amoeba Sisters video.) Activity: • Students will determine which group each specimen belongs to by using the dichotomous key. • Once the students have completed the dichotomous key, they will program their "robot" to: ➢ Travel through the specimen's name first, then the group that it belongs with. ➢ For each specimen, the student will need to begin at the specified start area. Wrap-Up: • Students will need to record their programs into the chart provided. If time allows, have students share their projects.
Lesson links/resources:	• Dichotomous Key Robotics • Dichotomous Keys: Identification Achievement Unlocked
CS standards addressed:	**AP.2.4a** Students should break down problems into subproblems, which can be further broken down to smaller parts. **AP.2.5a** Students will create procedures and/or functions that are used multiple times within a program to repeat groups of instructions. **AP.2.8a** Students will test programs by considering potential error, such as what will happen if a user enters invalid input (e.g., negative numbers and zero instead of positive number). **AP.2.10b** Students should incorporate comments in their product (comments in the code).
Time needed:	**Total time: 60 Mins** • Warm Up 10 mins • Activity 40 mins • Wrap Up 10 mins
Materials needed:	**Teacher:** • Computer • Projector/smartboard with sound **Students:** • Computer/tablet with internet access
Subject integrated:	Science
Other standards addressed:	**L.6.4.2** Use classification methods to explore the diversity of organisms in kingdoms (animals, plants, fungi, protists, bacteria). Support claims that organisms have shared structural and behavioral characteristics.

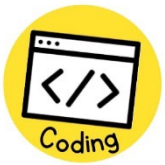
Vocabulary:	
Notes:	

Coding Kingdoms

Lesson overview:  	**Purpose:** • Students will apply their knowledge of kingdoms to develop a Scratch animation or clicker game. **Lesson:** Warm Up: • Review kingdoms with the class as a whole group. • Review the Scratch website and how to create an animation or clicker game. Activity: • Students will create an animation or clicker game using Scratch. • Students must include the following kingdoms: animal, plant, fungi, protist, eubacteria, and archaea. • Students will include the following information for each kingdom: characteristics, at least 1 "fun fact", and at least 2 examples. Wrap Up: • Students will submit their Scratch animation or clicker game to the teacher. • Students will share their creation with classmates to view or play. • If time permits, students can share their creation to the class as a whole group.
Lesson links/resources:	• The 6 Kingdoms of Classification in 3 minutes • How to Make a Clicker Game in Scratch • How to make a presentation in Scratch **Example Projects** • https://scratch.mit.edu/projects/220707555/ • https://scratch.mit.edu/projects/456053811/
CS standards addressed:	**AP.2.4a** Students should break down problems into subproblems, which can be further broken down to smaller parts. **AP.2.5a** Students will create procedures and/or functions that are used multiple times within a program to repeat groups of instructions.
Time needed:	**Total time: 60 Mins** • Warm Up: **10 Mins** • Activity: **40 Mins** • Wrap Up: **10 Mins** (This activity can be started in about 15 minutes. Students can complete their work as time allows.)
Materials needed:	**Teacher:** • Computer • Projector/smartboard with sound **Students:** • Computer/tablet with internet access
Subject integrated:	Science
Other standards addressed:	**L.6.4.2** Use classification methods to explore the diversity of organisms in kingdoms (animals, plants, fungi, protists, bacteria). Support claims that organisms have shared structural and behavioral characteristics.

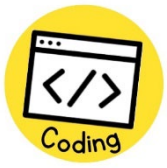
Vocabulary:	
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Scratch

Scratching the Surface of Newton's Laws

Lesson overview: 	**Purpose:** • Students will create an animation discussing all aspects of Newton's Law. **Lesson:** Warm-Up: • Review Newton Law's with the class as a whole group. • Review the Scratch website and how to create an animation. Activity: • Students will create a Scratch animation to provide information on Newton's Law • Each law should be defined, with an example. • Students should utilize various tools, such as Sprites, backdrops, events, sounds, etc. Wrap-Up: • Students will submit their Scratch animation or clicker game to the teacher. • Students will share their creation with classmates to view or play. • If time permits, students can share their creation to the class as a whole group.
Lesson links/resources:	• How to make a presentation in Scratch • scratch.mit.edu **Example Projects** • https://scratch.mit.edu/projects/181039882/
CS standards addressed:	**AP.2.4a** Students should break down problems into subproblems, which can be further broken down to smaller parts. **AP.2.5a** Students will create procedures and/or functions that are used multiple times within a program to repeat groups of instructions.
Time needed:	**Total time: 60 Mins** • Warm Up: **10 Mins** • Activity: **40 Mins** • Wrap Up: **10 Mins** (This activity can be started in about 15 minutes. Students can complete their work as time allows.)
Materials needed:	Teacher: • Computer • Projector/smartboard with sound Students: • Computer/tablet with internet access
Subject integrated:	Science
Other standards addressed:	**P.6.6.1** Use an engineering design process to create or improve safety devices (e.g., seat belts, car seats, helmets) by applying Newton's Laws of motion. Use an engineering design process to define the problem, design, construct, evaluate, and improve the safety device*
Vocabulary:	

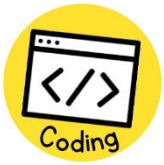
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Scratch

Energy and Motion

Lesson overview:  	**Purpose:** • Students will create a program that will provide information about potential energy, kinetic energy, and thermal energy, as well as how they are connected. **Lesson:** Warm Up: • Review potential energy, kinetic energy, and thermal energy, as well as how they are connected. • Watch the "Introduction to Sprite Lab" video. • This activity can be completed individually, or in small groups. Activity: • Students will log-on to code.org and open a new Sprite Lab that they will rename to "Energy and Motion". • Instruct students to watch the Sprite Lab: Introducing Sprite Lab Video. • Students will create a program that will provide information about the potential energy, kinetic energy, and thermal energy, as well as how they are connected. • Students will utilize the world tool to set an appropriate background. (There is a larger selection of backgrounds if you click "set background to" and click the down arrow. At the bottom of the background menu students will see "more". They can click that and have access to more backgrounds. • Students will utilize the sprites tool for characters and objects. (There is a larger selection of sprites if you click "Make new..." and click the down arrow. At the bottom of the sprite menu students will see "costumes" they will click that. Then they will click "New Costume" and select something from the costume's library, draw their own, or upload and image. • Students will utilize tools such as events, behaviors, loops, variables, text, etc. to generate an informative sprite lab that explains each type of energy, as well as their relationships to one another. Wrap Up: • Students will share their presentations with the class.
Lesson links/resources:	• Sprite Lab • Introduction to Sprite Lab Tutorial
CS standards addressed:	**AP.2.4a** Students should break down problems into subproblems, which can be further broken down to smaller parts. **AP.2.5a** Students will create procedures and/or functions that are used multiple times within a program to repeat groups of instructions.
Time needed:	**Total time: 60 Mins** • Warm Up: **10 Mins** • Activity: **40 Mins** • Wrap Up: **10 Mins** (This activity can be started in about 15 minutes. Students can complete their work as time allows.)
Materials needed:	Teacher: • Computer

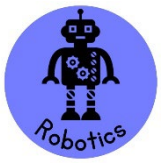
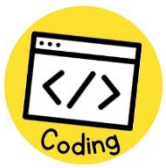
	• Projector/smartboard with sound Students: • Computer/tablet with internet access
Subject integrated:	Science
Other standards addressed:	**P.6.6.7** Determine the relationships between the concepts of potential, kinetic, and thermal energy.
Vocabulary:	
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Code.org

Solar System

Lesson overview:  	**Purpose:** • Students will create a Scratch program that will compare characteristics and movements of planets in our solar system. **Lesson:** Warm Up: • Review the solar system and the movements of planets. Activity: • Students will be placed into groups of 2-4 and assigned one of the following: Sun, Mercury, Venus, Mars, Earth, Jupitar, Saturn, Uranus, Neptune, or Pluto. • Students will research information on their assigned topic. • Each group should include the following information: the distance from Earth (if it is Earth, then distance from the sun), the diameter, how many moons it contains, period of revolution, period of rotation, and include at minimum 3 fun facts. • Each group should incorporate the following in their scratch program: sprites, backdrop, sound, and "say" functions. • Students should document where they found the information for their project. Wrap Up: • Students will share their presentations with the class.
Lesson links/resources:	• scratch.mit.edu • Introduction to an Interactive Presentation • Build Your Own Presentation **Example Projects** • https://scratch.mit.edu/projects/481187872 • https://scratch.mit.edu/projects/141674426
CS standards addressed:	**AP.2.4a** Students should break down problems into subproblems, which can be further broken down to smaller parts. **AP.2.5a** Students will create procedures and/or functions that are used multiple times within a program to repeat groups of instructions. **AP.2.9a** Students will work collaboratively in groups. **AP.2.9b** Students should assume predefined roles within their teams and manage the project workflow using structured timelines. **AP.2.9c** Students should give attribution to the original creators to acknowledge their contributions.
Time needed:	**Total time: 60 Mins** • Warm Up: **10 Mins** • Activity: **40 Mins** • Wrap Up: **10 Mins** (This activity can be started in about 15 minutes. Students can complete their work as time allows.)
Materials needed:	Teacher: • Computer • Projector/smartboard with sound Students: • Computer/tablet with internet access

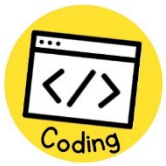
Subject integrated:	Science
Other standards addressed:	**E.6.8.4** Obtain and evaluate information to model and compare characteristics and movements of objects in the solar system (including planets, moons, asteroids, comets, and meteors.
Vocabulary:	
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Scratch and CS First - Google

Moon Phases

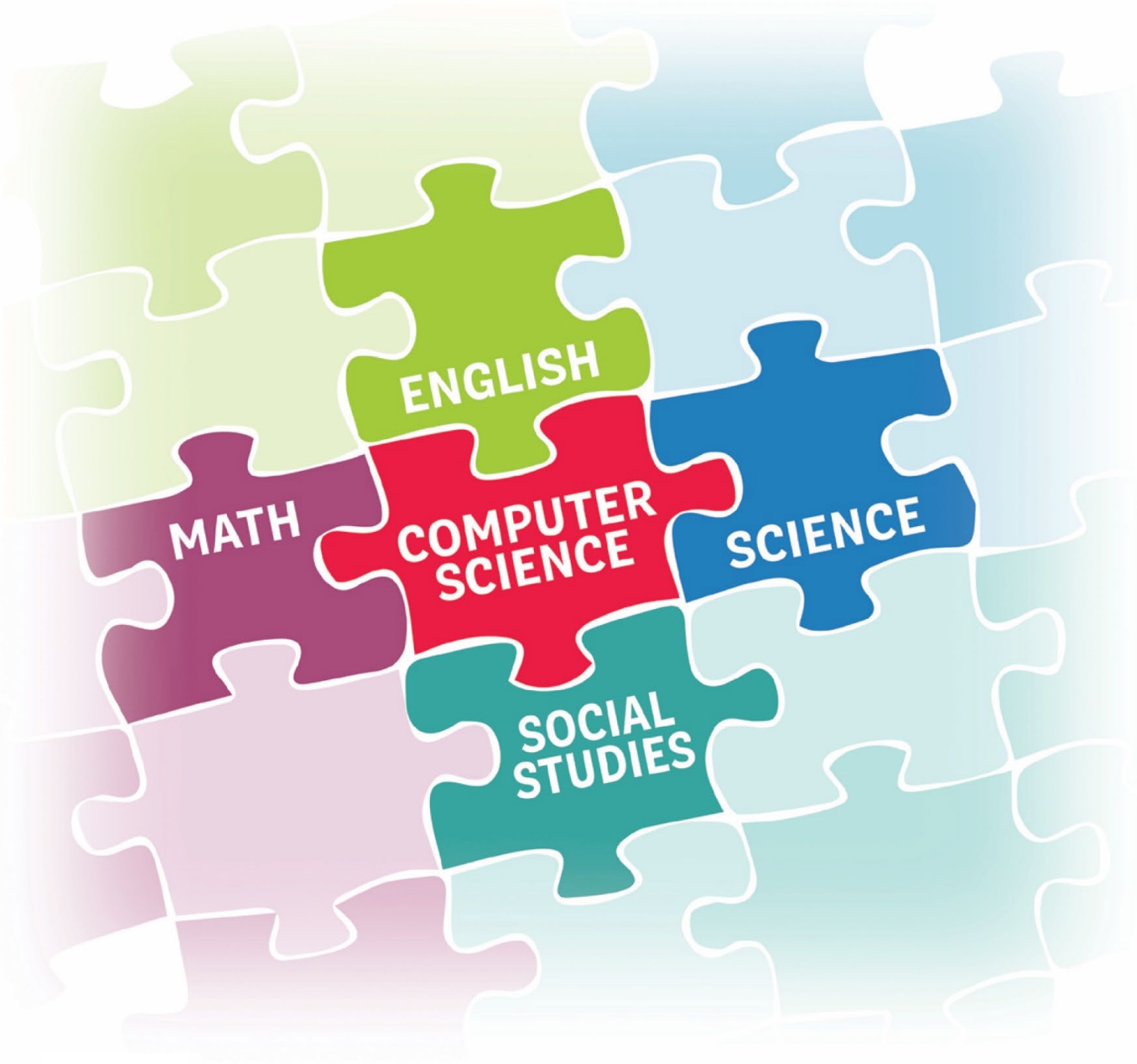
Lesson overview:  	**Purpose:** • Student will program a robot to reach correct destinations on a grid depicting/ labeling the phases of the moon. **Lesson:** Pre-Lesson Prep Work: • The teacher will create a grid 5 x 5 grid. (This grid can be taped out on a table or the floor for a codable robot, or it can be taped out on the floor to use a student as the robot.) There should be a clearly defined start square at the top, left corner of the grid. • The teacher will print pictures of the earth, moon phases, and sun to use in the activity. Warm Up: • Review moon phases with the class. • Students can complete this activity individually, or in small groups. Activity: This activity can be implemented in two ways: <u>Moon Phase Images</u> • To begin, you can have a blank grid that only has the start square, Earth, and sun depicted. • The teacher will show students an image of a moon phase, and the students must provide the program to place the image in the correct position. *Example: If the teacher showed an image of a full moon, then I would program my robot in the following manner:* *Forward, turn right, forward, forward.* <u>Labeling Moon Phases</u> • The teacher will have the moon phases placed in the correct positions on the grid surrounding the earth. • The teacher will call out a moon phase, and the students will have to program the robot to reach the correct image. *Example: If the teacher called out full moon, then I would program my robot in the following manner:* *Forward, turn right, forward, forward.* Wrap Up: • Students will submit their grids and programming to the teacher as a formative assessment.
Lesson links/resources:	• Moon Phases Robot Grid • Printable Arrows
CS standards addressed:	**AP.2.4a** Students should break down problems into subproblems, which can be further broken down to smaller parts. **AP.2.5a** Students will create procedures and/or functions that are used multiple times within a program to repeat groups of instructions.
Time needed:	**<u>Total time:</u> 60 Mins** • Warm Up: **10 Mins** • Activity: **40 Mins** • Wrap Up: **10 Mins** (This lesson will require a full 60 minutes to complete)

Materials needed:	Teacher: • Computer • Projector/smartboard with sound • Moon Phases Robot Grid • Printable Arrows Students: • Computer/tablet with internet access
Subject integrated:	Science
Other standards addressed:	**E.6.8.6** Design models representing motions within the Sun-Earth-Moon system to explain phenomena observed from the Earth's surface (positions of celestial bodies, day and year, moon phases, solar and lunar eclipses, and tides.)
Vocabulary:	
Notes:	

Cell Explorations

Lesson overview:  	**Purpose:** • Students will create an animation that reviews the parts of different cells. **Lesson:** Warm Up: • The teacher will introduce the lesson by reviewing with the students the components and their purposes of a certain type of cell (i.e. animal, plant, etc.). Activity: • Students will start a new program in Scratch. Students will load an image of an unlabeled cell diagram as the backdrop (see links in resources for options). • Students will program their sprite to travel to each part of the cell diagram when that part is clicked. • When a component is clicked, the component name will appear, and the Sprite will describe its function/purpose to the user. Cell component names will remain on the screen until all components have been clicked. *(Another option is to have students program their Sprite to travel to all the cell components in a particular order and at each, the name and purpose will appear. This option removes the user input of clicking but instead will run the cell diagram the exact same way every time the green flagged is clicked.)* • Students should provide instructions for users on how to use their program in the instructions box. Wrap Up: • Students will test out another student's cell program to ensure it operates as intended and provide a peer review. **Extension:** • *Have students alter their program to be interactive where the user can type in the name of a cell component and the sprite will go to that component on the screen and tell the purpose. This will require students to learn about user inputs, variables, and sensing.*
Lesson links/resources:	• scratch.mit.edu • Blank animal cell diagram: option 1 • Blank animal cell diagram: option 2 • Blank plant cell diagram • Blank bacteria cell diagram
CS standards addressed:	**AP.2.3a** Students will design and develop programs that combine control structures. **AP.2.8a** Students will test programs by considering potential errors, such as what will happen if a user enters invalid input. **AP.2.10a** Students should provide documentation for end users that explains their artifacts and how they function.
Time needed:	**Total time: 60 Mins** • Warm Up: **10 Mins** • Activity: **40 Mins** • Wrap Up: **10 Mins** • (This activity can be started in about 15 minutes. Students can complete their work as time allows.)

