Level One: Fundamentals

Facilitator Guide

PEER REVIEWED

2300DD

18 USC 707

Discovering Computer Science & Programming through Scratch

Updated for Scratch 3.0

About this Guide

This facilitator's guide is designed to be a tool that further explains activities introduced in ***Discovering Computer Science & Programming through Scratch: Level One***. This guide suggests many activities which are "unplugged": off-computer activities designed to help learners understand computer science concepts and augment the on-computer programming experience, much in the spirit of *Computer Science Unplugged* (**csunplugged.org**). These unplugged activities are to be used as an introduction before students begin a coding activity. This guide also includes discussion questions, skills introduced, blocks introduced, and hints or suggestions to help youth better understand each activity and concept. Youth and facilitators are encouraged to explore, learn, and talk about Scratch together.

Acknowledgments

Authors:

Lenny Pitt, Professor Emeritus of Computer Science, University of Illinois

Judy Rocke, Curriculum Development Specialist, Office for Mathematics, Science, and Technology Education (MSTE), University of Illinois

Jana Sebestik, Assistant Director STEM Curriculum Design, (MSTE), University of Illinois

Support for this guide is provided by the 4-H Computing Connections (CS4H) project funded by the University of Illinois Extension and Outreach Initiative and also by the Department of Energy and the Department of Homeland Security under Award Number DE-OE0000780.

Principal Investigator: Lenny Pitt, University of Illinois

Co-Principal Investigators:

Maya Israel, Assistant Professor, College of Education, University of Illinois

George Reese, Director of the Office for Mathematics, Science, and Technology Education, University of Illinois

Layout and Design:

Christina Tran, Graphic Designer, MSTE, University of Illinois, iamchristinatran.com

This curriculum is available for purchase from the National 4-H Supply Service, the e-commerce and mail order catalog unit of National 4-H Council.
Visit shop4-h.org or call (301) 961-2934.

Table of Contents

Your Role as Facilitator

This computer science curriculum introduces sequence, iteration, conditionals, variables, and modularization – the most fundamental principles in any programming language. It uses Scratch, an MIT Media Lab project, to explore these concepts while creating a foundation for coding in any language.

This curriculum provides youth with a series of tutorials and challenges within the Scratch environment. Young people can work on the activities individually, with partners, or in a guided instructional setting.

This facilitator guide further explains the activities introduced in the youth guide. It suggests many “unplugged” activities. It also includes discussion questions, previews of skills and blocks introduced, and hints or suggestions to help youth better understand each activity and concept.

As a facilitator of this project, encourage youth to talk about what they learn as they try new scripts and find new blocks. Youth will learn faster and more when they discuss their projects with others. The Scratch community encourages users to share their projects on the Scratch website and to remix other’s projects. Just be sure to give credit to the original project creator. There are nearly 100,000,000 registered Scratch users. Join the fun!

This curriculum was written for youth in Grades 5-12, but may be used and adapted for younger and older audiences. Ideally, each Learner working with a group should have his/her own guidebook and Scratch Student Notebook to document their progress.

