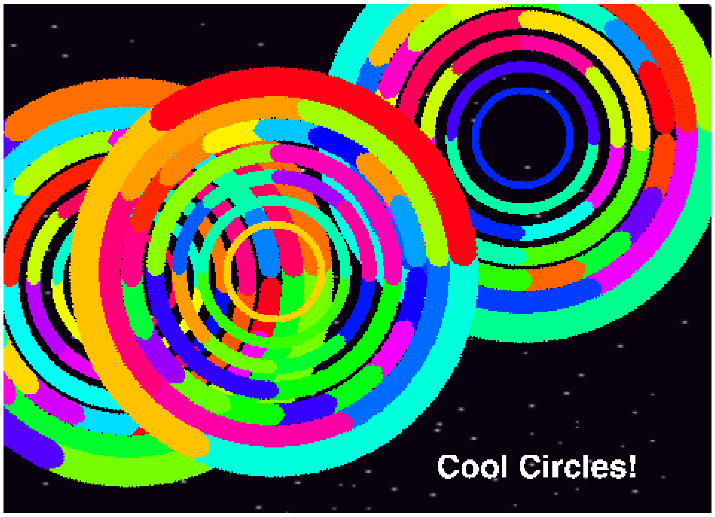
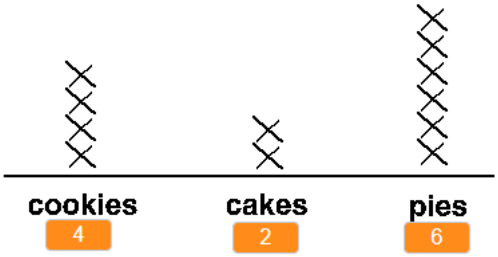
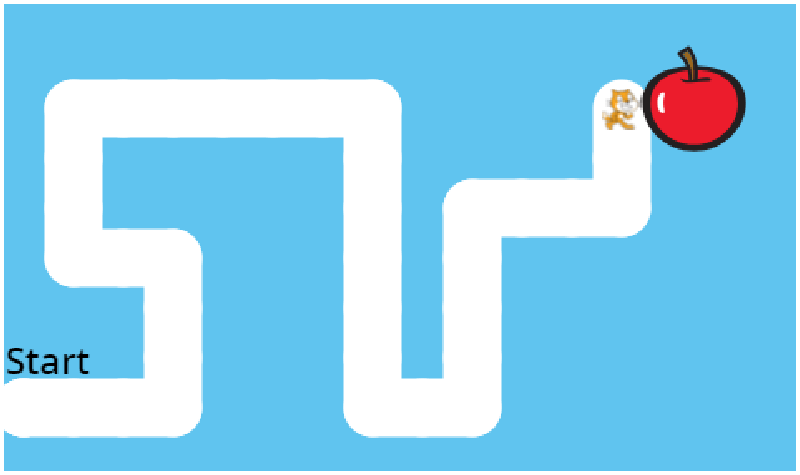
Understand Coding by Teaching Others to Code!

By:

Judy Rocke







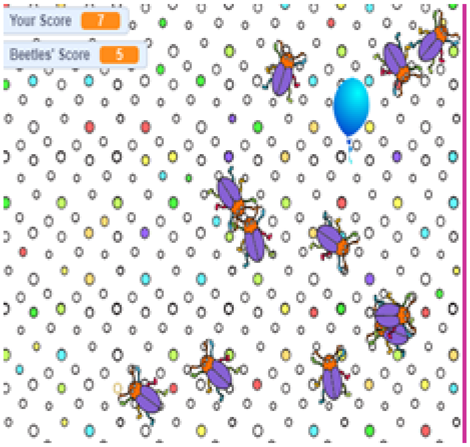
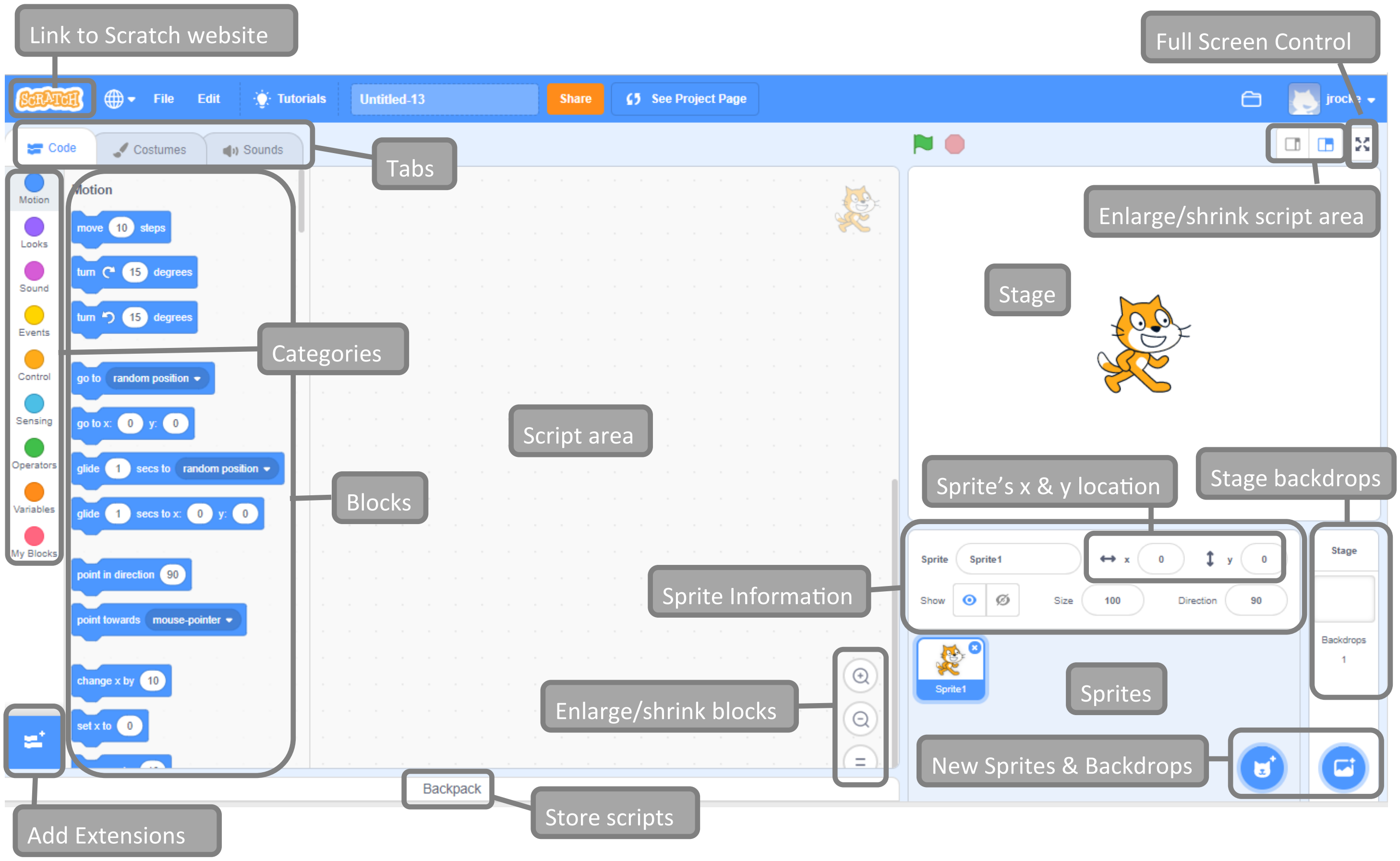


Table of Contents

Scratch 3.0 Environment



New Sprites & Backdrops: Click to choose items from a library or create your own.

Script area: Drag blocks here to build scripts. Right click in the Script area for "Undo" and "Redo".

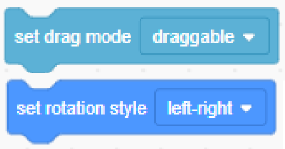
Stage: Sprites appear here. This is where your program is shown when it runs.

Categories: To make it easier to find a block, all categories are color coded.

Enlarge/shrink blocks: The size of blocks are large and hard to fit onto the script area. You can shrink these by clicking the *Enlarge/shrink (+ and —)* in the Script area.

Sprite information: All information about a sprite shows above the sprite when that sprite is clicked.

- In order to be able to drag a sprite around the stage in full screen mode, use this block from the Sensing category.
- A sprite's rotation style is set with this block from the Motion category.



Add Extensions: There are several categories that do not show on the Category list until you add them by first clicking on *More Categories*, then selecting the category you wish to add to the list. For example, the Pen category does not show until you add it to the list.

Tabs: Click one of these to switch between codes, a sprite's costume, and sound/recordings.

Backpack: Store a script or a sprite here so that it can be used in other Scratch projects.

Full Screen Control: Click to enlarge the stage to full screen. Click again to shrink stage.

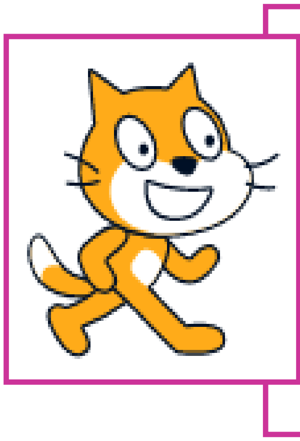
Understand How to Code by Teaching Others to Code

> "Those who know, do. Those that understand, teach."
> — Aristotle

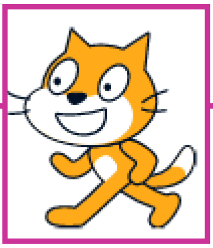
When you teach or explain something to someone, the topic often becomes clearer for you, too. Teaching coding works in the same way. Here is a list of benefits that might occur when you teach someone to code:

- Your coding skills become stronger.
- Your understanding of the principles of programming become stronger.
- Your communication skills become stronger.
- You are taking responsibility to improve your community.
- You are contributing to your community by offering your time and expertise in order to help more people learn to code.
- You can take joy in knowing that someone learned or improved their coding skills because of you.

The range of activities in this book are designed to provide fun projects for different ability learners. The lesson activities are color coded Bronze, Silver, and Gold for beginning, intermediate, and more advanced learners. You do not have to teach every activity to every learner, but you could if you wish. Look over the activities. Decide which you want to teach. Make sure you understand an activity well before you teach it to someone else. Use the sections called **Understand How This Works** to help with your understanding. As you develop your understanding of the activity, always be on the lookout for problems that might occur as you teach it. Refer back to these sections if the learner needs more help understanding the activity.



Teaching is a special skill. It is important that you provide encouragement without doing all the work for the learner. For example, it is tempting for you to take control of the learner's mouse or screen and do the work for the learner. Guard against doing that. Let the learner control the device! You also want a project to be challenging enough so the learner can problem solve and experiment, but the challenge cannot be too difficult or the learner may become frustrated and give up. In the same way, you do not want the project to be too easy. And you want the project to be fun! That is a big challenge for you. This book can help. Let's get started.



1. Create Your own Scratch Account

If you do not have a Scratch Account, create one now.

- Go to this address to create your own account: https://scratch.mit.edu
- Click **Join Scratch** and follow the instructions.
- Be sure to open your email and look for the message sent from Scratch. Use this to confirm your Scratch account. You will not be able to share your projects until you confirm your account.
- **Explore!** Try dragging blocks into the script area. Link the blocks together. Click the script you created. Add more sprites to the stage. Create scripts for the new sprites. **Explore some more!**

2. Answer these Questions

- Look over all the projects found in this book. Notice that each project has a Bronze, Silver, and Gold activity.
- Make sure you are comfortable with **all** aspects of each project. Ask yourself these questions about each project:

1. Which computer science concepts will a learner understand because of this project?
2. How can I use the **Understand How This Works** sections to help with a learner's understanding?
3. What are the important parts of this Scratch project?
4. What parts of this project might cause problems for a new learner?
5. What extensions could be added to this project?

3. Find a Learner

- Ask an adult to help you find a learner that is willing to learn about coding from you. Maybe a younger brother or sister, a cousin, or someone in your school or community? Maybe a grandparent, aunt, or uncle?
- If the new learner is under 18, get permission from the learner's parents. Be sure they are aware that you want to help their child learn to code.
- Ask them to help set up a Scratch account for the new learner. Show them how to do this by showing them how you set up your Scratch account. This is important because each Scratch on-line account must be tied to an email address. Parents will need to provide an email address in order to confirm their child's new Scratch on-line account.

4. Make Decisions

- Decide which activities you want to teach. Which activity do you find interesting?
- Consider the age and ability of the learner. It is a good idea to do Bronze and Silver lessons with the learner before doing the Gold lesson.
- Try to predict problems a learner might have with the activity you select. Ask yourself what problems you had as you worked through the activity.
- Set up a time and a place for you and the new learner to meet. Make sure your parents and the new learner's parents agree with the time and place.
- Start teaching.

5. Self-evaluate

- Be sure to **self-evaluate** each time you finish meeting with the learner. Ask yourself these questions:

1. First, and most importantly, did you and the learner have fun?
2. How successful was this lesson? Explain.
3. What went well during the lesson?
4. Were extensions added to the project?
5. What will you do differently when you teach this lesson again?
6. What coding concepts did **you** learn or understand more clearly because you taught this lesson?

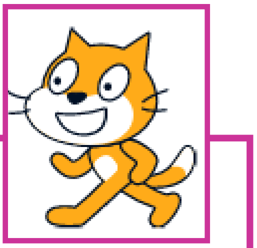
Basic Programming Concepts

This book will help you and the learner understand **sequence**, **iteration**, **modularization, variables**, **generalization**, and **conditionals.** These are all basic computer programming concepts. On the following page each of these are explained.

On the same page, there is also a chart showing in which Scratch activity these basic programming concepts are used.

It is helpful to use the correct vocabulary when teaching a person how to code.

Basic Programming Concepts



Sequence: Sequence is the order of the script. Sometime sequence is very important. In what sequence should blocks be placed? Does it make a difference if the order or sequence of the blocks are changed?

Iteration: Iterations is the number of times something is repeated. Does it repeat a set amount of times? Does it repeat until something else happens? Does it repeat forever?

Modularization: Modularization is used to “chunk” long scripts into organized, manageable parts. Modularization makes it easier for others for understand your scripts. And, by “chunking” long scripts into small parts, it is easier to find and fix any problems.