Materials needed:	Teacher: • Computer • Projector/smartboard with sound • Blank cell template Students: • Computer/tablet with internet access
Subject integrated:	Science
Other standards addressed:	**L.6.1.3** Develop and use models to explain how specific cellular components (cell wall, cell membrane, nucleus, chloroplast, vacuole, and mitochondria) function together to support the life of prokaryotic and eukaryotic organisms to include plants, animals, fungi, protists, and bacteria (not to include biochemical function of cells or cell part).
Vocabulary:	
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Scratch



2023

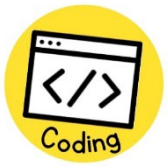
Elementary Integration Guide

SOCIAL STUDIES

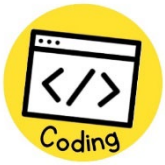
MISSISSIPPI STATE UNIVERSITY™
CENTER FOR CYBER EDUCATION

Native American Star Quilts - (Code.org) **Star Quilts Module: Lesson 1**	
Lesson overview:  	**Purpose:** • In this Unplugged lesson, students will be introduced to Native American Star Quilts and their significance. **Lesson:** Warm Up: • "Star Quilt" video Activity: • Graphic Organizer and coloring utensils • Students will make a 6-pointed star shape on paper. One of the main goals is to discuss the patterns within the shape. These ideas will be key for the online coding portions of the activity later. Wrap-up: • Debrief
Lesson links/resources:	• Code.org Native American Star Quilts Lesson 1
CS standards addressed:	**AP.2.3a** Students will design and develop programs that combine control structures. For example, when programming an interactive story, students could use a compound conditional within a loop to unlock a door only if a character has a key AND is touching the door. **AP.2.4a** Students should break down problems into subproblems, which can be further broken down to smaller parts. For example, animations can be decomposed into multiple scenes, which can be developed independently.
Time needed:	**Total time: 45 Mins** • Warm-Up: 10 minutes • Main Activity: 25 minutes • Wrap-Up: 10 minutes
Materials needed:	Teacher: • Computer • Projector/smartboard with sound • Star Quilt Article - Resource • Star Quilts - Slides (Download) • Star Quilts - Video Students: • Star Quilts Worksheet - Handout • Virtual Pattern Blocks (Optional) - Resource
Subject integrated:	Social Studies, Math
Other standards addressed:	**H.6.1** Explain the characteristics and development of culture. **H.6.1.1** Describe the major aspects of culture (religion/belief systems, language, ethnicity, institutions, technology, art, architecture, dress, foods, traditions, etc.). **H.6.1.2** Explain how culture changes as it is passed from one generation to the next. **H.6.1.3** Identify major culture regions of the world and explain how the characteristics of each set it apart from the others.

	6.G.1 Find the area of right triangles, other triangles, special quadrilaterals, and polygons by composing into rectangles or decomposing into triangles and other shapes; apply these techniques in the context of solving real-world and mathematical problems.
Vocabulary:	**Pattern** - Something that happens or appears in a regular and repeated way
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Code.org

Getting Started with The Artist - Code.org **Star Quilts Module: Lesson 2**	
Lesson overview:  	**Purpose:** • This skill-building lesson provides an opportunity for students to become familiar with the Artist Tool and to build important coding skills. Students learn how the stamping tool works, how to turn x degrees in an angle, and how to use loops. • Students will create a simple star shape. By the end of the levels, students will manipulate angle measurements to see the effect on the number of points on the star. Computers allow for accuracy and precision in studying shapes and patterns because images are generated in a matter of seconds instead of relying on hand-drawn shapes. **Lesson:** Warm Up: • What patterns do you see, what patterns can you think exist in the world around us and discuss vocabulary. Activity: • Students go through skill building lessons to create patterns. Wrap Up: • Debrief
Lesson links/resources:	• Code.org Native American Star Quilts Lesson 2
CS standards addressed:	**AP.2.3a** Students will design and develop programs that combine control structures. For example, when programming an interactive story, students could use a compound conditional within a loop to unlock a door only if a character has a key AND is touching the door. **AP.2.4a** Students should break down problems into subproblems, which can be further broken down to smaller parts. For example, animations can be decomposed into multiple scenes, which can be developed independently.
Time needed:	**Total time: 50 Mins** • Warm Up: 10 minutes • Main Activity: 30 minutes • Wrap Up: 10 minutes
Materials needed:	Teacher: • Computer • Projector/smartboard with sound • Star Quilts - Slides (Download) Students: • Computer/tablet with internet access • Virtual Pattern Blocks (Optional) - Resource
Subject integrated:	Social Studies, Math
Other standards addressed:	**H.6.1** Explain the characteristics and development of culture. **H.6.1.1** Describe the major aspects of culture (religion/belief systems, language, ethnicity, institutions, technology, art, architecture, dress, foods, traditions, etc.).

	H.6.1.2 Explain how culture changes as it is passed from one generation to the next. **H.6.1.3** Identify major culture regions of the world and explain how the characteristics of each set it apart from the others. **6.G.1** Find the area of right triangles, other triangles, special quadrilaterals, and polygons by composing into rectangles or decomposing into triangles and other shapes; apply these techniques in the context of solving real-world and mathematical problems.
Vocabulary:	**Artist** - You can write code to make him draw almost anything. **Loop** - A sequence of code that is repeated. **Point** - One part of the star shape
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Code.org

Code Your Star Quilt **Star Quilts Module: Lesson 3**	
Lesson overview:  	**Purpose:** • In the final lesson for this activity, students are ready to build their own Star Quilt. • Students will build one point of the star and then loop their code to end up with an 8 point traditional Star Quilt. Star Quilts are meant to be gifted so students can share their project with someone special. **Lesson:** Warm Up: • What is the difference between a 6-pointed star shape and an 8-pointed Star Quilt? Activity: • Star Quilts – Skill Building Wrap Up: • Reflection and Sharing
Lesson links/resources:	• Code.org Native American Star Quilts Lesson 3
CS standards addressed:	**AP.2.3a** Students will design and develop programs that combine control structures. For example, when programming an interactive story, students could use a compound conditional within a loop to unlock a door only if a character has a key AND is touching the door. **AP.2.4a** Students should break down problems into subproblems, which can be further broken down to smaller parts. For example, animations can be decomposed into multiple scenes, which can be developed independently.
Time needed:	**Total time: 50 Mins** • Warm-Up: 10 minutes • Main Activity: 30 minutes • Wrap-Up: 10 minutes
Materials needed:	Teacher: • Computer • Projector/smartboard with sound • Star Quilts - Slides (Download) • Virtual Pattern Blocks - Resource Students: • Computer/tablet with internet access
Subject integrated:	Social Studies, Math
Other standards addressed:	**H.6.1** Explain the characteristics and development of culture. **H.6.1.1** Describe the major aspects of culture (religion/belief systems, language, ethnicity, institutions, technology, art, architecture, dress, foods, traditions, etc.). **H.6.1.2** Explain how culture changes as it is passed from one generation to the next. **H.6.1.3** Identify major culture regions of the world and explain how the characteristics of each set it apart from the others.

	6.G.1 Find the area of right triangles, other triangles, special quadrilaterals, and polygons by composing into rectangles or decomposing into triangles and other shapes; apply these techniques in the context of solving real-world and mathematical problems.
Vocabulary:	**Comments** - Notes to yourself that explain a section of code. They do not alter the program but help keep your code in order.
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Code.org

Designing for Accessibility

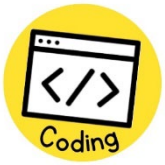
Lesson overview: 	**Purpose:** • Through learning about accessibility, students recognize the impacts of computing beyond their own lives. Accessibility might not seem like a relevant CS topic but creating technology that is accessible for underserved users helps make tech better for everyone else as well. **Lesson:** Warm Up: • What is an app? Why have apps become so popular? Activity: • Designing for Accessibility Wrap Up: • Reflection and sharing
Lesson links/resources:	• Code.org Designing for Accessibility
CS standards addressed:	**CS 2.1** Recommend improvements to the design of computing devices based on an analysis of how users interact with the devices. **CS2.1.a** Make recommendations for existing devices (e.g., a laptop, phone, or tablet) or design their own components or interface (e.g., create their own controllers). T
Time needed:	**Total time: 50 Mins** • Warm Up (10 min) • Activity (35 min) • Wrap Up (5 min)
Materials needed:	Teacher: • Computer • Projector/smartboard with sound • Types of Disabilities - Resource Students: • Designing for Accessibility - Slides • crayons/pencils • Journals
Subject integrated:	Social Studies
Other standards addressed:	**G.6.7** Compare and contrast ways that humans and the physical environment are impacted by the extraction of resources. **G.6.7.3** Describe examples of how the physical environment provides opportunities and constraints for human activities.
Vocabulary:	**Accessibility** – the extent to which a service, device, or product is usable by as many individuals as possible, including people who have disabilities. **Empathy** – being able to know how someone is feeling, even when you aren't in the same situation.
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Code.org

Finding Credible News:
How do we find credible information on the internet?

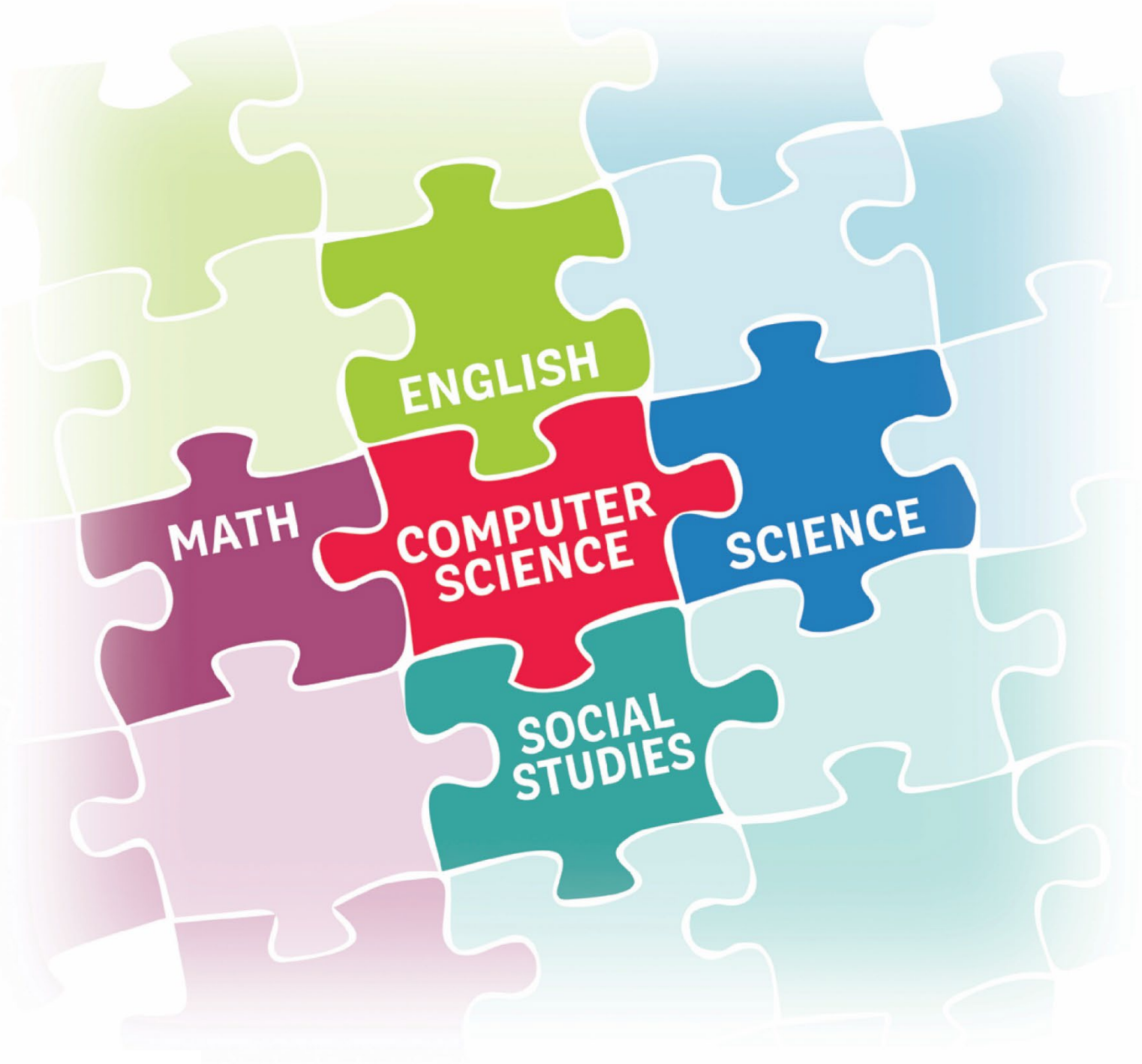
Lesson overview: 	**Purpose:** • Help students dig into why and how false information ends up online in the first place, and then practice evaluating the credibility of what they're finding online. **Lesson:** This activity can be completed individually, or in small groups. Warm Up: • "Tricky Wiki" Analyze: • "News or Fake News?" Wrap Up: • "Fighting Fake"
Lesson links/resources:	• Common Sense Education: Finding Credible News
CS standards addressed:	**IC2.4** Describe tradeoffs between allowing information to be public and keeping information private and secure. **IC.2.1a** Students should consider current events related to broad ideas, including privacy, communication, and automation.
Time needed:	**Total time: 45 minutes** • Warm Up: 10 mins • Analyze: 25 mins • Wrap Up: 10 mins
Materials needed:	Teacher: • Computer • Projector/smartboard with sound • Lesson Slides • News or Fake News? • Example #3 Article Handout (Student Version) • Lesson Quiz Students: • Computer/tablet with internet access Take Home Resources: • Family Activity • Family Tips • Family Engagement Resources
Subject integrated:	Social Studies
Other standards addressed:	**CI.6.2** Examine the challenges of civic engagement in the contemporary world. **CI.6.2.1** Compare the positive and negative impacts of changing technologies on expanding the role of citizens throughout the world and the challenges posed by new media sources to obtaining reliable information upon which to make decisions.
Vocabulary:	**Credible** – able to be believed.

	Corroboration – an additional source that confirms or supports a news story, article, or piece of information. **Bias** – showing a strong opinion or preference for or against something or someone. **Evaluate** – to carefully examine something to figure out its value.
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at https://www.commonsense.org/education

Cybersecurity - Simple Encryption

Lesson overview: 	**Purpose:** • "Encryption" is a process for transforming a message so that the original is "hidden" from anyone who is not the intended recipient. Encryption is not just for the military and spies anymore. We use encryption every day on the Internet, primarily to conduct commercial transactions, and without it our economy might grind to a halt. • This lesson gives students a first taste of the kind of thinking that goes into encrypting messages in the face of computational tools. Computational tools dramatically increase the strength and complexity of the algorithms we use to encrypt information, but these same tools also increase our ability to crack an encryption. Developing strong encryption relies on knowledge of problems that are "hard" for computers to solve and using that knowledge to encrypt messages. As a resource, you may wish to read all of Chapter 5 of Blown to Bits. It provides social context which you may want to bring to your classroom. **Lesson:** Warm Up: • Classic Encryption – Caesar Cipher Activity: • Crack a Caesar Cipher • Crack a Random Substitution Cipher Wrap Up: • "Encryption and Public Keys" video • Discussion • Career discussion
Lesson links/resources:	• Code.org Cybersecurity - Simple Encryption
CS standards addressed:	**NI.2.3a** Students should encode and decode messages using a variety of encryption methods, and they should understand the different levels of complexity used to hide or secure information.
Time needed:	**Total time:** • Warm Up: 10 mins • Main Activity: 35 mins • Wrap Up: 15 mins
Materials needed:	Teacher: • Computer • Projector/smartboard with sound Students: • Computer/tablet with internet access
Subject integrated:	Social Studies
Other standards addressed:	**H.6.1.1** Describe the major aspects of culture (religion/belief systems, language, ethnicity, institutions, technology, art, architecture, dress, foods, traditions, etc.).

