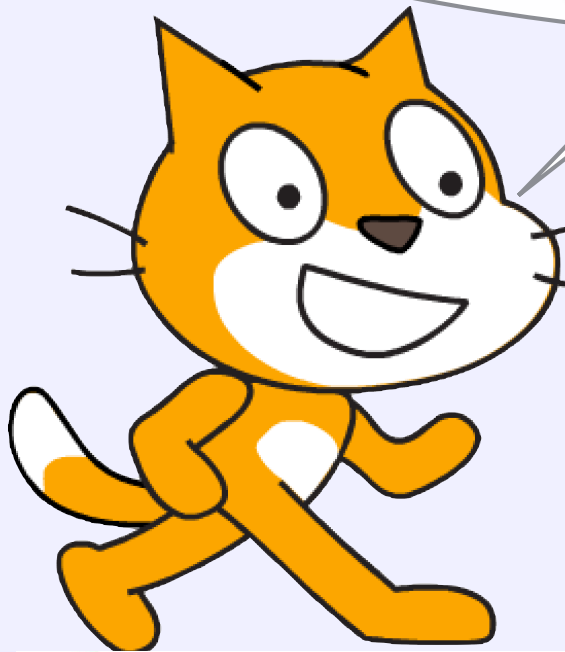


Scratch that Coding Itch with Scratch!



Barry Webster, computer science teacher
bwebster@barrywebster.com
<http://barrywebster.com>



Michigan
Computer
Science
Teachers
Association



A Chapter of the
Computer Science
Teachers Association



Scratch that Coding Itch

- Install scratch offline editor
- Scratch, from Scratch
- Guide your students' interests
- Where to go after Scratch?
- How Scratch compares to other programming languages
- Hour of Code K-8 Resources and HS resources
- Resources

Introductions

- Name
- Grades and/or subjects you teach
- Where you teach
- What would you most like to get from this workshop?



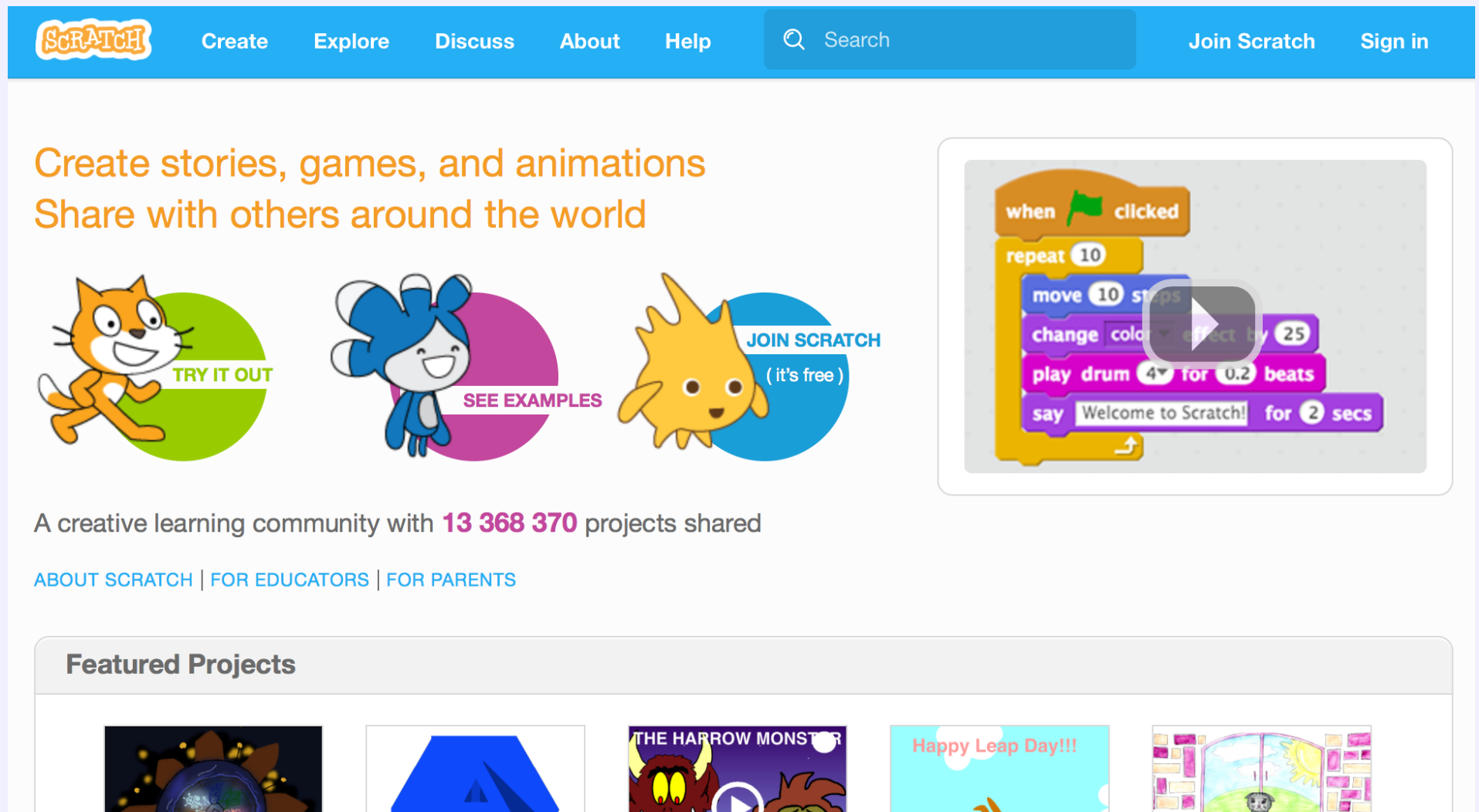


Install Offline Editor.

- If you don't want to login to the scratch website
- If your Internet access has glitches (or you want to save your files locally)
- If you want capability to work offline
- If you want to use an older version (a standard app)
- To have experience for setting up student computers

To Install Offline Editor..

- Go to the scratch website <https://scratch.mit.edu/>



The image shows the Scratch website homepage. At the top is a blue navigation bar with the Scratch logo, links for 'Create', 'Explore', 'Discuss', 'About', and 'Help', a search bar, and links for 'Join Scratch' and 'Sign in'. Below the navigation bar, the main content area features the text 'Create stories, games, and animations' and 'Share with others around the world'. There are three circular icons: a cat icon with 'TRY IT OUT', a blue character icon with 'SEE EXAMPLES', and a yellow character icon with 'JOIN SCRATCH (it's free)'. To the right is a video player showing a Scratch script with blocks: 'when green flag clicked', 'repeat 10', 'move 10 steps', 'change color effect by 25', 'play drum 4 for 0.2 beats', and 'say Welcome to Scratch! for 2 secs'. Below this is the text 'A creative learning community with 13 368 370 projects shared' and links for 'ABOUT SCRATCH | FOR EDUCATORS | FOR PARENTS'. At the bottom is a 'Featured Projects' section with five project thumbnails.

Scratch

Create Explore Discuss About Help Search Join Scratch Sign in

Create stories, games, and animations
Share with others around the world

TRY IT OUT

SEE EXAMPLES

JOIN SCRATCH
(it's free)

when green flag clicked
repeat 10
move 10 steps
change color effect by 25
play drum 4 for 0.2 beats
say Welcome to Scratch! for 2 secs

A creative learning community with **13 368 370** projects shared

[ABOUT SCRATCH](#) | [FOR EDUCATORS](#) | [FOR PARENTS](#)

Featured Projects

THE HARROW MONSTER

Happy Leap Day!!!

To Install Offline Editor..

- Go to the scratch website <https://scratch.mit.edu/>
- Scroll to the bottom of the page

About

[About Scratch](#)
[For Parents](#)
[For Educators](#)
[Credits](#)
[Jobs](#)
[Press](#)

Community

[Community Guidelines](#)
[Discussion Forums](#)
[Scratch Wiki](#)
[Statistics](#)

Support


[Help Page](#)
[FAQ](#)
[Offline Editor](#)
[Contact Us](#)
[Donate](#)

Legal

[Terms of Use](#)
[Privacy Policy](#)
[DMCA](#)

Scratch Family

[ScratchEd](#)
[ScratchJr](#)
[Scratch Day](#)
[Scratch Conference](#)
[Scratch Foundation](#)

English 

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

To Install Offline Editor..