Variables: A variable can hold and change information used in a script. For an example, in a game your score can be stored as a variable. This variable can change each time you score or lose a point.

Generalization: Using variables allows you to generalize any script. With generalization instead of creating many scripts every time the input changes, you only need to create one script that will handle all the changes.

Conditionals: Conditionals allow you to control when events will occur in your activity. Will the event occur when something is clicked? When something is broadcast? When something is touching something? After a certain length of time?

	Sequence	Iteration	Modularization	Variables/ Generalization	Conditionals
What’s in the Box?	X	X	X		X
Make & Solve a Maze	X	X	X		X
Take a Survey	X	X		X	X
Pop the Balloon	X	X		X	X
Concentric Circles	X	X	X		X

1. What's in the Box?

Summary of Activity: When the box is clicked, the box changes its costume to a random picture of another sprite. The new sprite appears, and the box disappears making it look as if the random sprite had been in the box. The new sprite shows for a short time. Then it hides. The box reappears, ready to be clicked again.

Bronze: https://scratch.mit.edu/projects/119775892

Silver: https://scratch.mit.edu/projects/297354738

Gold: https://scratch.mit.edu/projects/297329606

What's in the Box?

Getting Started - Unplugged Activity

- Get a small box.
- Get six small "things" that each fit into the box. These might be a paperclip, a penny, a small cookie, a small plastic animal, a rock, and a leaf. Keep these hidden.
- Put one thing in the box. Pass the box to the new learning and ask, "What's in the box?" Let the learner guess, then open the box to see what is in the box.
- Repeat several times.

Make a Connection

- Open the Silver Scratch activity.
- Ask the learner to guess what is in the box.
- Click the box.
- Repeat several times.

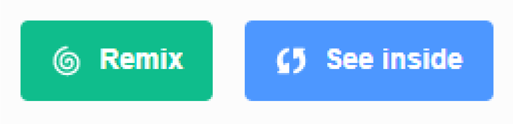
Ask:

1. How is this activity similar the game you played in *Getting Started*?
2. How is this activity different?

Use **Understand How This Works** to help you understand how the activity works. Later use it again to help the learner understand how the activity works.

Bronze Lesson - Understand How This Works

Open the Silver Scratch activity. Click **Remix.** This allows you to make changes to and copy someone's Scratch project. Read more about **Remixing and Copying** at: https://scratch.mit.edu/info/faq



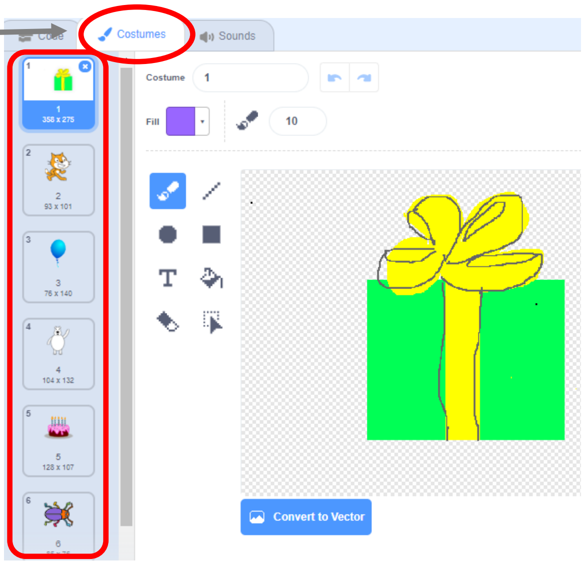
- Click **See inside.** This lets you look at the code used to run a Scratch activity.

- Click the **Costume** tab.

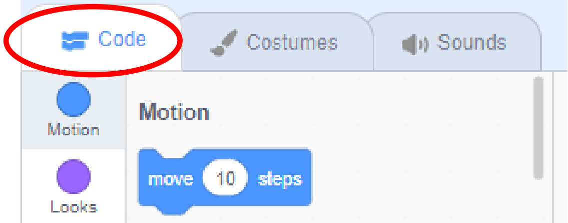
- Notice there are 9 different costumes for this sprite. Each costume came from the sprite library. Each costume is numbered.

- Click each costume.

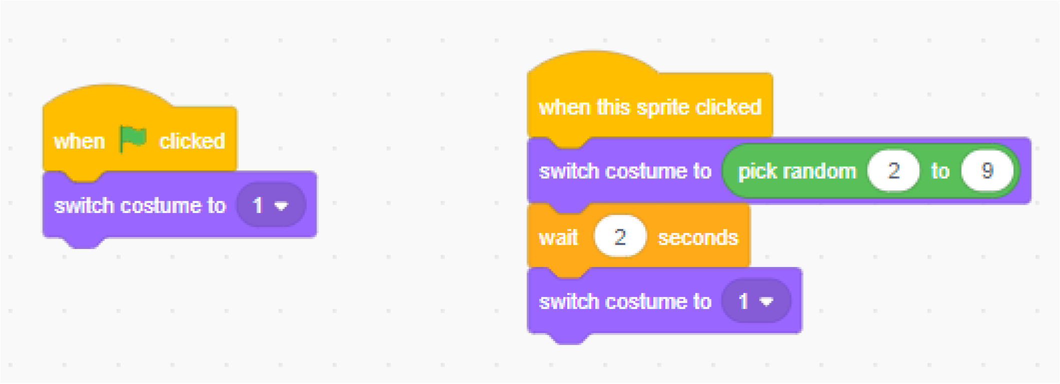
- What happens?



- Click the **Code** tab.



- Notice the blocks in the Script area. The color of the blocks indicate in which category they are found. They are color coded to make them easier to find.



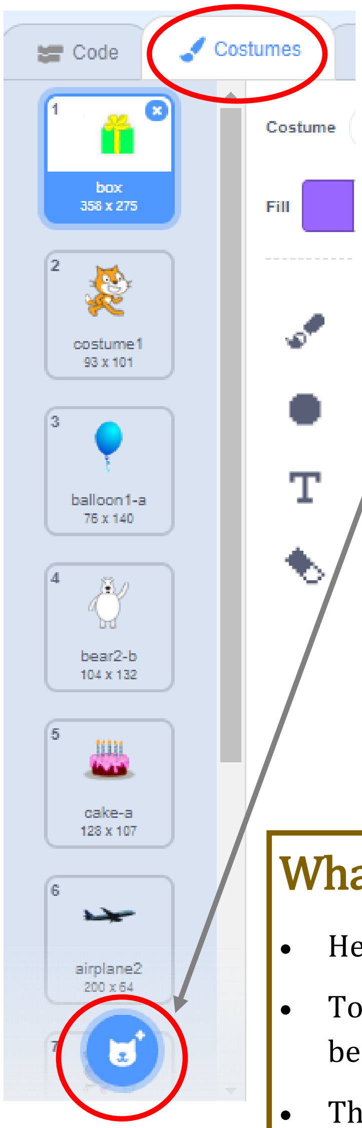
```
when 🏁 clicked
switch costume to 1 ▾
```

```
when this sprite clicked
switch costume to pick random 2 to 9
wait 2 seconds
switch costume to 1 ▾
```

- Practice creating scripts by recreating both of the scripts shown above. Drop a pick random __ to __ block into the switch costume to ___ block. Can you find each block in its category? Remember blocks are color coded. Drag the blocks to the Script area and click them together. Try pulling them apart. Pull straight down to separate blocks.

- When you finish you can delete the extra copies of the scripts by dropping them back into the script area, or you can right click on the script and select “Delete Block”.

- When your right click in the script area a choice is “Clean up Blocks”. This organizes and cleans up the script area. Why would this be helpful?



- Create a new costume.
  - ⇒ Click the **Costume** tab.
  - ⇒ Click the cat head to search the sprite library for a new costume.
  - ⇒ Double click a costume you would like to add.
  - ⇒ Remember to change the input number in the pick random __ to __ block.
- What happens if you do not change the input number in the pick random __ to __ block? Without changing the input number, will the new costume ever appear when the box is clicked?

Remember, the new learner has not completed these steps. Before you begin the Bronze Lesson Activity below, think about which of these steps helped you better understand the activity. Which parts caused you problems? Be ready to explain these ideas during the lesson.

What's in the Box? - Bronze Lesson Activity

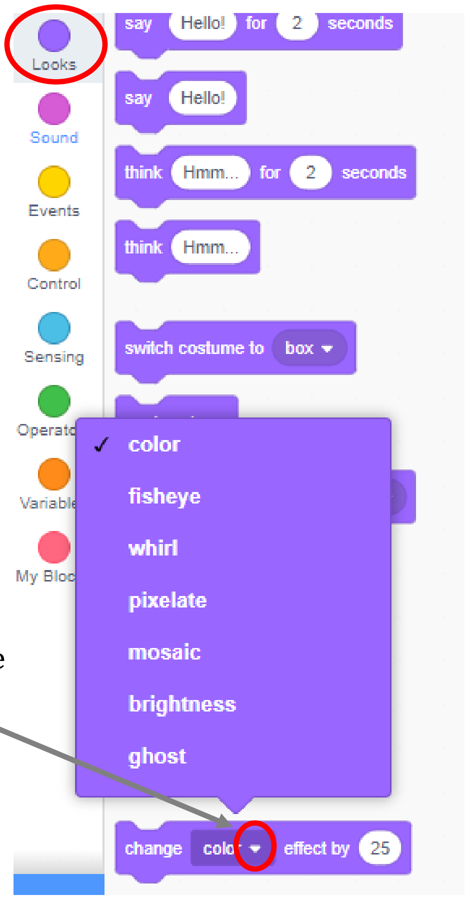
- Help the learner open and **Remix** the Bronze Scratch activity.
- Together, notice all the blocks needed to complete this activity have already been placed in the script area. Discuss what each block does.
- This activity does not work. The blocks are not in the correct **sequence.** Ask the learner to find a way to connect the blocks in the proper **sequence** so that the activity works correctly.
- Use the word "**sequence**" as you talk about the activity.
- Let the learner explore.
- Allow time for the learner to share their activity with others.

What's in the Box? - Silver Lesson Activity

- Help the learner add more costumes to their **Remixed** activity from the Bronze lesson.
- Ask the learner what happens if the input number in the pick random __ to __ block is not changed.
- Allow time for the learner to extend their project.
- Encourage the learner to share their project with others.

What's in the box?

What's in the Box? - Gold Lesson
Understand How This Works

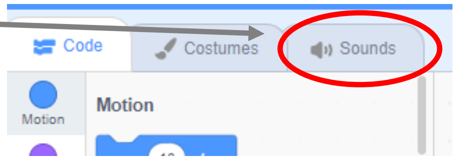


This activity uses **modularization** to break the script into "chunks". It uses recordings from the **Sounds** tab. It uses the special effects block, change (__) effect by __. This block is found in the LOOKS category. It uses **Conditional** blocks. It uses if__ then__ blocks found in the EVENTS category. It drops OPERATORS blocks which are used as input into other blocks. And it uses **iteration**.

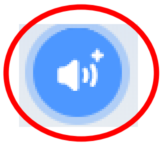
Effects:

- Open and **Remix** the Gold Scratch activity.
- Click the LOOKS category.
- Click the small triangle in the change (__) effect by __ block. Notice a menu appears. There are 7 different effects that can be used.
- Experiment with the special effects.

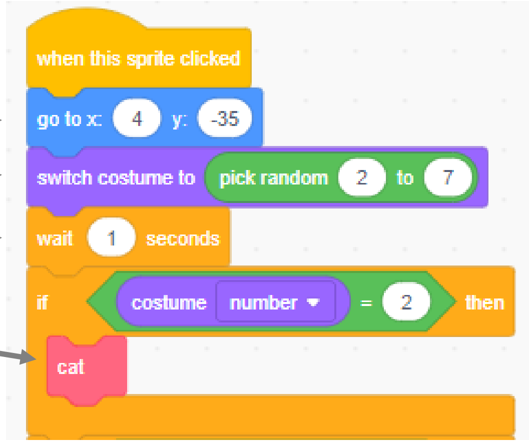
Recordings:



- Click the **Sounds** tab.
- Notice this activity already uses three recordings. To hear each sound, click the sound, then click the blue triangle shown below the sound's recording.
- Click the speaker below the sounds to open the sound library.
- Experiment with the recordings.



Modularization:

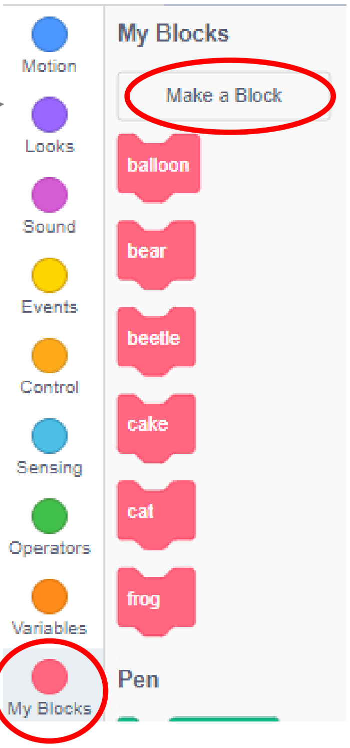


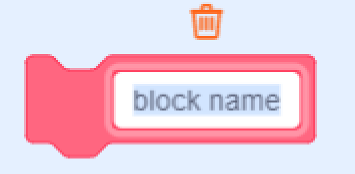
```
when this sprite clicked
go to x: 4 y: -35
switch costume to pick random 2 to 7
wait 1 seconds
if costume number = 2 then
    cat
```

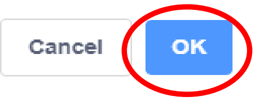
- Look at the partial script shown here.
- When the sprite is clicked, it goes to a certain location on the stage.
- It switches to a random costume.
- It waits for 1 second.
- **If** the random costume is #2, **then** it will cat.
- Look at the code for this activity. Notice **if** the costume number is #3, **then** it will balloon.
- Cat, balloon, and the other pink blocks are blocks that you create and define. "Chunking" the script into short parts named and defined by the pink blocks is an example of how to use **modularization**. If there is a problem with the cat, you know where to look in the script to fix the problem.