Vocabulary:	**Caesar Cipher** - a technique for encryption that shifts the alphabet by some number of characters. **Cipher** - the generic term for a technique (or algorithm) that performs encryption. **Cracking encryption** - When you attempt to decode a secret message without knowing all the specifics of the cipher, you are trying to "crack" the encryption. **Decryption** - a process that reverses encryption, taking a secret message and reproducing the original plain text. **Encryption** - a process of encoding messages to keep them secret, so only "authorized" parties can read it. **Random Substitution Cipher** - an encryption technique that maps each letter of the alphabet to a randomly chosen other letters of the alphabet.
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Code.org



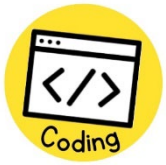
2023

Elementary Integration Guide

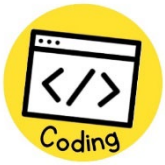
SOCIAL EMOTIONAL LEARNING

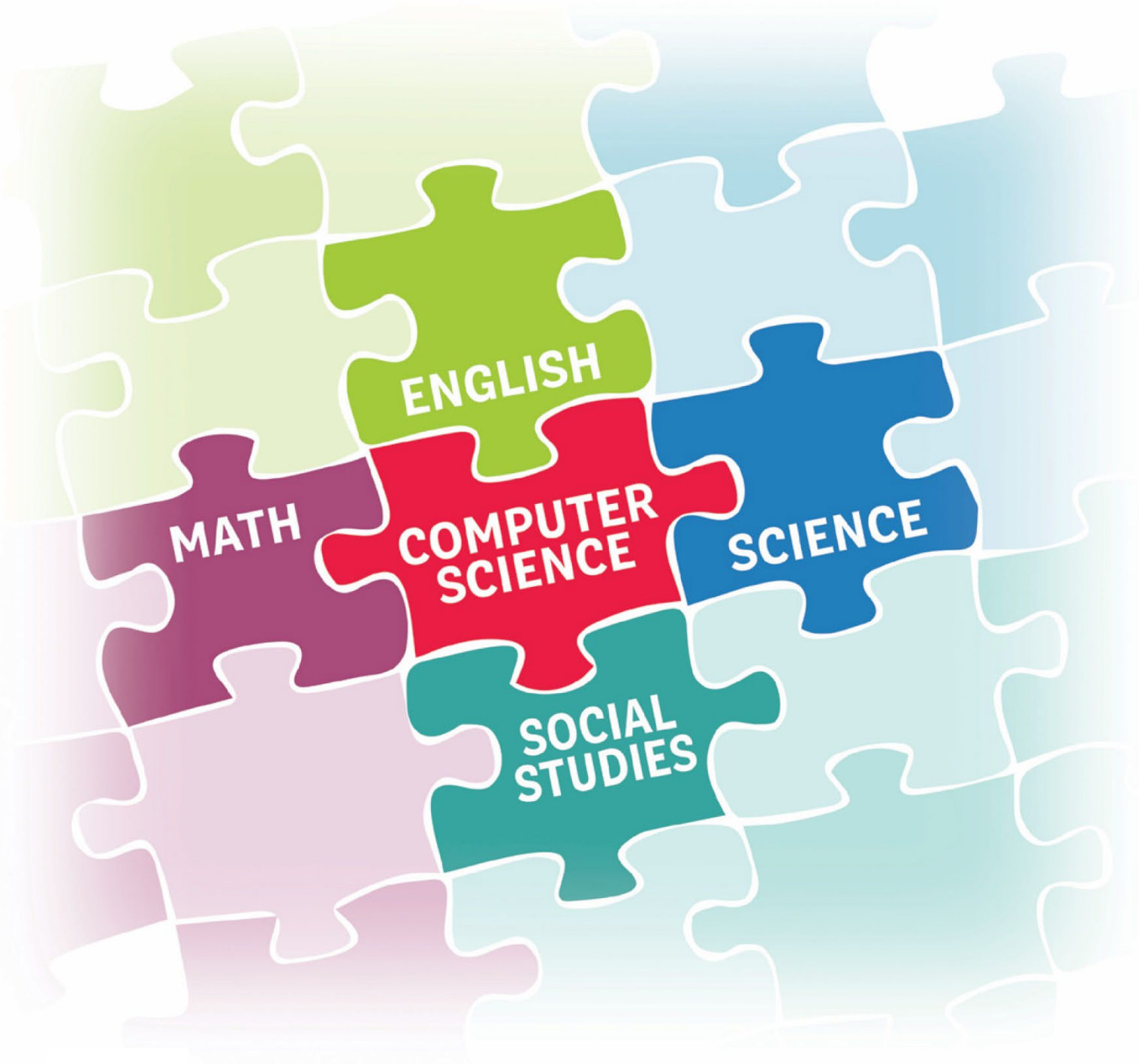
MISSISSIPPI STATE UNIVERSITY™
CENTER FOR CYBER EDUCATION

"All About Me" Animation

Lesson overview:  	**Purpose:** • Students will create an animation on Scratch describing their talents, skills, and other positive attributes. **Lesson:** Warm Up: • What are talents? What are skills? What are positive attributes? Activity: • Students will make a list, on paper, of their own talents, skills, and other positive attributes. • Students will use Scratch to insert a Sprite and animate a list or story of their talents, skills, and other positive attributes. The students can build their animation to best represent them (i.e., Sprite choice, background, etc.). Wrap Up: • Reflection and sharing with classmates.
Lesson links/resources:	• https://scratch.mit.edu/
CS standards addressed:	**AP.2.7a** Students should use portions of code, algorithms, and/or digital media in their own programs and websites
Time needed:	**Total time: 60 Mins** • Warm Up: 10 Mins • Activity: 40 Mins • Wrap up: 10 Mins
Materials needed:	Teacher: • Teacher/Student Scratch accounts (optional) Students: • Paper and pencil • Internet connected device
Subject integrated:	SEL
Other standards addressed:	**1B.** Develop an accurate perception of oneself (i.e., beliefs, values, skills, talents, and interests) **1B.8** Identify positive attributes and qualities about oneself including talents, interests, physical characteristics, etc. **1B.9** Describe characteristics that are important to oneself (i.e., loyalty, honesty, etc.) **1B.10** Describe how one's personal qualities, interest, beliefs and academic/career goals impact decision making.
Vocabulary:	**Talents** – The natural ability to do something better than most people. **Skills** – ability that comes from training or practice. **Positive Attributes** – qualities, character traits, and strengths that are considered good or help us in some way.
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Scratch

Conflict Resolution

Lesson overview: 	**Purpose:** • Students will learn about conflict and conflict resolution. Students will create a conflict and proper resolution in a Scratch animation. **Lesson:** Warm up: • Discuss the meaning of conflict and conflict resolution. Provide scenarios for the students to understand. • This activity can be completed individually, or in pairs. Activity: • Student(s) to work to create a Scratch animation depicting a conflict (provided by the teacher) and the appropriate resolution. Wrap Up: • Have students share their stories with classmates.
Lesson links/resources:	• https://scratch.mit.edu/
CS standards addressed:	**AP.2.7a** Students should use portions of code, algorithms, and/or digital media in their own programs and websites
Time needed:	**Total time: 60 Mins** • Warm Up: 10 mins • Activity: 40 mins • Reflection and sharing: 10 mins
Materials needed:	Teacher: • Teacher/Student Scratch accounts (optional) Students: • Paper and pencil • Internet connected device
Subject integrated:	SEL
Other standards addressed:	**4C.** Demonstrate the ability to successfully manage and resolve conflict in relation. **4C.11.** Identify behaviors that create conflict (e.g., spreading rumors, inappropriate posts or texts on social media, wrongful accusations, and insult or put downs). **4C.12.** Apply conflict resolution skills in order to de-escalate, defuse and resolve a conflict. 13. Determine strategies for avoiding or resolving conflicts related to destructive peer pressure.
Vocabulary:	**Conflict** – a challenge to the way a person thinks or behaves. **Conflict Resolution** – the process that two or more people use to agree on a solution to a problem. **De-escalate** – to lessen they intensity or anger in a conflict.
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Scratch



2023

Elementary Integration Guide

DIGITAL CITIZENSHIP

MISSISSIPPI STATE UNIVERSITY™
CENTER FOR CYBER EDUCATION

Finding Balance in a Digital World

Lesson overview:  	**Purpose:** • Reflect on their common online and offline activities. • Identify ways to "unplug" to maintain balance between online and offline activities. • Use the Digital Habits Checkup routine to create a personal challenge to achieve more media balance. **Lesson:** Warm Up: • "It's a Digital World!" Reflect: • "My Online and Offline Life" Apply: • "Balance It Out" Wrap Up: • "Finding Media Balance"
Lesson links/resources:	• Common Sense Education: Finding Balance in a Digital World
CS standards addressed:	**IC.2.4** Describe tradeoffs between allowing information to be public and keeping information private and secure.
Time needed:	**Total time: 60 Mins** • Warm Up: 10 mins • Reflect: 10 mins • Apply: 15 mins • Wrap Up: 15 mins • Quiz: 10 mins
Materials needed:	Teacher: • Computer • Projector/smartboard with sound Students: • Computer/tablet with internet access
Subject integrated:	ELA
Other standards addressed:	**W.6.4** Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience. **W.6.8** Gather relevant information from multiple print and digital sources; assess the credibility of each source; and quote or paraphrase the data and conclusions of others while avoiding plagiarism and providing basic bibliographic information for sources.
Vocabulary:	**digital media** – Information that comes to us through the internet, often through a table, smartphone, or laptop. **media balance** – using media in a way that feels healthy and in balance with other life activities (family, friends, school, hobbies, etc.) **red flag feeling** - when something happens on digital media that makes you feel uncomfortable, worried, sad, or anxious. **unplug** - to engage in activities that don't involve devices, apps, or the internet.

	digital habits - behaviors we do often or regularly with digital media and devices
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Common Sense Education

Don't Feed the Phish

Lesson overview:   	**Purpose:** • Compare and contrast identity theft with other kinds of theft. • Describe different ways that identity theft can occur online. • Use message clues to identify examples of phishing. **Lesson:** Warm Up: • "Safe or Unsafe?" Explore: • "How Identity Theft Happens" Analyze: • "How to Catch a Phish" Wrap Up: • "Stay Safe from Scams"
Lesson links/resources:	• Common Sense Education: Don't Feed the Phish
CS standards addressed:	**IC.2.4** Describe tradeoffs between allowing information to be public and keeping information private and secure.
Time needed:	Total time: 60 Mins • Warm UP: 10 mins • Explore: 15 mins • Analyze: 15 mins • Wrap Up: 5 mins
Materials needed:	Teacher: • Computer • Projector/smartboard with sound Students: • Computer/tablet with internet access
Subject integrated:	ELA
Other standards addressed:	**W.6.4** Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience. (Grade-specific expectations for writing types are defined in standards 1–3 above.) **W.6.9** Draw evidence from literary or informational texts to support analysis, reflection, and research.
Vocabulary:	**Private information** – information about you that can be used to identify you because it is unique to you (e.g., your full name or your address). **Identify theft** – a type of crime in which your private information is stolen and used for criminal activity. **Phishing** – when someone poses as an institution, like a bank or school, and sends you a personalized message asking you to provide private information. **Internet scam** – an attempt to trick someone, usually with the intention of stealing money or private information. **Shortened URL** – a web address that has been condensed and which could mislead a user into going into a risky website.

Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Common Sense Education

"Who Are You Online?"

Lesson overview:   	**Purpose:** • What does it mean to "be yourself" or to "be "real"? Those are deep thoughts for any middle schooler. For kids today, these questions matter online, too. Help your students explore why some people create different or alternate personas for themselves online and on social media. • Reflect on reasons why people might create fake social media accounts. • Identify the possible results of posting from a fake social media account. • Debate the benefits and drawbacks of posting from multiple accounts. **Lesson:** Warm Up: • Why "Finstas"? Explore: • "Which Me Should I Be?" Debate: • "The Finsta Debate"
Lesson links/resources:	• Common Sense Education: "Who Are You Online?"
CS standards addressed:	**NI.2.2a** Students will explain how physical and digital security measures protect electronic information. **IC.2.4a** Students should discuss and describe the benefits and dangers of allowing information to be public or kept private and secure.
Time needed:	**Total time: 45 Mins** • Warm Up: 10 Mins • Which Me Should I Be: 15 Mins • The Finsta Debate: 20 Mins
Materials needed:	Teacher: • Computer • Projector/smartboard with sound Students: • Computer/tablet with internet access
Subject integrated:	ELA
Other standards addressed:	**W.6.1** Write arguments to support claims with clear reasons and relevant evidence. **W.6.1a** Introduce claim(s) and organize the reasons and evidence clearly. **W.6.1b** Support claim(s) with clear reasons and relevant evidence, using credible sources and demonstrating an understanding of the topic or text. **W.6.1c** Use words, phrases, and clauses to clarify the relationships among claim(s) and reasons.
Vocabulary:	**Anonymous** – without a name or other information that identifies who you are. **Affinity Group** – a group of people linked by a common interest or purpose.

	Curate – to select, organize and look after a collection (e.g., content posted to a social media profile) **Finsta** – a fake Instagram account used for posting to a specific group of people or to post anonymously
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Common Sense Education

Chatting Safely Online

Lesson overview:  •  	**Purpose:** • Games, social media, and other online spaces give kids opportunities to meet and chat with others outside the confines of their real-life communities. But how well do kids know the people they're meeting and interacting with? Help students consider whom they're talking to and the types of information they're sharing online. • Analyze how well they know the people they interact with online. • Reflect on what information is safe to share with different types of online friends. • Learn to recognize red flag feelings and use the Feelings & Options thinking routine to respond to them. **Lesson:** Warm Up: • Who You're Talking to Online Evaluate: • Two Online Chats Analyze: • Red Flag Feeling Wrap Up: • Exit Ticket
Lesson links/resources:	• Common Sense Education: Chatting Safely Online
CS standards addressed:	**IC.2.4a** Students should discuss and describe the benefits and dangers of allowing information to be public or kept private and secure.
Time needed:	**Total time: 60 Mins** • Warm Up: 10 mins • Evaluate: 15 mins • Analyze: 15 mins • Wrap Up: 10 mins • Discuss "Take-Home" Resources: 10 mins
Materials needed:	Teacher: • Computer • Projector/smartboard with sound Students: • Computer/tablet with internet access
Subject integrated:	ELA
Other standards addressed:	**W.6.4** Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience. (Grade-specific expectations for writing types are defined in standards 1–3 above.) **W.6.10** Write routinely over extended time frames (time for research, reflection, and revision) and shorter time frames (a single sitting or a day or two) for a range of discipline-specific tasks, purposes, and audiences.

