

The Scratch logo, consisting of the word "SCRATCH" in a white, pixelated font on a grey rectangular background.

SCRATCH

A solid yellow circle.

for

A solid yellow circle.

Young Learners

A resource for teaching  
Scratch to kids ages 5-8.

# Table of Contents

Overview

Activity Map

A guide through the activities

Before you start

A collection of practical advice and guiding principles

Optional Opening Activity

Drawing before beginning in Scratch

Getting Started in Scratch

A basic starting point: a sprite and the stage

Explorations

Activities to introduce different aspects of Scratch

# Overview

This resource is a series of activities and pointers for teaching Scratch to young students. After these activities alone, the students will have enough tools to create their own stories. You should be able to give as much structure as you think is necessary, like requirements for number of scenes, characters, or scripts. Stories are a great way to introduce all the features of Scratch, and they allow for lots of creativity from the kids. Scratch is a creative tool for any age, and even kids this young can learn to use it to express themselves.

For the 2008-2009 school year, I helped teach Scratch to a group of first and second grade students, and this resource is partially based on my experience. I worked at Cardinal Forest Elementary School under the direction of Fred Allard and in partnership with Jessica Gorman. I hope that your experience will be more successful by learning through mine.

In this resource, I work through the creation of a simple underwater scene to show how Scratch can be made easy for kids in this age group. As an educator, you need only minimal experience in Scratch to be able to use this resource. Some time spent experimenting on your own or reading through the available online guides will more than suffice.

([http://info.scratch.mit.edu/Support/Get\\_Started](http://info.scratch.mit.edu/Support/Get_Started)) I have given only one of many ways to introduce Scratch, and it is not the only way. Feel free to explore other areas of Scratch or center learning around a different goal, like developing math skills. This guide is also by no means expansive; there is so much you can do with Scratch!

For each activity, I have broken the steps down by purpose to help you understand them.

**Follow** - the kids follow while you show them what to do; they copy your actions

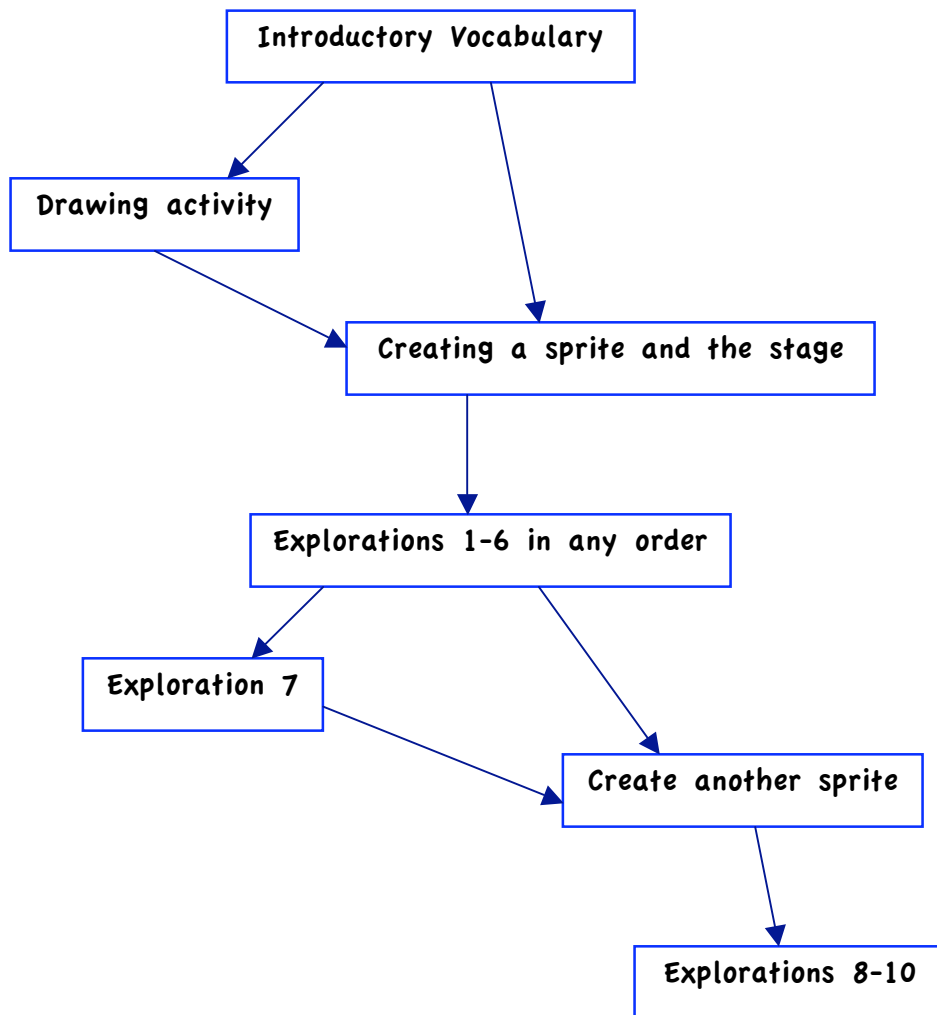
**Play** - a chance for the kids to explore and discover on their own