Create Blocks:

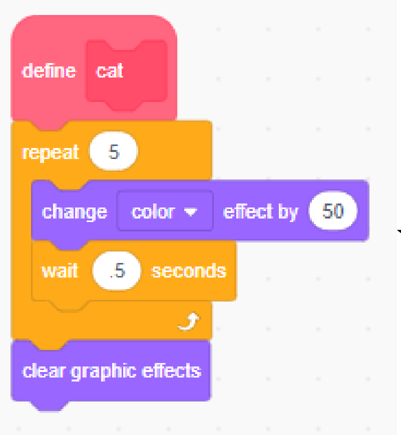
- Click the My Blocks category.
- Click **Make a Block.**
- A window will open. Click **block name** and type a name for your new block.
- Click **OK.**
- Two blocks appear:





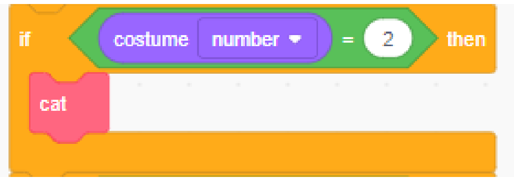


⇒ One is used to define the block.



```
define cat
repeat 5
    change color ▾ effect by 50
    wait .5 seconds
clear graphic effects
```

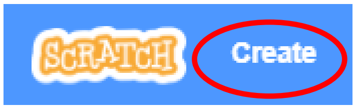
⇒ One is used in the script.



```
if costume number ▾ = 2 then
    cat
```

Remember, the learner has not completed these steps. Before you begin the Gold Lesson Activity below, think about which of these steps helped you better understand the activity. Which parts caused you problems? Be ready to explain these ideas during the lesson.

What's in the Box? - Gold Lesson Activity



- Help the learner open a new Scratch activity. Click **Create**.
- Experiment with the recordings, special effects, dropping OPERATORS blocks into **conditional** if__ then__ blocks, and creating and defining your own blocks from the MY BLOCKS category.
- Use the words **sequence**, **conditionals**, **modularization**, and **iteration** as you talk about the activities.
- Help the learner open and **Remix** What's in the Box? - Gold Scratch activity.
- Discuss how each new pink block is defined, what the does, and how it is used in the scripts.
- Notice how **iteration** is used for the new blocks, bear, beetle, cat, and balloon.
- Allow time for the learner to add special effects and **modularization** to their saved What's in the Box? - Silver Scratch activity.
- Allow time for the learner to celebrate their success by sharing their activity with others.

Other Scratchers cannot see your projects until you click **Share.** You cannot **Share** until you use the email message from Scratch to confirm your Scratch account.

Self-Evaluate Remember to self-evaluate. Review the questions on page 5.

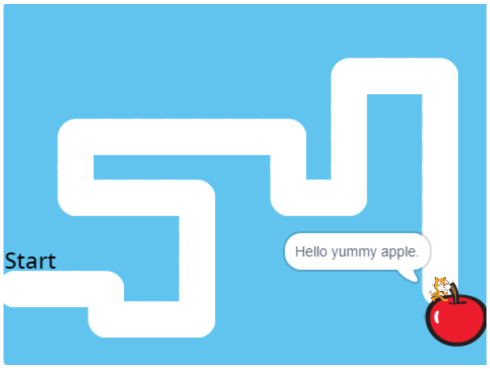
2. Create & Solve a Maze

Summary of Activity: The Bronze activity lets you use **modularization** to solve a maze. The Silver activity lets you create a maze using **modularization**. The Gold activity lets you create a maze and write code using **modularization** to solve the maze.

Bronze: https://scratch.mit.edu/projects/168085452

Silver: https://scratch.mit.edu/projects/168208132

Gold: https://scratch.mit.edu/projects/315743870



Create & Solve a Maze

Getting Started - Unplugged Activity

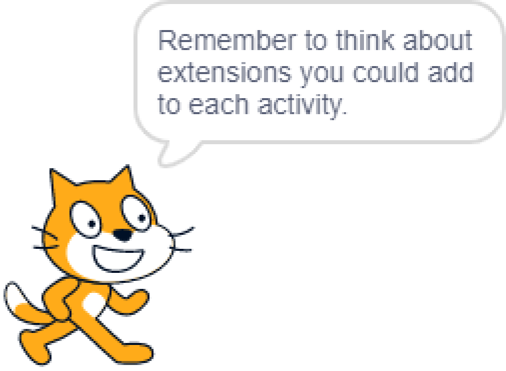
- Use post-it notes to create a path of a maze on the table. Or, if you want to be more active, use sheets of paper to create a path of a maze on the floor. Label the start and the finish.
- Ask the learner to stand on the start of the maze or to put their pointer finger on the start of the table-top maze.
- Give direction to complete the maze. For example, you might say, "step, step, turn left, step, step, step, turn right, step" to finish the maze. If you use the post-it notes make your fingers move like they are taking steps.
- Does the **sequence** of the directions matter?
- Spread the post-its or paper farther apart, but keep the same path. Do the directions for solving the maze change?
- Take turns giving directs and walking the maze. Pay attention to the **sequence** of directions. Change the maze. Do this several time.

Make a Connection

- Open the Bronze Scratch Activity.
- Click the green flag. Notice the cat is at the start of the maze. The yellow box is at the finish.
- Click the cat.

Ask:

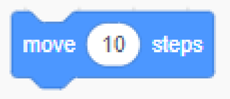
1. How is this activity similar to the game you played in *Getting Started*?
2. How is this activity different?


Remember to think about extensions you could add to each activity.

Use **Understand How This Works** to help you understand how the activity works. Later, use it again to help the learner understand how the activity works.

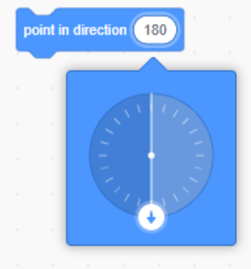
Understand How This Works - Bronze Lesson

- Open the Bronze Scratch activity. Click **See inside.** Notice the squares on the maze. Even though it looks like each square is only one step for the cat, this is not the case. The cat needs to move 30 steps to go to the next square on the maze.
- Test this. Pull out this block from the MOTION category. Try inputting different numbers into this block. Click the block each time you change the input. Notice how the cat moves. How many steps does it take the cat to move from one square to the next?

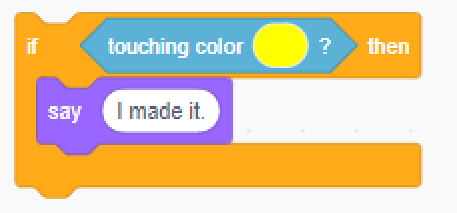


```
move 10 steps
```

- Instead of "step", the code uses "move forward", "move back", "move up", and "move down". This a good example of **modularization.** Look at the script to see how each of these blocks are defined and used. How many steps does the cat move each time?
- Also notice that instead of "turn left" or "turn right" the code uses a point in direction __ block.
- Find this block and experiment with it. How is each direction shown?



```
point in direction 180
```



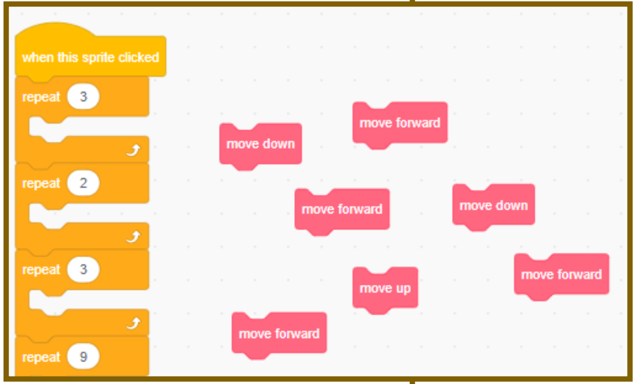
```
if touching color ? then
  say I made it.
```

- Notice this **conditional** block at the end of the long script. What does it do?
- Find the touching color__? block in the SENSING category. How can you change the yellow to a different color?

Remember, the learner has not completed these steps. Before you begin the Bronze Lesson Activity below, think about which of these steps helped you better understand the activity. Which parts caused you problems? Be ready to explain these ideas during the lesson.

Create & Solve a Maze - Bronze Lesson Activity

- Open and **Remix** the Bronze Scratch activity.
- Pull all the pink blocks from the script.
- Let the learner place the pink blocks back in the correct locations. Discuss how **modularization** is used.
- Next, delete all the input numbers in all the repeat blocks. Discuss how **iteration** is used.
- Let the learner input the correct numbers.
- Allow time for the learner to share their project with others.



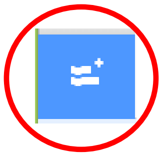
```
when this sprite clicked
repeat 3
repeat 2
repeat 3
repeat 9

move down
move forward
move forward
move down
move forward
move up
move forward
```

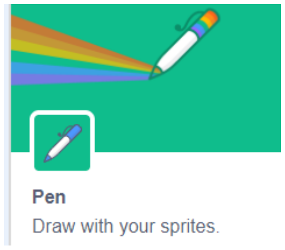
Create & Solve a Maze - Silver Lesson

Understand How this Works

This activity uses blocks from the PEN category. This category does not show in the list of categories when you open a new Scratch project. You can open this category.

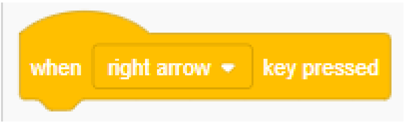
- Open a new Scratch project.
- Click the blue box below the list of categories.
- This opens a new window. Click the Pen.
- Now look at the list of blocks. Nine Pen blocks show.
- Experiment with each of these blocks. What does each block do?
- Open the Create & Solve a Maze - Silver Scratch activity. Notice how each of the pen blocks are used in this script.
- For an example of a completed maze, press the right arrow. Scroll down in the script area to find the script that starts with when right arrow key is pressed.
- How does the cat use a pen block to draw a maze?

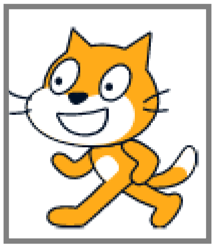




Create & Solve a Maze - Silver Lesson Activity

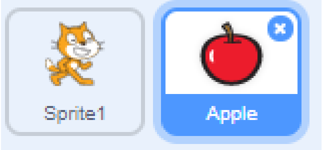
- Open the Create & Solve a Maze - Silver Scratch activity.
- Press the right arrow to see an example of a maze. Find the script in the script area that starts when right arrow key pressed and discuss what it does.
- Click the green flag.
- Drag the apple sprite to a new location on the stage.
- Ask the learner to click the move forward", "move back", "move up", and "move down" blocks to create their own maze which finds the apple in its new location. As they use these blocks, they should record on a sheet of paper the number of times each is used. For example, they might write, "Move forward three times. Move up six times. Move back five times."
- Click the green flag. Their maze will disappear and the cat will return to the start.
- Look at the list of commands they wrote. Ask the learner to use the list, with **modularization** and **iteration,** to make the cat recreate that maze when left arrow key pressed. This script will look similar to the script that starts with when right arrow key pressed. Does the maze lead the cat to the new location of the apple?



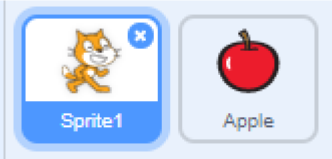


Create & Solve a Maze - Gold Lesson Activity

- Open and **Remix** the Create & Solve a Maze - Gold Scratch activity.
- Click the apple sprite below the stage. Notice how it becomes active. Now you can see the scripts for the apple sprite. There is only one script for the apple.

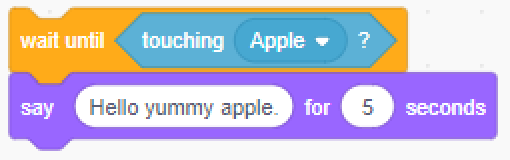


- Click the cat sprite shown below the stage. Notice how it becomes active. Now you can see all the scripts for the cat.

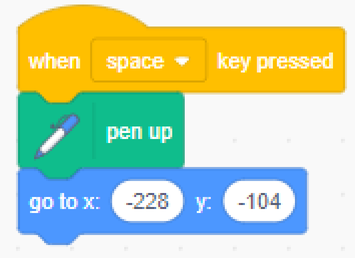


If there is more than one sprite, be sure you have the correct sprite active before you create scripts for that sprite.

- Look at these two scripts for the cat.
- What does each do?
- Why is each helpful?



```
wait until touching Apple ?
say Hello yummy apple. for 5 seconds
```



```
when space key pressed
pen up
go to x: -228 y: -104
```

- Allow time for the learner to create a maze, find a way for the cat to go back to the start of the maze, and finally solve the maze.
- How will the learner start each script? Will one start with the when right arrow key pressed block?
- Encourage the learner to share what they have created with others.

Self-Evaluate

Remember to self-evaluate. Review the questions on page 5.

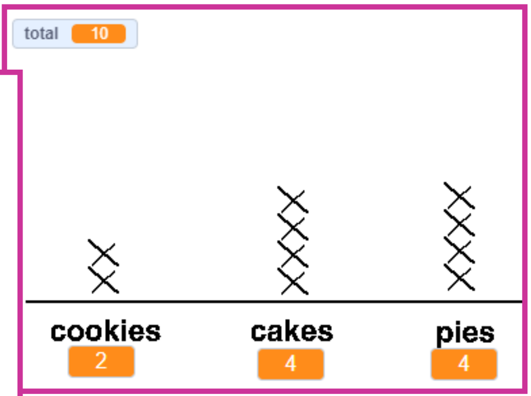
3. Take a Survey

Summary of Activity: This activity asks the user to choose which is their favorite desert. The choices are cookies, cakes, or pies. The results are recorded in a list that the computer uses to create a line plot. This shows that 10 people took the survey. Two liked cookies. Four liked cakes. Four liked pies.

Bronze: https://scratch.mit.edu/projects/145934607

Silver: https://scratch.mit.edu/projects/146771090

Gold: https://scratch.mit.edu/projects/317025907



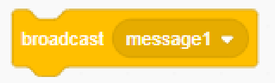
Take a Survey

Getting Started - Unplugged Activity

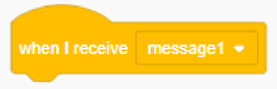
- Play a game. Listen to the TV or the radio for about 2 or 3 minutes. Stand up, then sit down every time you hear the word “the”.
- Play the game again. You and the learner can each pick a different word to use as the signal for you each to stand and sit. Who stands the most?
- Play the game again. This time pick a new word for a signal and decide on a new thing to do when you hear the signal. Make it something fun. Will you bark like a dog? Sing a song?