Vocabulary:	**Private Information** – information about you that can be used to identify you because it is unique to you (e.g., your full name or your address). **Red Flag Feeling** – when something happens on digital media that makes you feel uncomfortable, worried, say, or anxious. **Inappropriate** – not acceptable in the situation; not okay **Risky** – potentially harmful to one's well-being
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Common Sense Education

Digital Drama Unplugged	
Lesson overview:  	**Purpose:** • Miscommunication is a common occurrence online and on social media. Plus, being behind a screen makes it easier to say things they wouldn't say in person. So how do we help students avoid the pitfalls of digital drama? Help them learn tips on avoiding online drama in the first place and de-escalating drama when it happens. • Reflect on how easily drama can escalate online. • Identify de-escalation strategies when dealing with digital drama. • Reflect on how digital drama can affect not only oneself but also those around us. **Lesson:** Warm Up: • "What is Digital Drama?" Explore: • "Where do People Stand?" Wrap Up: • "Act It Out"
Lesson links/resources:	Common Sense Education: Digital Drama Unplugged
CS standards addressed:	**IC.2.1a** Students should consider current events related to broad ideas, including privacy, communication, and automation.
Time needed:	**Total time: 60 Mins** • Warm Up: 5 mins • Explore: 25 mins • Wrap Up: 15 mins • Discuss "Take-Home" Resources: 15 mins
Materials needed:	Teacher: • Computer • Projector/smartboard with sound Students: • Computer/tablet with internet access
Subject integrated:	ELA
Other standards addressed:	**W.6.8** Gather relevant information from multiple print and digital sources; assess the credibility of each source; and quote or paraphrase the data and conclusions of others while avoiding plagiarism and providing basic bibliographic information for sources. **W.6.9** Draw evidence from literary or informational texts to support analysis, reflection, and research.
Vocabulary:	**De-escalate** – to lessen they intensity or anger in a conflict. **Digital Drama** – when people use devices, apps, or websites to start or further a conflict between people.
Notes:	→Teachers will need to create FREE teacher and/or student accounts (when applicable) at Common Sense Education

Appendix A: Code.org

I'd like to start using Code.org in my classroom. How should I start?

https://support.code.org/hc/en-us/articles/228116468-I-d-like-to-start-using-Code-org-in-my-classroom-How-should-I-start-

How to create a teacher account:

https://support.code.org/hc/en-us/articles/228116468-I-d-like-to-start-using-Code-org-in-my-classroom-How-should-I-start-

How to create a classroom section:

https://support.code.org/hc/en-us/articles/115000488132-Creating-a-classroom-section

Finding curriculum and lesson plans:

https://support.code.org/hc/en-us/articles/115001595051-Finding-curriculum-and-lesson-plans

Code.org Support

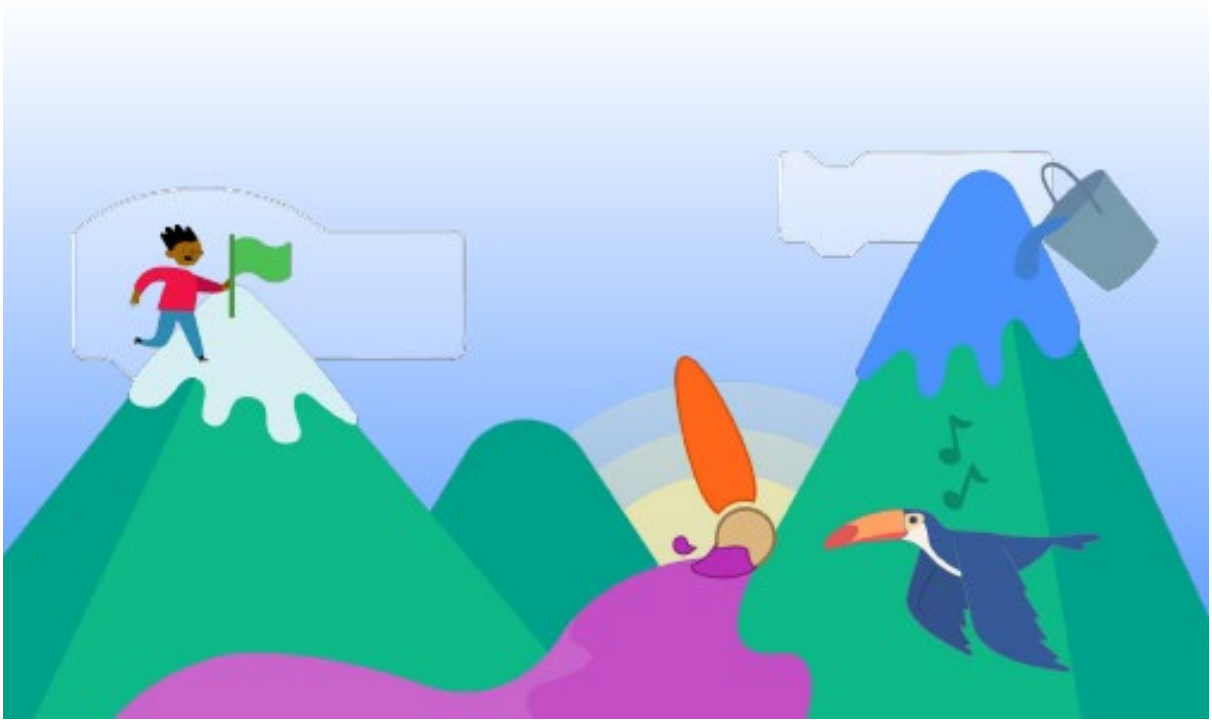
https://support.code.org/hc/en-us

Appendix B: Scratch

SCRATCH

Educator's Guide

- Teacher Accounts
- Beginner's Guide
- Lesson Guides



Teacher Accounts

As an educator, you can request a Scratch Teacher Account. A Scratch Teacher Account provides educators with additional features to manage student participation on Scratch, including the ability to create student accounts, organize student projects into studios, and monitor student comments. This guide will walk you through creating an account, creating a class, adding and managing your students, and creating class studios. You can also see our Scratch for Educators page and our Teacher Account FAQ page for additional information.

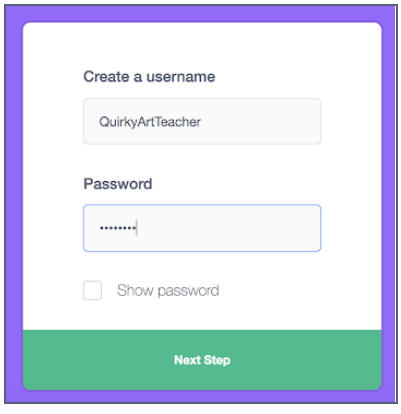
Create Your Teacher Account

Visit this link to get started: https://scratch.mit.edu/educators/register

You'll be prompted to create a username and password. ***Make sure that your username does not contain your name or personal information***, like your school, location, or email address.

Within the Scratch community, all users are asked to refrain from sharing personal information through their usernames. ***It's important that both you and your students follow these guidelines. Accounts that do not adhere to these guidelines will be deleted.***

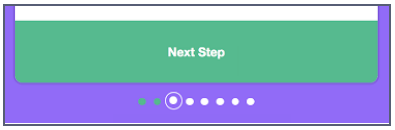
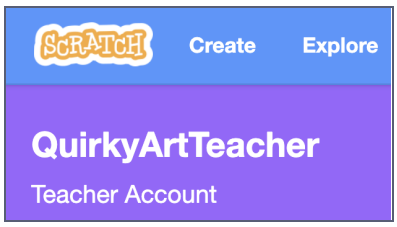
Creating your teacher account



Tips for making your username

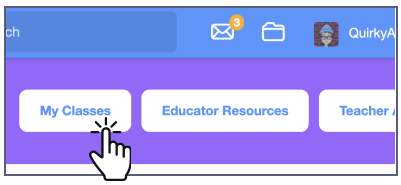
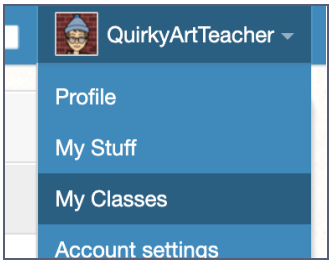
- **Incorporate the name of the subject you teach**
  - ex: QuirkyArtTeacher
- **Use a tool or term from the subject you teach**
  - ex: MetamorphicRocks
- **Add an important date, be unique**
  - ex: Bibliophile1440
- **Make it memorable with a pun or an alliteration!**
  - ex: TyranoTeacher

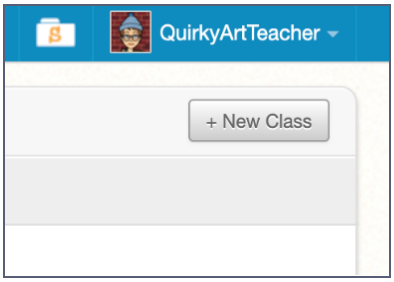
Be sure to make a note of your username and password.

	Click through each step to **complete registration**.
	Log into your email and confirm your email address. Check your spam folder if you do not see the email. Once you have **confirmed your email address**, we'll review your account.
	Once your account has been reviewed and approved, **you will receive a welcome email**. Then, you can **log into your teacher account at scratch.mit.edu**!

Create a Class

Creating classes allows you to manage groups of students, and create studios where your students can add their projects.

Creating your class	
 	Once you have successfully logged into your Teacher Account, if you are looking at the homepage, there will be a bar at the top of the screen with three options. Select "**My Classes**." You can also access your classes from the dropdown under your username.

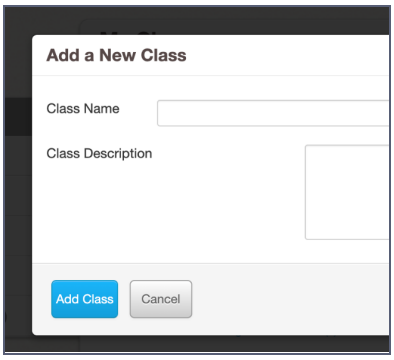


To create a class, click the "**+ New Class**" button at the top right of the page.

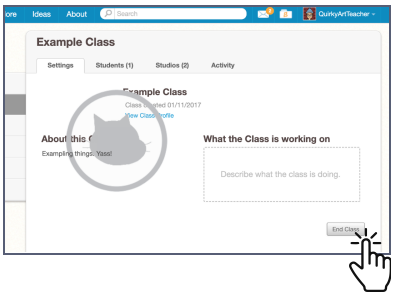
Enter the class name and description.

Warning: Do not include real names and locations, like the name of your school or city/town.

Once you have created a class, you can add students.



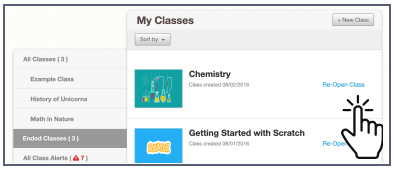
Ending your class



To end a class, under "My Classes," choose your class and on the Settings tab, click the "End Class" button.

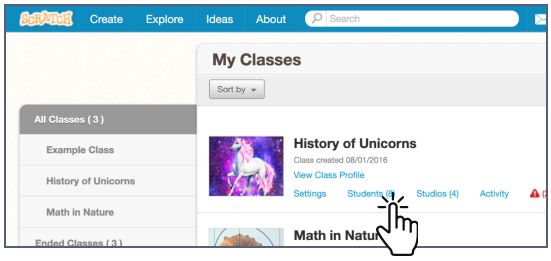
When you end a class, your class profile page will be hidden and your students will no longer be able to log in (but their projects and the class studios will still be visible on the site).

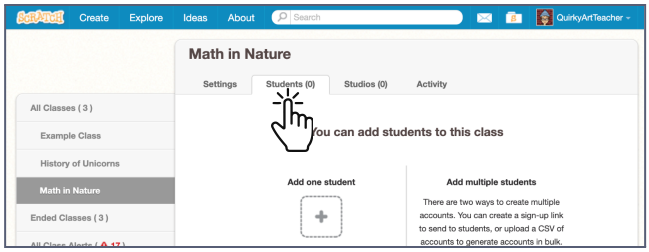
You may re-open the class at any time. By going to the "Ended Classes" tab and clicking the "Re-Open Class" link near the class you want to reopen.



Add Students to Your Class

While on “My Classes,” select the class and then click on “Students” (either the link under the class name or the Students tab). Once created, your student accounts will appear here.

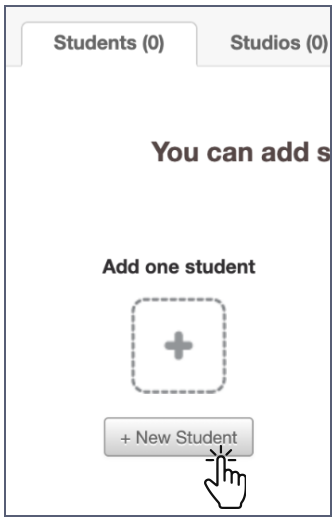




There are three ways to add students to your class. The first method allows you to add an individual student to a class. Methods 2 and 3 allow you to add multiple students to a class.

Tip: Create a naming convention as a guideline for generating usernames. For example, you may want each name to include an abbreviation for the course name, the class section, and the student’s number on your roster (ex: VisArts-02-17). Use the Student Username List we have created to record the usernames and passwords your students have created.

Method 1: Add Individual Students



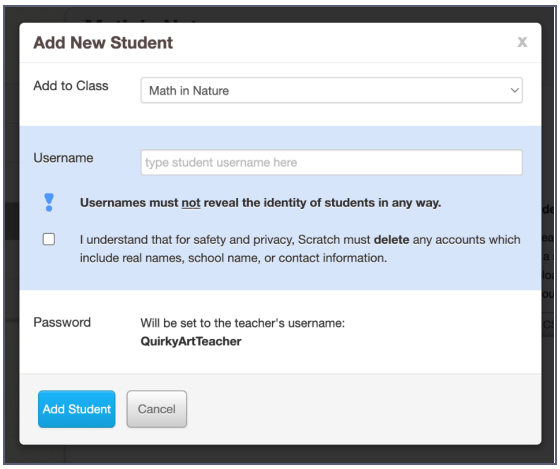
Click the “+ New Student” button to add students individually.

Confirm the correct class is showing in the “Add to Class” dropdown menu.

You will be prompted to create a username for this student.

Warning: Make sure that the usernames you create do not contain identifying information about yourself, your students, or your school. Accounts that do not adhere to these guidelines will be deleted.

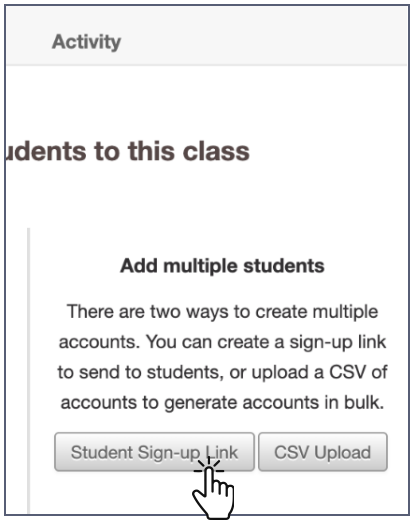
The password for this student username will automatically be set as the username of your teacher account.

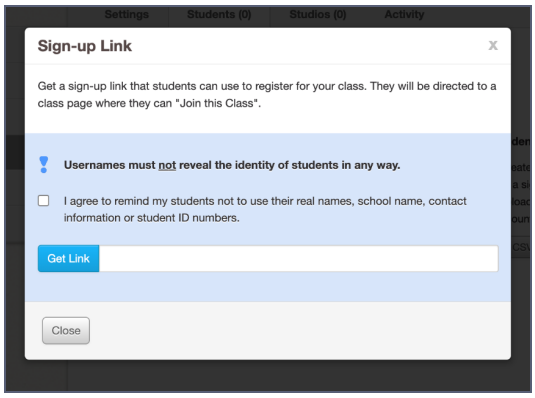


Have students log into their accounts and change their passwords as soon as possible.

Tip: It is not possible to add an existing Scratch account to a classroom. You will need to create a new Student Account for them using your Teacher Account. A student can only be a part of one class, and it is not possible to transfer students from one class or teacher to another.

Method 2: Student Sign-up Link



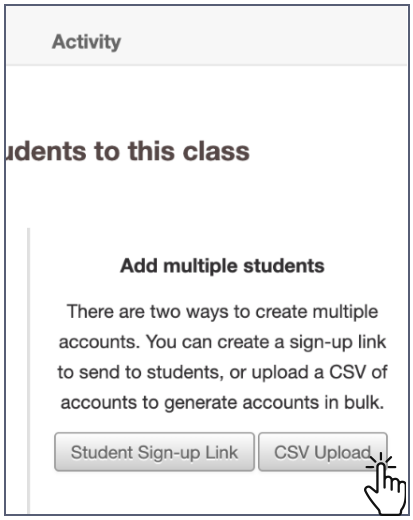


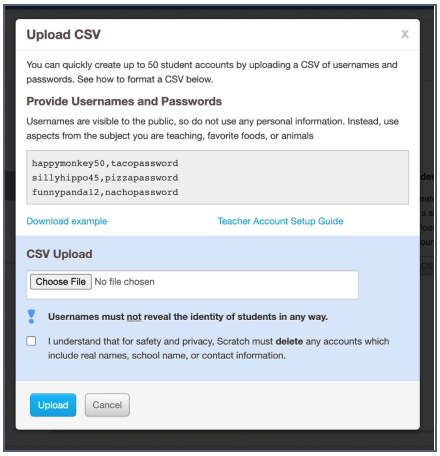
Clicking the "Student Sign-Up Link" button brings you to another window and clicking the "Get Link" button will generate a link that will allow your students to join the class you have just created. The link will start with "http://scratch.mit.edu/signup…"

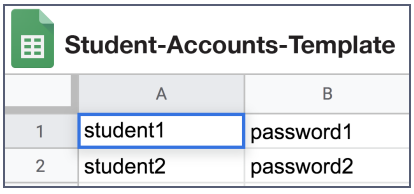
Students can then create their own usernames and passwords.