- Go to the scratch website <https://scratch.mit.edu/>
- Scroll to the bottom of the page
- Click on Offline Editor

About

[About Scratch](#)
[For Parents](#)
[For Educators](#)
[Credits](#)
[Jobs](#)
[Press](#)

Community

[Community Guidelines](#)
[Discussion Forums](#)
[Scratch Wiki](#)
[Statistics](#)

Support

[Help Page](#)
[FAQ](#)
[Offline Editor](#)
[Contact Us](#)
[Donate](#)

Legal

[Terms of Use](#)
[Privacy Policy](#)
[DMCA](#)

Scratch Family

[ScratchEd](#)
[ScratchJr](#)
[Scratch Day](#)
[Scratch Conference](#)
[Scratch Foundation](#)

English



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

To Install Offline Editor..

- Follow instructions 1 and 2 on page

Scratch 2 Offline Editor

You can install the Scratch 2.0 editor to work on projects without an internet connection. This version will work on Mac, Windows, and some versions of Linux (32 bit).

Note for Mac Users: the latest version of Scratch 2.0 Offline requires Adobe Air 20. To upgrade to Adobe Air 20 manually, go [here](#).

Adobe AIR	Scratch Offline Editor	Support Materials
1	2	3
If you don't already have it, download and install the latest Adobe AIR	Next download and install the Scratch 2.0 Offline Editor	Need some help getting started? Here are some helpful resources.
Mac OS X - Download ↓ Mac OS 10.5 & Older - Download ↓	Mac OS X - Download ↓ Mac OS 10.5 & Older - Download ↓	Starter Projects - Download ↓ Getting Started Guide - Download ↓
Windows - Download ↓ Linux - Download ↓	Windows - Download ↓ Linux - Download ↓	Scratch Cards - Download ↓



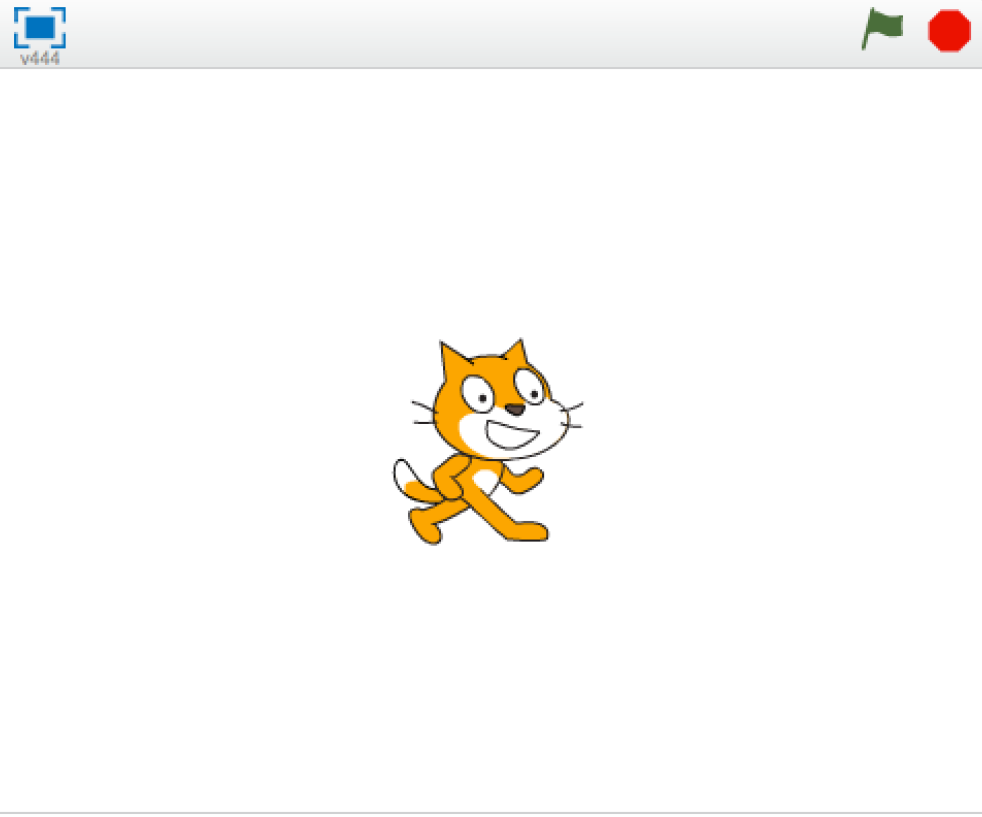
Scratch, from Scratch

- Computer Science and Scratch
- Editor tips
- Build a project, with add-ons
- (...)
- How might you guide students using Scratch?
- Learning more Scratch
- Scratch teaching resources

Now, let's explore

- Launch your offline editor **OR**
- Sign in (click “Sign in” at top right of scratch webpages) so you can save projects, then click “create” at top left for online editor.

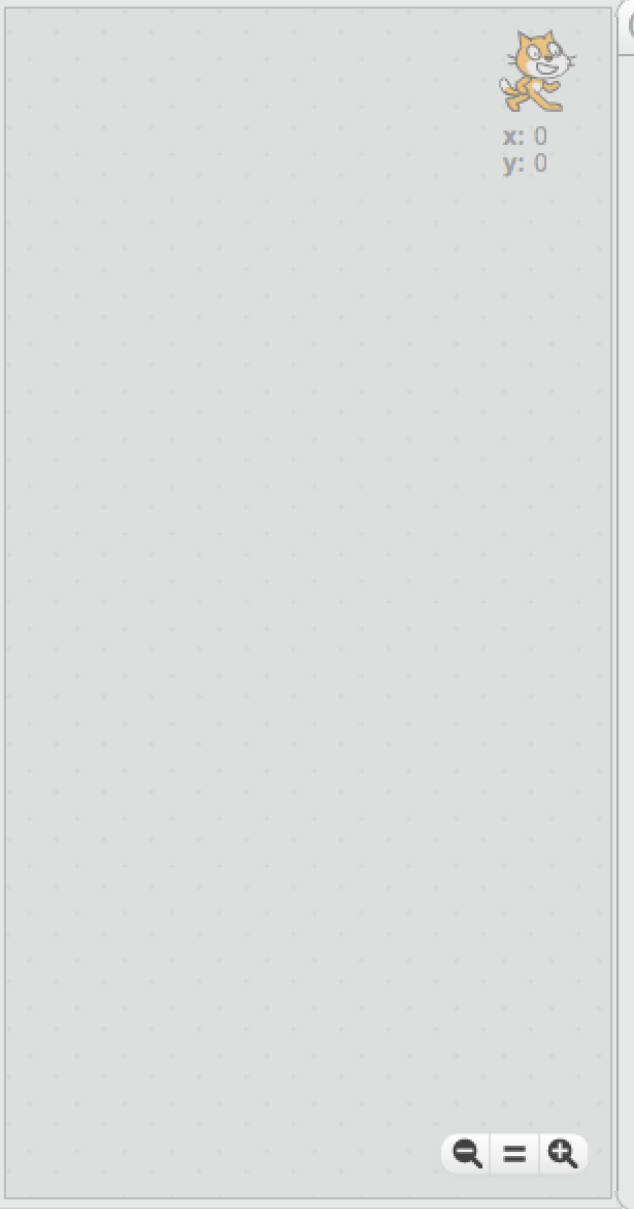




Scripts Costumes Sounds

- Motion
- Looks
- Sound
- Pen
- Data
- Events
- Control
- Sensing
- Operators
- More Blocks

```
move 10 steps
turn 15 degrees
turn 15 degrees
point in direction 90
point towards
go to x: 0 y: 0
go to mouse-pointer
glide 1 secs to x: 0 y: 0
change x by 10
set x to 0
change y by 10
set y to 0
if on edge, bounce
```



Sprites

New sprite: [cat icon] [brush icon] [upload icon] [camera icon]

Stage 1 backdrop

New backdrop: [image icon] [brush icon] [upload icon] [camera icon]

Sprite1

Scratch 2 Offline Editor

Scratch File Edit Tips About

v444

Scripts Costumes Sounds

Motion Events
Looks Control
Sound Sensing
Pen Operators
Data More Blocks

move 10 steps
turn 15 degrees
turn 15 degrees
point in direction 90
point towards
go to x: 0 y: 0
go to mouse-pointer
glide 1 secs to x: 0 y: 0
change x by 10
set x to 0
change y by 10
set y to 0
if on edge, bounce

x: -130 y: 10

Sprites New sprite: [] [] [] []

Stage 1 backdrop
New backdrop: [] [] [] []

Sprite1

x: 0 y: 0

Show Tips

Show Tips


The image shows the Scratch 2 Offline Editor interface. The main stage area contains a Scratch cat sprite. The Scripts area on the right lists various motion blocks. The Sprites area at the bottom left shows the current sprite, 'Sprite1'. Two blue callout boxes with red borders and the text 'Show Tips' are overlaid on the interface. One callout points to the 'Tips' menu item in the top navigation bar. The other callout points to a small question mark icon in the top right corner of the Scripts area.