Make a Connection

- **Take a Survey** activity uses this block broadcast (__) from the EVENTS category to send messages to other sprites.



- You can give the broadcast any name.
- When a sprite receives a broadcast, it is a signal which makes a sprite do something.

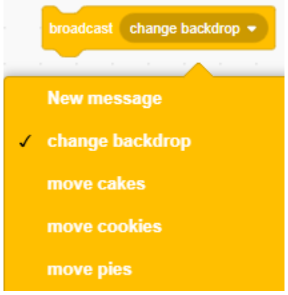


Ask:

1. How is this broadcast similar to the game you played in *Getting Started*?
2. How is this activity different?

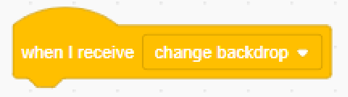
Understand How This Works - Bronze Lesson

This activity uses **new sprites** that you create, a **list** and **variables** to hold survey responses and data, **broadcasts** that send and receive messages, and a way to **ask** each user the survey question.



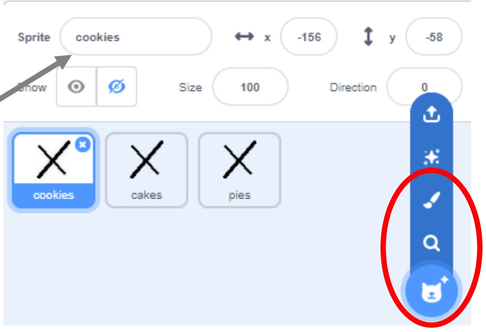
Broadcasts:

- Create a new broadcast by clicking **New message** in the drop-down menu.
- Use the when I receive (__) block, for a sprite to receive the broadcast.



New Sprites:

- Open and **Remix** the **Bronze** Scratch activity.
- Notice this activity has 3 sprites named cookies, cakes, and pies.
- Practice creating and naming new sprites. Click the cat head, then click the paint brush to draw sprites.
- Click here to change the name of any sprite.

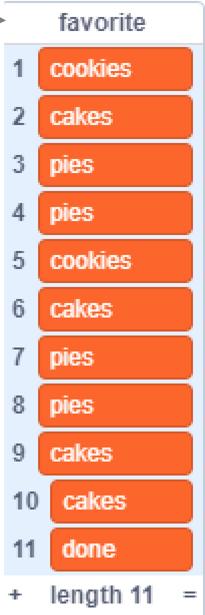


Sprite cookies x -156 y -58

Show Size 100 Direction 0

A list is used to collect and store large amounts of data.

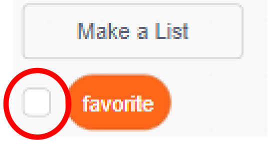
- This activity uses a list called *favorite* to collect all the answers to the survey. Each row number of the list is stored in a **variable** called list item #. Cookies are located at list item # 1 and 5. Pies are located at list item # 3, 4, 7, and 8. Which list item # shows where cakes are located?
- The *item* at list item # 11 is "done". This shows that the survey is over.
- Create a new list:
  - ⇒ In the VARIABLES category, click **make a list**.
  - ⇒ Give the list a name. Then click **ok**.
  - ⇒ Hide or show a list on the stage by clicking in the box before the list name.
- In this activity the list is filled when people input their answer to the survey question. But there are other ways to fill a list. You can click in the list and type data, or you can fill a list with blocks shown in the VARIABLES category. **Explore** all the ways you can find to fill, delete, and move items in a list.



favorite

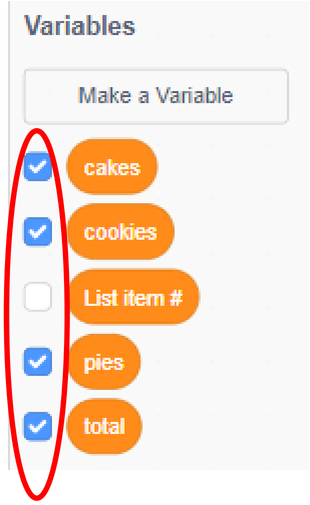
1	cookies
2	cakes
3	pies
4	pies
5	cookies
6	cakes
7	pies
8	pies
9	cakes
10	cakes
11	done

+ length 11 =



Variables:

- This activity uses five variables shown here.
- The total number of times cakes, cookies, or pies were chosen during survey are counted in their variables named cakes, cookies, or pies.
- Each row number of a list is identified by the list item # variable.
- The total number of responses to the survey is shown in the variable called total.
- Variables will show on the stage if the box before its name is checked. Without a check, the variable does not show on the stage.
- Make variables in the same way you made a list. In the VARIABLES category, click **make a variable**. Always choose meaningful names for your lists and your variables.



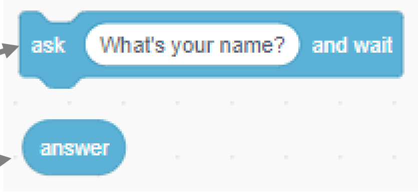
Variables

Make a Variable

☑ cakes
☑ cookies
□ List item #
☑ pies
☑ total

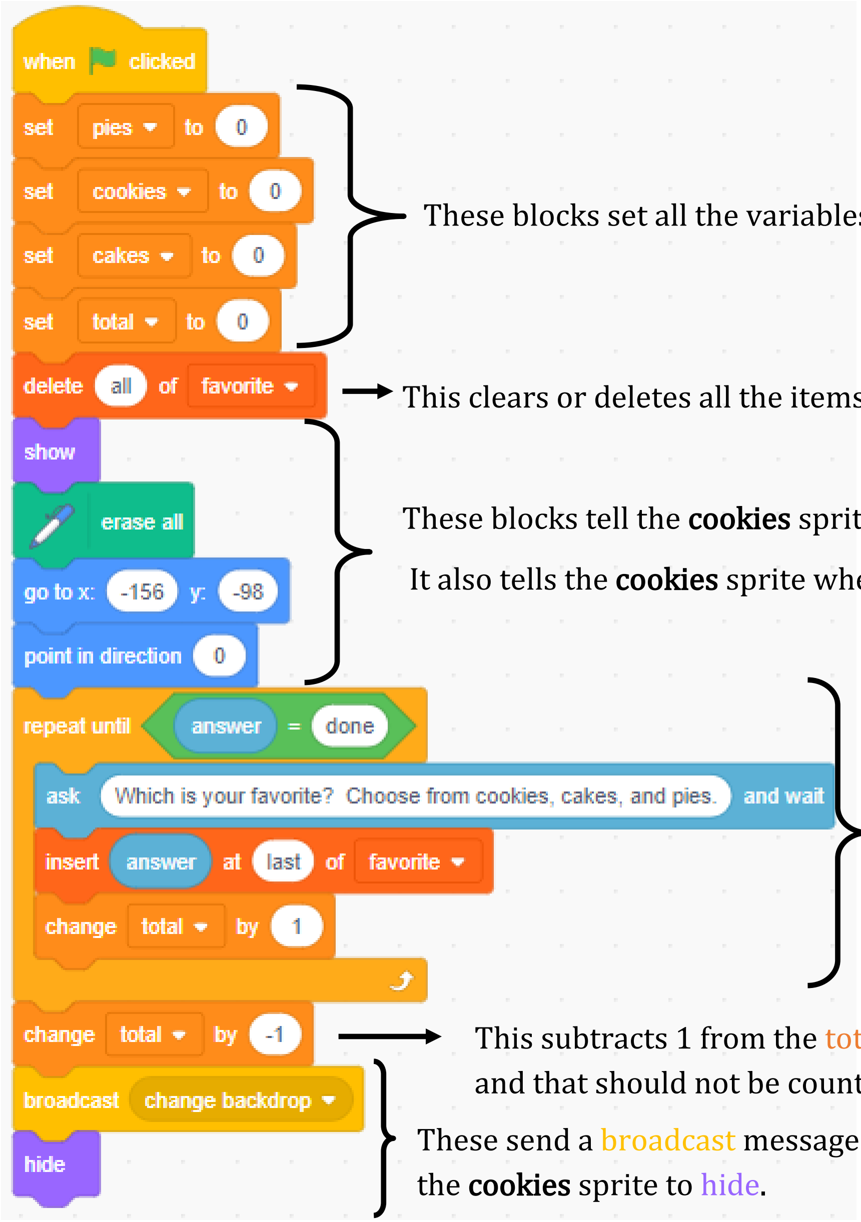
Ask and Wait:

- Use this ask (__) and wait block from the SENSING category to ask the user a question. Type your question into the input area. The program waits until an answer is given.
- This answer block holds the user's answer like a variable. It allows you to use the answer in your script.



```
ask (What's your name?) and wait

(answer)
```

What does the **cookies** sprite do when the green flag is clicked? Look at one of its scripts:



```
when flag clicked
set [pies] to (0)
set [cookies] to (0)
set [cakes] to (0)
set [total] to (0)
delete (all) of [favorite]
show
erase all
go to x: (-156) y: (-98)
point in direction (0)
repeat until <(answer) = (done)>
    ask (Which is your favorite? Choose from cookies, cakes, and pies.) and wait
    insert (answer) at (last) of [favorite]
    change [total] by (1)
change [total] by (-1)
broadcast (change backdrop)
hide
```

These blocks set all the variables to zero.

This clears or deletes all the items in the list *favorite*.

These blocks tell the **cookies** sprite to show and erase all **X**s on the plot.

It also tells the **cookies** sprite where to go and in which direction to point.

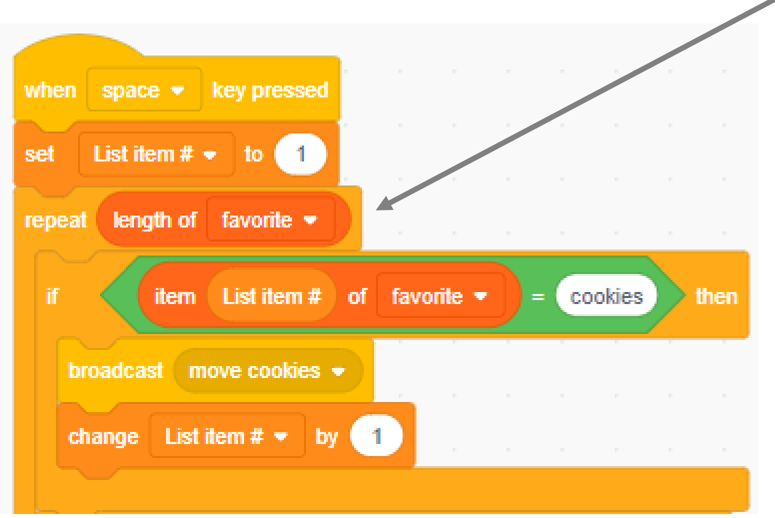
These blocks make the computer continue to **repeat**, "Which is your favorite? Choose from cookies, cakes, and pies.", **until** the answer is, "done." Each time an answer is given the answer is added to the last spot in the *favorite* list, and the total number of survey responses is changed by 1.

This subtracts 1 from the total because one response was, "done", and that should not be counted as a survey response.

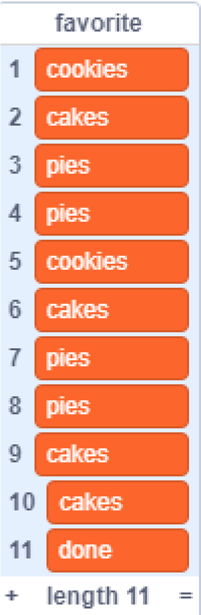
These send a broadcast message to change the backdrop and tell the **cookies** sprite to hide.

- Is the "repeat until answer = done" block a **conditional** block, an **iteration** block, or both?
- Why is the answer block considered a **variable**?

- Find the **cookies** sprite's script that start with when space key is pressed. A part of that script is shown below.
- Notice when space key is pressed, this script sets the list item # 1. This is the start of the *favorite* list.
- Notice the first item in the list is **cookies**. The scripts says, if the item is **cookies**, then broadcast move cookies and change the list item # by 1. Now the list item # is set to 2, and the next item in the list is **cakes**.
- Look at the script. What happens if the item is **cakes**? What happens if the item is **pies?**
- Notice this process of looking at each item in the list and broadcasting a message repeats the length of the list. "Done" indicates the list is finished.

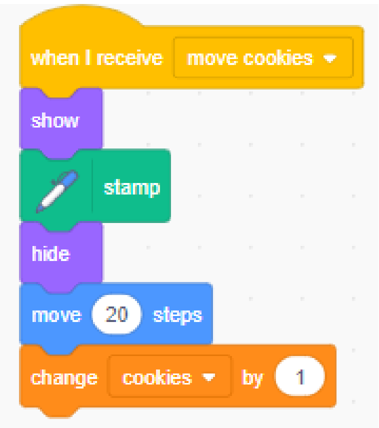


```
when space key pressed
set List item # to 1
repeat length of favorite
    if item List item # of favorite = cookies then
        broadcast move cookies
        change List item # by 1
```

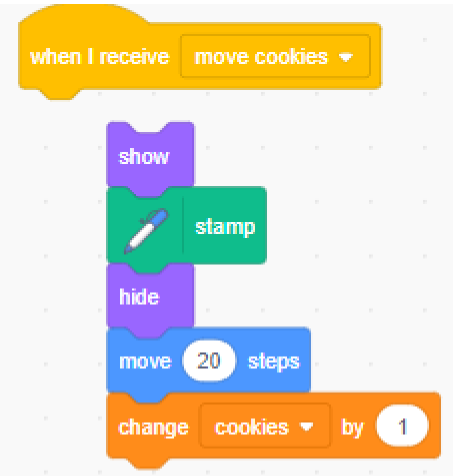


favorite	
1	cookies
2	cakes
3	pies
4	pies
5	cookies
6	cakes
7	pies
8	pies
9	cakes
10	cakes
11	done
+	length 11 =

- Notice the scripts shown so far are only for the **cookies** sprite. The other sprites do not have all of these scripts. Only one sprite needs to control all this activity. It is not necessary to repeat these directions again, but the other two sprites do have a script similar to the one for the **cookies** sprite shown on the left below. Why? What does it do?
- To see what this script does, pull it apart. Then click show. What happens? Why do the other sprites need a similar script?