Warning: Remind your students that, when making their usernames, the username should not contain identifying information about themself, their teacher, or their school. Accounts that do not adhere to these guidelines will be deleted.

Method 3: CSV Upload







Click the “CSV Upload” button on the class page.

Using the template provided by clicking the “Download example” link, create a username and password for each of your students. You can use the template provided or create your own spreadsheet with student usernames in column A and passwords in column B. To upload your own template, you’ll need to save the file as a CSV file.

Once you’ve created usernames and passwords for each student and saved the file, click the “Choose file” button to locate the file, then click the “Upload” button.

It is not possible to add more than 250 students to a single class. You can, however, create a new class and add another 250 student accounts to each new class.

Warning: Make sure that the usernames you create do not contain identifying information about yourself, your students, or your school. Accounts that do not adhere to these guidelines will be deleted.

You can add students via any of these methods at any time under the “Students” tab.

Creating Studios for Student Work

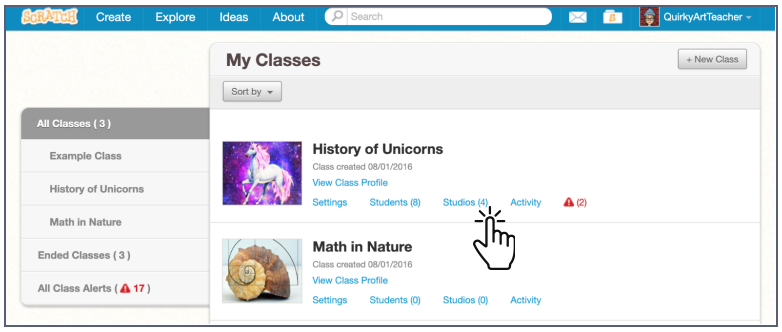
Studios allow you to create collections of student projects for specific classes or assignments. This makes it easier for you to view their projects throughout their creative process. It also makes it easier for students to collaborate and be inspired by each other's work.

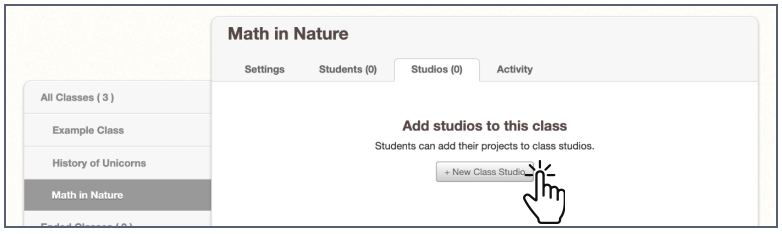
Scratcher status is required in order to create a studio, and the person who created the studio is automatically assigned the role of "host." There is only one host per studio, and only studio hosts can edit the title, thumbnail, and description.

Studios are immediately public, even those created in the context of a class. Unlike Scratch projects, there is no share/unshare option for studios. Everyone can follow a studio, see studio comments and projects, and leave a comment or add a project (unless commenting or the ability to add projects is turned off).

There are two ways to create a studio on a teacher account. Method one creates studios that automatically add all students in a class as curators. Method two creates studios without automatically adding students as curators, and students or any Scratcher can be individually added as curators.

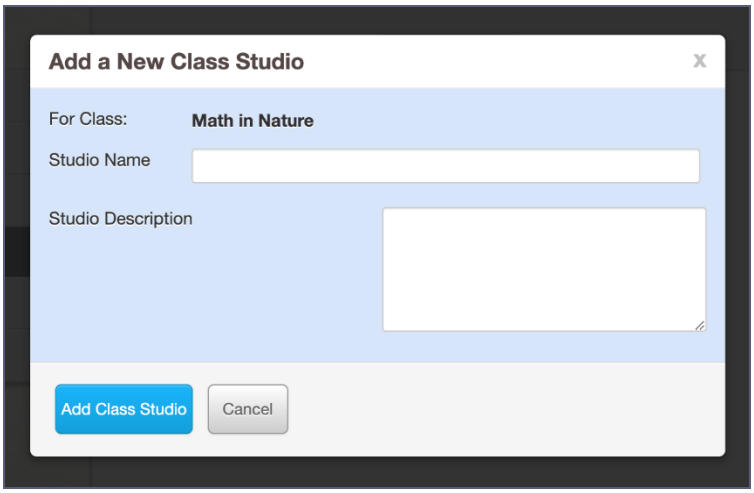
Method 1: Create a studio that automatically adds all students in a class as curators



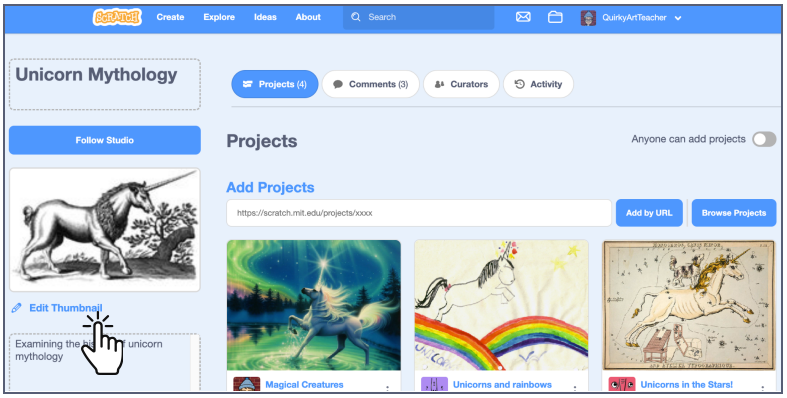


Once logged into your Scratch account, go to "My Classes."

Choose the class to assign the studio to, then click on "Studios" (either the link under the class name or the Studios tab). Then click the "+ New Class Studio" button.



On the window that appears, you will be asked to **give the studio a name and description**. (These can always be adjusted in the studio later.) In the description, you can share the theme of the studio, what kinds of projects you are looking to include… Just be sure your title and description don't reveal any personal information (like school name or first and last name).

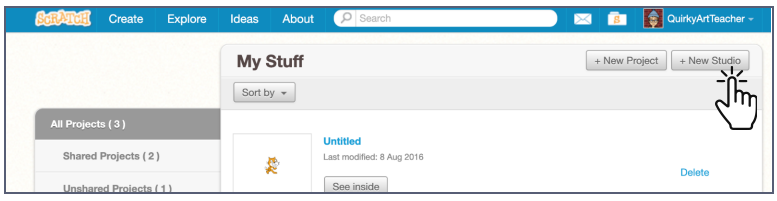


Then, click the "Add Class Studio" button.

Once in the studio, click the "Edit Thumbnail" button to change the default gray cat image in the upper left-hand corner. **Upload your own studio thumbnail image**. The maximum file size for a thumbnail is 512 KB and your image must be less than 500x500 pixels.

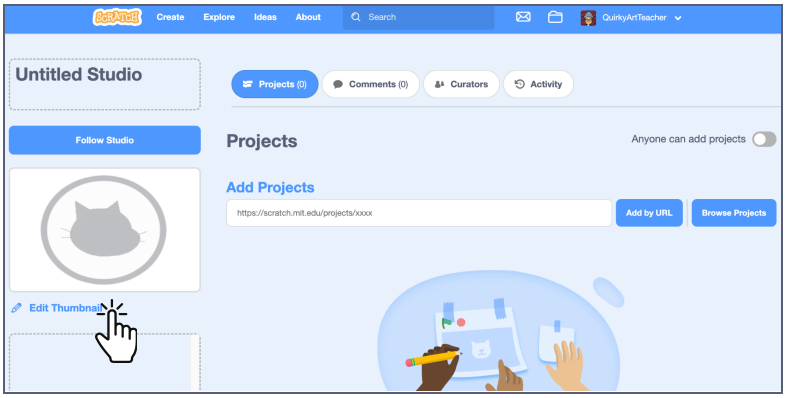
When you click on the "Curators" tab, you should see all the class students have been set as studio curators.

Method 2: Create a studio without automatically adding students as curators



Once logged into your Scratch account, go to "My Stuff."

Choose the "**+ New Studio**" button at the top right.



Click on “Untitled Studio” to **give your studio a name and description**. In the description, you can share the theme of the studio, what kinds of projects you are looking to include… Just be sure your title and description don’t reveal any personal information (like school name or first and last name).

Click the “Edit Thumbnail” button to change the default gray cat image in the upper left-hand corner. **Upload your own studio thumbnail image**. The maximum file size for a thumbnail is 512 KB and your image must be less than 500x500 pixels.

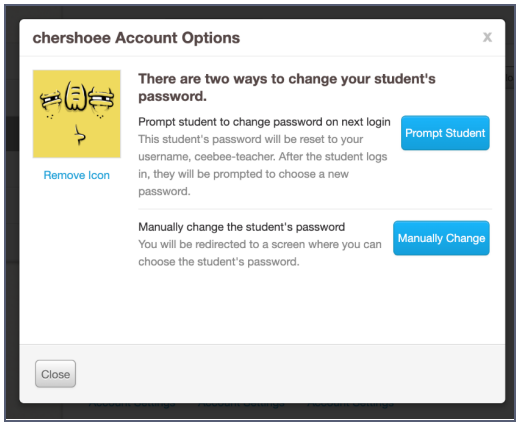
When you click on the “Curators” tab, you should see no curators have been assigned yet.

See our Studio Guide for detailed information on:

- Studio Definitions
- How to Manage a Studio
- How to Add Projects to a Studio

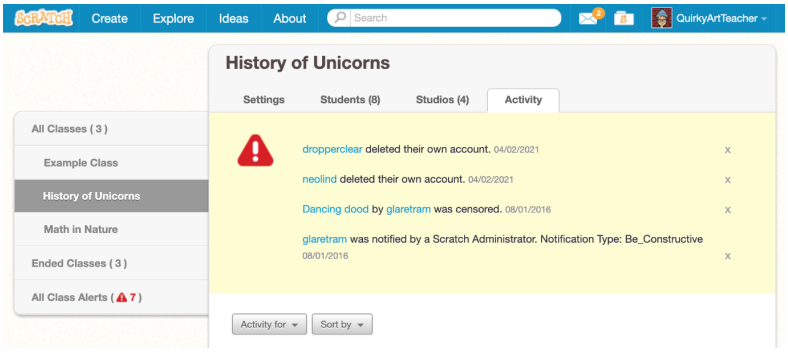
Managing Your Students

Managing a student



You can manually **reset a student password** from within your Scratch Teacher Account. First, navigate to "My Classes" and choose the class and go to the "Students" tab. Then click on the "Account Settings" link below the student's account.

You cannot delete a student's account by using a Teacher Account, but students can delete their own account.



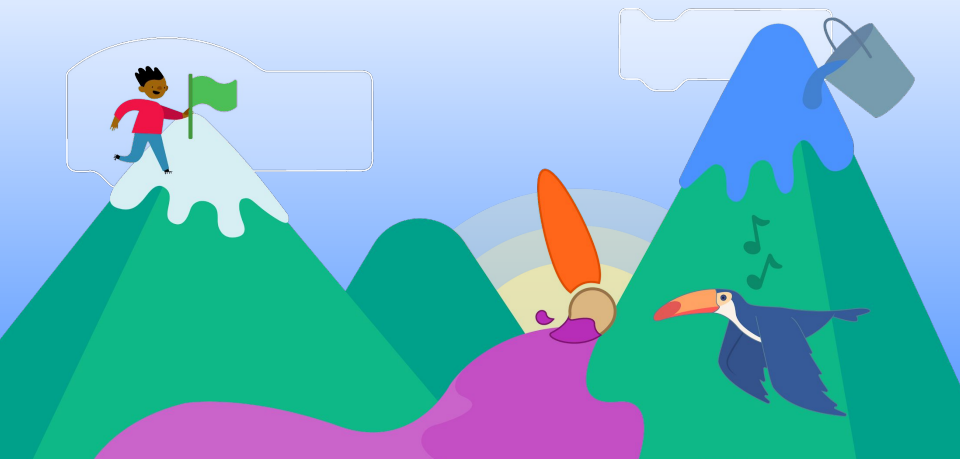
You can see alerts about notifications your students receive on the "Activity" tab of a class or the "All Class Alerts" tab.

Tip: If you'd like to translate this guide, **click here to make a copy** of this Google doc.

Getting Started with SCRATCH

Beginner's Guide

Create your own games, animations, interactive stories, and more.

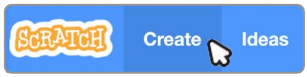


GETTING STARTED

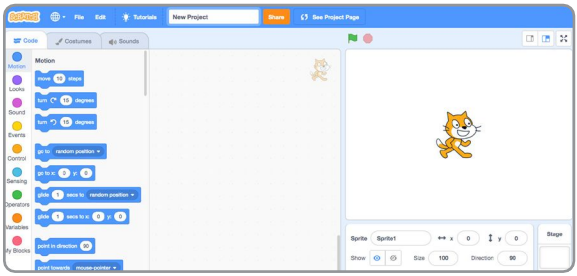
You can use Scratch online at: **scratch.mit.edu**

scratch.mit.edu

Once you've navigated to scratch.mit.edu, click **Create**.



This will bring you to the **Scratch Editor,** where you can start creating projects.



If your computer uses an older operating system, or your internet connection is unreliable, you can download Scratch and use it offline.

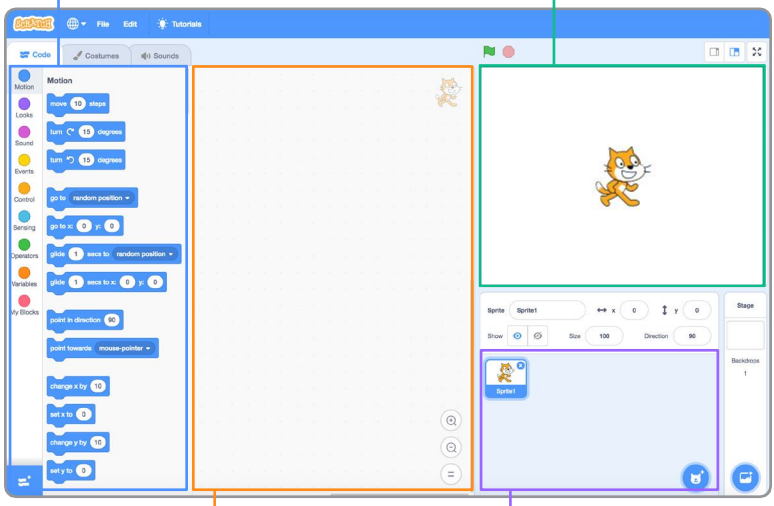
Visit: **https://scratch.mit.edu/download** for information on downloading and installing the Scratch app.

THE SCRATCH EDITOR

The Scratch Editor is where you create projects in Scratch. Here are its main parts:

Blocks Palette
Blocks for coding your projects

The Stage
Where your creations come to life

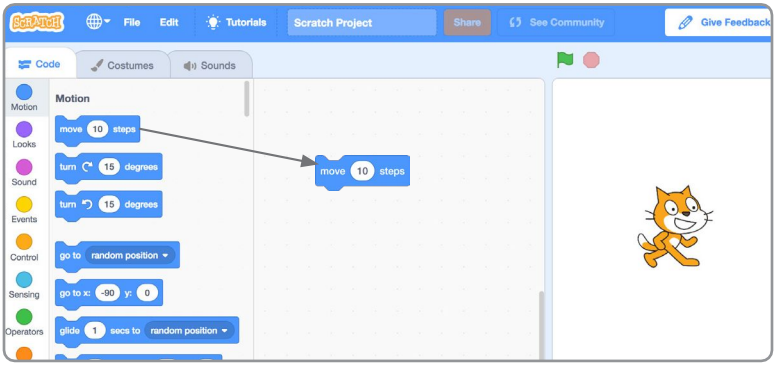


Coding Area
Drag in blocks and snap them together to code your sprites

Sprite List
Click the thumbnail of a sprite to select it

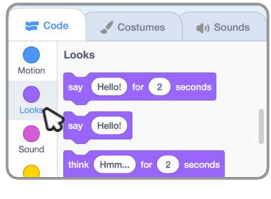
LET'S CODE!

To code projects in Scratch, you snap together blocks.
Start by dragging out a **move** block.

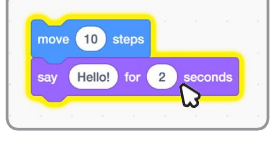


Click the block to try it.
Does your cat move?

Now say something!
Click the **Looks** category.

