

# Scratch that Coding Itch with Scratch!



Barry Webster, computer science teacher  
[bwebster@barrywebster.com](mailto: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
- Choose languages / programming environments
- Resources

# Introductions

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

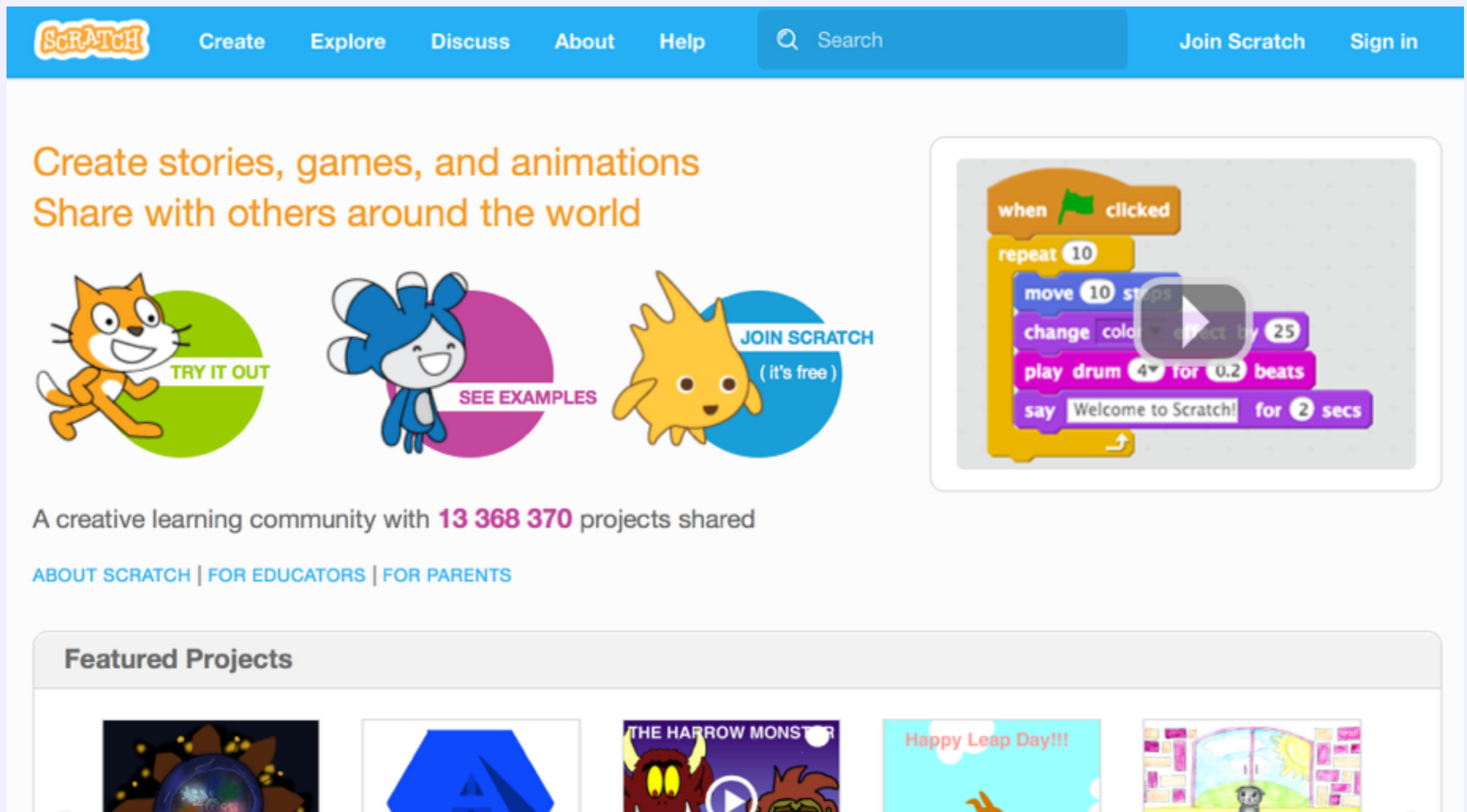


# Install Offline Editor..

- 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: an orange cat with 'TRY IT OUT', a blue girl with 'SEE EXAMPLES', and a yellow character with 'JOIN SCRATCH (it's free)'. To the right is a code editor window showing a script: '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, it says 'A creative learning community with 13 368 370 projects shared' and 'ABOUT SCRATCH | FOR EDUCATORS | FOR PARENTS'. At the bottom, there is a 'Featured Projects' section with five project thumbnails.

# 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
<b>1</b>	<b>2</b>	<b>3</b>
If you don't already have it, download and install the latest <a href="#">Adobe AIR</a>	Next download and install the Scratch 2.0 Offline Editor	Need some help getting started? Here are some helpful resources.
Mac OS X - <a href="#">Download</a> ⬇ Mac OS 10.5 & Older - <a href="#">Download</a> ⬇ Windows - <a href="#">Download</a> ⬇ Linux - <a href="#">Download</a> ⬇	Mac OS X - <a href="#">Download</a> ⬇ Mac OS 10.5 & Older - <a href="#">Download</a> ⬇ Windows - <a href="#">Download</a> ⬇ Linux - <a href="#">Download</a> ⬇	Starter Projects - <a href="#">Download</a> ⬇ Getting Started Guide - <a href="#">Download</a> ⬇ Scratch Cards - <a href="#">Download</a> ⬇