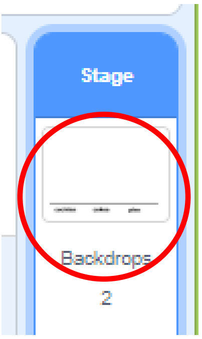


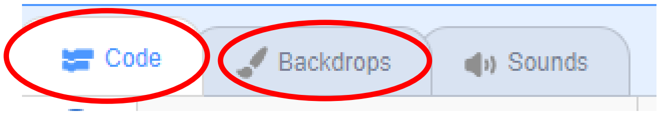
```
when I receive move cookies
show
stamp
hide
move 20 steps
change cookies by 1
```



```
when I receive move cookies

show
stamp
hide
move 20 steps
change cookies by 1
```

- Notice even the backdrop has two scripts. To see these, first click the backdrop below the stage. Then click **Code** above the block area to see the code. What do the scripts each do?





- Click **Backdrop** to see the all the backdrops. This is also where you can edit the way a backdrop looks using the paint tools.

Take a Survey - Bronze Lesson Activity

- Open and **Remix** the Bronze Scratch activity. Click the green flag.
- Make the list *favorite* show by clicking the box in front of *favorite* under **Make a List** in the block area.
- Survey several people. Notice as you enter their responses, the list *favorite* fills. After the last person's response, type "done" as the answer to the question. This will end the survey.
- Press the space key and watch the line plot fill.
- Ask the learner to think of another question that has three responses which could be use to survey people.
- Explain they can use their **Remix** to create a new survey. Only a few changes need to be made to the **Remix**:

1. Rename the sprites.

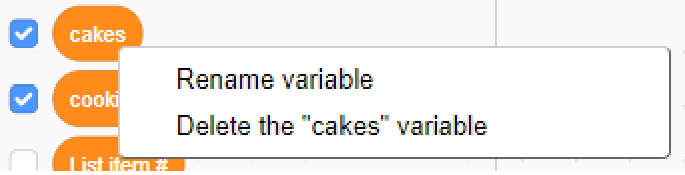
⇒ Change names in the Sprite Information box shown above the sprite when it is active.

2. Change the question.

⇒ Type your new question and the choices for the user in the ask (__) and wait block.

3. Change the names of the variables.

⇒ Right click the name of the variable in the block area. Select **Rename variable**.



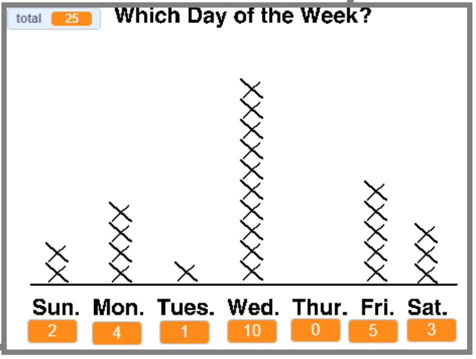
4. Change the backdrop to show the new names of the variables.

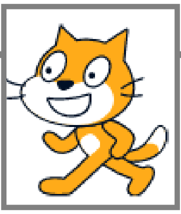
⇒ Change a backdrop with the paint tools.

- Allow time for the learner to survey people and share their work with others.

Take a Survey - Silver Lesson Activity

- Open the Scratch *Day of the Week Calculator* activity found at: https://scratch.mit.edu/projects/51552484
- Use the activity to find on which day of the week you and the leaner were born.
- Let several people use the *Day of the Week Calculator* to find on which day they were born.
- Open the Silver Scratch activity. Let the learner record their responses in the activity. This will fill the list with days of the week. Type "done" when the last response is entered.
- Press the space bar to see the line plot fill.
- Ask the learner to think of a new survey question with more than three choices.
- Help the learner create a new Scratch activity that will ask the new question, give the new choices, and create a new line plot of the survey's results.
- Allow time for the learner to survey others and show them their work.





Take a Survey - Gold Lesson

Understand How This Works

This Gold Scratch activity collects a survey's data in a **list**, and it uses it to create a line plot. It also uses the data with blocks from the OPERATORS category and **variables** to **generalize** and calculate the **percent of times** each answer was given. If you have questions, you can look at the Gold Scratch activity for help, but first try this on your own.

Change any Fraction into a Percent

- Any fraction can be changed into a percent with a few calculations.

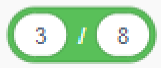
⇒ For example, if the fraction is 2/5 you can turn it into a percent by first finding the answer to 2 ÷ 5. Then multiply the answer by 100. Finally round this to the nearest whole number. So 2 ÷ 5 = .40. And .40 x 100 = 40. So 2/5 = 40%.

⇒ If the fraction is 3/8 then, 3 ÷ 8 = .375. And .375 x 100 = 37.5, which rounds to 38. So 3/8 is approximately 38%.

OPERATORS Blocks in Scratch

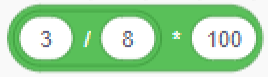
- How does Scratch handle these calculations?

⇒ Use this block to show 3 ÷ 8.



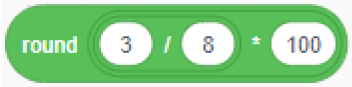
```
3 / 8
```

⇒ Drop that block into the multiplication block to multiply the answer by 100.



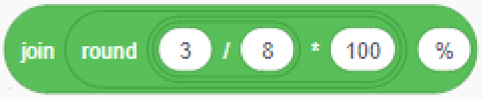
```
3 / 8 * 100
```

⇒ Drop that block into the round block to round to the nearest whole number.



```
round 3 / 8 * 100
```

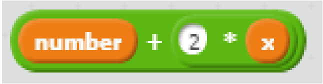
⇒ This will give the correct calculation. To add a percent sign to the answer use a join block. Notice how the percent is now joined with a % sign.



```
join round 3 / 8 * 100 %
```

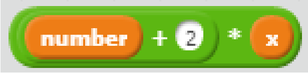
Notice how Scratch handles Order of Operations.

Order of Operations in Scratch



number + 2 * x

Notice in this block (2 * x) is on the top layer. It fits into the (number + ___). Because it is on the top layer, Scratch will first multiply 2 * x , then it will add the product to number.

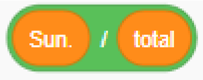


number + 2 * x

Notice in this block (number + 2) is on the top layer. Scratch will first add number + 2 , then multiply the sum by x .

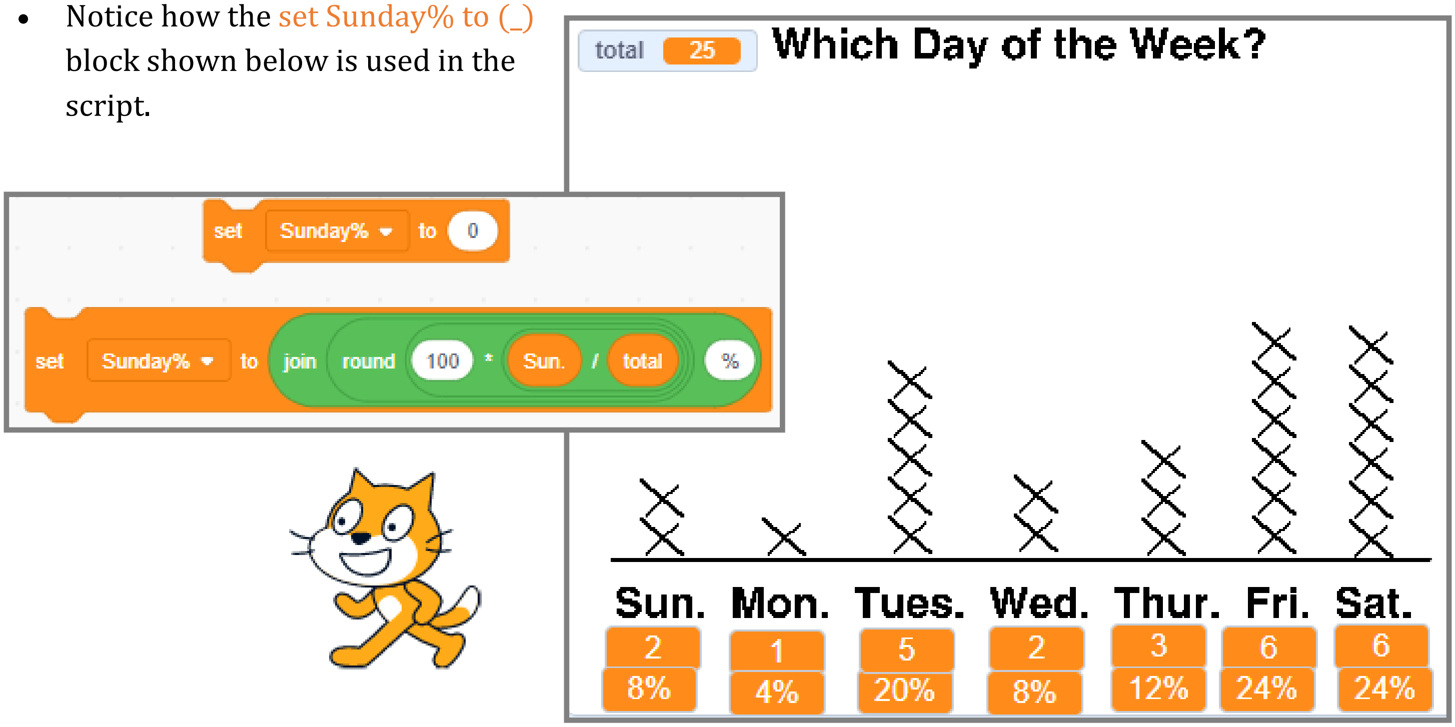
Be careful how you layer OPERATORS blocks.

- OPERATORS blocks will also take a **variable** as input. For example, look at the line plot below. Sunday has 2 out of 25 total responses. Sunday's fraction would be 2/25. Using **variables** you can **generalize** that fraction to Sun. ÷ total.



Sun. / total

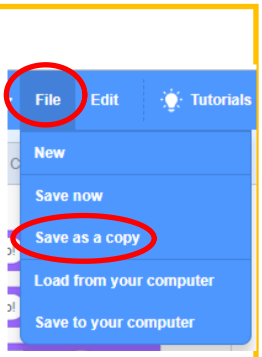
- Using variables as input instead of numbers allows Sunday's count and the total count to change with the results of the survey. This is a good example of **generalization**.
- Notice how the set Sunday% to (_) block shown below is used in the script.



set Sunday% to 0

set Sunday% to join round 100 * Sun. / total %

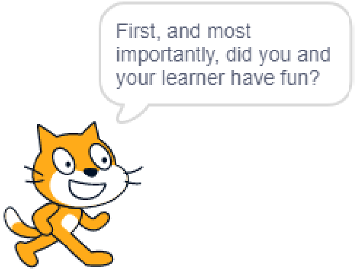
Take a Survey - Gold Lesson Activity



- Open and copy the learner's Silver Scratch activity which they created in the Silver lesson.

⇒ Click **File** above the block area. Then click **Save as a copy**.

- Help the learner find a way to display a percent for each choice in their survey on their line plot.
- Be sure to use the vocabulary **sequence**, **iteration**, **conditionals**, **variables**, and **generalization** as you help the learner.
- Encourage the learner to **Share** their project so other Scratchers can see their work.
- Allow time for the learner to use their Survey and share the results with others.

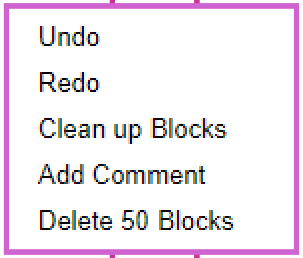

First, and most importantly, did you and your learner have fun?

Self-Evaluate

Remember to self-evaluate. Review the questions on page 5.

You can add comments to your scripts. Comments help others understand your scripts. It is a good way to communicate with the person using your activity.

⇒ To add a comment, right click in the script area. This box will appear. Click, "Add Comment".



⇒ This allows you to attach a comment to a particular script or leave a comment in the script area. Experiment with leaving and deleting comments.

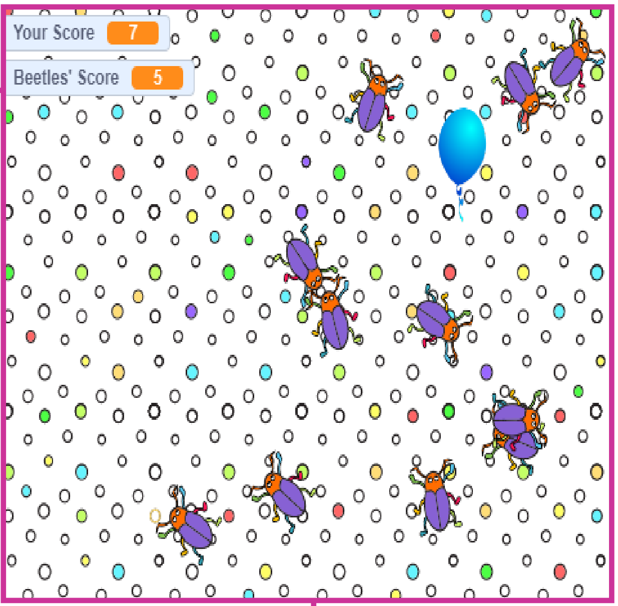
Besides adding comments to your scripts, this box is very helpful.

⇒ If your blocks are on top of one another and hard to read, click "Clean up Blocks". This organizes the scripts for you and makes them easier to follow and understand. Why is that important?

⇒ If you are creating scripts for one sprite and you want to change something, you can "Undo" or "Redo" scripts for that sprite. (Caution, once you click to another sprite, you can no longer "Undo" or "Redo" blocks for the original sprite.) Experiment with this. What did you learn?

4. Pop the Balloon

Summary of Activity: In the Bronze activity, use the mouse-pointer to pop the balloon. You earn a point for each balloon you pop. When you earn 20 points, you win. In the Silver activity be careful of the beetle. If the mouse -pointer touches the beetle you loose a point. If the number of times the mouse-pointer touches a beetle reaches 5, you loose and the beetle wins. In the Gold activity the beetle creates clones. Avoid the beetle and the clones!



