

**Learning Target:** I can explore the big ideas of universal programming with Scratch Creative Computing tutorials

**Success Criteria/Objectives:**

- ☐ Use Variables, Conditionals, Loops, Events, Procedures in Scratch Creative Computing tutorials

**Standards:**

7-8.CT.4 Write a program using functions or procedures whose names or other documentation convey their purpose within the larger task.

7-8.CT.5 Identify multiple similar concrete computations in a program, then create a function to generalize over them using parameters to accommodate their differences

7-8.CT.6 Design, compare and refine algorithms for a specific task or within a program.

7-8.CT.7 Design or remix a program that uses a variable to maintain the current value of a key piece of information.

7-8.CT.9 Read and interpret code to predict the outcome of various programs that involve conditionals and repetition for the purposes of debugging.

**Key Ideas:**

- sequence- identifying a series of steps for a task
- loops- running the same sequence multiple times
- parallelism- making things happen at the same time
- events- one thing causing another thing to happen
- conditionals- making decisions based on conditions
- operators- support for mathematical and logical expressions
- being iterative and incremental- developing a little bit, then trying it out, then developing some more
- testing and debugging- making sure that things work – and finding and fixing mistakes
- reusing and remixing- making something by building on what others – or you – have done
- abstracting and modularizing- building something large by putting together collections of smaller parts

**1. Introduction**

- What are the different ways you interact with computers?
- How many of those ways involve you creating with computers?

**2. Mini Lesson:**

- Explain that over the next several sessions they will be creating their own interactive computational media with Scratch.
- Show a basic demo of Scratch

- You build projects by snapping blocks together, just as you can build things in the physical world by snapping LEGO bricks together.
- There are more than 100 blocks in 8 different categories.
- As a small example, let's make the cat do a dance.
- Start by dragging out the "move 10 steps" block from the "Motion" blocks palette to the scripting area.
- Every time you click on the block the cat moves a distance of 10.
- You can change the number to make the cat move a greater or smaller distance.
- From the "Sound" palette, drag out the "play drum" block. Click on the block to hear its drum sound. Drag and snap the "play drum" block below the "move" block. When you click on this stack of two blocks, the cat will move and then play the drum sound.
- Copy this stack of blocks (either using the Duplicate toolbar item or by right clicking the stack and selecting "duplicate") and snap the copy to the already placed blocks.
- Change the second "move" block to -10 steps, so the cat moves backward. Every time the stack of four blocks is clicked, the cat does a little dance forward and back.
- Go to the "Control" blocks palette and grab the "repeat" block. Wrap the "repeat" block around the other blocks in the scripting area. Now when you click on the stack, the cat dances forward and back 10 times.
- Finally, drag the "when Sprite clicked" block and snap it to the top of the stack. Click on the cat (instead of the blocks stack) to make the cat dance.
- Show the range of projects they will be able to create, by reviewing the self paced tutorials
- Tell students they will maintain a design notebook, for recording their coding notes, ideas and plans, as well as for responding to the questions

### **3. Activity**

Give students 10 minutes to explore the Scratch interface in an open-ended way. One prompt is: "You have 10 minutes to make something surprising happen to a sprite." Students are encouraged to work together, ask each other for help, and share what they are figuring out during the 10 minutes.

### **4. Summary**

Ask for 3 or 4 volunteers to share with the entire group one thing that they discovered. Optionally, after the volunteers have shared, offer several challenges to the students: Did

anyone figure out how to add sound? Did anyone figure out how to change the background? Did anyone figure out how to access the help screens for particular blocks?

## **5. Out of class practice/homework**

Students complete the Scratch tutorials at home and in class choose any 13 of the 26 tutorials.

- Students explore the arts by creating projects that include elements of music, design, drawing, and dance. The computational concepts of sequence and loops, and the computational practices of being iterative and incremental are highlighted.
- Students explore storytelling by creating projects that include characters, scenes, and narrative. The computational concepts of parallelism and events and the computational practices of reusing and remixing are highlighted.
- Students explore games by creating projects that define goals and rules. The computational concepts of conditionals, operators, and data, and the computational practices of testing and debugging are highlighted.
- Students develop independent projects by defining a project to work on, collaborating with others to improve the project, and presenting the project and its development process. The computational practices of abstracting and modularizing are highlighted.

## **6. Resources**

- a. <https://scratched.gse.harvard.edu/sites/default/files/curriculumguide-v20110923.pdf>
- b. <https://ccl.northwestern.edu/netlogo/docs/programming.html>
- c. UDL for CS  
[https://ctrl.education.ufl.edu/wp-content/uploads/sites/5/2020/05/Copy-of-UDL-and-CS\\_CT-remix.pdf](https://ctrl.education.ufl.edu/wp-content/uploads/sites/5/2020/05/Copy-of-UDL-and-CS_CT-remix.pdf)
- d. <https://www.helloruby.com/educators>
- e. <http://www.nysed.gov/common/nysed/files/programs/curriculum-instruction/computer-science-digital-fluency-standards-k-12.pdf>
- f. <https://www.windham-schools.org/docs/DOK%20Wheel%20Slide%20for%20Teachers-0.pdf>
- g. [https://code.org/curriculum/docs/csf/CSF TeacherGuide CoursesA-F v2a\\_small.pdf](https://code.org/curriculum/docs/csf/CSF%20TeacherGuide%20CoursesA-F%20v2a_small.pdf)
- h. [Exploring CS](#) curriculum
- i. <https://www.dummies.com/article/technology/programming-web-design/coding/helping-kids-coding-dummies-cheat-sheet-252828>