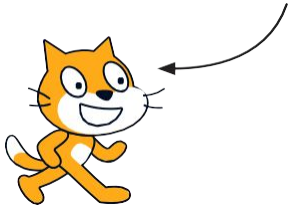


Drag out a **say** block.
Snap it onto the **move** block. Click on your blocks to try them.



WHAT IS A SPRITE?

In Scratch, any character or object is called a sprite. Every new project in Scratch starts with the Cat sprite.



Want to choose a different sprite?

Click the New Sprite icon.

Or, hover over the **New Sprite** icon to see more options.

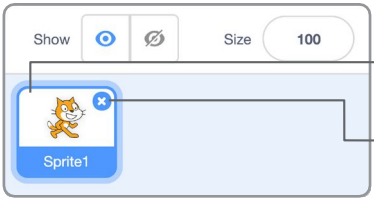
Upload an image from your computer.

Draw your own sprite.

Click for a surprise sprite!

Choose a sprite from the library.

Want to **delete a sprite** from your project?



First, select the sprite by clicking on its thumbnail in the Sprite List.

Then, click here to delete the sprite.

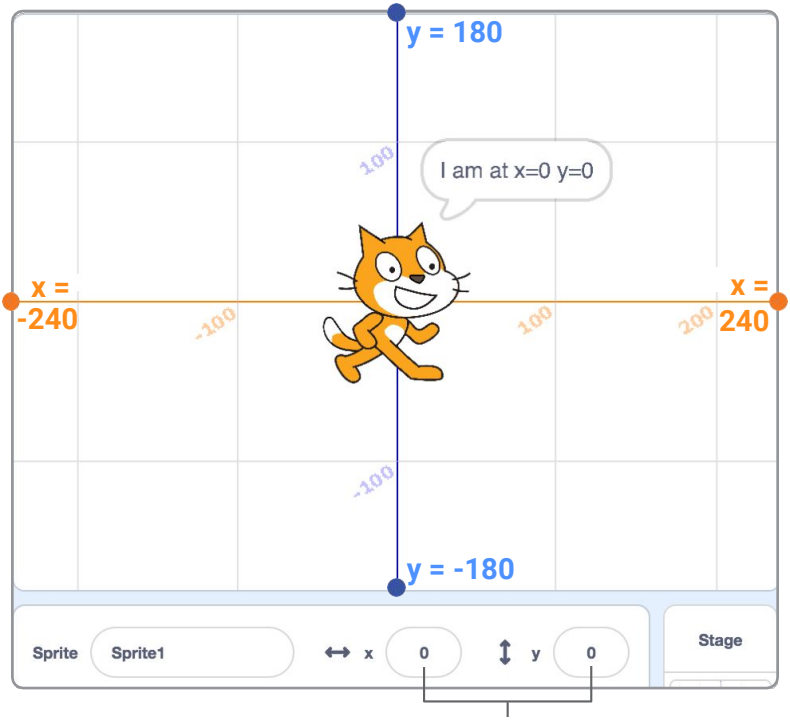
WHERE IS YOUR SPRITE?

Every sprite has an **x** and **y** position on the Stage.

x is the position of the sprite from left-to-right.

y is the position from top-to-bottom.

At the very center of the stage, **x** is 0 and **y** is 0.

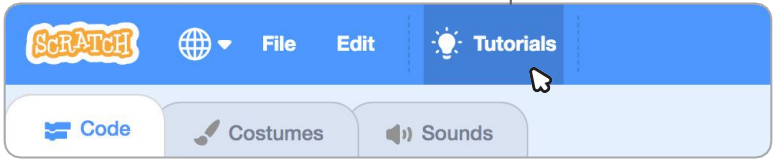


When you move your sprite, you can see its **x** and **y** position change.

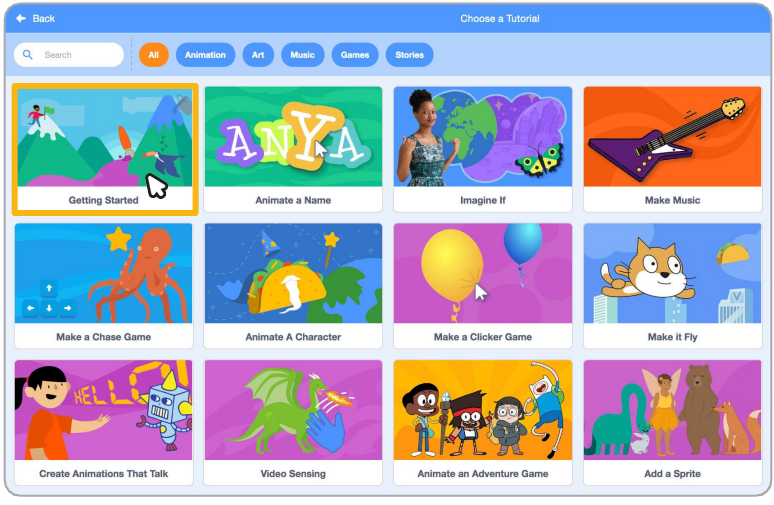
TUTORIALS

There are a range of tutorials available in the Scratch **Tutorials Library**, which guide learners in creating projects with Scratch. Students can get started making their own stories, animations, and games.

You can get to the Tutorials Library from the Scratch Editor by clicking the **Tutorials** button.



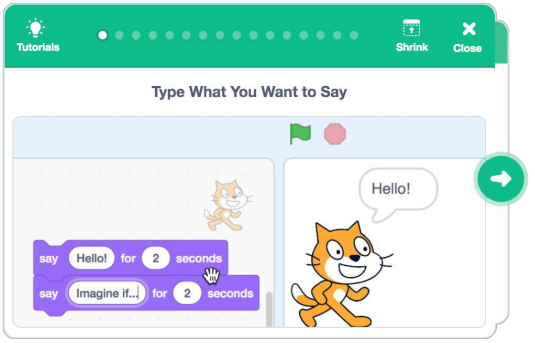
The **Getting Started** tutorial will walk you through the basics.



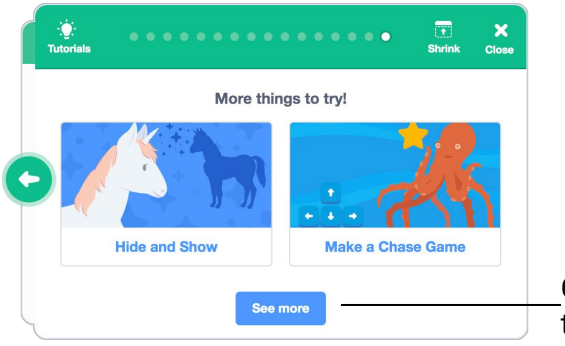
TUTORIALS

Once you've selected the tutorial, it will open in the Scratch Editor.

Click the green arrow to see each step.



When you've reached the end of a tutorial you can select another tutorial, and keep adding to your project.



Click here to see all the Tutorials.

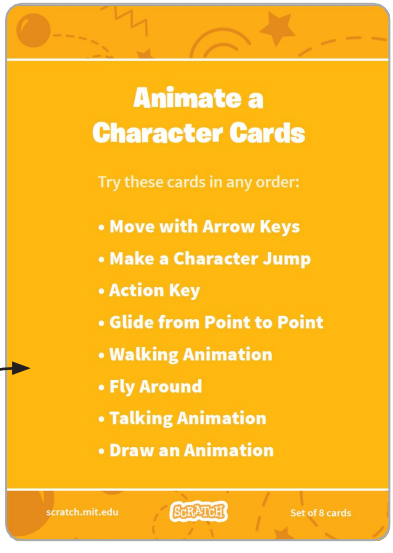
CODING CARDS

The Scratch **Coding Cards** provide another way to learn to create projects with Scratch. Download the cards at **scratch.mit.edu/ideas**.

Each set of cards starts with a title card, which shows you what you can create.

The **Animate a Character** cards are a great set to start with.

Examples of what you can create

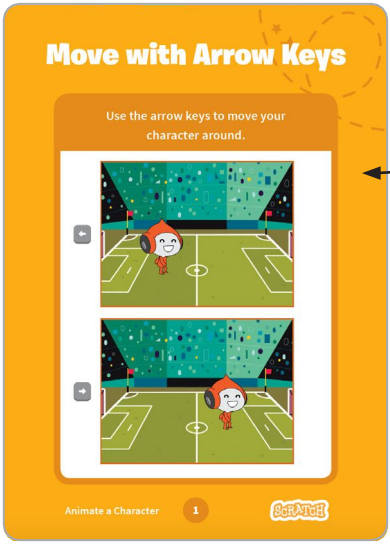


A list of all the cards in this set

USING THE CODING CARDS

After each title card is a series of cards walking you through each step of creating a project.

Add your own sprites, backdrops and more!



The front of each card shows you what you can create.

The back shows you how to do it.

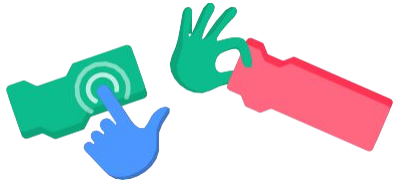
GET CREATIVE!

Encourage students to use their imagination as you create projects. There are many different ways they can make their Scratch projects unique.

You can choose or draw your own characters.

Choose a sound or record your own.

Try changing numbers or adding blocks to your code to see what happens.

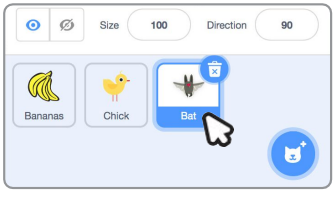


Experiment and customize your project however you want!

GET CREATIVE WITH SPRITES!

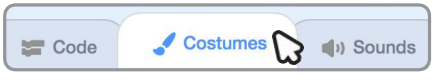
Scratch has its own paint tools, which allow you to customize sprites from the library, or even create sprites of your own.

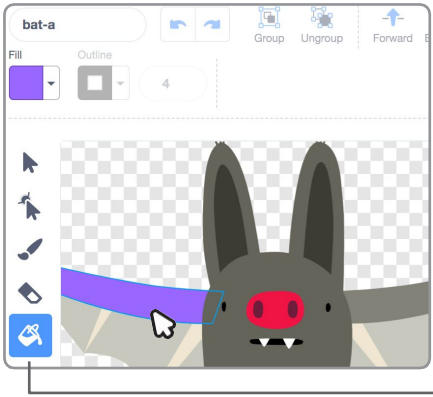
Let's start by editing a sprite from the library.



Select a sprite to edit by clicking on it in the Sprite list.

Click the **Costumes** tab at the top left to see the paint tools.





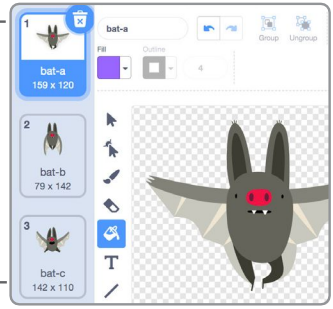
The paint tools allow you to recolor sprites, add to them with a paint brush, and change them in a variety of ways.

You can use the **paint bucket** tool to recolor different parts of a sprite.

GET CREATIVE WITH SPRITES!

Some sprites, like the Bat sprite have multiple costumes, or poses.

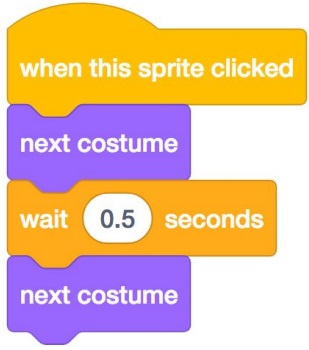
You can see a sprite's costumes by clicking the **Costumes** tab.



If your sprite only has one costume, right click on the costume to duplicate it (On Mac control + click).

Now you can modify the second costume using the paint tools, so your sprite has two different poses or facial expressions.

Click the **Code** tab, then try adding these blocks.

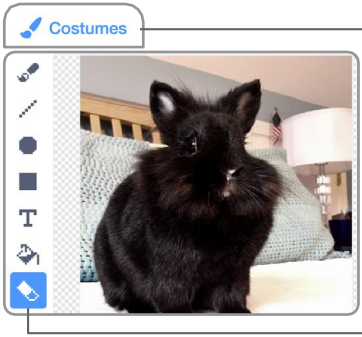


ADD YOUR OWN PHOTOS

There are many ways to create your own sprites and artwork using the Scratch paint tools.

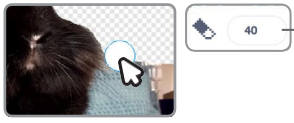
You can create your own sprites by uploading photos or images and erasing the background.

Hover over the New Sprite button, then select **Upload Sprite**.



Next click the **Costumes** tab. You will see bitmap tools for editing your image.

Click the **eraser** icon and use the eraser tool to remove the background from your photo.



Tip: to adjust the size of the eraser, type a larger or smaller number.

There are two modes for drawing in Scratch:

1. **Bitmap Mode** allows you to edit photos and paint with pixels.
2. **Vector Mode** allows you to create and edit shapes.

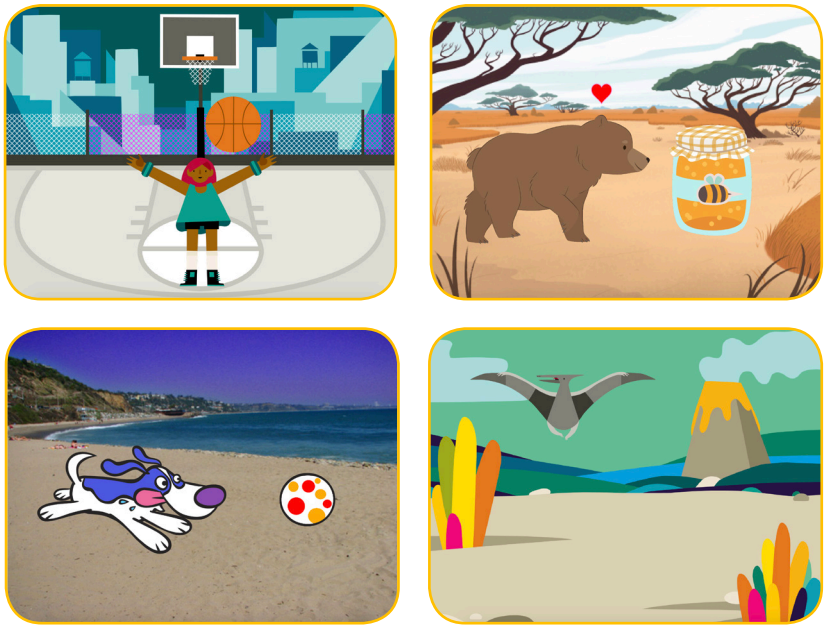
Tip: If you'd like to remix and customize this guide, **click here to make your own copy** of the Google Slides template.

COLOR SCHEME

EDUCATOR GUIDE

Animate a Character

With this guide, you can plan and lead a 55-minute lesson using Scratch. Students will gain experience with coding as they bring characters to life with animation.



ANIMATE A CHARACTER / EDUCATOR GUIDE

Lesson Outline

Objective: Students will become familiar with the Scratch environment by animating a character.

IMAGINE
10 minutes

First, gather as a group to introduce the theme and spark ideas.

CREATE
40 minutes

Next, help students as they animate characters, working at their own pace through the tutorial.

SHARE
5 minutes

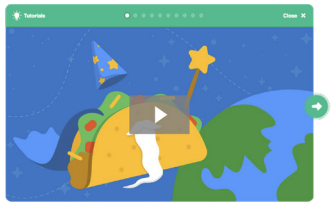
At the end of the session, gather together to share and reflect.

Get Ready for the Lesson

Use this checklist to prepare for the lesson.

☐ Preview the Tutorial

The *Animate a Character* tutorial shows students how to create their own projects. Preview the tutorial before your lesson and try the first few steps: scratch.mit.edu/tutorials



☐ Print the Activity Cards (optional)

Print a few sets of *Animate a Character* cards to have available for students during the lesson.
scratch.mit.edu/ideas

☐ Make sure students sign into their Scratch accounts

Have students sign into their own Scratch accounts at scratch.mit.edu.

☐ Set up computers or laptops

Arrange computers so that students can work individually or in pairs.

☐ Set up a computer with projector or large monitor

You can use a projector to show examples and demonstrate how to get started.