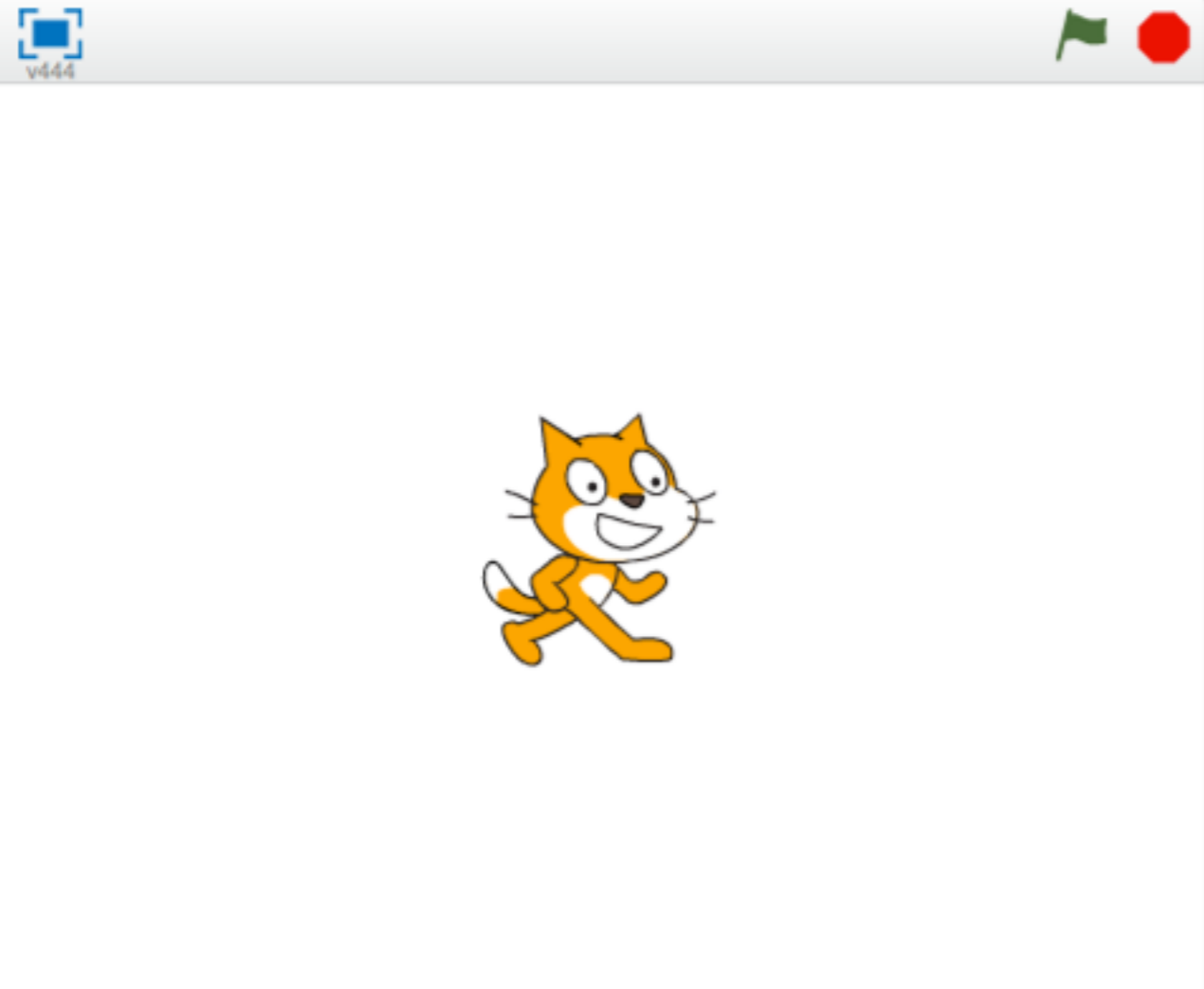
# Scratch, from Scratch

- Computer Science and Scratch
- Editor tips
- Build a project, with add-ons
- (break)
- 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.

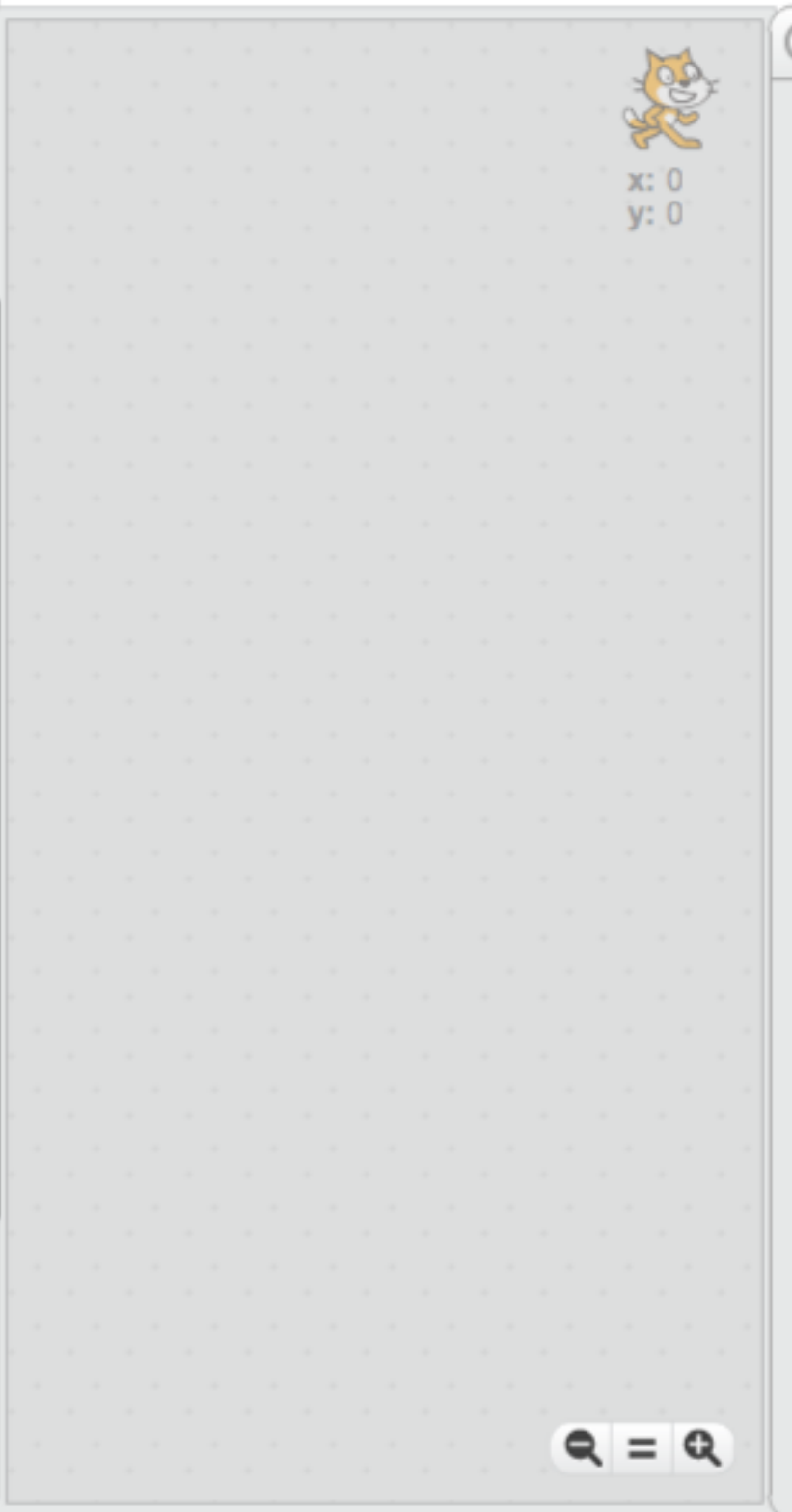




Scripts Costumes Sounds

- 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: [Icons for new sprite]

Stage 1 backdrop

New backdrop: [Icons for new backdrop]

Sprite1



v444

Show Tips



x: -130 y: 10

Sprites

New sprite: [Icons for creating a new sprite]



Sprite1

Stage  
1 backdrop

New backdrop:



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

Show Tips

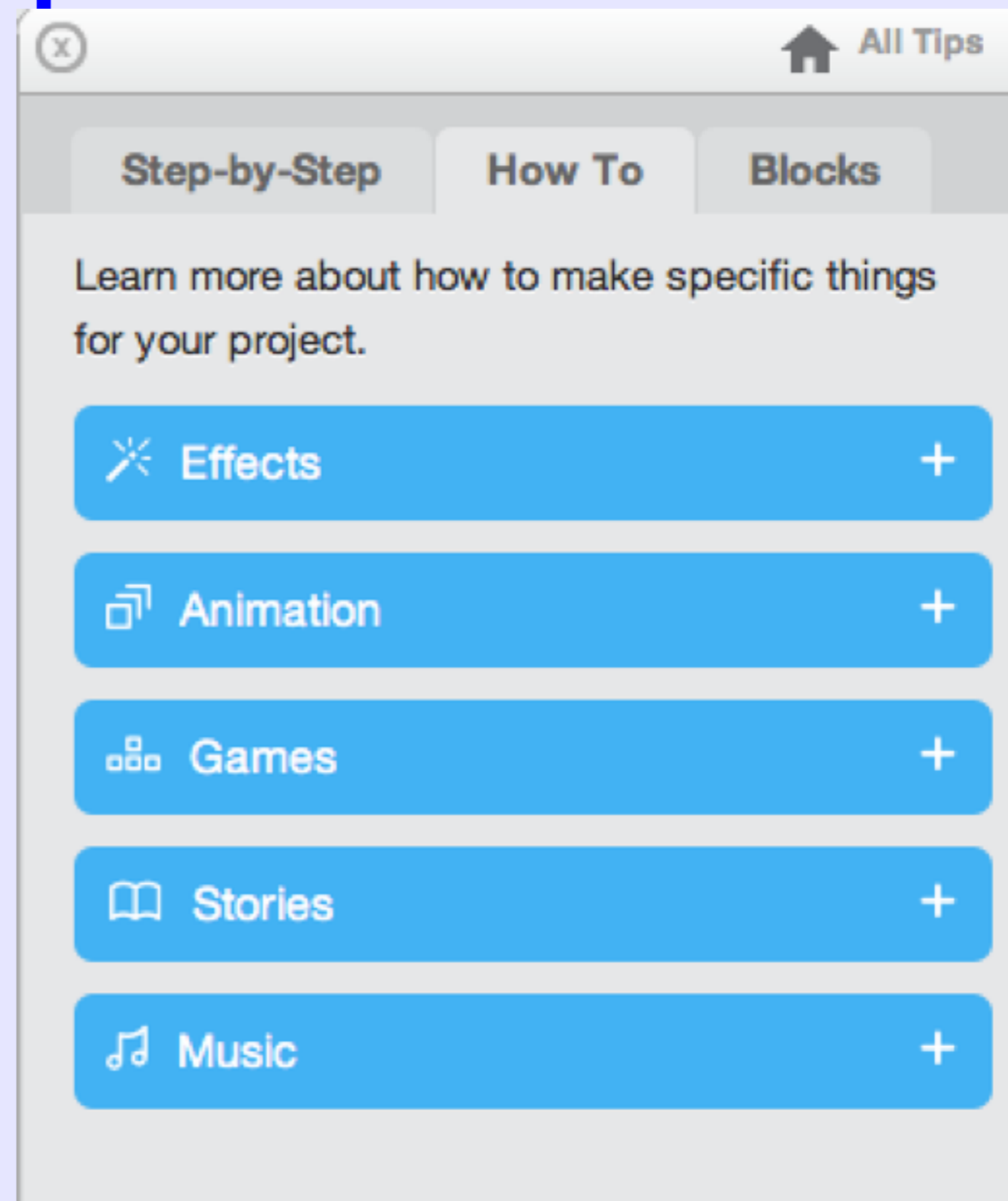


# Tips

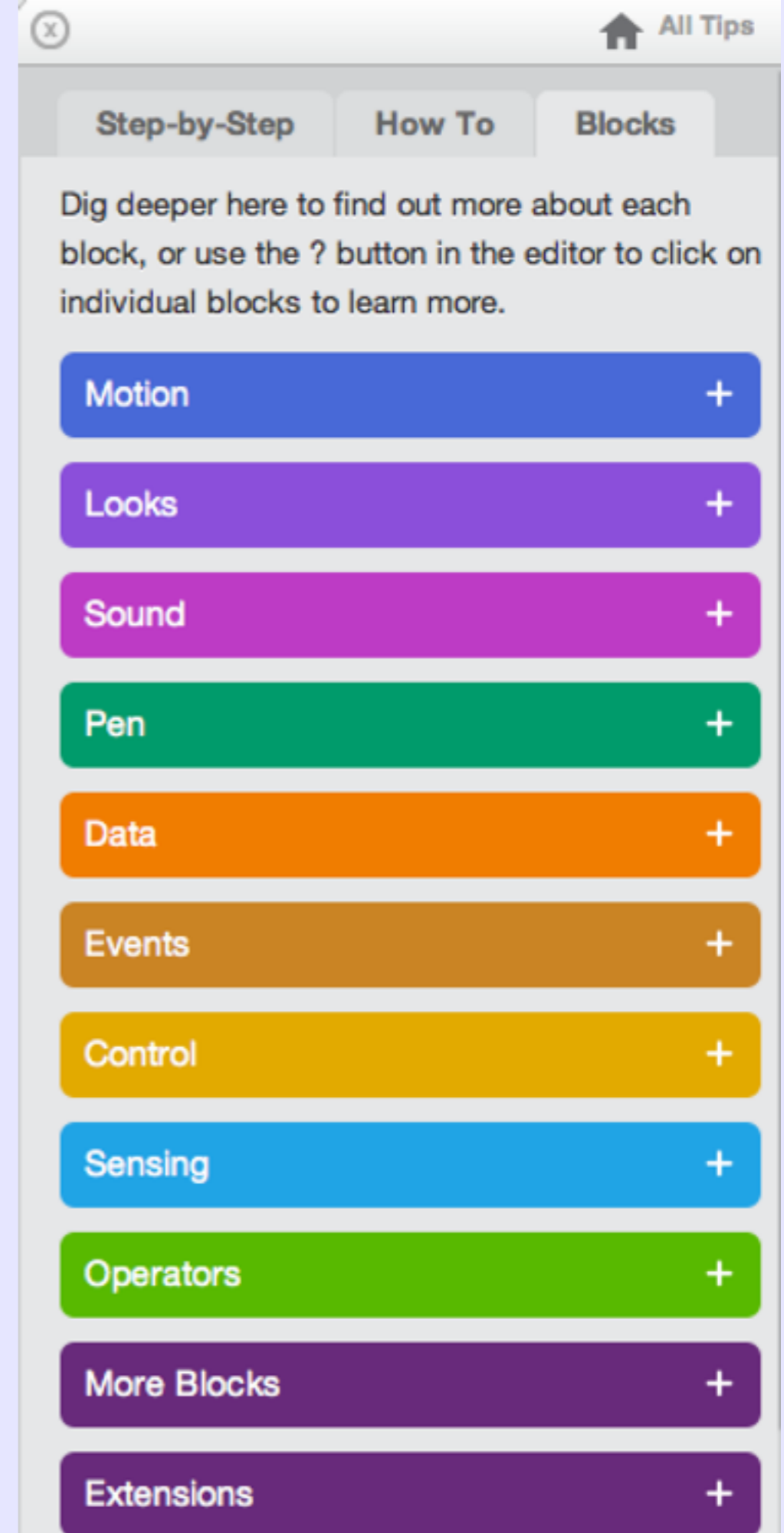
Scratch 2 Offline Editor

The image shows the Scratch 2 Offline Editor interface. At the top, the title bar reads "Scratch 2 Offline Editor". Below it is a menu bar with "ScrATCH", "File", "Edit", "Tips", and "About". The main workspace is titled "Untitled" and contains a single orange cat sprite. The Scripts palette is open, showing a list of motion 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", and "if on edge, bounce". A "Tips" window is open on the right, titled "All Tips", with tabs for "Step-by-Step", "How To", and "Blocks". It contains the text "Follow these tutorials to get started with your project." and a list of tutorial links: "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", and "Favorite Things".