Bronze: https://scratch.mit.edu/projects/317896911

Silver: https://scratch.mit.edu/projects/317149575

Gold: https://scratch.mit.edu/projects/317516153

Pop the Balloon

Getting Started - Unplugged Activity

- What is the difference between "**and**" and "**or**"?

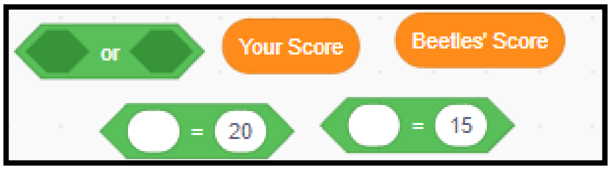
⇒ "**And**" means all requirements must be met.

⇒ "**Or**" means only one of the requirements must be meet.

- Touch your nose if you are under 10 years old **or** over 10 years old. (Everyone should touch their nose unless they are exactly 10 years old.)
- Blink your eyes if you have a dog **and** a cat for a pet. (If you have both a dog and a cat, blink your eyes. If you only have one, do not blink your eyes. "**And**" means you must have both.)
- Snap your fingers if you were born in Illinois **and** California. (Both requirements must be met, so no one should snap their fingers.)
- Continue making up "**and**" and "**or**" questions to ask each other.

Make a Connection

- The Silver Scratch activity uses these blocks:



or

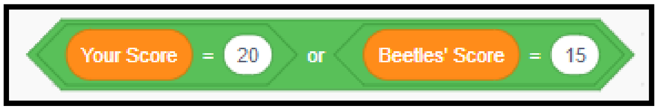
Your Score

Beetles' Score

( ) = 20

( ) = 15

To create this combined block:



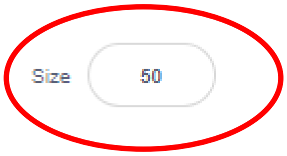
Your Score = 20 or Beetles' Score = 15

Ask:

1. What does this combined block mean?
2. How is this combined block is similar to the game you played in *Getting Started*?

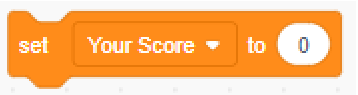
Understand How This Works - Bronze Lesson

- Open the Bronze activity.
- Notice the balloon size is set to 50%. This is found below the stage.



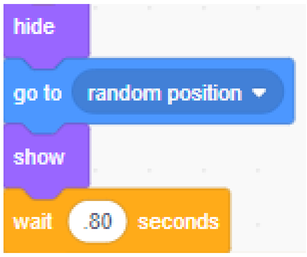
```
Size 50
```

- Click the green flag to start the activity.
- Notice Your score is set to zero.



```
set Your Score to 0
```

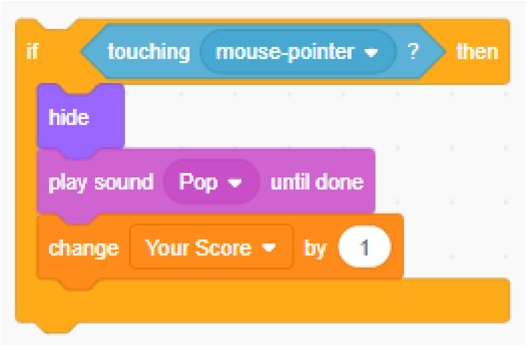
- Notice the balloon hides, moves to a random position, shows, and waits a short amount of time.



```
hide
go to random position
show
wait .80 seconds
```

- Find these blocks in the script.
- What does each block do?

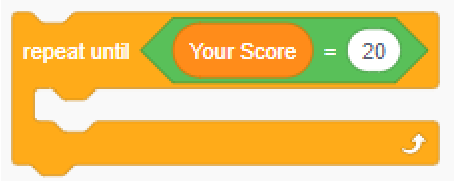
- Touch the balloon with the mouse-pointer.
- What happens?
- Notice each time the pointer touches the balloon, the balloon hides, a Pop sound is played, and Your Score is changed by 1.



```
if touching mouse-pointer ? then
    hide
    play sound Pop until done
    change Your Score by 1
```

- Find these blocks in the script.
- What does each block do?

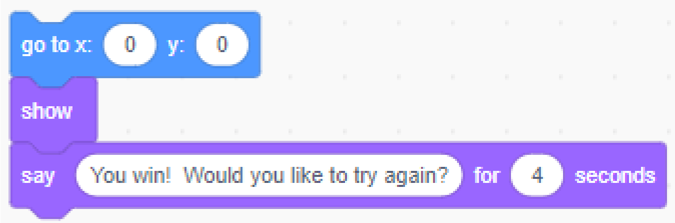
- Find this **conditional** block in the script.



```
repeat until Your Score = 20
```

- Notice how it expands to include the scripts above.
- Why are the scripts above inside this **conditional** block?
- Notice **iteration** is used in this block. It repeats until Your Score equals 20.

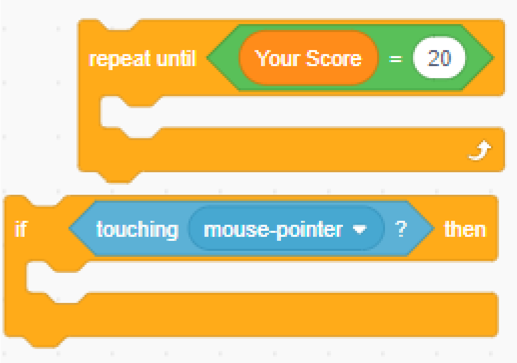
- Continue touching the balloon with the mouse-pointer until Your score equals 20.
- What happens?
- Find these blocks in the script.



```
go to x: 0 y: 0
show
say You win! Would you like to try again? for 4 seconds
```

- What does each block do? Notice these blocks are not included inside the repeat until block.
- Because of the **sequence** of the blocks these three blocks do not start until Your Score equals 20.
- Try moving these blocks inside the repeat until block. What happens? Why?

Pop the Balloon - Bronze Lesson Activity



```
repeat until <(Your Score) = 20>
    
end

if <touching (mouse-pointer ▾) ?> then
    
end
```

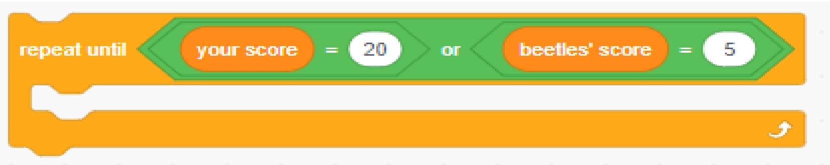
- Open the Bronze Scratch activity.
- Click the green flag and talk about what this activity does.
- Write the two **conditionals** this script has on a piece of paper.
- Talk about what each does.
- Open and **Remix** the Pop the Balloon Bronze Copy Scratch activity found at: https://scratch.mit.edu/projects/327203983
- Notice the blocks are in the script area, but they are not in the correct **sequence**.
- Encourage the learner place the blocks in the correct **sequence**.
- Use the words, "**sequence**, **iteration, conditional**, and **variable**" as you talk about the activity.
- Allow time for the learner to share their activity with others.

Pop the Balloon - Silver Lesson

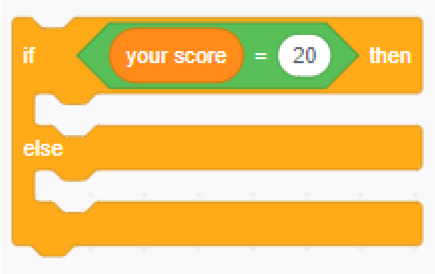
Understand How this Works

Whenever you create new sprites, blocks, broadcasts, variables, or lists, it is helpful to use meaningful names.
Also, think about the titles you give to your Scratch activities.

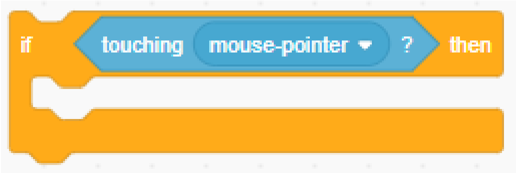
- Open the Silver Scratch activity.
- Notice this activity has two sprites.
- Click the green flag.
- What happens if the mouse-pointer touches the balloon? What happens if it touches the beetle?
- Look at the script for the balloon sprite. Compare its scripts to its scripts in the Bronze Scratch activity. How are they the same? How are they different?
- Look at the script for the beetle.
- Notice the beetle has three **conditional** blocks in its script. What does each do?



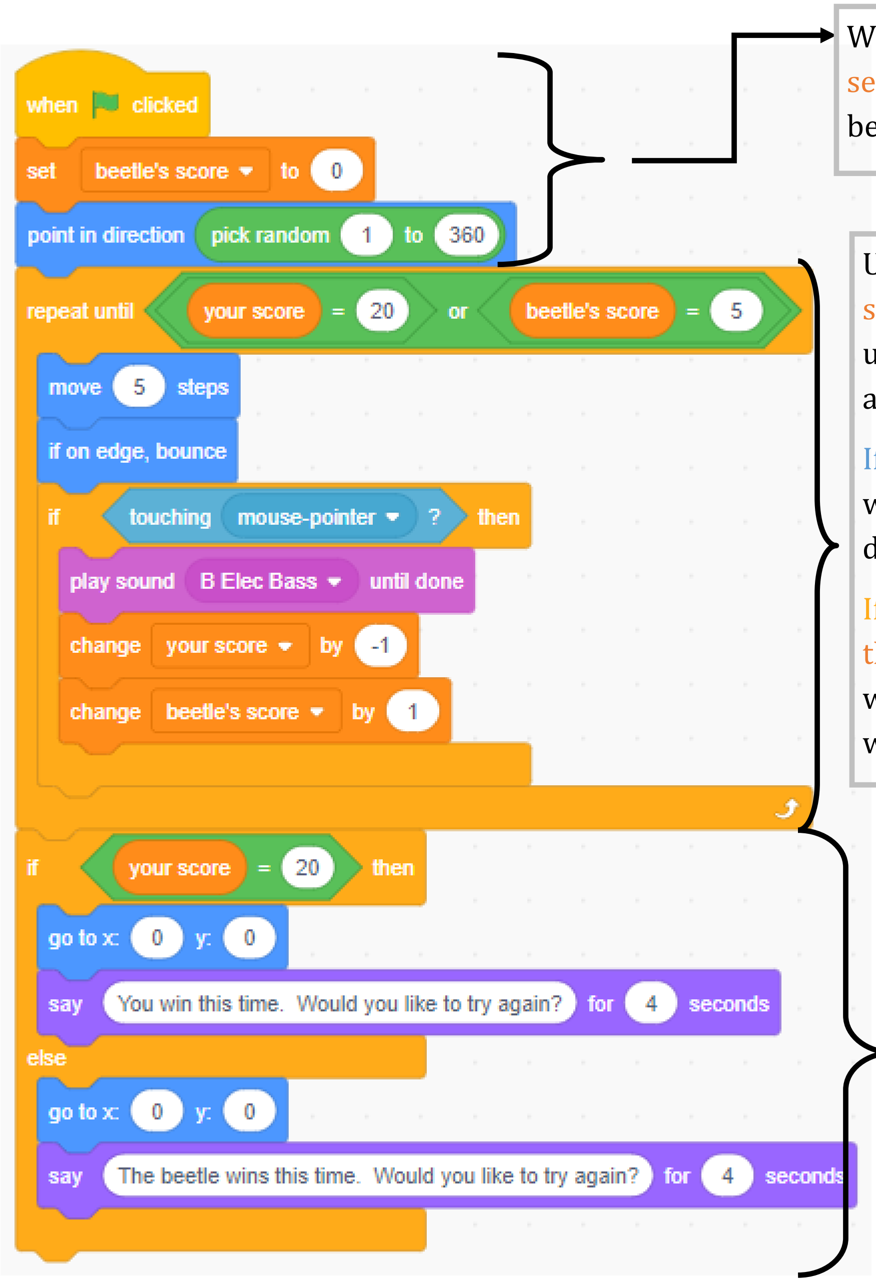
```
repeat until <<(your score) = 20> or <(beetles' score) = 5>>
    
end
```



```
if <(your score) = 20> then
    
else
    
end
```



```
if <touching (mouse-pointer ▾) ?> then
    
end
```



```
when ⚑ clicked
set beetle's score ▾ to 0
point in direction pick random 1 to 360
repeat until your score = 20 or beetle's score = 5
    move 5 steps
    if on edge, bounce
    if touching mouse-pointer ▾ ? then
        play sound B Elec Bass ▾ until done
        change your score ▾ by -1
        change beetle's score ▾ by 1
if your score = 20 then
    go to x: 0 y: 0
    say You win this time.  Would you like to try again? for 4 seconds
else
    go to x: 0 y: 0
    say The beetle wins this time.  Would you like to try again? for 4 seconds
```

When the green flag is clicked, these blocks set the beetle's score to 0 and points the beetle in a random direction .

Until your score equals 20 **or** the beetle's score equals 5, this repeat until block uses **iteration** to make the beetle move around the stage with certain **conditions**:

If the beetle is on the edge of the stage, it will bounce back and go a different direction.

If the mouse-pointer touches the beetle, then a Bass sound will play, your score will change by –1, and the beetle's score will change by 1.

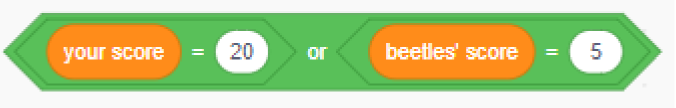
When you score equals 20 **or** the beetle's score equals 5 this If __ then else __ **conditional** controls the beetle.

If your score equals 20, the beetle goes to the center of the stage and says, "You win this time. Would you like to try again?"

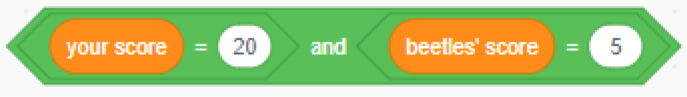
Else, if you score does not equal 20, then the beetle must have 5 points. It goes to the center of the stage and says, "The beetle wins this time. Would you like to try again?"