**Learn** - help the kids with a difficult concept or idea, or simply explain a nuance of Scratch

**Reflect** - ask the kids a question to reflect on their understanding or expand on an idea

Good Luck!

# Activity Map



# Before You Start...

## Some things you should think about.

- 1 **Vocabulary:** Choose your words in the beginning and stick to them. When you explain something, make sure you use the simplest terms you can think of so that the kids can relate. Start by opening Scratch and pointing out important features, but not every feature.
  - a. **Sprite:** a character, like in a story or a game
  - b. **Stage:** like the television screen that you can put different images on. Avoid the term "background," since that applies to what the stage 'wears.'
  - c. **Script:** an action or command; tells the sprite or stage what to do
- 2 **Saving:** Immediately show the kids how you want them to save. Saving their projects is essential to teaching a continuous set of lessons in Scratch; it is very difficult to catch up a student in later lessons. Saving their projects also allows ownership over what they are doing in Scratch, encouraging the creative process. You may want to have a default project available on the common server (maybe one that you create as you go along), and then when a student doesn't save, he or she doesn't have to start completely over.
- 3 **Keeping Interest:** Show the students some kind of end product that they might create. Make it a feasible version of what they are doing, like a short story involving the scripts that they will learn. Try not to show them anything that they will not be able to accomplish in the time you have with them, or it will be all they want to do.
- 4 **Reflection:** At this age, most kids are just learning how to write sentences. A great exercise is for them to write a sentence at the end of a few lessons. Ask them what their favorite thing is about Scratch, or what they might want to make in Scratch someday. Reflection is a good way to reinforce the lessons of the day and encourage learning.
- 5 **Set up:** Be careful about how you set up the computer lab. When you are trying to show kids something on a screen in the front, it is difficult for them to follow if they can't look at their screens and you at the same time. Experiment until you find the best set-up.
- 6 **Planning:** Stay flexible in your plans; you may not finish everything in the time you have. Some days will be better than others, so have multiple stopping points available throughout the lesson plan in case of ending early. Try to finish an idea before ending a lesson; it is difficult to restart in the middle of something. Also - don't be afraid to start over and re-do everything, repetition will only be helpful. More important than anything, have a specific plan and steps in mind. There are many details that you may look over when

planning a lesson that the kids will come up with, so try and stay a step ahead of them.

- 7 Incorporating educational material: It can be easy to include educational material if that is the setting you are in. Try using a story from history as your introductory activity. You can also delve deeper into x and y coordinates. Have the students animate a science process that they are learning, like the water cycle.
- 8 Collaboration: Encourage kids to help each other before they have to ask you, but be sure to define what it means to "help." For many students, that may mean just doing it for them. I like the rule that they are not allowed to touch another student's mouse.
- 9 Repetition: Go ahead and repeat yourself. Come up with as many ways as possible to reinforce things, especially vocabulary. Ask the kids questions, give mini "quizzes," or go over everything at the end of the day.