# Tips



# Tips



# 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>



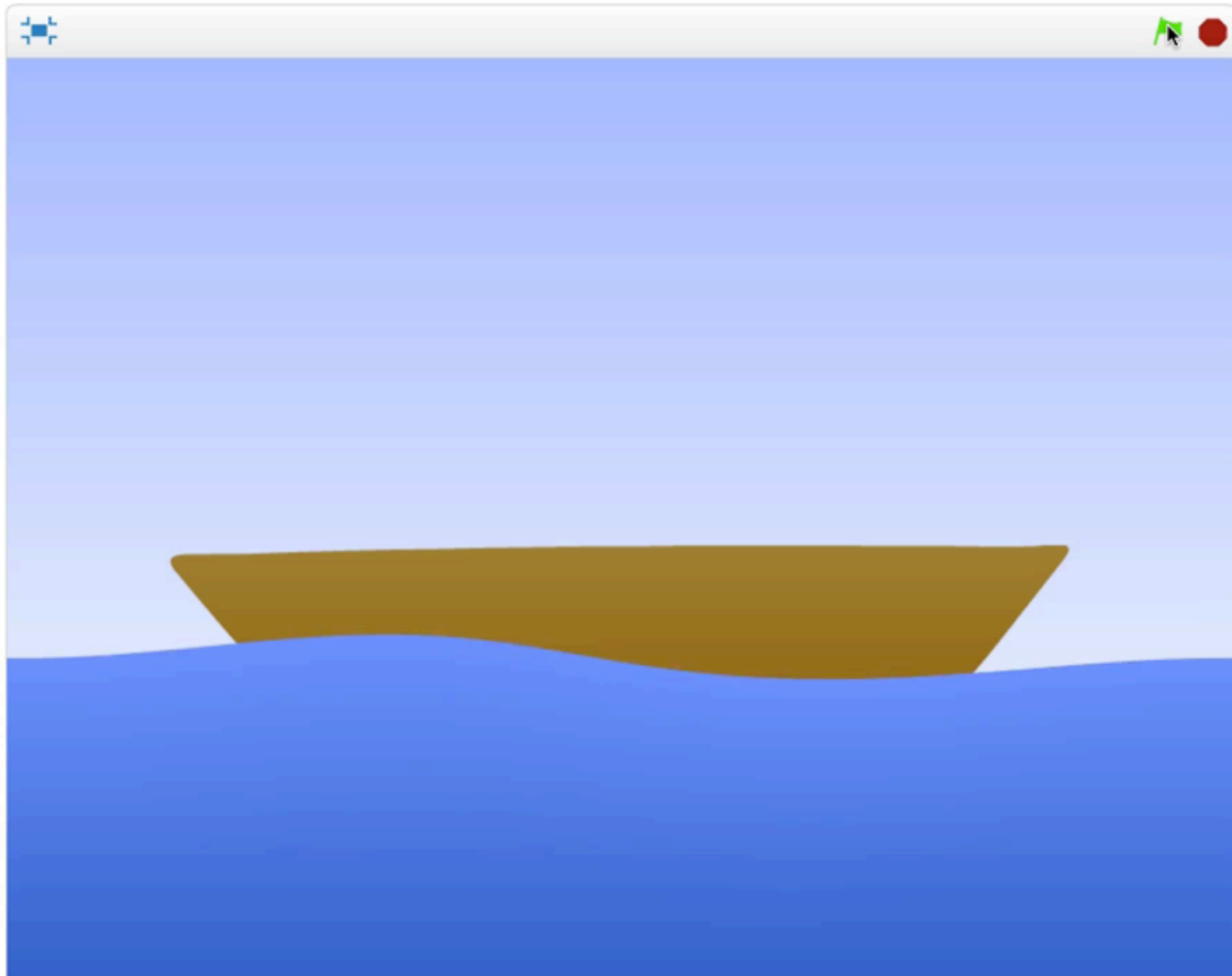
# **Sample Activity**

## High Seas Introduction

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

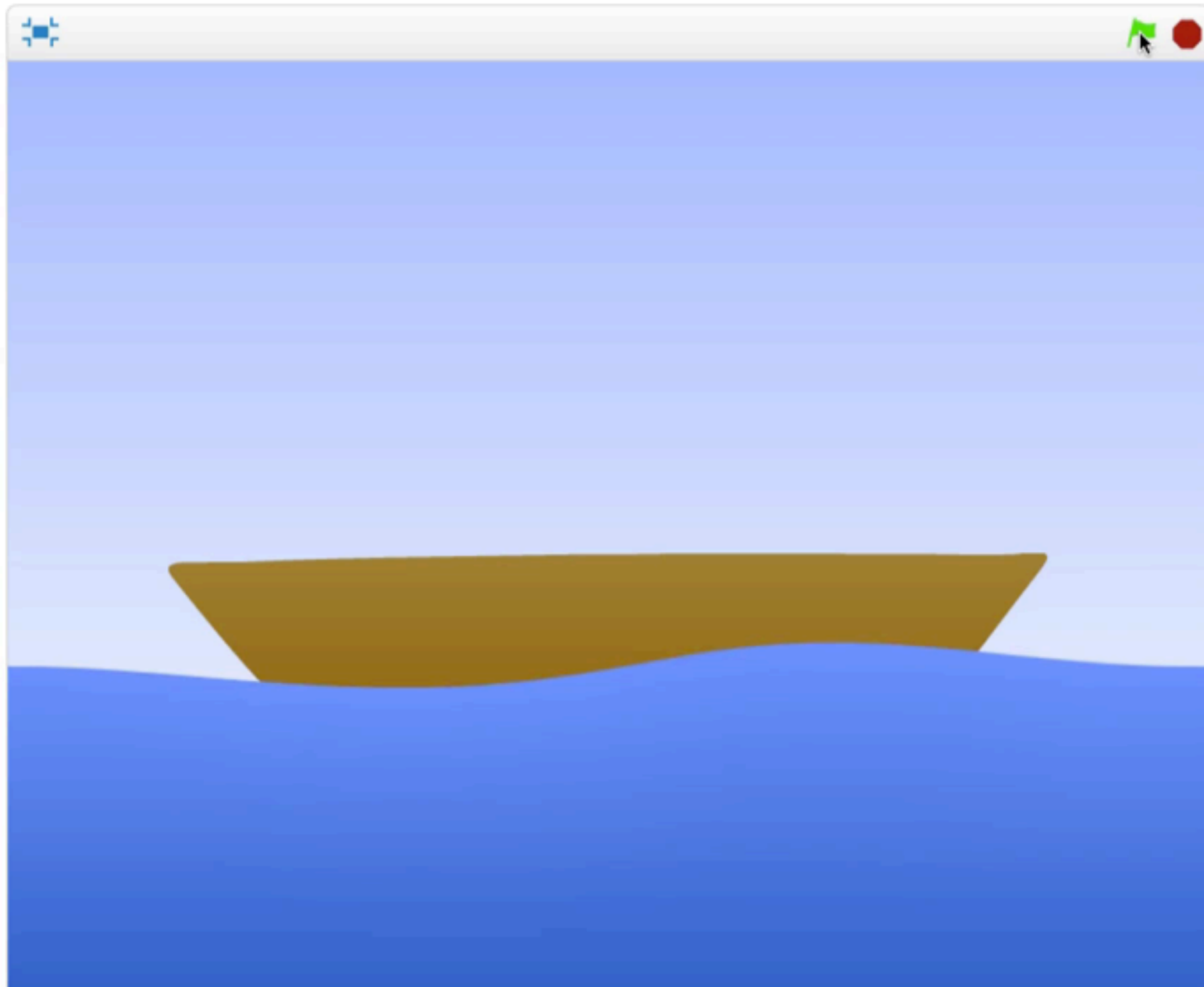
This video will guide you through opening a starter project and creating a Scratch sign in!