If__, then, else **conditional** blocks are very helpful. In this activity there are only 2 possible outcomes to end the game. Either you score 20 points, **or else** the beetle scores 5 points.

- Why is this __ or__ block from the OPERATORS category used in the script instead of this __ and__ block?



```
your score = 20 or beetles' score = 5
```



```
your score = 20 and beetles' score = 5
```

Pop the Balloon - Silver Lesson Activity

- Open the Silver Scratch activity.
- Click the green flag and help the learner play the game.
- Notice what the beetle sprite does as you play the game. As you play the game, help the learner write the answers to the questions about the beetle in the boxes below.

What does the beetle do until your score equals 20 **or** the beetle's score equals 5?

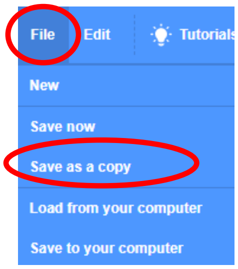
If the mouse-pointer touches the beetle, then what does the beetle do?

If your score equals 20, then what does the beetle do?

Else what does the beetle do?

- Open and make a copy the learner's **remixed** **Bronze** activity that was saved in the **Bronze Copy** lesson.
- Get a beetle sprite from the sprite library.
- Create a new variable called beetle's score.
- Ask the learner what the beetle does when the green flag is clicked. (Find blocks to make the beetle set the variables to 0 and point in a random direction.)
- Help the learner use the three lists above to create the **conditional** blocks needed for the beetle's script. Be sure the learner understands the if __ then, else block and the **sequence** needed in the scripts.
- Notice the beetle, not the balloon, ends the game in this activity. Make changes needed to the balloon's script.
- Use the words, "**sequence**, **iteration, conditional**, and **variable**" as you talk about the activity
- Allow time for the learner to share their game with others.

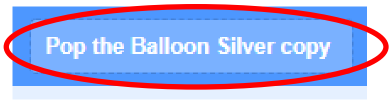
If you want to make changes to one of your Scratch activities, and you want to keep your original activity, you can create a copy of the activity.



1. Click "File".
2. Click "Save as a copy".

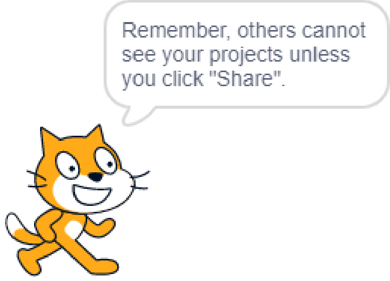
You can change the name of any Scratch activity.

1. Click the title of the activity found above the script area.
2. Delete the old name and type a new name.



Pop the Balloon - Gold Lesson

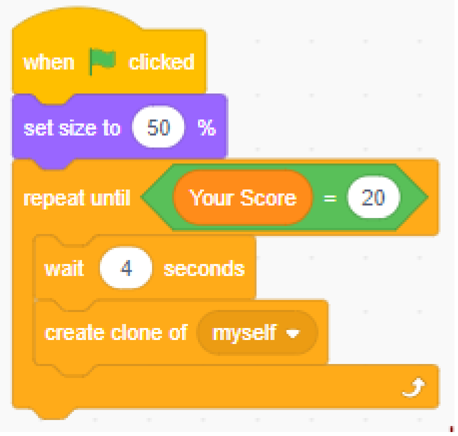
Understand How This Works


Remember, others cannot see your projects unless you click "Share".

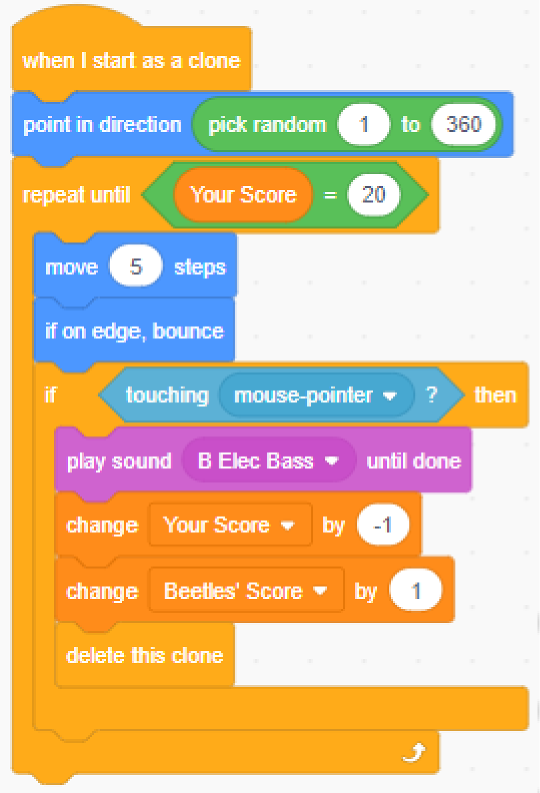
- Open the Gold Scratch activity.
- Click the green flag. Play the game.
- Notice every 4 seconds the beetle makes a clone of itself.

A clone is a copy of a sprite. It is not exactly a sprite. It does not appear below the stage like a sprite. To find out which of the beetles shown on the stage is the original sprite, click the beetle sprite below the stage. This will cause a blue box to appear around the original beetle. All clones use the same script, but by using randomness, you can make them do different things.

- Look at the scripts for the beetle.
- Notice it has two scripts that start when the green flag is clicked. One is almost the same script used in the Silver activity. What is the difference?
- The other script that starts with the green flag is shown here.
- What does it do?



```
when 🏴 clicked
set size to (50) %
repeat until <(Your Score) = (20)>
    wait (4) seconds
    create clone of (myself ▾)
↺
```



```
when I start as a clone
point in direction (pick random (1) to (360))
repeat until <(Your Score) = (20)>
    move (5) steps
    if on edge, bounce
    if <touching (mouse-pointer ▾) ?> then
        play sound (B Elec Bass ▾) until done
        change (Your Score ▾) by (-1)
        change (Beetles' Score ▾) by (1)
        delete this clone
↺
```

- Find this script shown on the left. It controls what each clone does when it starts. It tells the clones what to do when they start.
- Each time a clone starts, it points in a random direction.
- Until Your Score equals 20 the clone moves and if on the edge, it bounces.
- Also until Your Score equals 20, if the mouse-pointer is touching the clone, it will play a sound, change Your Score by –1, change Beetles' Score by 1, and delete this clone.

- Try changing the **sequence** of the script. Does it still work? Why or why not?

Pop the Balloon - Gold Lesson Activity

- Open the Gold Scratch activity.
- Play the game with the learner.
- Notice what the beetle does. Talk about what a clone is and how you can identify the original sprite on the stage.
- As the learner plays the game, ask the learner to answer the questions about the clones found in the boxes below.

Until Your Score equals 20 **or** the Beetles' score equals 15, how often does the beetle create a clone of itself?

What does each clone do when it starts as a clone?

If it touches the mouse-pointer, what does each clone do?

- Open and make a copy the learner's **remixed** Silver activity that was saved in the Silver lesson.
- Help the learner use the boxes above to create scripts for the beetle and its clones.
- Help the learner make changes needed to the balloon's script.
- Allow time for the learner to share their activity.

Self-Evaluate

Remember to self-evaluate. Review the questions on page 5.

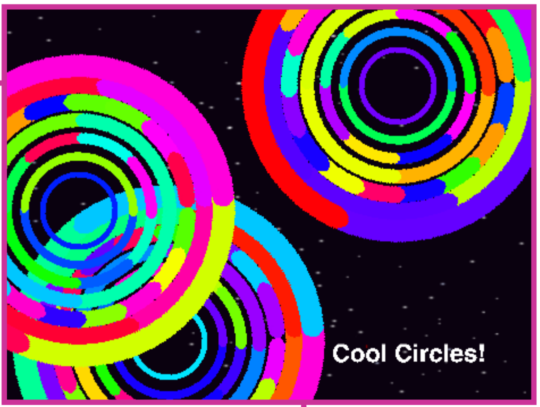
5. Concentric Circles

Summary of Activity: Modularization and **iteration** are used to create colorful, concentric circles. In the Bronze and Silver activity you complete scripts to design circles of your choice. The Gold activity forever creates your design of seven different size concentric circles that are each divided into fractional parts.

Bronze: https://scratch.mit.edu/projects/328060693

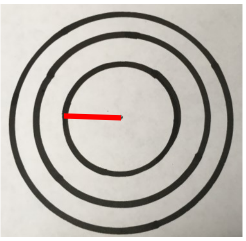
Silver: You and the learner will create an original Silver activity.

Gold: https://scratch.mit.edu/projects/328152154

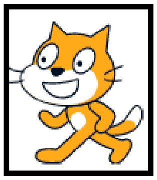


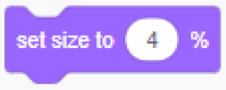
Concentric Circles

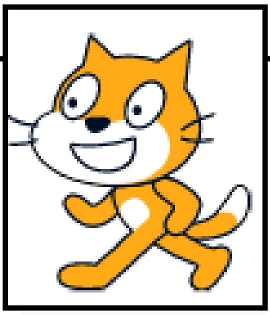
Getting Started - Unplugged Activity

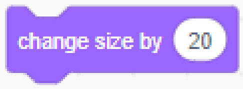


- Concentric circles are different size circles with the same center.
- Get a compass and use it to draw concentric circles like the ones shown here.
- The point of the compass is located at the center of the circle. The pencil or marker of the compass draws the circle.
- A line segment that starts at the center of the circle and goes to a point on a circle is called a radius.
- A radius is shown with a red segment in the small circle above.
- When you open the compass wider, the radius increases and the circle become larger.



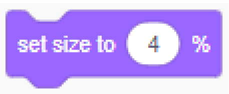

set size to 4 %

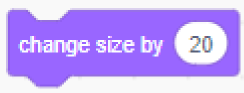



change size by 20

Make a Connection

- In Scratch, you can create a sprite using the paint tools. If you create the sprite in a certain way, you can use it to draw concentric circles. (This will be shown in the Bronze Scratch activity.)
- If drawn correctly, a sprite can rotate around its center just like a compass can rotate around a point.
- If you set the size of the sprite to 4%, it can be used to draw a small circle around its center of rotation. If you change the size of the sprite by 20, it can be used to draw a larger circle around its center of rotation.
- These blocks are found in the LOOKS category

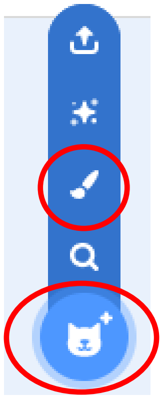

set size to 4 %


change size by 20

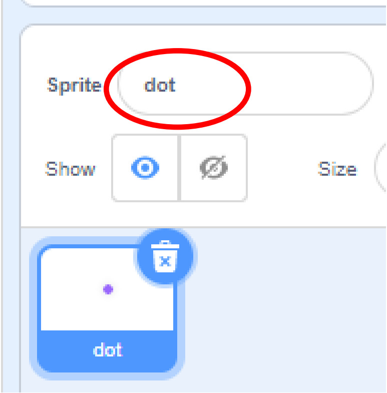
Ask:

1. How is this similar to the circles you drew in *Getting Started*?
2. How would you use a compass to draw colorful, concentric circles like the ones shown above?

Understand How This Works - Bronze Lesson

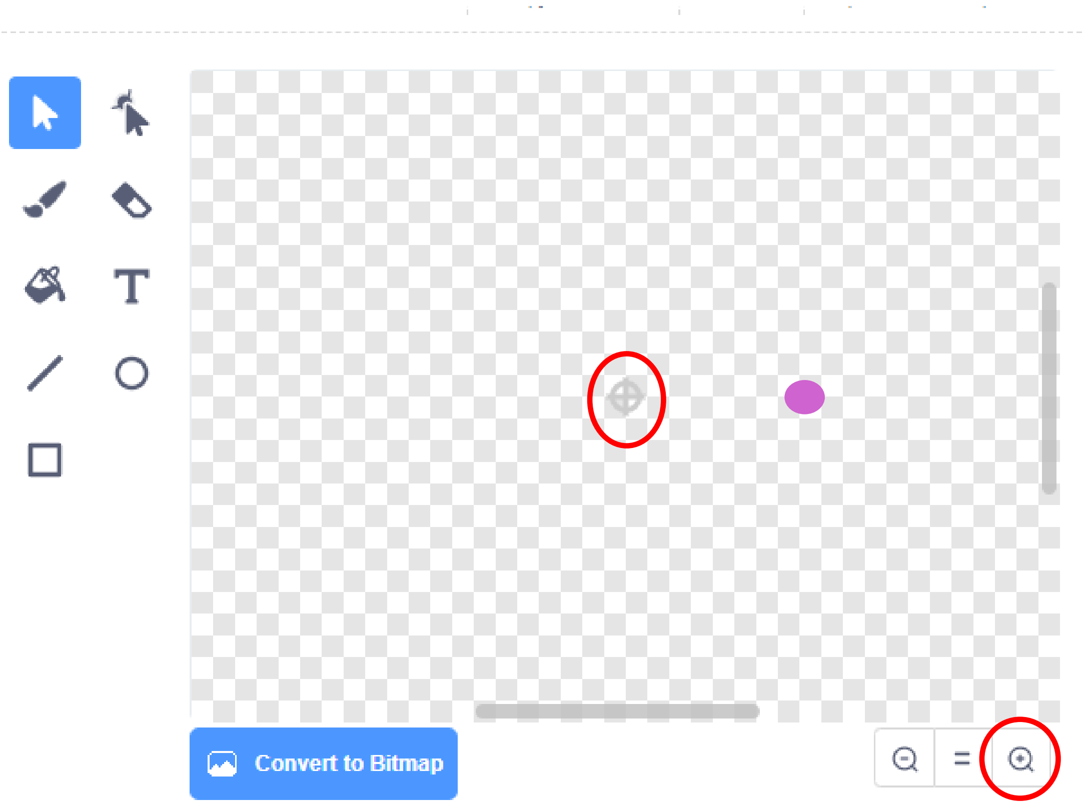


- Open a new Scratch activity. (You will use this new activity again in the Silver lesson.)
- Delete the cat sprite.
- To draw a new sprite, click the cat head in the sprite area. Then click the paint brush. This will open the costume tab shown far below. You will use those paint tools to create a new sprite.
- Use the + tool on the bottom right side of the screen to enlarge the area .
- Notice a circle with a cross in its center. This circle is the center of rotation for any sprite you draw.
- Use the paint brush to make a small dot away from the circle and cross. It is shown in purple below, but you can use any color except black to make your dot. (If you use black, you will not be able to change the color of your sprite.)