# Optional Opening Activity

## Drawing before coding

### Materials:

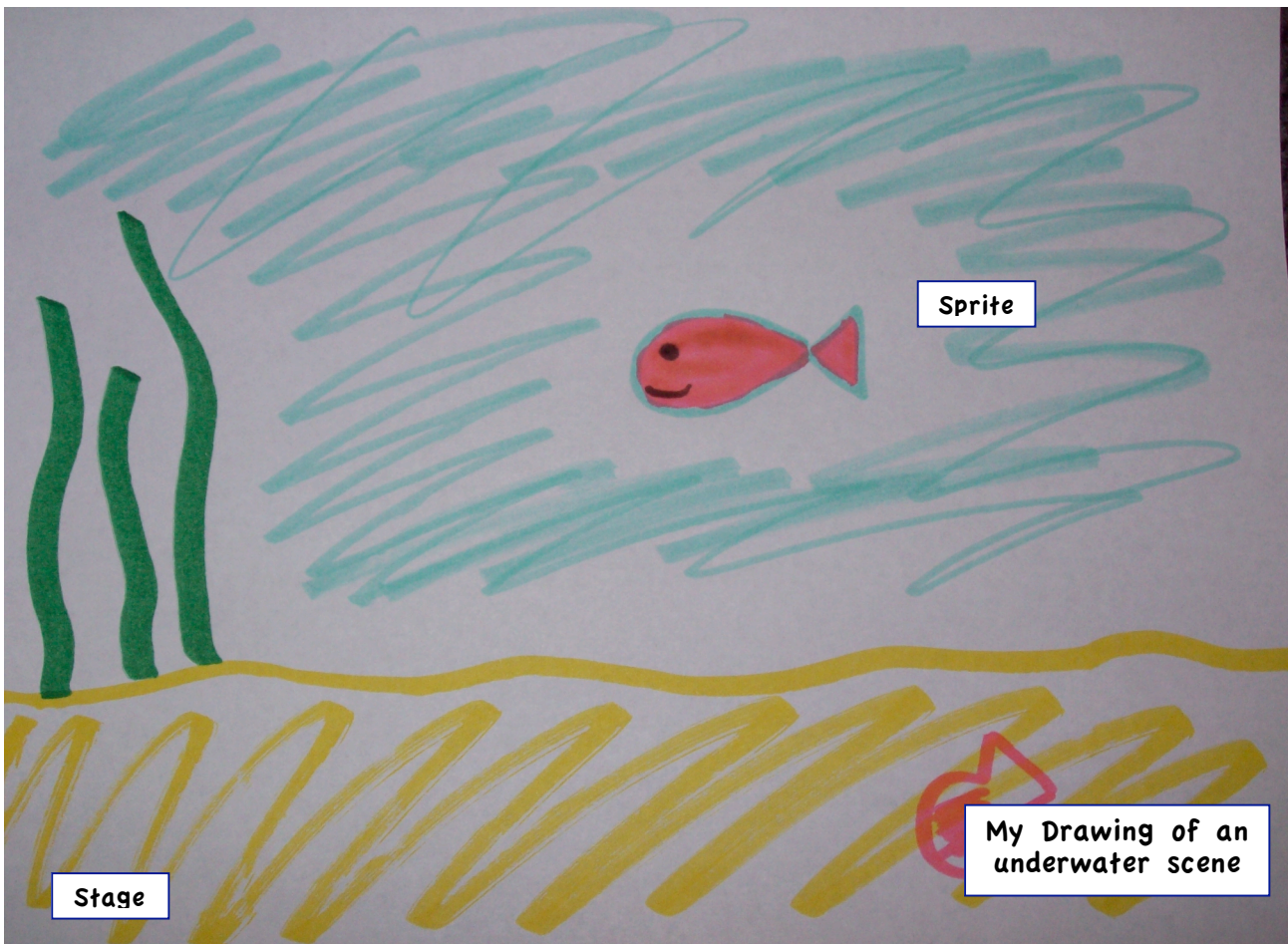
Large pieces of paper  
Crayons

Goal: Draw what you will make in Scratch

- o **Play** start with a foundation or a theme, like an underwater scene. Draw the ocean with seaweed, shells, and sand.
- o **Play** draw a character in the scene.
- o **Follow** label the drawing with Scratch terms. The background is the stage, and the character is the sprite.

Drawing is something they know how to do, so it provides a good introduction into what they can do in Scratch.

You might want to give them the outline of a fish to elaborate on.

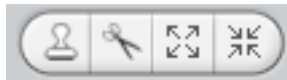


# Getting Started in Scratch

## Creating a sprite and the stage

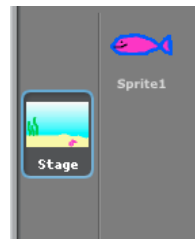
### Tools:

#### Paint Editor



### Stage:

- o **Follow** click between the sprite and the stage in the space below the stage, and go to the stage
- o **Follow** go to the backgrounds tab and click on Paint
- o **Play** create a background (preferably the one you drew)
- o **Reflect** How do you use all the tools in the Paint Editor?