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

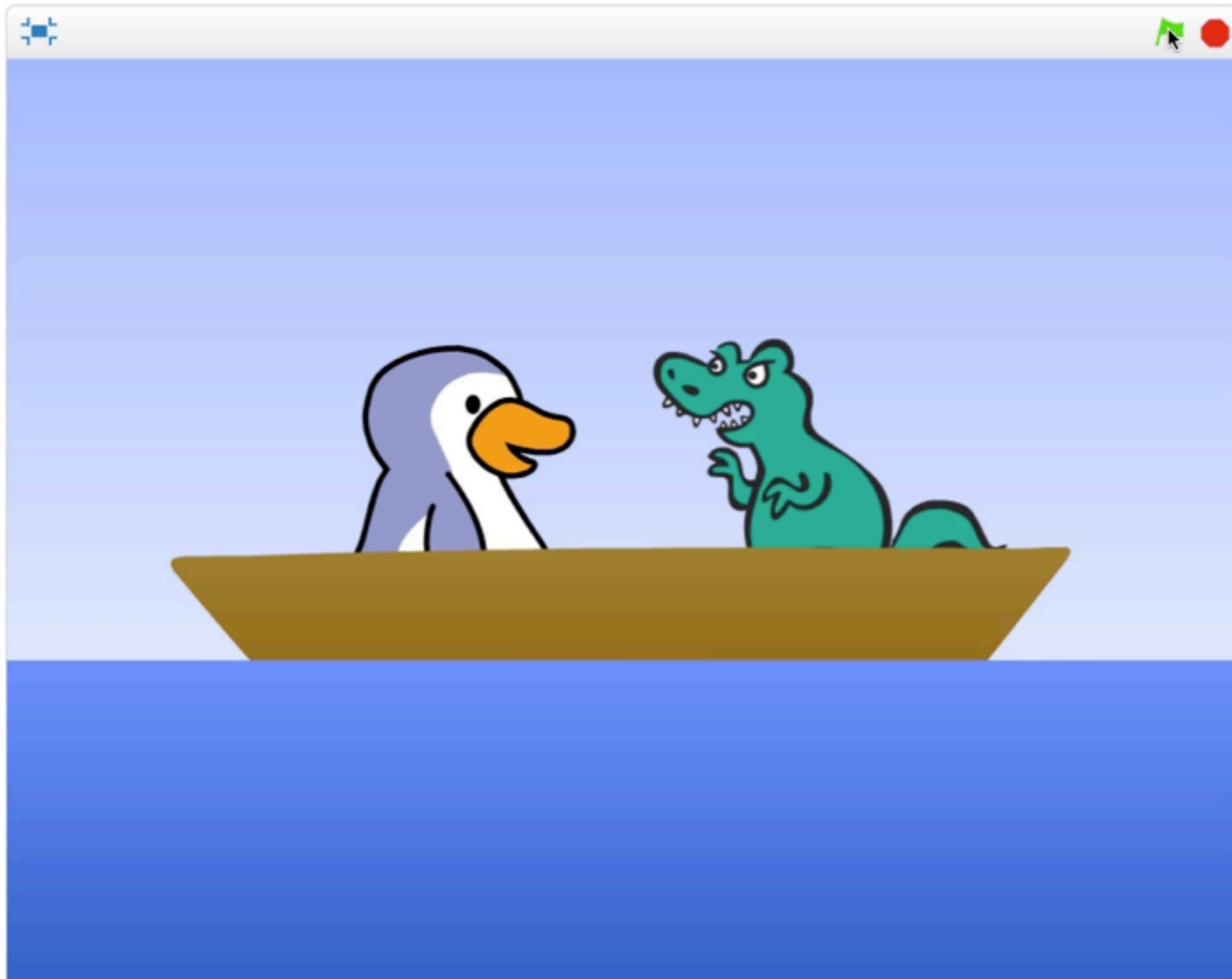


Video 3

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



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









Video 5

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

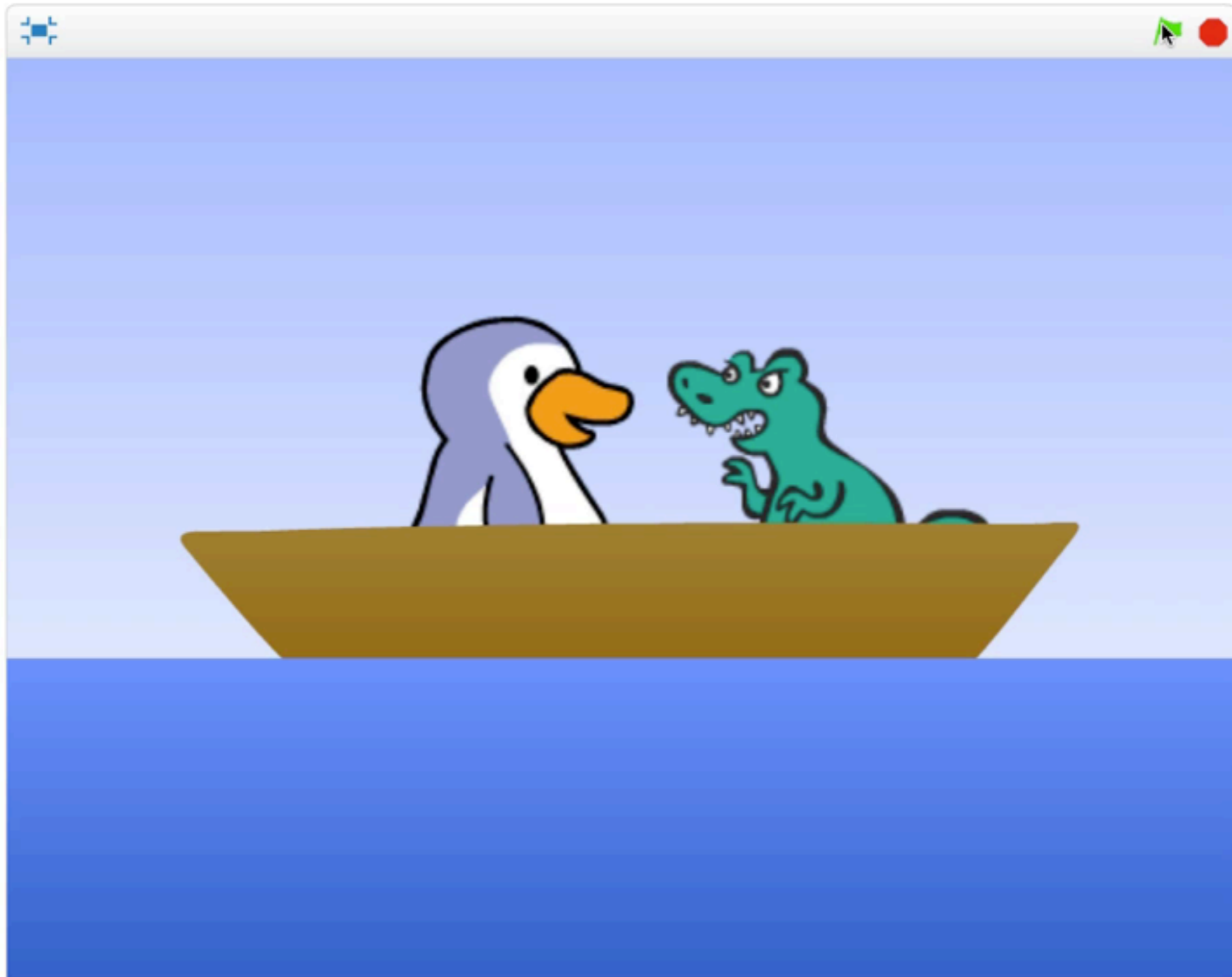