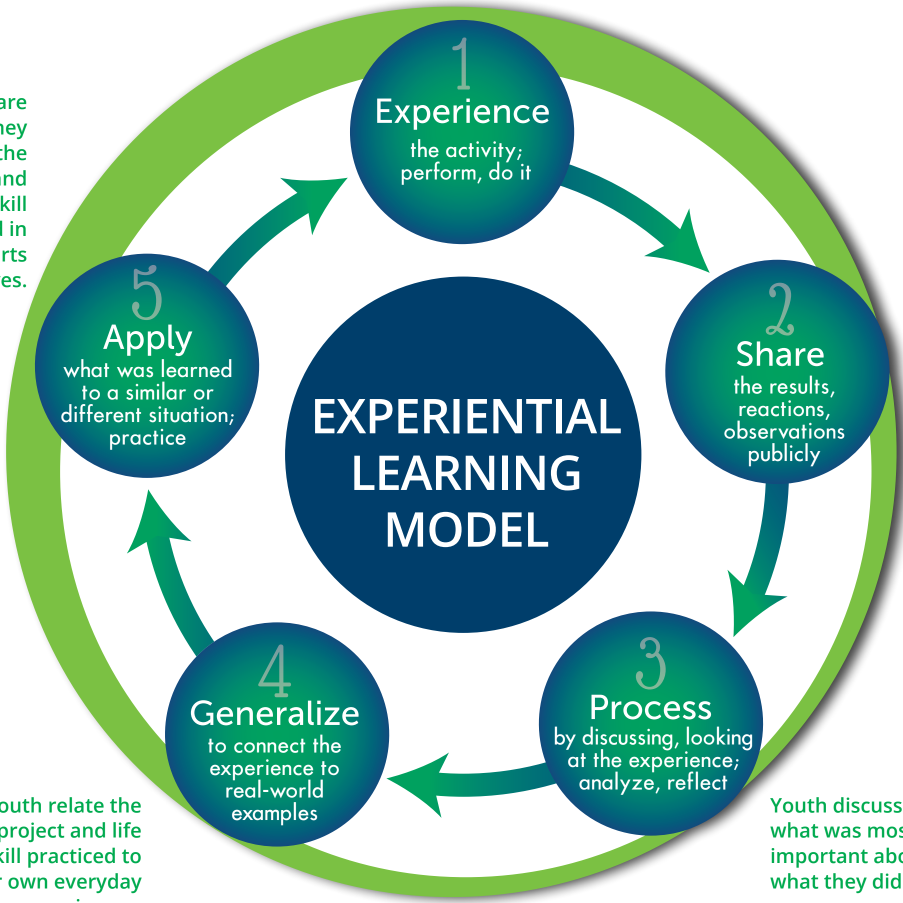
Experiential Learning

The Experiential Learning Model of Instruction provides learners an opportunity to become familiar with the content (Experience), explore a deeper meaning of the content (Share and Process), connect the learning to other examples or opportunities (Generalize), and apply it in real world situations.

The facilitator will guide youth by helping them to focus on the activities, provide support and feedback for the learning, and debrief with them about their learning experience: what went well, what they could have done differently, what they could do next.

This debriefing process fits hand-in-glove with the engineering design process used throughout the curriculum.

Youth do before being told or shown how.



Youth describe the experience and their reaction.

Youth discuss what was most important about what they did.

Youth relate the project and life skill practiced to their own everyday experiences.

Youth share how they will use the project and life skill practiced in other parts of their lives.

Pfeiffer, J.W., & Jones, J.E., "Reference Guide to Handbooks and Annuals"
©1983 John Wiley & Sons, Inc. reprinted with permission of John Wiley & Sons, Inc.

Getting Started with Scratch

Using Scratch

Join Scratch at: **scratch.mit.edu**

People need time to explore Scratch! They will discover many things on their own. Encourage them to share and problem solve together. Encourage them to listen to each other and make suggestions as they share the successes and problems they encounter as they complete the activities.

Using this Guide

Each chapter of ***Discovering Computer Science & Programming through Scratch: Level One*** explores one of these fundamental computer science concepts: **Sequence**, **Iteration**, **Conditionals**, **Variables**, and **Modularization**.

The facilitator notes for each chapter includes:

- Unplugged activities (off-computer activities to help develop a deeper understanding of the new concept in that chapter)
- Skill lists for each activity
- Lists of the blocks introduced in each activity
- Suggestions to help with group discussions
- Hints for helping youth understand a concept

Privacy

When you put your project on the Scratch website, it is private unless you click on the **SHARE** button. If you share your project, other Scratchers can see it. The Scratch Team encourages Scratchers to remix projects within guidelines that respect the work of the creator. Note the following FAQ's from **scratch.mit.edu/info/faq/#remix**

What is a remix?
When a Scratcher makes a copy of someone else's project and modifies it to add their own ideas (for example, by changing scripts or costumes), the resulting project is called a "remix." Every project shared to the Scratch website can be remixed. We consider even a minor change to be a valid remix, as long as credit is given to the original project creator and others who made significant contributions to the remix.