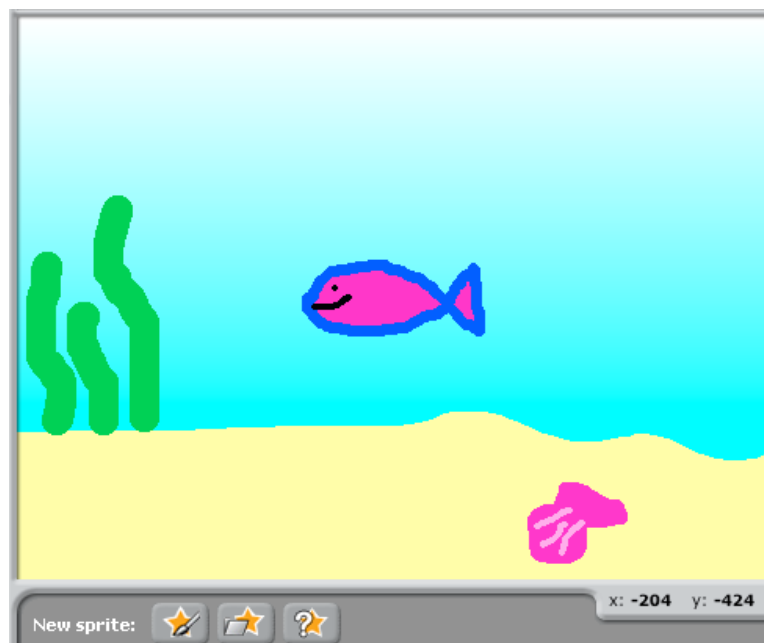


Instead of telling students how to use all the tools, encourage them to ask each other and explore.

### Sprite:

- o **Follow** use the scissors button above the stage to cut kitty by clicking on him in the stage
- o **Follow** import a new sprite, but cut it!
- o **Play** paint a new sprite
- o **Play** use the other buttons above the stage on your sprite (Stamp and Resize)

Point out how silly it is that there is a kitty in the middle of the ocean!



If you gave the students a fish outline on the drawing, they should import that same outline as a sprite that you have created and saved. Show them how to edit it.



# Explorations

## 1 Motion: Move and Turn

### Scripts:



- **Follow** drag in a *move* script
- **Play** change the number of steps
- **Follow** connect a *turn* script
- **Reflect** What happens when the scripts are not connected?
- **Follow** use the *if on edge, bounce* script so that kitty doesn't leave the stage
- **Play** turn the other way
- **Play** add more of all four scripts

## 2 Looks: Say and Think

### Scripts:

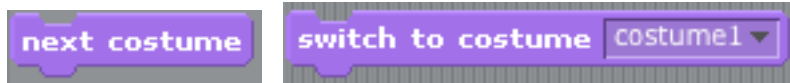


- **Follow** where the *say* and *think* scripts are
- **Play** connect the *say* and *think* scripts
- **Reflect** What is the difference between *say* and *think*?
- **Reflect** What happens when you don't use the script that says how long to *say* or *think*?



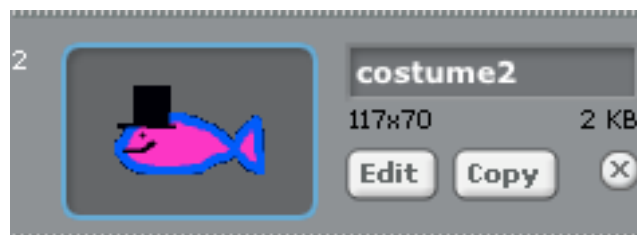
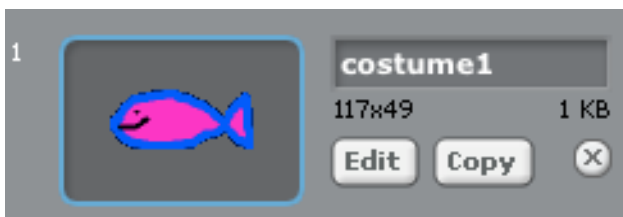
### 3 Looks: Costumes

#### Scripts:



- **Follow** three ways to get a new costume for a sprite:
  - > Import
  - > Paint
  - > Copy and edit
- **Learn** a costume is like your sprite putting on different clothes
- **Follow** click the different costumes and the sprite changes costumes
- **Follow** delete any costume with the x-button
- **Learn** looks scripts for costumes don't go in the costumes tab (they are scripts)
- **Play** *next costume* and *switch to costume* scripts
- **Reflect** What if you want your sprite to start in the same costume every time?
- **Follow** *switch to costume* starting script

Introduce the idea of starting conditions!