Tips

Scratch 2 Offline Editor

Scratch File Edit Tips About

Untitled v444



Scripts Costumes Sounds

- Motion
- Looks
- Sound
- Pen
- Data
- Events
- Control
- Sensing
- Operators
- More Blocks

move 10 steps

turn 15 degrees

turn 15 degrees

point in direction 90

point towards

go to x: 0 y: 0

go to mouse-pointer

glide 1 secs to x: 0 y: 0

change x by 10

set x to 0

change y by 10

set y to 0

if on edge, bounce

x: 28 y: 129

New sprite: [cat] [pen] [upload] [camera]

Sprites

Stage 1 backdrop

New backdrop: [image] [upload] [camera]

Sprite1

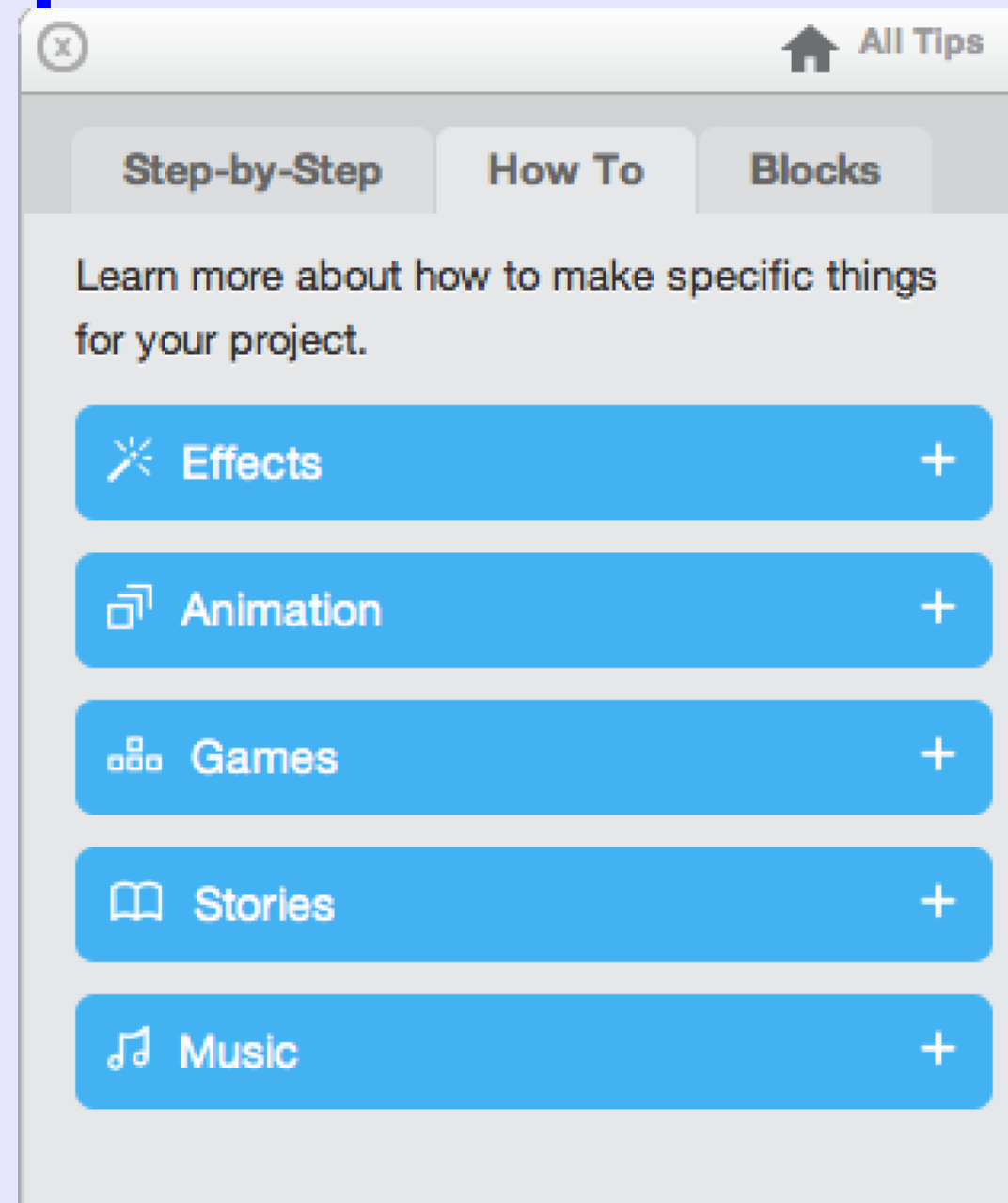
All Tips

Step-by-Step How To Blocks

Follow these tutorials to get started with your project.

- Getting Started with Scratch
- Animate Your Name
- Design a Valentine
- Dance, Dance, Dance
- Create a Pong Game
- Race to the Finish
- Hide-and-Seek Game
- Favorite Things

Tips



Tips

⌵ All Tips

Step-by-Step How To **Blocks**

Dig deeper here to find out more about each block, or use the ? button in the editor to click on individual blocks to learn more.

- Motion +
- Looks +
- Sound +
- Pen +
- Data +
- Events +
- Control +
- Sensing +
- Operators +
- More Blocks +
- Extensions +

There are several ways to learn scratch

- Explore the editor and its tips
- Scratch website resources
- Other websites
- Workshops