Congratulations!

# [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: Back (blue arrow), 1, 2, 3, 4, 5, 6 (selected), Wrap Up (blue button), Next (green arrow)

# Sea Sounds



Sea Sounds

# Sun Ray Animation



# Sink the Ship



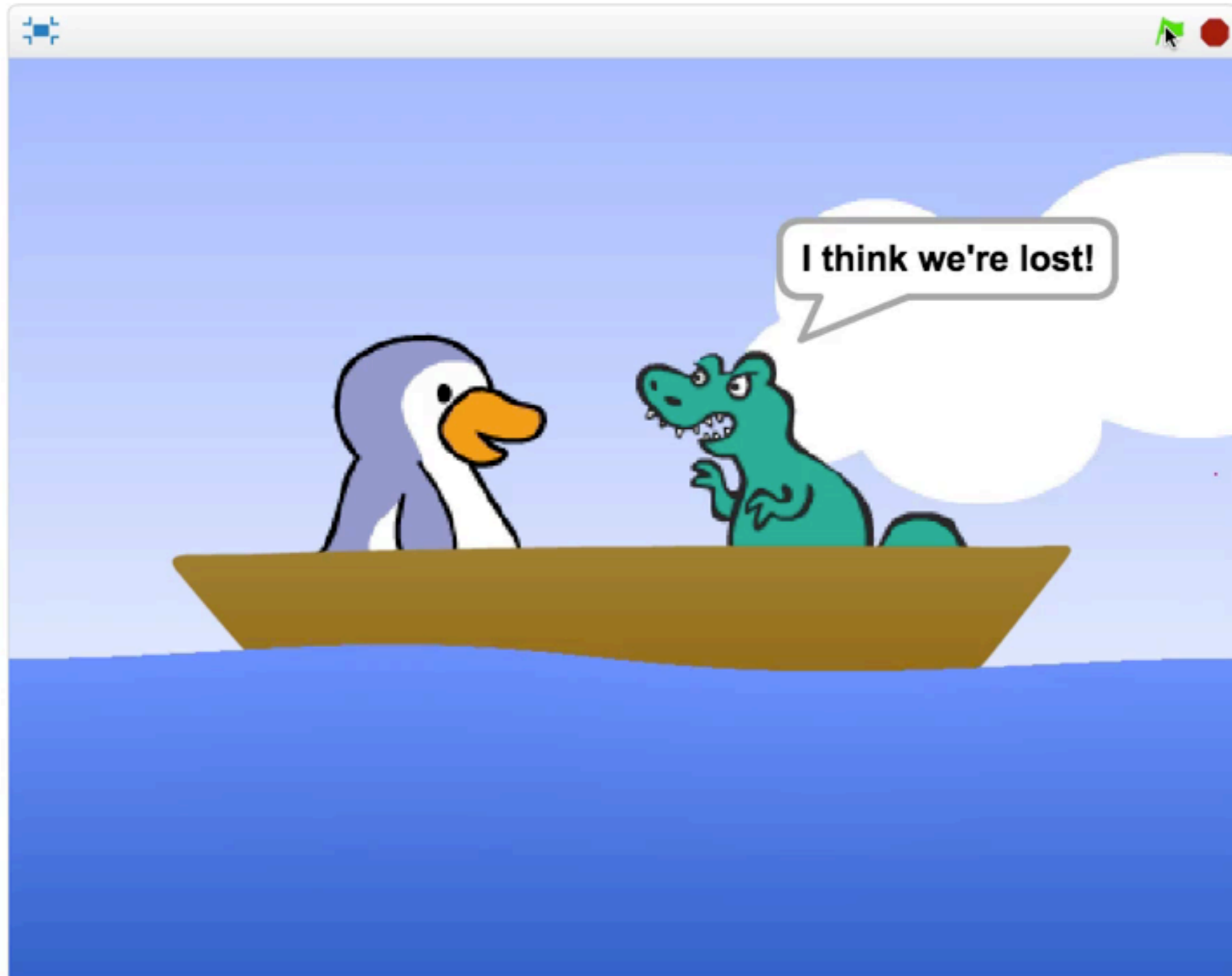
Sink the Ship!