## 4 Looks: The Stage

### Scripts:

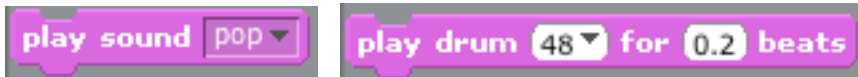


- **Follow** switch between the stage and the sprite in the space beneath the stage
- **Learn** a background is like a costume
- **Play** create a new background
  - > Import
  - > Paint
  - > Copy and edit
- **Play** switch between backgrounds with the *switch to background* and *next background* scripts
- **Reflect** Why might you want to change the background in a story?

Think about starting conditions again.

## 5 Sound: Play Sound

### Scripts:



- **Follow** from the Sounds tab, record or import a new sound for the sprite
- **Follow** add a *play sound* script
- **Play** add other sound scripts like *play drum* or *play note*

## 6 Control and Sensing:

### Scripts:



- **Play** add control scripts like *when space key pressed*, *when stage clicked*, *when sprite clicked*, or *wait 1 sec*
- **Learn** examples of an if statement like “if it is time for lunch, then you will eat”
- **Follow** add an if statement
- **Learn** the Sensing scripts fit into the space in the if statement
- **Follow** add touching color inside the if statement and set the color to something in the background like the sand or the seaweed by clicking in the script to use the dropper to get the color
- **Play** add a script inside the if statement
- **Reflect** How might you use an if statement in a story?



# Challenging Explorations

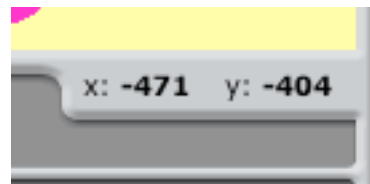
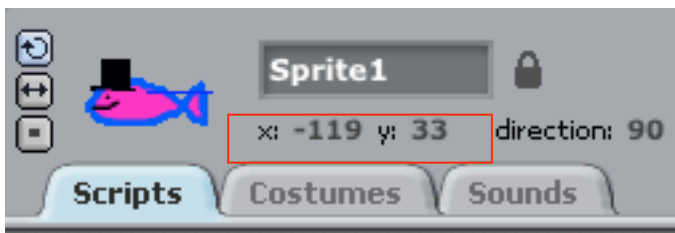
## 7 Motion: Simplifying x and y

### Scripts:



Use real life examples like describing where something is in a room by how many steps away from the center it is.

- **Learn** x and y is a way to say where a sprite is on the stage
- **Learn** a negative number is just a partner for another number that has a dash in front (this explanation is enough to use Scratch)
- **Follow** use these tricks for x and y:
  - > Double click on the sprite in the stage and the coordinates appear in the motion scripts
  - > Move the sprite to a spot on the stage and the coordinates appear in the space above the scripts column
  - > Move your mouse on the stage and the coordinates are shown to the bottom right of the stage
- **Play** add a script that uses x and y
- **Reflect** What is the difference between *go to* and *glide*?



Add a starting condition for the position of the sprite.

★ Before starting these explorations, you will need to add another sprite. See if the students remember how to do it on their own! ★

## 8 Control: Green flag

### Scripts:



- **Follow** start the scripts using the green flag instead of double clicking
- **Play** add scripts to the second sprite so that both sprites act when the green flag is clicked
- **Reflect** What is happening at the same time, and why?
- **Reflect** What is happening in order?

## 9 Control: Loops

### Scripts:



- **Reflect** What do you think the difference is between *repeat* and *forever*?
- **Follow** add a loop with a familiar script inside to the second sprite
- **Play** change which loop you use or which script is inside

## 10 Control: Broadcasting

### Scripts:



- **Learn** Special Activity:
  - > Have two students come up to a whiteboard or a SmartBoard. Tell them that they are going to work together to draw a smiley face.
  - > Assign one student the task of drawing the circle the eyes and the mouth, and the other gets the task of drawing the circle that encloses them.
  - > Position one student at the board and the other student facing away from the board. Tell the student facing away that he cannot draw his part until the first student is done.
  - > Tell the first student to draw his part, and wait to see what happens. The first student will likely tell the other that he or she is done; point this action out!
  - > Repeat the activity with the students switched and call them sprites, their actions, scripts, and their communication, broadcasting. They should be able to see how the sprites talk by broadcasting so that they know when to complete their scripts.
  - > Another play on this example is showing simultaneous action (as opposed to sequential) by you saying “Go,” which is like pressing the green flag.
- **Follow** insert a broadcast between the two sprites
- **Play** add broadcasting between a sprite and the stage
- **Reflect** How do you make two sprites do something at the same time?
- **Reflect** How do you make two sprites do things in order?