from <https://www.cs-first.com>



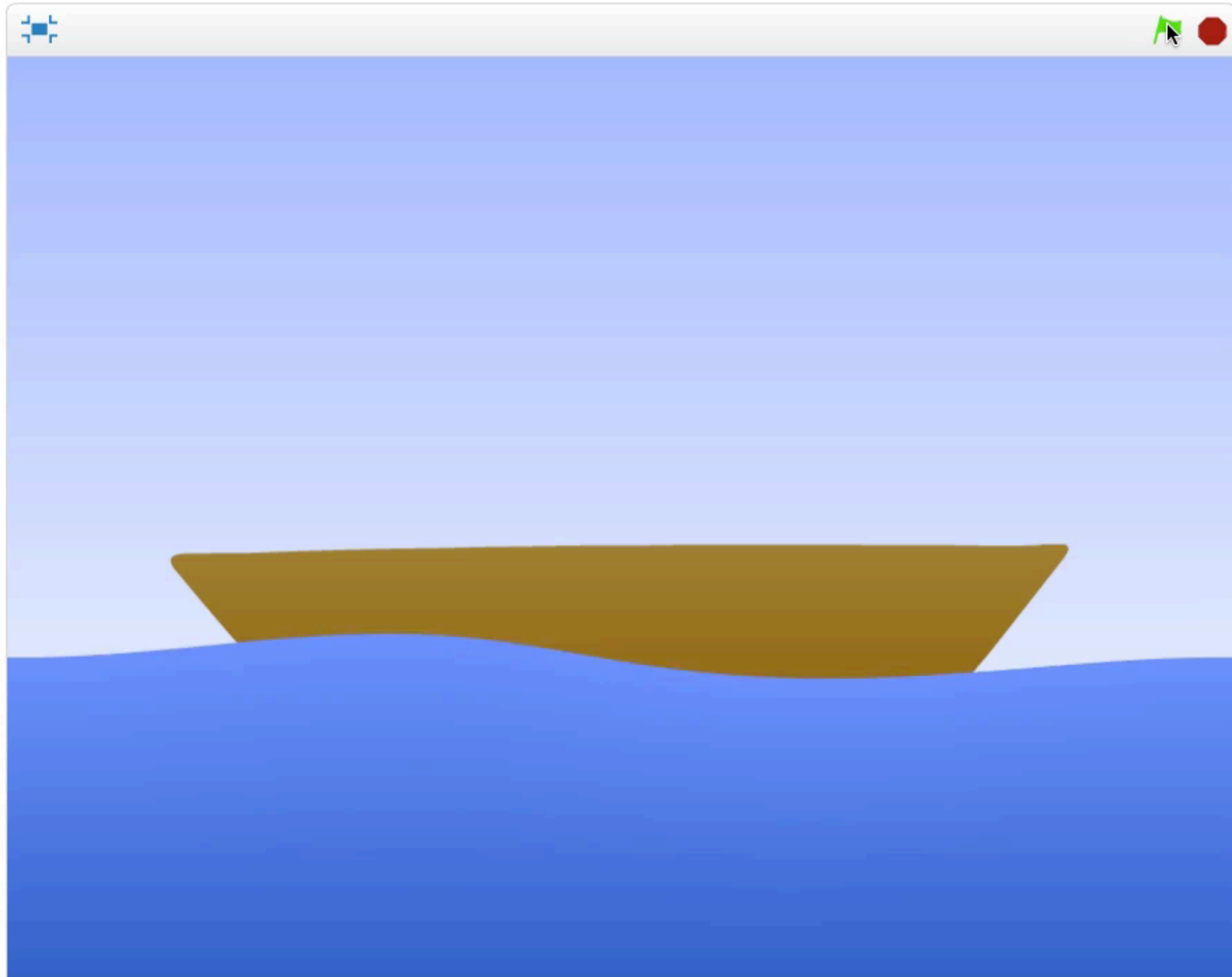
Sample Activity

High Seas Introduction

from <https://www.cs-first.com>

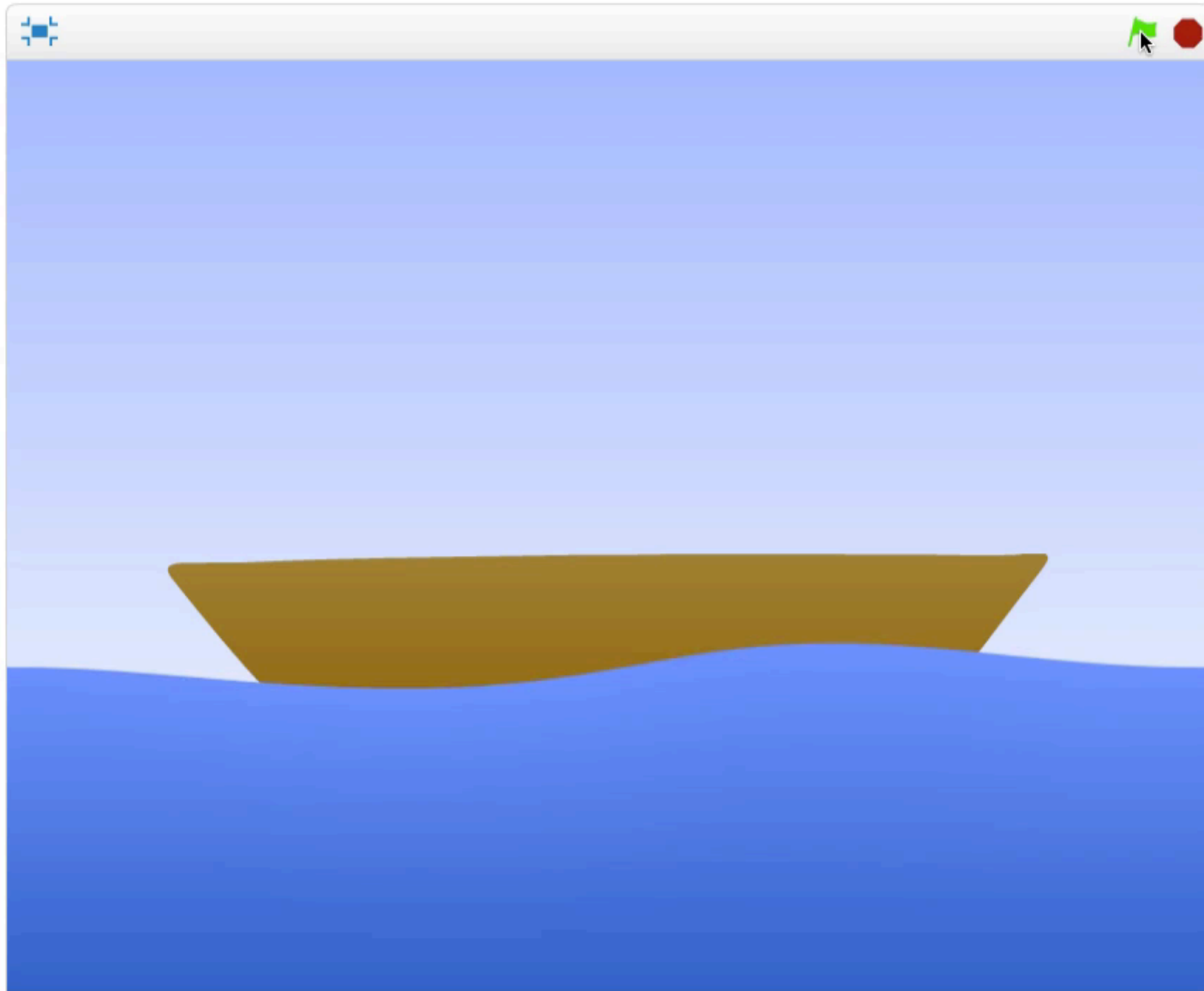
This video will guide you through opening a starter project and creating a Scratch sign in!