ANIMATE A CHARACTER / EDUCATOR GUIDE

Imagine

IMAGINE

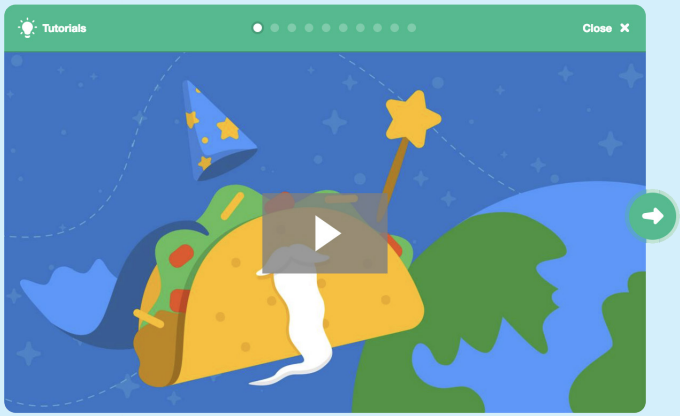
Begin by gathering the students to introduce the theme and spark ideas for projects.

Warm-up Activity: Favorite Characters

Gather the group in a circle. Ask each student to say their name, then share a favorite character from a book, movie, or TV show, and one or two of their favorite things about that character.

Provide Ideas and Inspiration

To spark ideas, watch the Animate a Character video at the start of the tutorial. The video shows a variety of projects to spark ideas and inspiration.



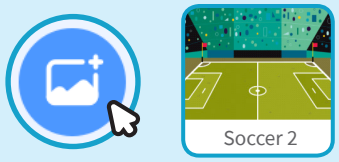
View the scratch.mit.edu/ideas

Demonstrate the First Steps

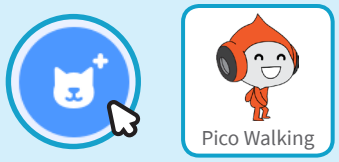
IMAGINE

Demonstrate the first few steps of the tutorial so students can see how to get started.

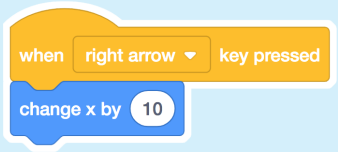
Choose a backdrop.



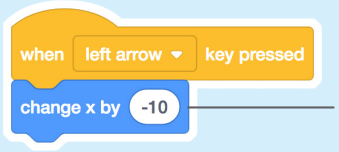
Choose a character to animate.



Make your sprite move right and left with arrow keys:



Choose right arrow from the menu.

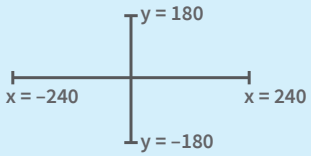


Choose left arrow from the menu.

Type a minus sign to move left.

Press the left arrow and right arrow keys on your keyboard to move.

Helpful Hint: Understanding x y coordinates will help students figure out how to move sprites around the stage.



y is the position on the Stage from top to bottom.

x is the position on the Stage from right to left.

Create

CREATE

Support students as they create animated Scratch projects.

Start with Prompts

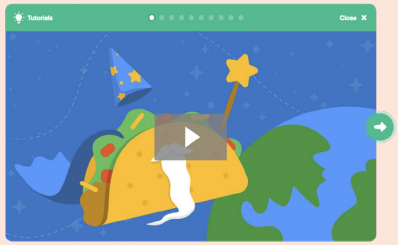
Ask students questions to get started

Which character would you like to animate?

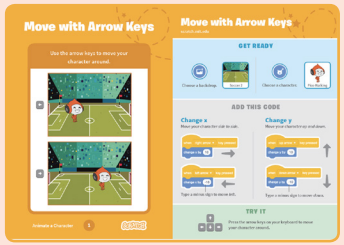
What do you want your character to do?

Provide Resources

Offer options for getting started



Some students may want to follow the online tutorial: scratch.mit.edu/animate



Others may want to explore using the activity cards: scratch.mit.edu/ideas

Suggest Ideas for Starting

- Choose a character to animate.
- Animate your character: make it jump, fly, glide or talk!
- Choose a backdrop.

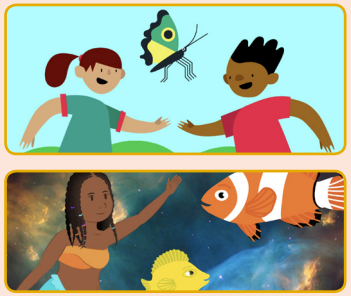
CREATE

More Things to Try

- Try combining more than one kind of animation.
- If you're not sure what to do, pick a card and try something new.
- Add a second character or object to animate.

Support collaboration

- When someone gets stuck, connect them to another participant who can help.
- See a cool idea? Ask the creator to share with others.



Encourage experimenting

The Animate a Character activity can be done in any order, with a range of different character and object sprites.

Encourage students to try new things:

What will your character do next?

How can you make your animation interactive?

Share

SHARE

Have students share their project with their neighbors.

Ask questions they can discuss:

What do you like best about the project you made?

What was the hardest part?

If you had more time, what would you add or change?

What's Next?

Students can use the ideas and concepts from this lesson to create a wide variety of projects. Encourage them to continue developing their projects into games, stories or interactive art with the resource listed below.

Video Sensing

Interact with characters and objects in Scratch with video sensing.

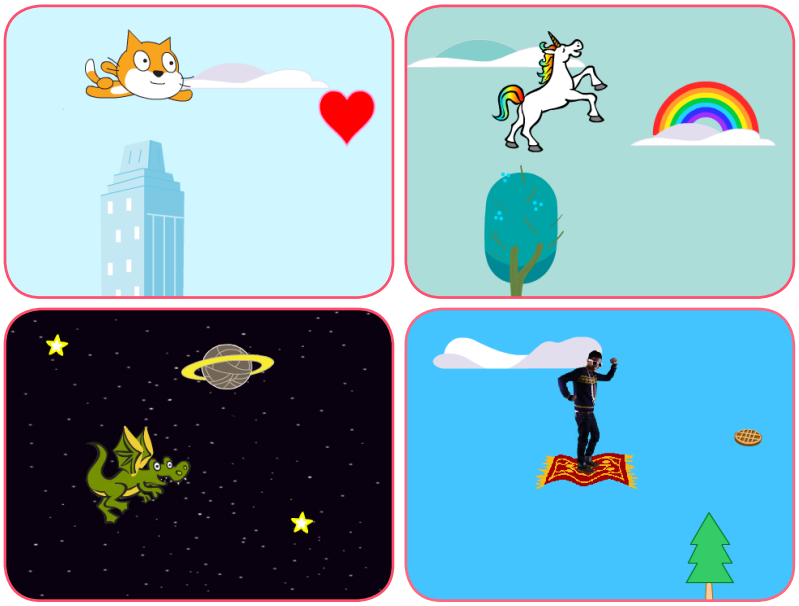
Find this project and more in the Tutorials library: **scratch.mit.edu/ideas**

Scratch is a project of the Lifelong Kindergarten Group at the MIT Media Lab.

EDUCATOR GUIDE

Make It Fly

With this guide, you can plan and lead a 55-minute lesson using Scratch. Students will choose a character and program it to fly.



MAKE IT FLY / EDUCATOR GUIDE

Lesson Outline

Objective: Students will create an animation with the illusion of a flying character.

IMAGINE
10 minutes

First, gather as a group to introduce the theme and spark ideas.

CREATE
40 minutes

Next, help students as they create a flying animation, working at their own pace through the tutorial.

SHARE
5 minutes

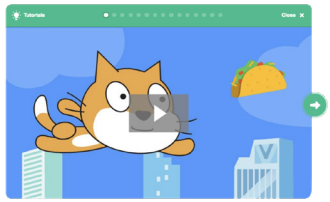
At the end of the session, gather together to share and reflect.

Get Ready for the Lesson

Use this checklist to prepare for the lesson.

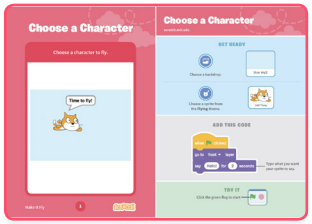
☐ **Preview the Tutorial**

The *Make It Fly* tutorial shows students how to create their own projects. Preview the tutorial before your lesson and try the first few steps: scratch.mit.edu/fly



☐ **Print the Activity Cards (optional)**

Print a few sets of *Make It Fly* cards to have available for students during the lesson.
scratch.mit.edu/fly/cards



☐ **Make sure students sign into their Scratch accounts**

Have students sign into their own Scratch accounts at scratch.mit.edu.

☐ **Set up computers or laptops**

Arrange computers so that students can work individually or in pairs.

☐ **Set up a computer with projector or large monitor**

You can use a projector to show examples and demonstrate how to get started.

Imagine

IMAGINE

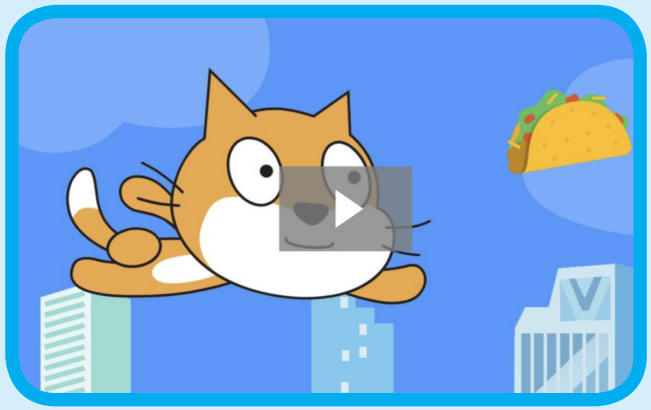
Begin by gathering the students to introduce the theme and spark ideas for projects.

Warm-up Activity: If I Could Fly...

Gather the group in a circle and ask, "If you could fly, where would you want to go?" Suggest that they close their eyes and imagine flying through their favorite place. Ask, "Where are you? What kinds of things do you see below you?" If there's time, have each person say where they imagined flying or something they saw on their flight.

Provide Ideas and Inspiration

Show the introductory video for the *Make It Fly* tutorial. The video shows a variety of projects for ideas and inspiration.



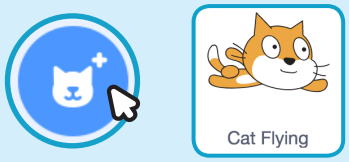
View at scratch.mit.edu/fly or vimeo.com/llk/fly

Demonstrate the First Steps

IMAGINE

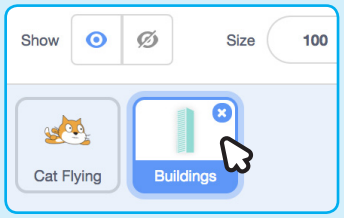
Demonstrate the first few steps of the tutorial so students can see how to get started.

In Scratch, click Create.
Choose a flying sprite from the library:



Choose a new sprite for your character to fly past:

Make the building move across the stage to make your character look like it's flying:



Create

CREATE

Support students as they make a flying animation.

Start with Prompts

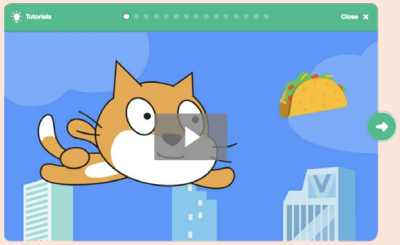
Ask students questions to get started

What character would you like to make fly?

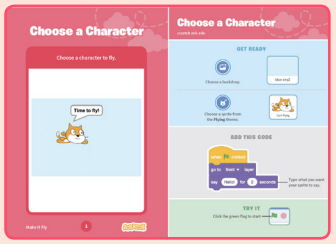
Where will your character go flying?

Provide Resources

Offer options for getting started



Some students may want to follow the online tutorial: scratch.mit.edu/fly



Others may want to explore using the activity cards: scratch.mit.edu/fly/cards

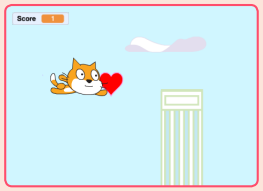
Suggest Ideas for Starting

- Choose a character
- Make the character say something
- Choose buildings or other scenery
- Make the scenery move

CREATE

More Things to Try

- Switch costumes to change the scenery.
- Make your character move when you press a key.
- Add clouds and other floating objects.
- Score points when touching an object.



Encourage Debugging

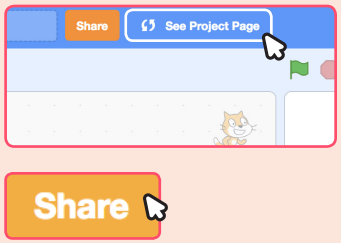
Here are some strategies to suggest to help students fix any bugs or difficulties they encounter:

- When stuck, talk out what you're working on with someone.
- Try out small bits of code at a time to figure out what's happening at each step.
- Look closely at the blocks on the tutorial or activity cards to see if they are the same or different from the blocks you're using.
- Remember that bugs always arise when creating a computer program. Debugging is a helpful skill to know not just in coding, but throughout life.

Prepare to Share

To add instructions and credits to a project, click the button: "See project page".

Give your project a title, add instructions and credits, then click Share.



Share

SHARE

Share projects with others in the room. Organize a flying character showcase. Ask half the room show their projects, while the others view them. Then switch.

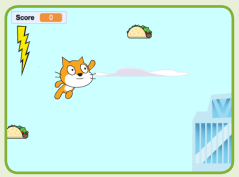
Suggest that they ask each other questions, such as:

What do you like best about the project you made?

What might you like to change or make next?

What's Next?

Students can use the ideas and concepts from this lesson to create other projects. Here are a couple of variations on the flying character project you could suggest.



Flying Game

Make a game where you avoid some objects and try to catch others. Add or subtract points based on what your character touches.

Flying Stories

Tell a story about your flying characters. You can record your voice and play sound clips. Or, use say blocks to make voice bubbles.

EDUCATOR GUIDE

Make a Chase Game

With this guide, you can plan and lead a 55-minute lesson using Scratch. Students will make a game that includes a variable to keep score.

Lesson Outline

Objective: Students will create a game using sensing.

IMAGINE
10 minutes

First, gather as a group to introduce the theme and spark ideas.

CREATE
40 minutes

Next, help students as they make chase games, working at their own pace through the tutorial.

SHARE
5 minutes

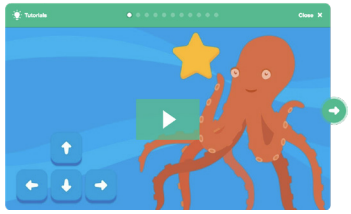
At the end of the session, gather together to share and reflect.

Get Ready for the Lesson

Use this checklist to prepare for the lesson.

☐ Preview the Tutorial

The *Make a Chase Game* tutorial shows students how to create their own projects. Preview the tutorial before your lesson and try the first few steps,



☐ Print the Activity Cards (optional)

Print a few sets of *Chase Game* cards to have available for students during the lesson. You can download the cards at: scratch.mit.edu/ideas

☐ Make sure students sign into their Scratch accounts

Have students sign into their own Scratch accounts at scratch.mit.edu.

☐ Set up computers or laptops

Arrange computers so that students can work individually or in pairs.

☐ Set up a computer with projector or large monitor

You can use a projector to show examples and demonstrate how to get started.

Imagine

IMAGINE

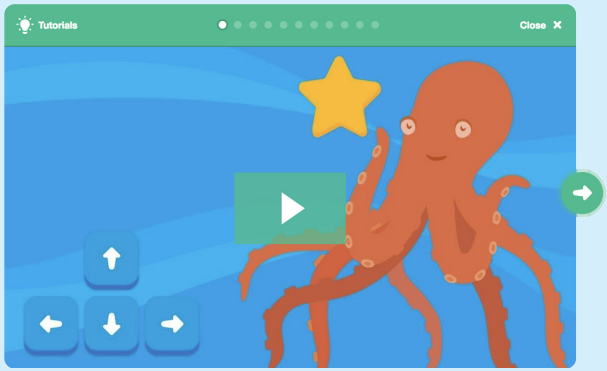
Begin by gathering the students to introduce the theme and spark ideas for projects.

Warm-up Activity: Imaginary Chase

Gather the students in a circle. Start by giving an example of one thing chasing another, such as "The dog is chasing the dinosaur." The next person adds on, such as, "The dinosaur is chasing a donut." The following person adds on by saying, "The donut is chasing a duck." or whatever creature or object they choose. Continue until each person has added on to this imaginary game of chase.

Provide Ideas and Inspiration

To spark ideas, watch the Make a Chase Game video at the start of the tutorial.



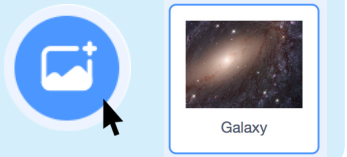
View the video at scratch.mit.edu/chase

Demonstrate the First Steps

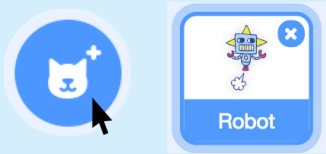
IMAGINE

Demonstrate the first few steps of the tutorial so students can see how to get started.