What if I don't want others to remix my projects?
Remixing is an important part of the Scratch community. If you don't want others to view or remix your creations, you can still create projects on the Scratch website, but don't share them on the website.

Sequence

★ Allow time to discuss successes and problems.

A computer does exactly what you tell it to do, and it follows the steps in the code in exactly the order that you give it. Order matters.

Unplugged Activity

Three people come to the front of the room. Person 1 gives instructions and the other persons follow the instructions.

- Person 1 tells Person 2 to go **forward 5 steps**, **turn 90 degrees**, **go forward 3 steps**.
- Next, Person 1 reorders the instructions and tells Person 3 to go **forward 3 steps**, **turn 90 degrees**, **go forward 5 steps**.
- Where do Persons 2 and 3 stop?
- Talk about the results. Emphasize that even though the instruction components are the same, arranging them in different sequences results in different outcomes.
- What is the result when these same instructions are executed in this order? **turn 90 degrees**, go **forward 3 steps**, go **forward 5 steps**.
- When Scratch blocks are arranged in an incorrect sequence the result may be incorrect or unexpected. Sequence matters in coding.

Sequence and Scratch

Calendar Activity

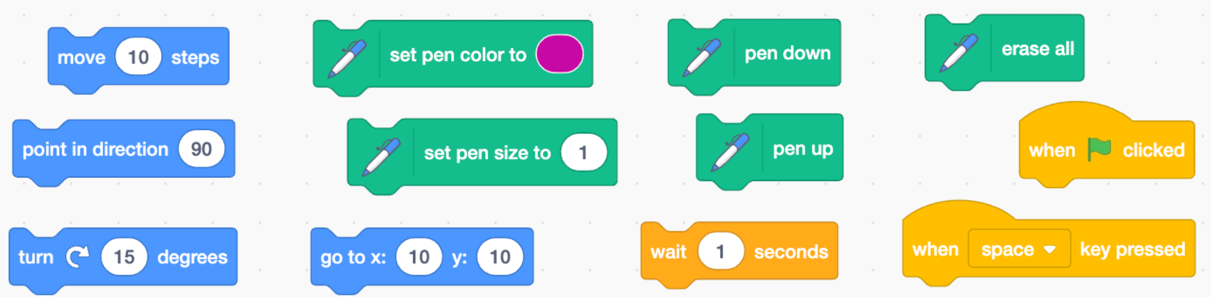
A calendar is used as a backdrop for these activities.

Open the project at **scratch.mit.edu/projects/52455886/**. Click **See inside** and then click **Remix**. Remember you must be signed into a Scratch account to **Remix** a project.

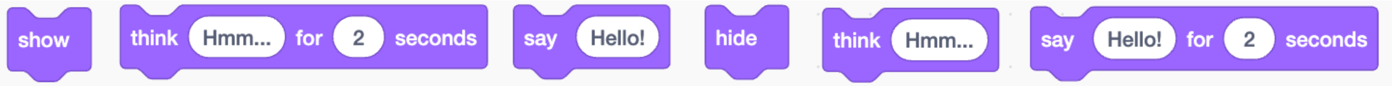
Skills

- Make a sprite travel to a specific location on the stage
- Draw the path a sprite makes as it moves around the stage
- Use **Events** blocks to start a script
- Create a reset script to return a sprite to its starting location
- Use (x,y) coordinates to position a sprite on the stage

Blocks introduced in this activity are in the **Motion**, **Pen**, **Events**, and **Control** categories.



Challenge Activities offer opportunities to apply new skills. Experiment and try new blocks from the **Looks** category.