from <https://www.cs-first.com>



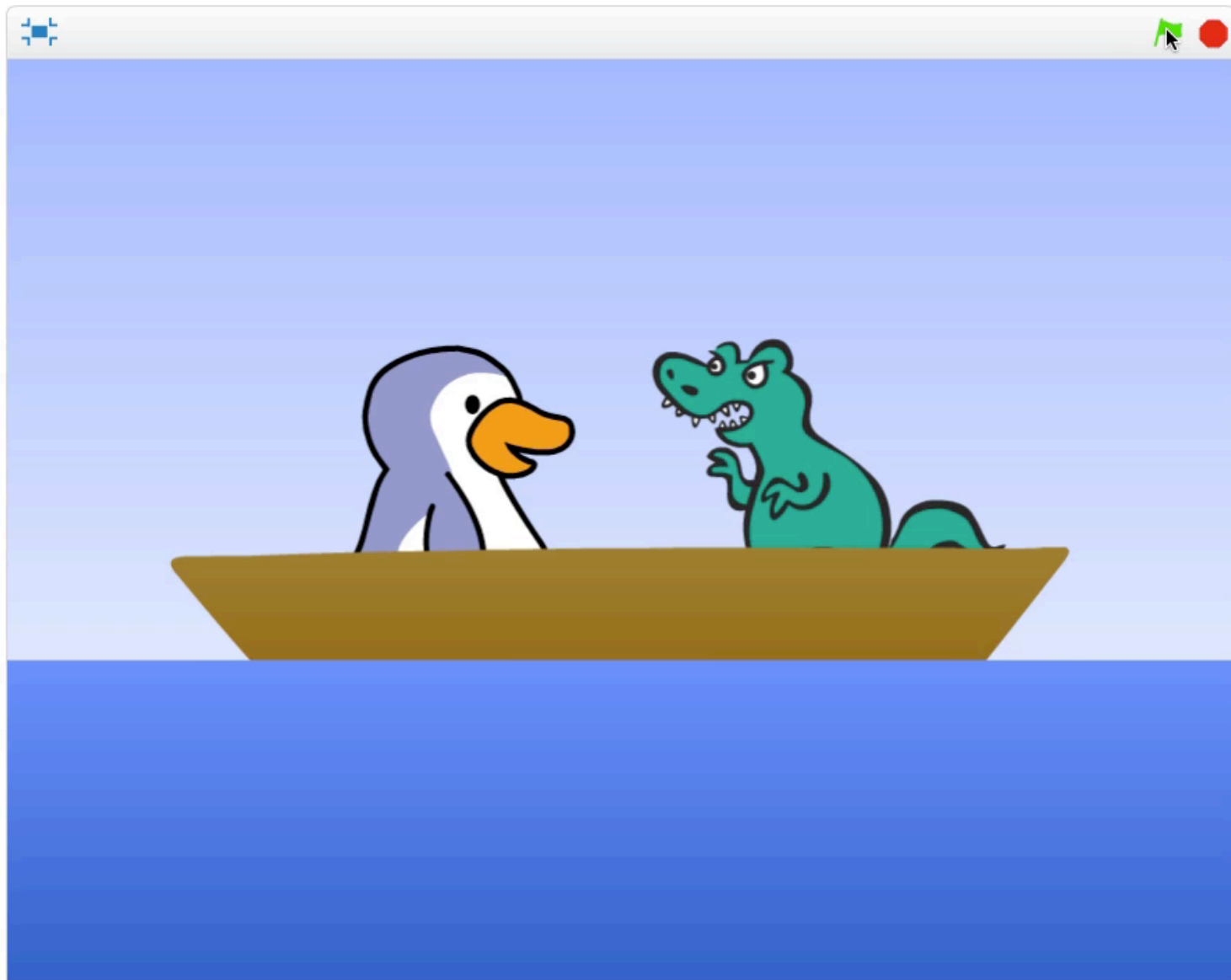
Video 3

from <https://www.cs-first.com>



Video 4

from <https://www.cs-first.com>











Video 5

from <https://www.cs-first.com>

Congratulations!

[https://www.cs-first.com/ studentpage/activity-add-ons](https://www.cs-first.com/studentpage/activity-add-ons)

<p>Sea Sounds</p>  <p>Add sound effects to your story.</p> <p>Sounds .</p>	<p>Sun Ray Animation</p>  <p>Draw and animate a sun sprite.</p> <p>SunRay .</p>	<p>Sink the Ship</p>  <p>Program a sinking ship in your story.</p> <p>SinkShip .</p>
<p>Second Scene</p>  <p>Program a Second Scene for your Story.</p> <p>2ndScene .</p>	<p>Clouds</p>  <p>Draw and program clouds that move.</p> <p>Clouds .</p>	<p>Gamify</p>  <p>Program one or more sprites to move when keys on the keyboard are pressed.</p> <p>Gamify .</p>

Navigation:  **1** **2** **3** **4** **5** **6** Wrap Up  Next

Sea Sounds



Sea Sounds

Sun Ray Animation



Sink the Ship



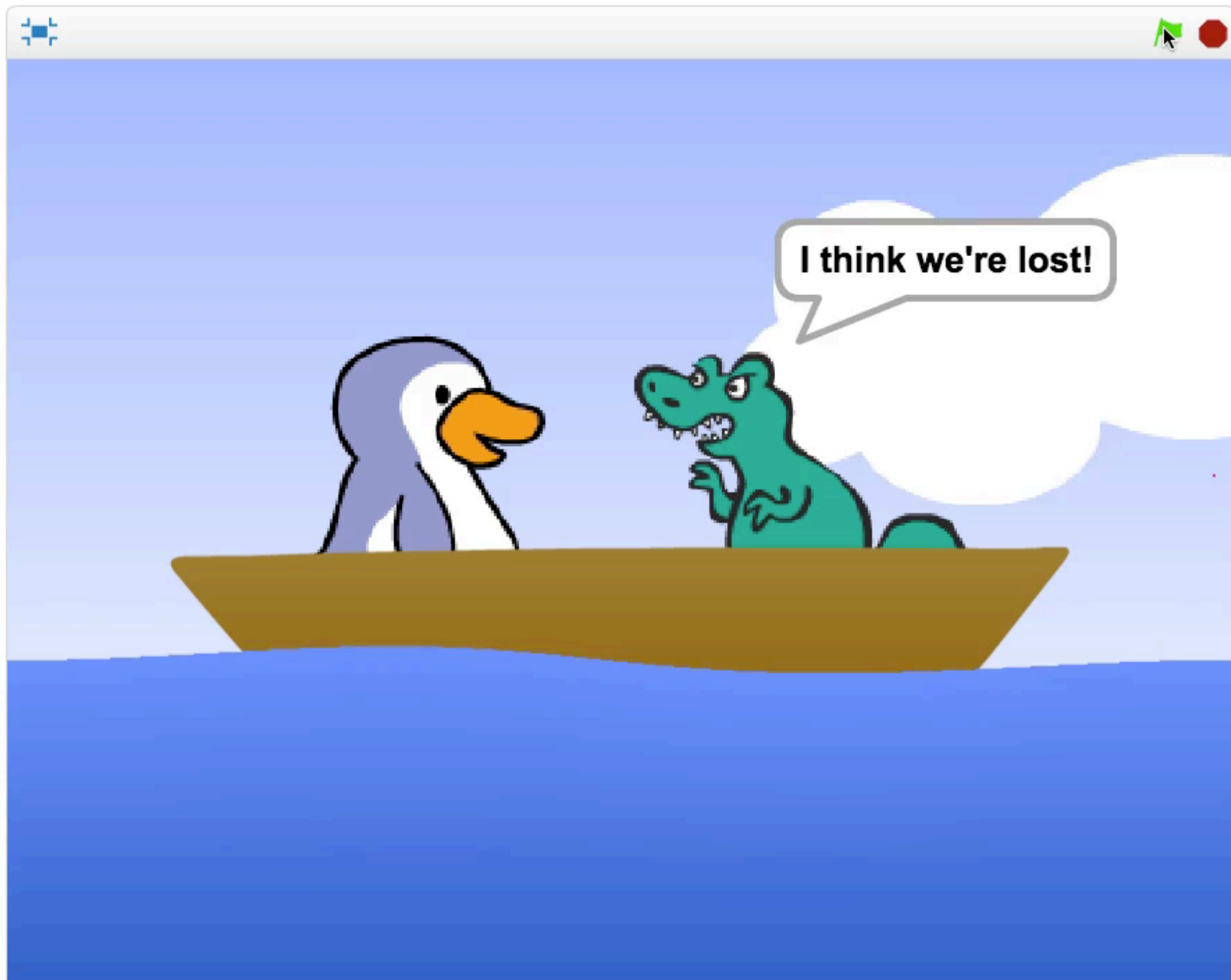
Sink the Ship!