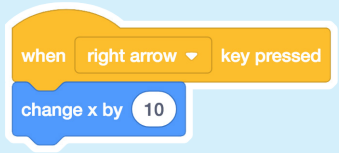
Choose a backdrop.



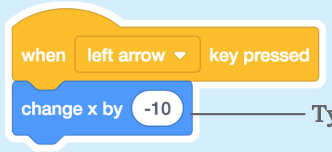
Choose a Sprite, like Robot.



Make your sprite move right and left with arrow keys.



Choose right arrow from the menu.

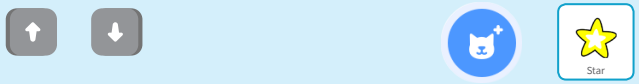


Choose left arrow from the menu.

Type a minus sign to move left.

Press the left arrow and right arrow keys on your keyboard to move.

Discuss next steps they can try, such as coding the sprite to move up and down and adding a sprite to chase.



Create

CREATE

Support students as they create catch games. Suggest working in pairs.

Start with Prompts

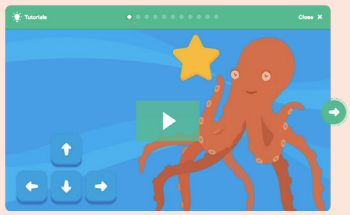
Ask students questions to get started

Which backdrop would you like to choose for your game?

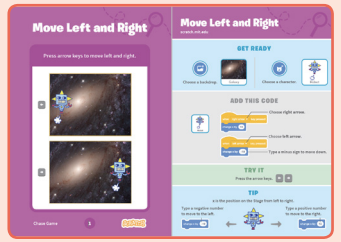
Who do you want as the main character in your game? What will it chase?

Provide Resources

Offer options for getting started



Some students may want to follow the online tutorial: scratch.mit.edu/chase



Others may want to explore using the printed cards: scratch.mit.edu/ideas

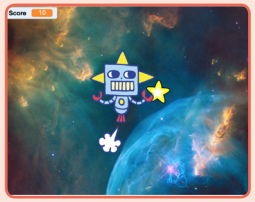
Suggest Ideas for Starting

- Choose a backdrop
- Choose or draw a main character
- Make it move with arrow keys.
- Select an object to chase.

CREATE

More Things to Try

- Code the star or other sprite to chase
- Add a variable to keep score
- Add sounds
- Add a level
- Show a message when reaching the new level



Encourage Tinkering

- Encourage students to feel comfortable trying combinations of blocks and seeing what happens.
- Suggest students look inside other chase games to see the code.
- If they find code they like, they can drag the scripts or sprites into the backpack to reuse in their own project.

Prepare to Share

To add instructions and credits to a project, click the button: *"See project page"*.

Share

SHARE

Have students share their projects with their neighbors.

Ask questions that encourage reflection:

What do you like best about your game?

If you had more time, what would you add or change?

What's Next?

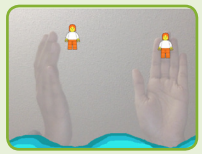
Chase Game projects provide an introduction to creating interactive games in Scratch. Here are a few ways that learners can build on the concepts they learned from this project.

Add Obstacles

For a more complex game, add obstacles to avoid. Subtract points when you hit the obstacles.

Make a Two-Player Game

For an extra challenge, make a version of the game that allows two players to play.



Video Sensing

If the computers have a web camera attached or built-in, learners can make a game that they interact by moving their bodies. See the Video Sensing tutorial and educator guide for support.

Created by the Scratch Team

EDUCATOR GUIDE

Pong Game

With this guide, you can plan and lead a 55-minute lesson using Scratch. Students will gain experience with coding as they design a bouncing ball game.

PONG GAME / EDUCATOR GUIDE

Lesson Outline

Objective: Students will develop an interactive game using variables to keep score.

IMAGINE
10 minutes

First, gather as a group to introduce the theme and spark ideas.

CREATE
40 minutes

Next, help students as they make games, working at their own pace through the tutorial.

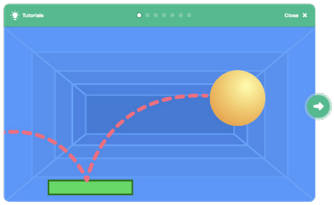
SHARE
5 minutes

At the end of the session, gather together to share and reflect.

Get Ready for the Lesson

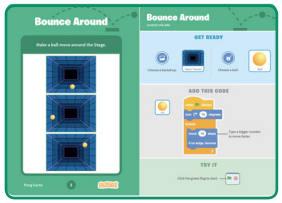
Use this checklist to prepare for the lesson.

☐ Preview the Tutorial



The *Pong Game* tutorial shows students how to create their own projects. Preview the tutorial before your lesson and try the first few steps: scratch.mit.edu/pong

☐ Print the Activity Cards (optional)



Print a few sets of *Pong Game* cards to have available for students during the lesson.
scratch.mit.edu/ideas

☐ Make sure students sign into their Scratch accounts

Have students sign into their own Scratch accounts at scratch.mit.edu.

☐ Set up computers or laptops

Arrange computers so that students can work individually or in pairs.

☐ Set up a computer with projector or large monitor

You can use a projector to show examples and demonstrate how to get started.

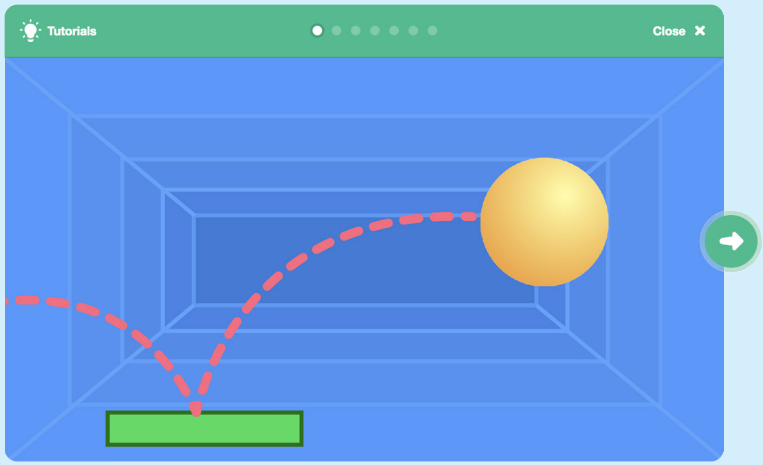
Imagine

IMAGINE

Begin by gathering the students to introduce the theme and spark ideas for projects.

Provide Ideas and Inspiration

Show the introductory video for the *Pong Game* tutorial. The video shows pong games with a variety of themes, including everything from soccer to a magic potion-themed Pong game.



View at scratch.mit.edu/pong

Warm-up Activity: Bouncing Ideas

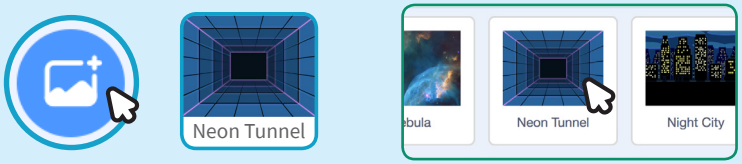
To get students thinking about a theme for their game, take turns calling out a theme, such as pizza pong or flower pong and brainstorming ideas for the type of images they could use to represent the theme.

Demonstrate the First Steps

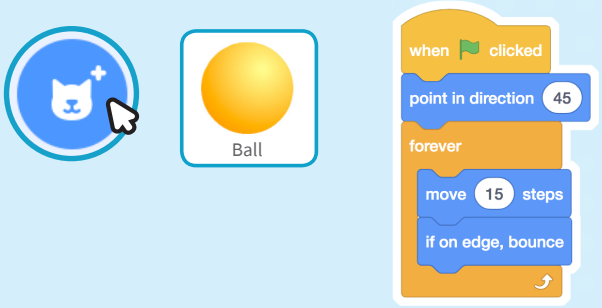
IMAGINE

Demonstrate the first few steps of the tutorial so students can see how to get started.

Go to the Scratch website. Click Create. Choose a new backdrop:



Choose a ball sprite and make it bounce around:



Add a paddle sprite and control it with the mouse:

Create

CREATE

Support students as they create pong games, on their own or in pairs.

Start with Prompts

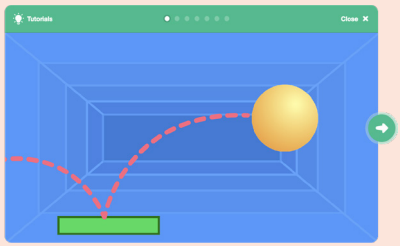
Ask students questions to get started

What background do you want for your game?

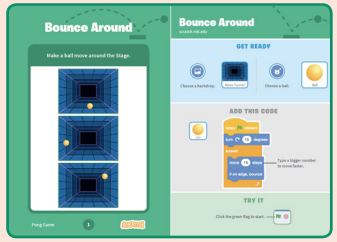
What color or type of ball?

Provide Resources

Offer options for getting started



Some particpants may want to follow the online tutorial: scratch.mit.edu/pong



Others may want to use the printed activity cards: scratch.mit.edu/ideas

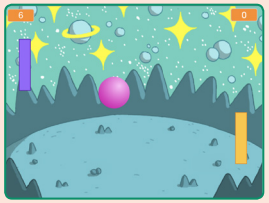
Suggest Ideas for Starting

- Choose a backdrop
- Choose or draw a ball sprite and make it bounce around
- Add a paddle sprite that you can control
- Make the ball bounce off the paddle

CREATE

More Things to Try

- Add sounds and color effects
- Keep score by adding a variable
- Add a way to win or lose the game
- Change the backdrop when you reach a certain number of points
- Duplicate the ball for an added challenge



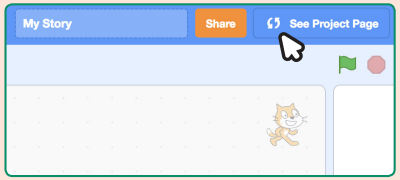
Offer strategies for problem solving

- Talk out what you're working on with someone
- Try out small bits of code at a time to figure out what's happening at each step
- Look closely at the blocks on the tutorial or activity cards to see if they are the same or different from the blocks you're using
- Look at the code for other pong games on the Scratch site

Prepare to Share

To add instructions and credits to a project, click the button: "See project page".

Then click the Share button if you want the project visible to others online.



Share

SHARE

Have participants share their projects with others in the room.

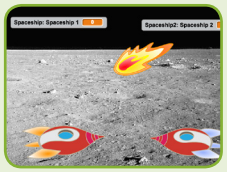
Ask questions to encourage reflection:

What did you notice about the games you tried?

What ideas might you add to your game?

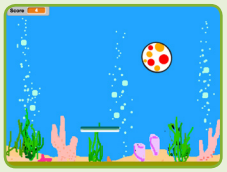
What's Next?

Here are a couple of other directions you could suggest:



Two-Player Game

For a more advanced project, try making a two-player game. To make a new version of your own project, click **File > Save as a Copy.**



Remix a Game

A different way to make a pong game is to remix someone else's project, adding images and ideas. Find a project to remix in the **Pong Game Studio**: scratch.mit.edu/studios/644508/ Click '**See inside**', then click the '**Remix**' button.

Scratch is a project of the Lifelong Kindergarten Group at the MIT Media Lab.

EDUCATOR GUIDE

Create a Story

With this guide, you can plan and lead a 55-minute lesson using Scratch. Students will create a story with settings, characters, and dialogue.

CREATE A STORY / EDUCATOR GUIDE

Lesson Outline

Objective: Students will create an animated story between at least two characters.

IMAGINE
10 minutes

First, gather as a group to introduce the theme and spark ideas.

CREATE
40 minutes

Next, help students as they create story projects, working at their own pace through the tutorial.

SHARE
5 minutes

At the end of the session, gather together to share and reflect.

Get Ready for the Lesson

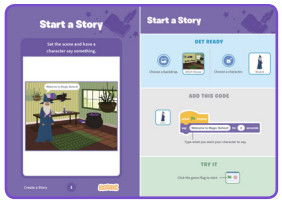
Use this checklist to prepare for the lesson.

☐ Preview the Tutorial

The *Create a Story* tutorial shows students how to create their own projects. Preview the tutorial before your lesson and try the first few steps: scratch.mit.edu/story

☐ Print the Coding Cards (optional)

Print a few sets of *Create a Story* cards to have available for students during the lesson. You can download from this page: scratch.mit.edu/ideas



☐ Make sure students sign into their Scratch accounts

Have students sign into their own Scratch accounts at scratch.mit.edu.

☐ Set up a studio for project sharing on Scratch

Set up a studio so students will be able to add their projects. Go to your ***My Stuff*** page, then click the ***+New Studio*** button. Type in a name for the studio.

☐ Set up computers or laptops

Arrange computers so that students can work individually or in pairs.

CREATE A STORY / EDUCATOR GUIDE

Imagine

IMAGINE

Begin by gathering the students to introduce the theme and spark ideas for projects.

Warm-up Activity: Story Starters in a Bag

Have students make up a brief story by giving them a bag with three objects in it, and asking them to include all of the items in the story. In each bag, you could include small objects, pictures of animals or characters, and/or words (people, places, or things). Divide students into groups of two or three, and have each pick a bag. Give them a few minutes to come up with a quick story.

Provide Ideas and Inspiration

You can show the Create a Story tutorial video to show students how they can start making stories in Scratch.

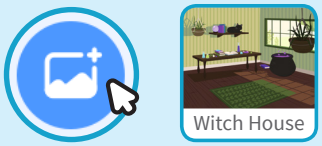
View the video at: scratch.mit.edu/story

Demonstrate the First Steps

IMAGINE

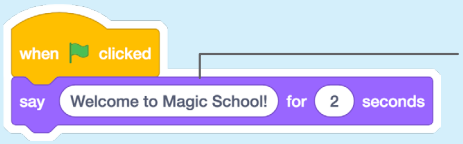
Demonstrate the first few steps of the tutorial so students can see how to get started.

In Scratch, click Create. Choose a backdrop.



Choose any character (in Scratch called a *sprite*).

Code your character to say something.



Type what you want your character to say.

Click the green flag to start.

Add another character.

Add code to the new character.

Use this block to have the second character wait before they say something.

Create

CREATE

Support students as they create Story projects, on their own or in pairs.

Start with Prompts

Ask students questions to get started

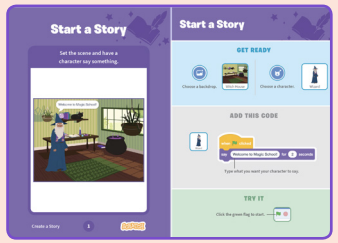
Where will your story take place?

What will happen first?

Provide Resources

Offer options for getting started

Some students may want to follow the online tutorial: scratch.mit.edu/story



Others may want to explore using the coding cards: scratch.mit.edu/ideas

Suggest Ideas for Starting

- Choose a backdrop.
- Choose a character.
- Make a character say something
- Make a character hide and show.

CREATE

More Things to Try

- Switch backdrops.
- Make your characters have a conversation.
- Move your characters.
- Change something when you click on it.

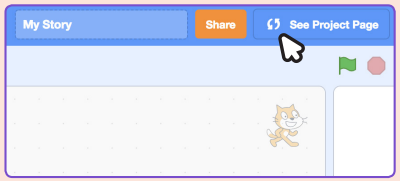
Support Tinkering

Scratch is designed to support creating by experimenting and tinkering. So, your students may want to start their stories without planning beforehand. As they create, one idea can spark another. Celebrate their sparks of creativity and the unexpected turns their stories may take.

Prepare to Share

To add instructions and credits to a project, click the button: "See project page".

Then click the Share button if you want the project visible to others online.



Share

SHARE

Help the students add their projects to a shared studio in Scratch. Give them a link to the studio. Then they can click 'Add Projects' at the bottom of the page.

Ask for volunteers to show their project to the group.

What's Next?

Students can use these ideas and concepts to create a variety of projects. Here are some variations on the story project you could suggest:

Retell a story

Start with a story you know and make it in Scratch. Imagine a new ending or a different setting.

Neighbourhood story

Take photos of your classroom, school, or neighborhood and use them as backdrops in your story.

Round-robin story

Give everyone 5 minutes to start a story. Then, have them switch to the next computer to add to the story. Repeat.

Created by the Scratch Team