Notes

- One step is a very short distance on the stage. The stage is 480 steps across and 360 steps from top to bottom.
- When Scratch opens the cat appears in the center of the stage at point (0,0) on a (x,y) coordinate graph.
- Blocks connect together to create a script. A script is a series of instructions for a sprite.
- Encourage learners to keep the script area organized so others can follow their codes more easily. Right click on the script area. There is a menu for Undo, Redo, Clean up Blocks, Add Comment, and Delete Blocks.
- By clicking "Clean up Blocks" Scratch will organize all the blocks and scripts in the script area. Experiment by adding some small scripts and blocks to the script area. Click "Clean up Blocks". Discuss how this might be helpful when others look at your scripts.
- By clicking "Add a Comment" users can add comments to help others see what each script does in their code. You can expand, shrink, or delete comments. Discuss why it is important for others to be able to follow a user's scripts and code. How might this help in finding and fixing problems or bugs in a script or code?

Talk About

1. What are the advantages of having a reset script?
2. What are some of the things you've learned in these activities?
3. What is the difference between a block and a script?
4. What ideas would you share with someone who has never used Scratch before?
5. How would you explain sequence?

Iteration

Allow time to discuss successes and problems.

A computer can not only do exactly what you tell it, it can also do the same task repeatedly. It never gets tired, and it never makes a mistake. Another word for repetition is iteration.

Unplugged Activity

Each learner needs a pen and a blank sheet of paper. Ask each to follow these steps:

- Move your pen to the left middle of the paper
- Repeat the following six times:
  - Put your pen tip down on the paper
  - Draw a circle with approximate radius of one inch
  - Lift your pen up
  - Move your pen one inch to the right

You are using iteration to draw this series of six circles. In the same way, you can use a repeat block to tell the computer how many times you want a specific task performed.

Iteration and Scratch

Box It In

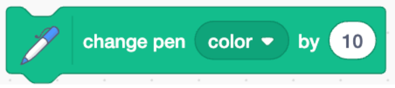
A calendar is used as a backdrop for some of these activities.

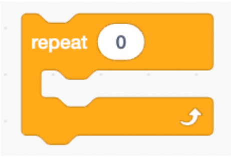
Open the project at **scratch.mit.edu/projects/52455886/**. Click **See inside** and then click **Remix**.

Skills

- Use the **repeat** block and blocks from the **Pen** and **Motion** categories to draw a square at a specific location on the stage
- Place blocks inside a **repeat** block

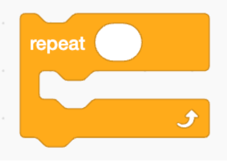
Blocks introduced in this activity are in the Control and Pen categories.



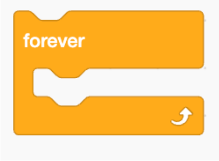


Challenge Activities offer time to create. They encourage linking new scripts to previously coded scripts and applying skills in new situations.

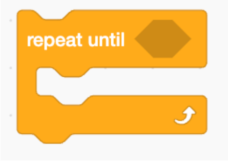
Notes



```
repeat
```



```
forever
```



```
repeat until
```

Scratch uses three kinds of repeat blocks. The first block tells the computer to **repeat** a task a **specified number of times.** The second tells the computer to **repeat the task forever**. The third tells the computer to **check the until condition and if it is <u>not</u> true, repeat the <u>entire task</u> inside the block.** Then check the until condition again and if still not true repeat the entire task inside the block again. The computer continues to check and repeat until the condition is true.

Calendar Cross Out

Skills

- Combine blocks in a correct sequence to recreate a drawing
- Use iteration to accomplish a task
- Use a negative number in a move block to make a sprite move in the opposite direction

Notes

- Use the repeat block whenever the same series of blocks are used over and over. Think about how a repeat block can eliminate this repetition.