Second Scene



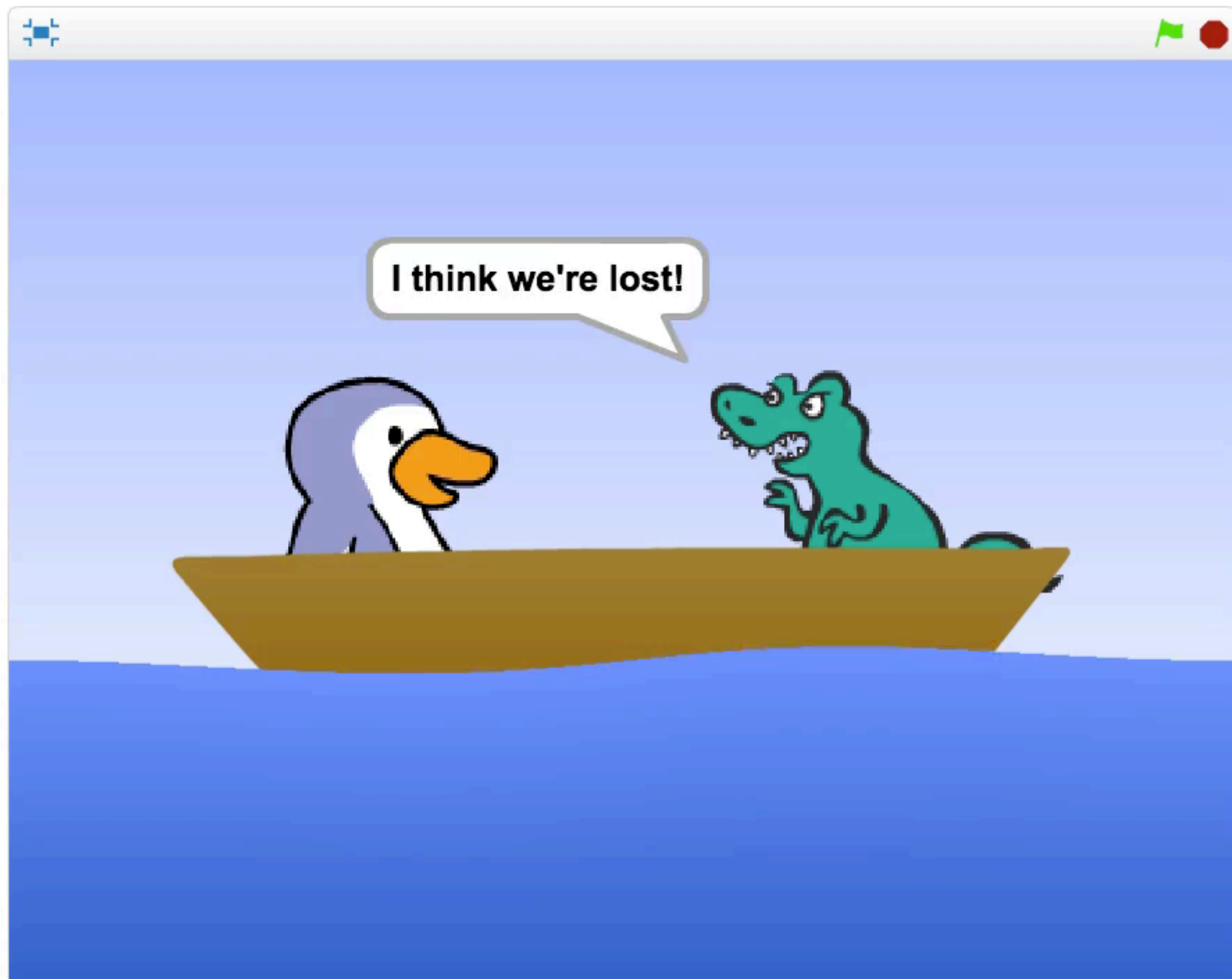
Second Scene

Clouds

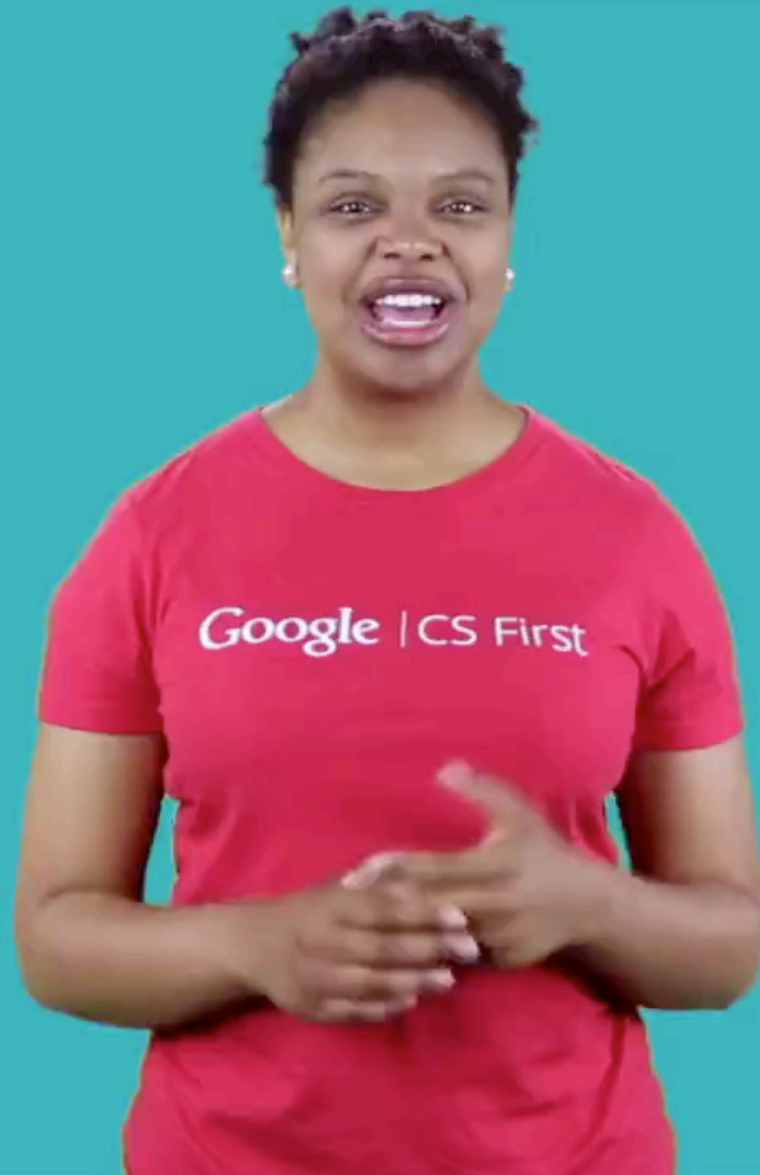


Clouds

Gamify



High Seas Wrap Up





Guide your students' interests

- Learn more about Scratch so you can
- Explore Scratch teaching materials so you can

Explore Scratch

- Explore the Scratch editor
- Explore <https://scratch.mit.edu/>

Explore Scratch Teaching

- Explore CS First: <https://www.cs-first.com/materials>

Google CS First

Students: Try Now!

Sign in

[About](#) [Start a Club](#) [Volunteer](#) [Advocate](#) [View Materials](#) [Resources](#) [Help](#)

View Materials



High Seas Activity

Sample CS First with "High Seas," an introductory activity designed for use in a classroom setting or at a conference, hackathon, or other event. "High Seas" is a one-time, standalone activity and not part of a regular CS First theme or club, so it does not use or provide printed materials, usernames, or passwords.

[Try Now](#)

[View Club Plans](#)



Storytelling

In Storytelling, students use computer science to tell fun and interactive stories. Storytelling emphasizes creativity by encouraging club members to tell a unique story each day.

Explore Scratch Teaching

- <https://www.cs-first.com/training/welcome-cs-first>

Training Materials

Welcome to CS First

[Celebrate Student Work using the Showcase Selector](#)

[Overview of a Club Day](#)

[Making CS First Work for You](#)

[How to Prepare for your Club Day](#)

[How to Work with a Self-Paced Club](#)

[Setting Up Computers for CS First](#)

[Print Materials](#)

Welcome to CS First

Google Computer Science First



Explore ScratchEd

- <http://scratched.gse.harvard.edu/>



[Join](#) [Sign In](#) [About](#) [Help](#) [Contact](#)

Stories

Resources

Discussions

Members

Events

What is Scratch?

Scratch is a programming language that makes it easy to create interactive art, stories, simulations, and games – and share those creations online.

[Learn more »](#)

What is ScratchEd?

ScratchEd is an online community where Scratch educators:



share stories



exchange
resources



ask questions



find people



Get Started with Scratch

Imagine the creative possibilities with Scratch and the online community in this intro video.