# Second Scene

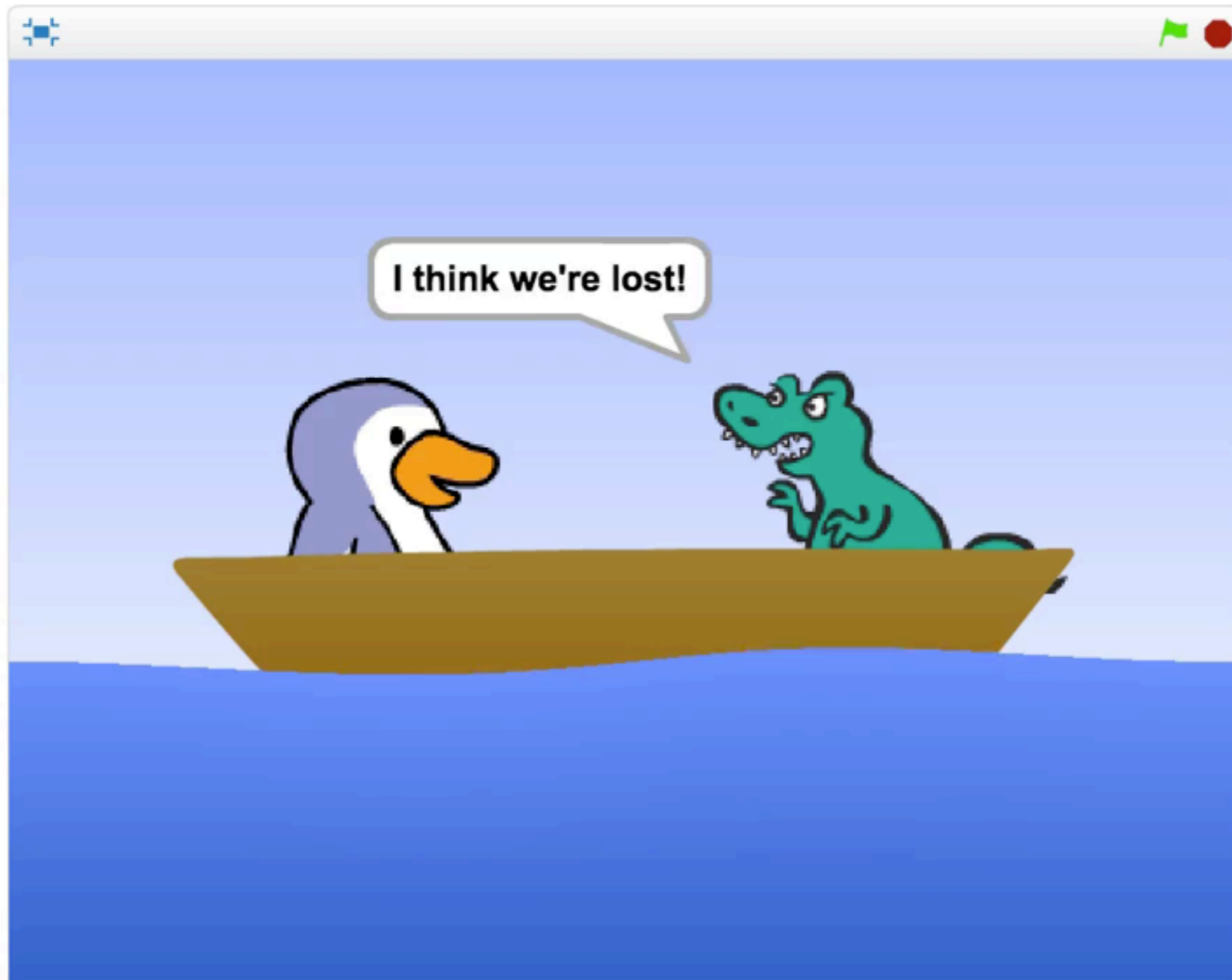


Second Scene

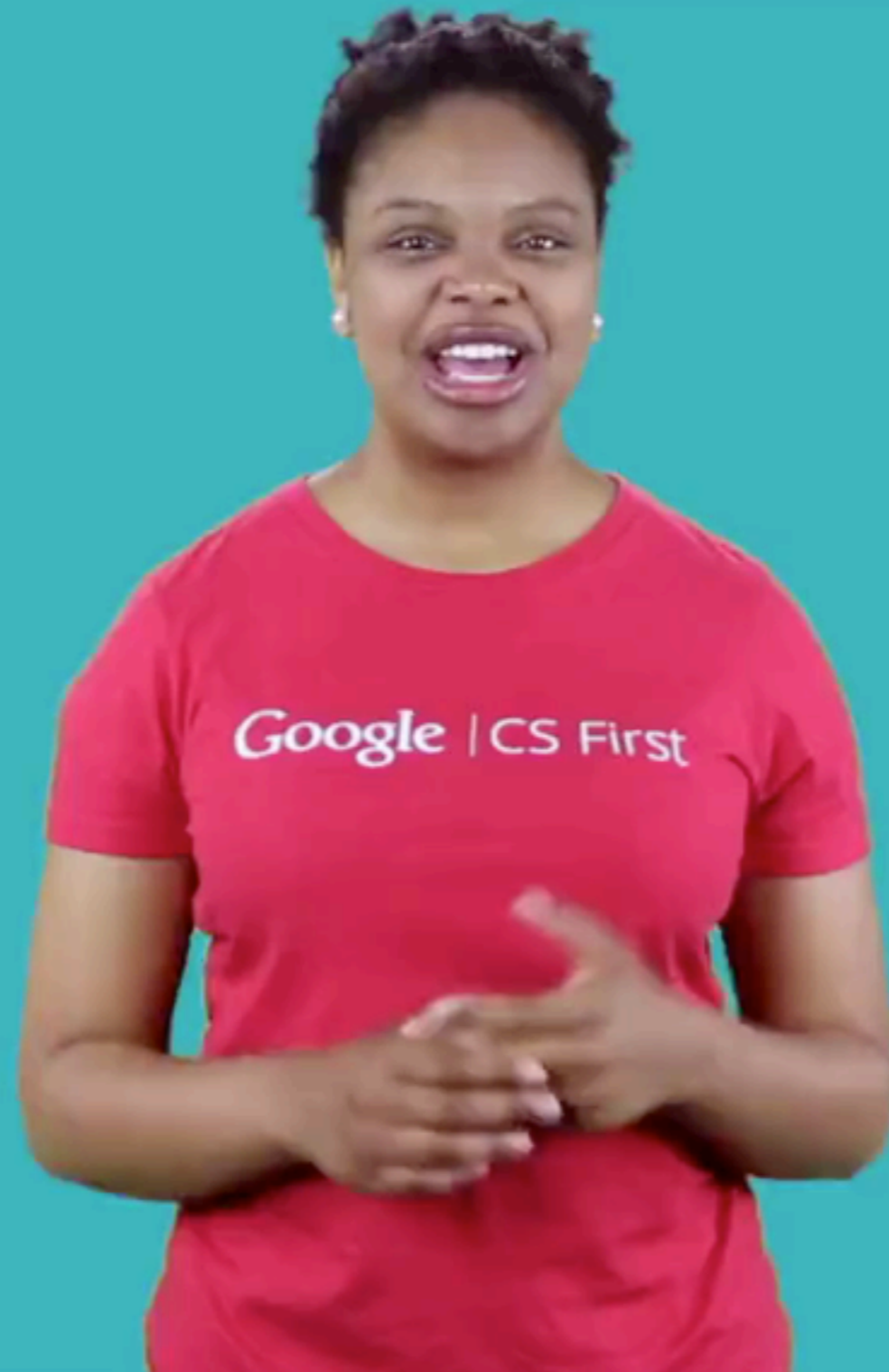
# Clouds



# Gamify



# High Seas Wrap Up



# Guide your students' interests

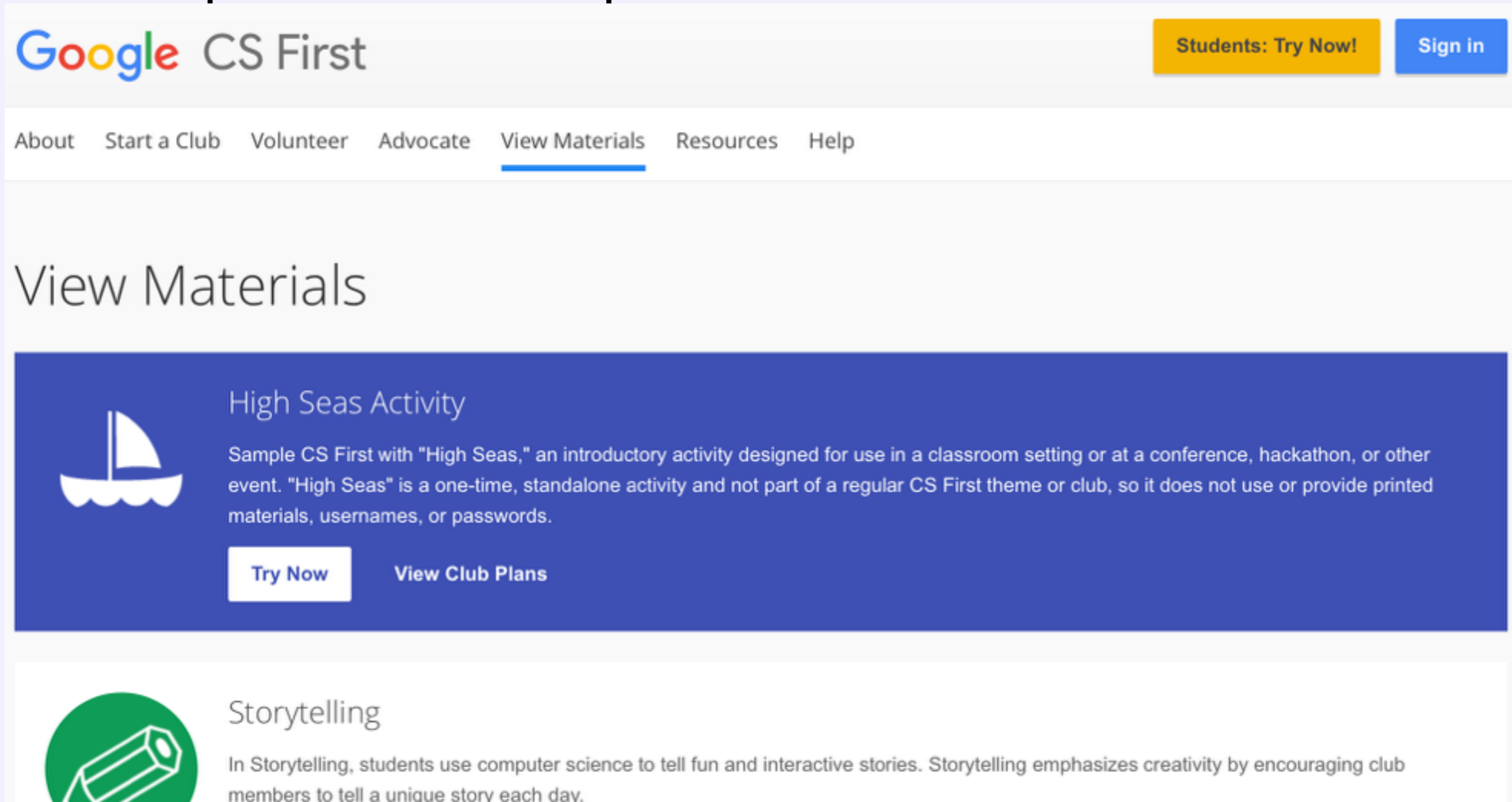
- Learn more about Scratch so you can
- Explore Scratch teaching materials so you can

# Explore Scratch

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

# Explore Scratch Teaching

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




The screenshot shows the Google CS First website. At the top left is the Google logo followed by 'CS First'. On the top right, there are two buttons: 'Students: Try Now!' in yellow and 'Sign in' in blue. Below the header is a navigation menu with links for 'About', 'Start a Club', 'Volunteer', 'Advocate', 'View Materials' (which is underlined), 'Resources', and 'Help'. The main content area is titled 'View Materials' and features two activity cards. The first card, 'High Seas Activity', has a white sailboat icon on a blue background and includes a 'Try Now' button and a 'View Club Plans' link. The second card, 'Storytelling', has a green pencil icon on a white background and includes a descriptive paragraph about using computer science for storytelling.

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

Search

All

GO

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](#)

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>



# 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 young 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

# 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>

## 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 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 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 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.

# 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](mailto:bwebster@barrywebster.com)  
<http://barrywebster.com>