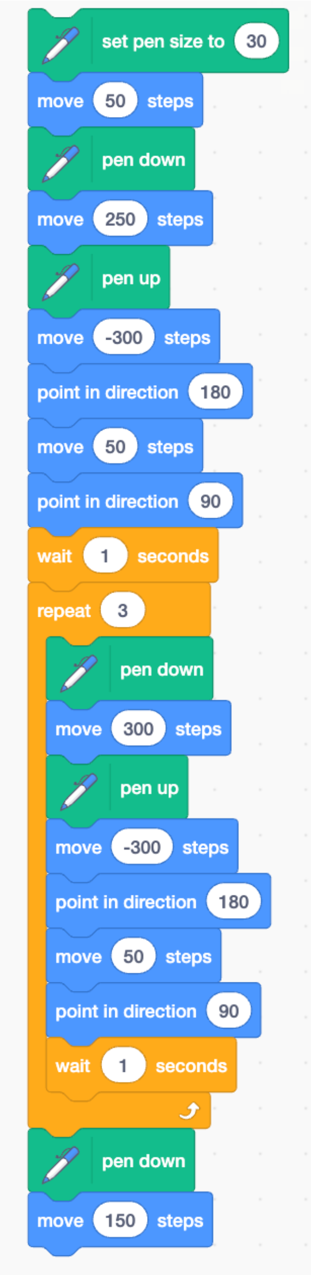
Bugs, Glitches, and Problems in Scripts

Skills

- Recreate a script
- Debug or fix the script so that it creates a specified picture
- Right click a script to make a duplicate copy of the script
- Pull a script apart one block at a time to test the script step by step while searching for errors

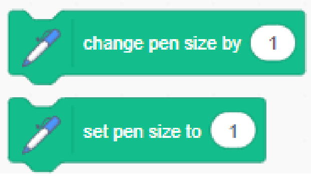
Notes

- Clicking and holding the script allows users to detach and drag all the blocks from the cursor location and below.
- One possible “debugged” script, for this activity, is shown here.



```
set pen size to 30
move 50 steps
pen down
move 250 steps
pen up
move -300 steps
point in direction 180
move 50 steps
point in direction 90
wait 1 seconds
repeat 3
    pen down
    move 300 steps
    pen up
    move -300 steps
    point in direction 180
    move 50 steps
    point in direction 90
    wait 1 seconds
pen down
move 150 steps
```

- These 2 blocks are often confused. One sets the size of the pen. With this block the size of the pen doesn’t change unless it is set again. The other changes the size of the pen each time the block is executed. Try out these two blocks to see how they affect the pen size.



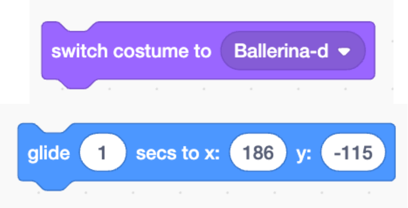
```
change pen size by 1
set pen size to 1
```

Ballerina

Skills

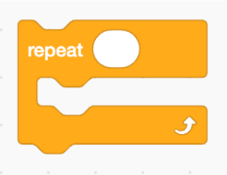
- Change a sprite's costume
- Make a sprite appear to dance by rapidly changing its costumes
- Make a sprite glide across the stage
- Use **repeat** and **forever** blocks
- Make a sprite seem to appear or disappear
- Select a sprite to make it active

Blocks introduced in this activity are in the **MOTION, LOOKS,** and **CONTROL** categories.



switch costume to Ballerina-d

glide 1 secs to x: 186 y: -115



repeat

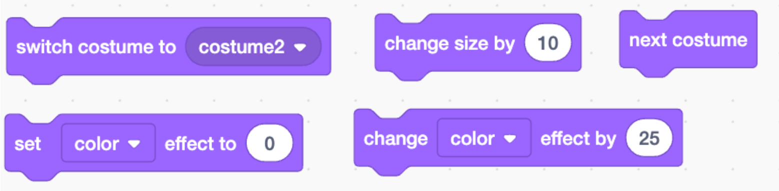
Challenge Activities

Make a sprite appear to dance forever.