Explore ScratchEd Resources

- <http://scratched.gse.harvard.edu/resources>



[Join](#) [Sign In](#) [About](#) [Help](#) [Contact](#)

All

GO ▶

Stories

Resources

Discussions

Members

Events

Resources Home

Explore Resources

Featured Resources



Scratch Educator Meetup Guide

Contributed by [ScratchEd Team](#), January 29, 2015

A guide to organizing local Scratch educator meetups

Content Types: Advocacy Material, Handout, Reference Guide

Education Level: Professional Development

Curricular Areas: Teacher Education

4 Comments 20 Bookmarks

More Featured Resources

[Scratch Curriculum Guide](#)

Contributed by: [ScratchEd Team](#)

[Best of both worlds: Issues of structure and agency in computational creation, in and out of school](#)

Contributed by: [Karen Brennan](#)

[Hour of Code Activity & Handout: Interactive Holiday Card](#)

Contributed by: [ScratchEd Team](#)

[View More »](#)

Explore ScratchEd Resources

Search Resources

Education Level

Preschool and Kindergarten (262)
Elementary School (517)
Middle School (478)
High School (370)
College and University (319)
Professional Development (355)
Other (253)

Content type

Activity (338)
Advocacy Material (54)
Assessment (86)
Audio and Video (158)
Curriculum (111)
Handout (210)
Lesson Plan (187)
Presentation (107)
Reference Guide (51)
Research (67)
Sample Scratch Project (249)
Textbook (40)
Tool (50)
Tutorial (187)
Website (253)
Other (54)

Curricular Area

Computer Science (539)
Engineering (241)
Language Arts (273)
Mathematics (268)
Music (212)
Science (244)
Social Studies (222)
Teacher Education (305)
Technology (386)
Visual Arts (248)
Other (271)

Language

English (532)
Español (128)
Français (22)
Deutsch (9)
Português (12)
Ελληνικά (3)
Nederlands (Dutch) (7)
日本語 (1)
正體中文 (3)
Other (35)

Explore ScratchEd Resources

- <http://scratched.gse.harvard.edu/guide/files/CreativeComputing20141015.pdf>



CREATIVE COMPUTING

Karen Brennan | Christan Balch | Michelle Chung

Harvard Graduate School of Education

Guide your students' interests

- You may want to use other tools; where to go from Scratch
- What to consider when choosing



Where to go from Scratch



How Scratch compares to other programming languages

- Scratch is a graphical block language
- Several languages for beginning students use graphical blocks
- Several are more specialized (less general)

How Scratch compares to other programming languages

- Traditional computer languages are text based
 - Require precision
 - Can require patience for beginners
 - Are used by professionals
 - Are used in high school and college computer science

How Scratch compares to other programming languages

- **Lowers the bar to start programming**
 - Drag and drop (instead of pesky semicolons;)
 - Quick feedback and development
 - Visuals are easier
 - Sound is easier
- Find the right examples to engage students

Which programming environment(s)?

- Is it suitable for your students (interests, age, and stage of development)
- Will it work well on computing equipment (at school, home, personal)
- How easy is it to learn compared to how much learning is possible (or likely)
- How much does it and teaching materials cost

Which programming environment(s)?

- Is it under active development
- Who is the developer
- How widely is it being used
- How “professional” is it



Hour of Code tutorials

<https://code.org/educate/allhourofcode>

- Student-Led Hour of Code Tutorials
 - Tutorials that teach JavaScript
 - [Tutorials for Grades K-8](#)
 - Tutorial apps for phones and tablets
 - Tutorials in other programming languages
 - Make games or apps
- Teacher Led Hour of Code Lesson Plans
 - Elementary School
 - Middle School
 - [High School](#)
- Other activities



Hour of Code K-8 tutorials

<https://code.org/educate/allhourofcode>

Tutorials for Grades K-8

Tynker

Ages 5-13. Modern web browsers, iPad, Android. Learn to code by solving fun puzzles and build your own games.

- The Tynker Visual Programming Language is based on Open Web standards (HTML5, JavaScript, CSS) and works seamlessly across Web browsers and natively on mobile platforms (Android, iOS).

Scratch

Ages 8+. Desktop-only web browsers. Create your own interactive games, stories, and animations with Scratch!

- Offline editor runs on desktops and requires Adobe AIR. (Older version runs on older computers. ScratchJr runs on iPads.)

Lightbot

Ages 5-13. ALL browsers and iOS, Android, or Game Console. Program Lightbot to solve puzzles using procedures and loops!

- Solve Puzzles using Programming Logic

The Foos

Elementary (Pre-readers welcome). Modern web-browsers, iOS, Android. A fun game to learn about programming.

- Concepts: Problem Recognition, Conditionals, Critical Thinking, Perseverance, Sequencing, Algorithms, Loops, Commands/Parameters

Hour of Code K-8 tutorials

<https://code.org/educate/allhourofcode>

Tutorials for Grades K-8

Kodable

Elementary (Pre-readers welcome). Modern web-browsers, iPad. A fun iPad game to teach computer programming concepts.

- **Grab and Go K-5 Solution. No programming experience needed**

Monster Coding

Ages 5-13. Modern web browsers, iOS, Android. A colorful self-guided programming adventure for children.

- **We think coding is a great way to learn many things, not just programming. So we've incorporated a lot of cool math learning blocks, as well as shapes and patterns.**

AllCanCode

Ages 5-10. Modern web browsers, iOS. An immersive game to guide Marco with a visual programming language.

- **Run Marco! A coding adventure around the world.**

CS First

Ages 9-14. Modern web-browsers. Animate a story about two characters on the ocean. Add your own style!

- **Theme-Based Clubs**
- **Each CS First club is based on a real-world theme and offers about 10 hours worth of lessons and activities. The different club themes aim to attract and engage students of varying backgrounds and interests. All materials are targeted at students in 4th - 8th grades (or between the ages of 9 - 14) and are free and easy to use.**
- **CS First is a free program that increases student access and exposure to computer science (CS) education through after-school, in-school, and summer programs. All clubs are run by teachers and/or community volunteers.**
- **Our materials: Are completely free and available online, are targeted at students in grades 4th-8th (ages 9-14), can be tailored to fit your schedule and needs, involve block-based coding using Scratch and are themed to attract students with varied interests**

Hour of Code K-8 tutorials

<https://code.org/educate/allhourofcode>

Tutorials for Grades K-8

Inside Out - Made With Code

Ages 9-14. Modern web-browsers. Help Riley from the Pixar animated movie Inside Out, write code to help her make it past some of the life challenges she experiences during the movie.

- Less than 1% of girls study Computer Science. Let's change that.

NCLab: Karel the Robot

Elementary. Web-based. Learn basic concepts of Computer Science by typing programs for a robot.

- NCLab's mission to bring access and equity in STEM education to all learners, especially the under served, by providing them with technological opportunities that engage them, ignite their curiosity, and personalize & tailor their learning. NCLab makes sure that its users achieve their STEM education goals through the use of innovative self-paced online courses in computer programming, 3D modeling, and other essential STEM subjects. NCLab is passionate about training and supporting teachers who are and will remain an indivisible part of the educational process.

Alice Project

Elementary. Desktop or Game Console. Create an Alice animation with Garfield the Cat using two tutorials: Tutorial 1 sets up the scene. Tutorial 2 writes the program code.

- Alice is an innovative 3D programming environment that makes it easy to create an animation for telling a story, playing an interactive game, or a video to share on the web. Alice is a freely available teaching tool designed to be a student's first exposure to object-oriented programming. It allows students to learn fundamental programming concepts in the context of creating animated movies and simple video games. In Alice, 3-D objects (e.g., people, animals, and vehicles) populate a virtual world and students create a program to animate the objects.

Hour of Code K-8 tutorials

<https://code.org/educate/allhourofcode>

Tutorials for Grades K-8

Coding Pirates

Elementary. Web-based, Android, iOS. Learn to code with Captain Hack by visually programming with blocks.

- Requires app or unity webplayer-mini plugin.
- **Thimble** is an online code editor that makes it easy to create and publish your own web pages while learning HTML, CSS & JavaScript.



Hour of Code HS tutorials

<https://code.org/educate/allhourofcode>

High School

STEM Projects

Ages 5-18. Programming, Science (Ecology), Science (Space), Web-based. Code and animate a Solar System simulation, an interactive ecological pyramid, a working analog clock, and more.

- uses Tynker, at <https://www.tynker.com/hour-of-code/tynker-stem-teacher-guide.pdf>

CodeHS Pixel Art

Ages 14-18. Art, Math (Coordinates), Unplugged. Students learn about coordinates, what pixels are, and how to create drawings by setting pixels to be different colors.

- pdf pixel art resources with blanks, by CodeHS, <https://codehs.com>

10 Minutes of Code using a T.I. Calculator

Ages 13-18. Math (Algebra), Math (Functions), Programming, Unplugged, TI-84™ Plus graphing calculator required.

- it is <https://education.ti.com/en/us/solutions/ti-codes>

Scratch Animate Your Name

Ages 8 to 16. Programming, Creativity, Web-based. Students will animate the letters of their name, initials, or favorite word using Scratch!

- it is <https://scratch.mit.edu/scratchr2/static/pdfs/help/AnimateYourNameGuide.pdf>

Scratch Hide and Seek Game

Ages 8 to 16. Programming, Creativity, Web-based. Students will gain experience with coding as they make a hide-and-seek game.

- it is <https://scratch.mit.edu/scratchr2/static/pdfs/help/Hide-and-Seek-Guide.pdf>

Scratch Dance, Dance, Dance

Ages 8 to 16. Programming, Creativity, Web-based. Participants will create and code an animated dance scene.

- it is <https://scratch.mit.edu/scratchr2/static/pdfs/help/DanceGuide.pdf>

Hour of Code HS tutorials

<https://code.org/educate/allhourofcode>

High School

Looking at Data with Splunk

Ages 14-18. Math (data analysis), Web-based. Students will analyze the data from a theoretical game to find levels which are too easy or difficult.

- It is <https://www.dropbox.com/s/46ed5ilhvsam8ci/SplunkLessonPlan.pdf?dl=1>. Requires connection to <https://splunk.codeday.org>: blocked. Check out <https://codeday.org>.

Input and Output, Math Activity

Ages 12-16. Math (Algebra), Math (Functions), Unplugged. Connect JavaScript functions to both math and real world problems.

- It is <https://app.vidcode.io/doc/unplugged-activity-math.pdf>. Looks a bit sketchy.

Vizwik Voter App

Ages 13-18. Programming, App building. Learn how to build your own mobile app (iOS and Android) to share with friends to vote on a question that is important to you.

- It is www.vizwik.com/hoc. Requires sign in.

Climate Science

Ages 12-16. Science (Climate), Science (Environment), Unplugged. Students draw a picture, and take turns giving the class steps to recreate their drawing.

- It is <https://app.vidcode.io/hourofcode/science-teacher-guide>.

Mozilla Homework Excuse Generator

Ages 13-18. Programming, Language Arts, Web-based. Use Mozilla's code editor, Thimble, to edit strings inside JavaScript arrays and customize the homework excuse generators.

- It is <https://d157rqmrxj6ey.cloudfront.net/mozillalearning/11701/>. Thimble at <https://thimble.mozilla.org/en-US>.

Hour of Code HS tutorials

<https://code.org/educate/allhourofcode>

High School

Fact or Fiction?

Ages 16-18. Programming, Build an App, Web-based. Students create an app to survey whether their classmates think a statement is find a fact or fiction.

- It is www.vizwik.com/hoc

Oral History Project

Ages 14-18. Programming, Storytelling, Unplugged CEOHP has worked with a variety of educators to develop ideas for classroom activities, homework, and exam problems based on the interview materials.

- It is www.cs.southwestern.edu/OHProject/materials-overview.html (Computing Educators Oral History Project)

Best Technology Activity

Ages 14-18. History, Storytelling, Unplugged Wired.com ran a series of articles in 2013 on each decade of the past 100 years and the significant inventions of those decades. Most interesting to students are the past 2030 years, with the explosion of technology and the gadgets that ensued.

- It is <https://csedweek.org/csteacher/besttechnology.pdf>

Arduino Activity

Ages 12-15. Programming, Arduino Ever wonder how toys make noises and blink lights when you push buttons? Microcontrollers and circuits are used in all sorts of everyday objects. From remote controlled cars to robots and drones.

- It is <https://csedweek.org/csteacher/arduino.pdf>

Secret Codes Activity

Ages 12-15. Ciphers, Math (Cryptography), Scratch, Web-based. Turing has done many things for computer science (often called the father of computer science) but today we will focus on one very important one that helped with the invention of computers.

- it is <https://csedweek.org/csteacher/secretcodes.pdf>. Includes remixing and extending a simple Scratch project to create your own secret code at <http://scratch.mit.edu/projects/30127212>

Hour of Code HS tutorials

<https://code.org/educate/allhourofcode>

High School

Computer History Activity

Ages 12-15. History, Storytelling, Scratch, Google Docs, Unplugged Your class will be creating a 'history of computers' web page/Scratch project/video that we can share with the world. To make this web page, you and your partner will do research and write about one important event or person in computer history

- It is <https://csedweek.org/csteacher/computerhistory.pdf>. Has link to Computer History Museum (www.computerhistory.org): interesting.

Grace Hopper Debugging Activity

Ages 12-18. History, Language Arts, Storytelling, Unplugged Students will research Grace Hopper and learn the story of the first "bug".

- The link is <https://csedweek.org/csteacher/gracehopperdebugging.pdf>. Follow several links about Grace Hopper and computer history.

Globaloria MakeQuest

Ages 10-18. Programming, Game Design, English, Math, Creativity, Web-based. Learn to edit and write JavaScript code to defeat the 'Evil 404,' as you explore computer science concepts like variables and functions. Lesson Plan includes subject-matter extension activities for English, Mathematics, Science, History and Arts classes.

- The link is globaloria.com/courses-services/teacher-guides/. Game using code is at <http://code.globaloria.com>. Main <http://globaloria.com>.

Codesters Dream Sequence

Ages 11-16. Programming, Story Telling, Language Arts, Web-Based. Students write a story using transition words while learning computer science in this Common Core aligned English language Arts project.

- It is <https://www.codesters.com/hoc-classroom/>. See also the next item.

Codesters Transformation Puzzles

Ages 11-16. Programming, Math (Coordinates, Geometry), Web-Based. Students explore, identify, and perform transformations on the coordinate plane in this Common Core aligned Math project.

- <https://www.codesters.com/hoc-classroom/> TechJam is next step <https://www.codesters.com/TechJam/>. It uses Python. "Codesters offers a full 40-lesson Intro to Python curriculum as well as several standards-aligned Math modules. Visit our **marketplace** to see the modules we have available."



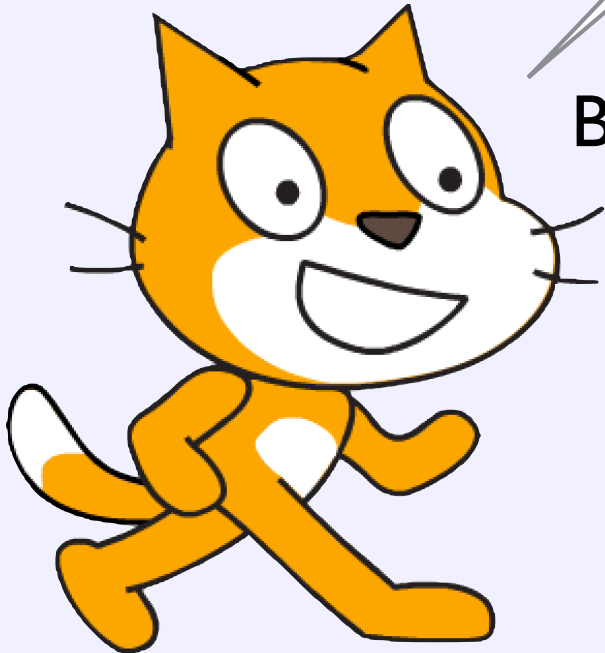
Acknowledgments

- Scratch: <https://scratch.mit.edu/>
- ScratchEd: <http://scratched.gse.harvard.edu/>
- CS First: <https://www.cs-first.com/>
- Code . org: <https://code.org/>
- Hour of code tutorials: <https://code.org/educate/allhourofcode>

Links

- MACUL SIGCS (Special Interest Group for Computer Science): <http://www.macul.org/sigs/sigcs/>
- MiCSTA (Michigan Computer Science Teachers Association): <http://barrywebster.com/micsta.html>
- CSTA: (Computer Science Teachers Association): <https://csta.acm.org/>
- My website: <http://barrywebster.com/>

Thank you for
participating!



Barry Webster, computer science teacher
bwebster@barrywebster.com
<http://barrywebster.com>