Combine scripts from previous activities to make a script for the ballerina. Choose a sprite from the library that has many costumes. Use concepts of sequence and iteration to write, and then test a script that makes the new sprite dance while it moves across the stage, changes colors, and says things.

Note: Sprites do not actually talk. They show a speech bubble on the stage. However, users can create a **sound block** of their own voice recording, by clicking the **Sounds** tab, then selecting **Record** in the sound menu on the bottom left.

Blocks introduced in these challenges are found in the **CONTROL** and **LOOKS** categories.



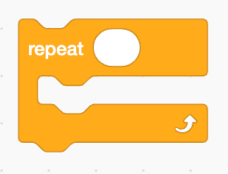
switch costume to costume2

change size by 10

next costume

set color effect to 0

change color effect by 25

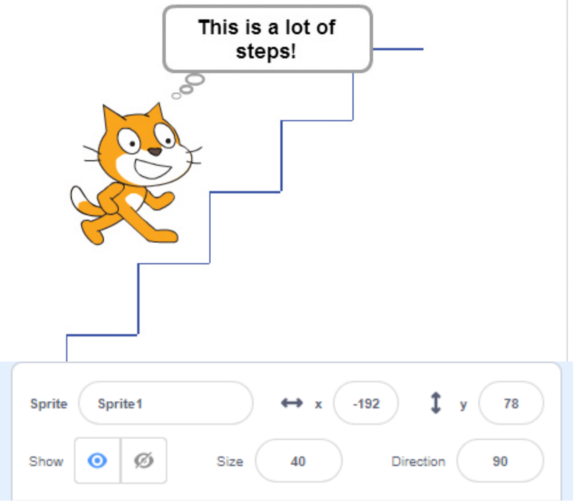


repeat

Notes

- If more than one sprite is on the stage, select the sprite you want active before beginning to create scripts for that sprite.
- Click, drag, and hover the script from the first spite you want to copy over the second sprite shown below the stage. When you select the second sprite to make it active, the copied script will automatically return to the first sprite's script area, and a copy of that script will also appear in the second sprite's script area.
- The **wait __ secs** block can use a decimal number.

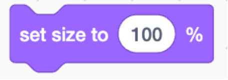
Cat Aerobics



Sprite: Sprite1 ↔ x: -192 ↕ y: 78
Show: ◉ ⊘ Size: 40 Direction: 90

Skills

- Use drawing tools to create a new sprite
- Use iteration to create a staircase
- Use blocks from the **Events** category to create a reset script for two different sprites
- To change the size of a sprite, change the input for the sprite's size in the Sprite's information box shown below the stage and above the sprite, or use this block from the **Looks** category.



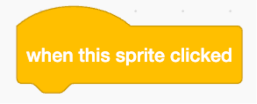
set size to 100 %

Challenge Activities

Make the cat walk up the stairs, pause half way up, say something, continue to the top, and say something else.

Draw the stairs and then make the cat wait to walk up the stairs until it is clicked.

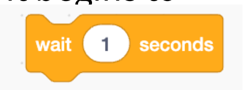
The **block** introduced in this challenge is in the **Events** category.



when this sprite clicked

Notes

- There are other ways to make the cat wait until the stairs are drawn before it begins to walk up. One option is to determine how long it takes to draw the steps and then use the **wait __ secs** block.



wait 1 seconds

- Remind learners to change the names of their sprites. Sprite 1, the cat, could be called "cat". And Sprite 2, the tiny dot, could be called "tiny dot". Change the names of sprites in the Sprite Information box below the stage and above the sprites.

Talk About

1. How has iteration helped with this activity, the Ballerina activity, and the Calendar activity?
2. Compare the script you wrote in these activities to others' scripts. Were there different ways to write the scripts for these activities?
3. How would you explain iteration to someone?
4. Using iteration, what other activities could you create with Scratch?

I pledge my head to clearer thinking,

my heart to greater loyalty,

my hands to larger service, and

my health to better living,

for my club, my community,
my country and my world.