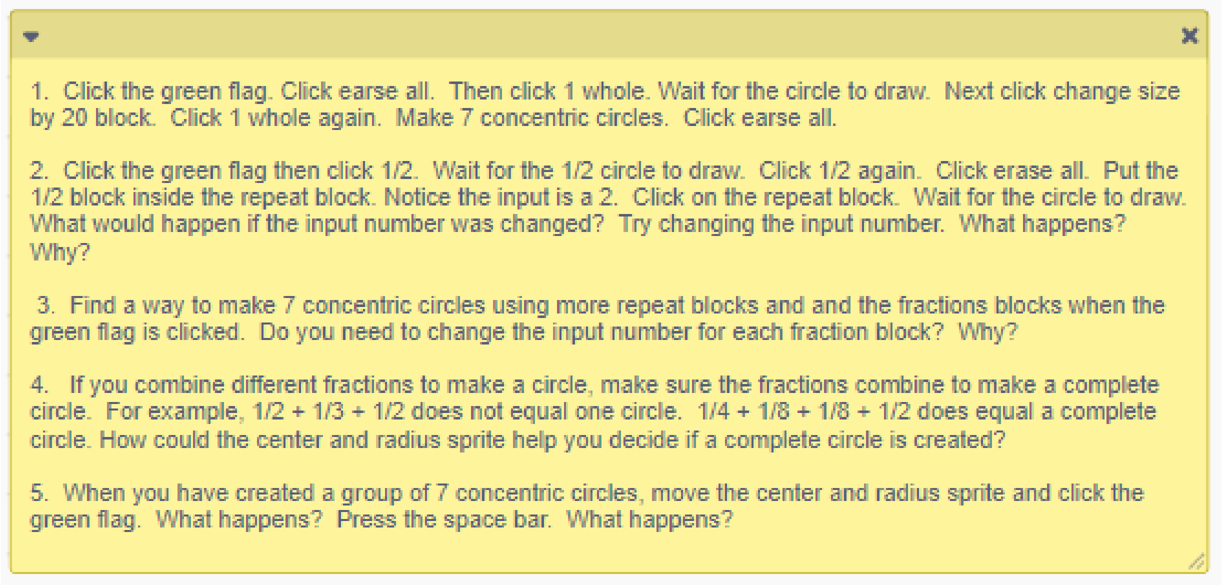


Sprite dot

- Look under the stage. Notice there is a new sprite under the stage. Change the name of the sprite by changing the name in the information box for the sprite below the stage. This new sprite is now named *dot.*
- If you increase the size of this sprite, the distance between the center of rotation and the *dot* sprite will increase. This is similar to increasing the radius of a circle.



- Open the Bronze Scratch activity.
- Notice this comment box is open in the script area. This comment box has directions for the user. A comment box is a good way to communicate to any user of your activity.
- Follow the directions given in this comment box.



1. Click the green flag. Click earse all. Then click 1 whole. Wait for the circle to draw. Next click change size by 20 block. Click 1 whole again. Make 7 concentric circles. Click earse all.

2. Click the green flag then click 1/2. Wait for the 1/2 circle to draw. Click 1/2 again. Click erase all. Put the 1/2 block inside the repeat block. Notice the input is a 2. Click on the repeat block. Wait for the circle to draw. What would happen if the input number was changed? Try changing the input number. What happens? Why?

3. Find a way to make 7 concentric circles using more repeat blocks and and the fractions blocks when the green flag is clicked. Do you need to change the input number for each fraction block? Why?

4. If you combine different fractions to make a circle, make sure the fractions combine to make a complete circle. For example, 1/2 + 1/3 + 1/2 does not equal one circle. 1/4 + 1/8 + 1/8 + 1/2 does equal a complete circle. How could the center and radius sprite help you decide if a complete circle is created?

5. When you have created a group of 7 concentric circles, move the center and radius sprite and click the green flag. What happens? Press the space bar. What happens?

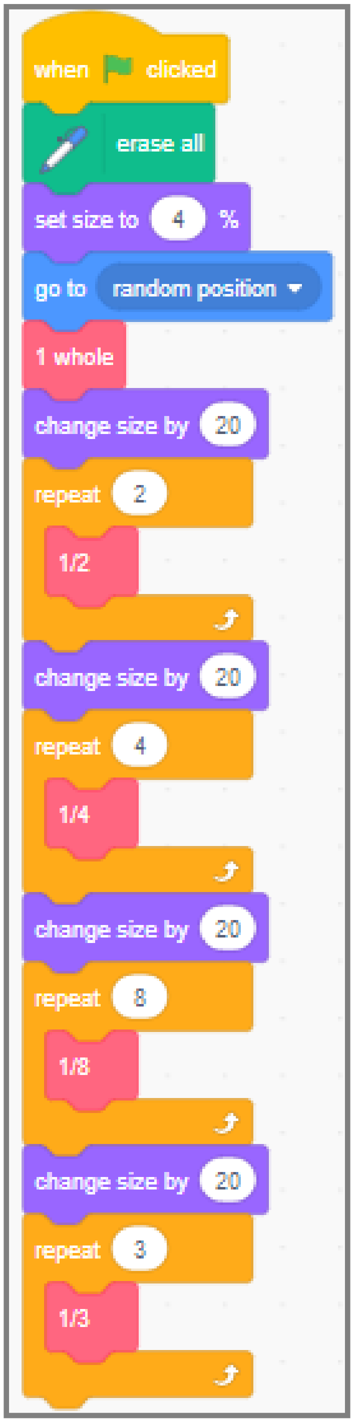
- What happened? What did you learn?
- This script for this Bronze activity uses **modularization**. (How these blocks are defined will be explained more clearly in the Silver Scratch activity.)

Concentric Circles - Bronze Lesson Activity

- Open a new Scratch activity. (The learner will use this new activity again in the Silver lesson.)
- Help the learner create a new *dot* sprite.
- Discuss this sprite's center of rotation and how to enlarge the sprite.
- Open and **Remix** the Bronze Scratch activity.
- It is okay to teach something to someone else by merely giving them directions to follow. This allows the learner to take responsibility for their own learning. If the learner is old enough to read and follow written directions, let them read and complete the comments given in the Bronze Scratch activity on their own. If you feel they need more help, read and discuss the directions with them.
- Allow time for the learner to share their work with others.

Concentric Circles - Silver Lesson

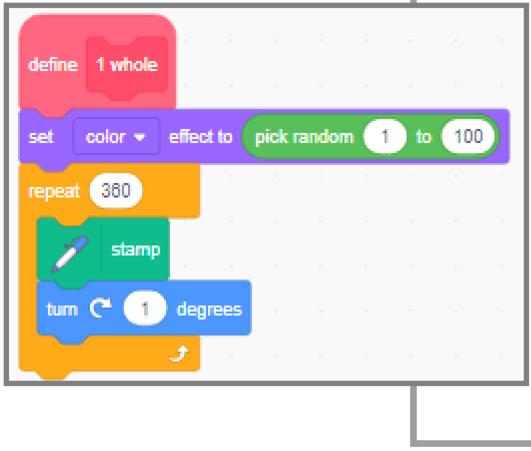
Understand How this Works



```
when [flag] clicked
erase all
set size to (4) %
go to (random position ▾)
1 whole
change size by (20)
repeat (2)
    1/2
end
change size by (20)
repeat (4)
    1/4
end
change size by (20)
repeat (8)
    1/8
end
change size by (20)
repeat (3)
    1/3
end
```

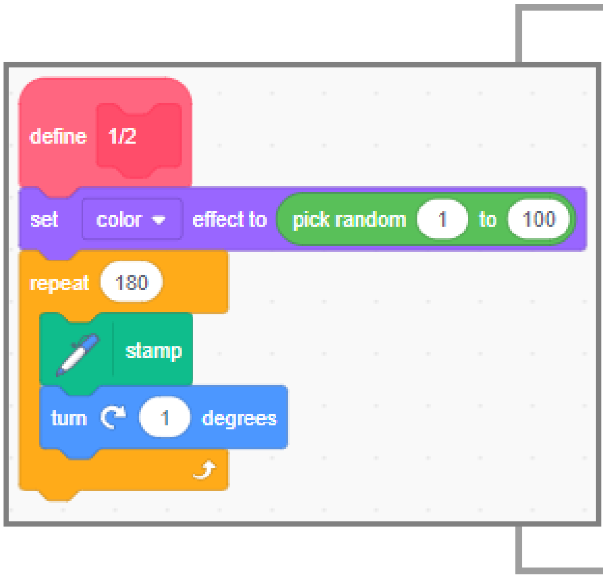
- Open the Scratch activity you created in the Bronze lesson in which you created a new sprite called *dot*.
- Look at the script to the left. It could be used to create 5 concentric circles.
  - ⇒ The smallest first circle would be a solid color circle.
  - ⇒ The next sized circle would be 1/2 one color, and the other 1/2 would be another color.
  - ⇒ The next sized circle would be four colors each 1/4 of the circle.
  - ⇒ The next sized circle would be 8 colors each 1/8 of the circle.
  - ⇒ The last circle would be 3 different colors each 1/3 of the circle.
- Notice the script uses **modularization**. Each of the pink blocks from the MY BLOCKS category need to be defined.
- What does each block do?
- Create this script. Then we will define the blocks.

- This script can be used to define the 1 whole block.
- First it sets the pen color to a random color.



```
define 1 whole
set (color ▾) effect to (pick random (1) to (100))
repeat (360)
    stamp
    turn ⟳ (1) degrees
end
```

- Inside the repeat block there are two blocks. One stamps a pen trail, and the other rotates the sprite.
- Because the sprite turns around its center of rotation, and *dot* is located away from its center, the pen stamps a circle as it turns one degree at a time.
- This repeats 360 times because there are 360 degrees in a circle.
- Create and try this script.



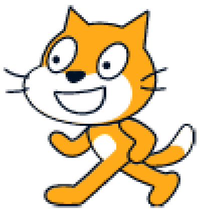
```
define 1/2
set color ▾ effect to (pick random 1 to 100)
repeat 180
    stamp
    turn ⟳ 1 degrees
```

- This script can be used to define the 1/2 block.
- First it sets the pen color to a random color.
- Inside the repeat block there are two blocks. One stamps a pen trail, and the other rotates the sprite.
- Because the sprite turns around its center of rotation, and *dot* is located away from its center, the pen stamps a circle as it turns one degree at a time.
- This repeats 180 times because there are 360 degrees in a circle and this is only drawing 1/2 of the circle.
- Create and try this script.

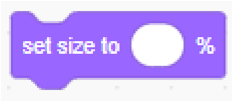
- Look at the scripts used to define 1 whole and 1/2.
- Notice the only difference in the two scripts is the input in the repeat block.
  - ⇒ 1 whole repeats 360 times.
  - ⇒ 1/2 repeats 180 times.
- How many times would it repeat for 1/4? For 1/3? For 1/8?
  - ⇒ Hint: 1 x 360 = 360. 2 x 180 = 360. 4 x 90 = 360. 3 x 120 = 360. 8 x 45 = 360.
  - ⇒ Can you think of other fractional parts of a circle you could use?
- Use this information to create and define more blocks which draw concentric circles.

Concentric Circles - Silver Lesson Activity

- Open the activity the learner created in the Bronze lesson in which a new sprite called *dot* was created.
- Discuss and help the learner use **modularization** to created a script that starts when the green flag is clicked. This scrip creates concentric circles using blocks from the MY BLOCKS category.
- Help the learner create and define scripts for the 1 whole and 1/2 blocks.
- Discuss how these scripts are similar.
- Help the learner create and define more blocks which draw concentric circles.
- Allow time for the learner to share their activity with others.

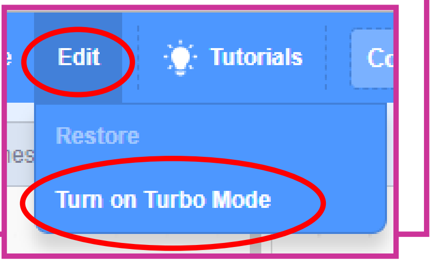


If the first circle is too large or too small, adjust the input number in the set size to (__) block. Try decimal numbers.



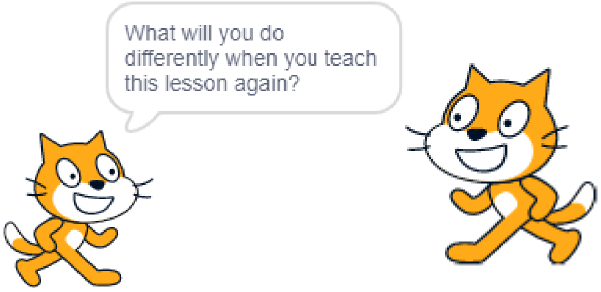
```
set size to ( ) %
```

It takes a long time to draw all the concentric circles in the Gold Scratch activity. You can speed up any Scratch activity by turning on **Turbo Mode**. Click "Edit" in the blue ribbon across the top of the page. A drop-down menu shows. Click "Turn on Turbo Mode". Turn off **Turbo Mode** in the same way. Try using **Turbo Mode** now.



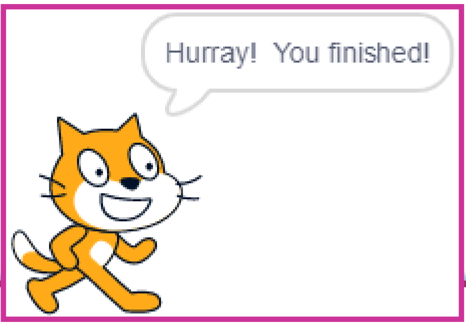
Concentric Circles - Gold Lesson Activity

- Open the Gold Scratch activity.
- Click the green flag. What happens?
- Find the forever block in this script. Discuss how it is used, and what it does in this activity.
- This script sends and receives a broadcast called "cool circles". Find where this happens in the script. What does it do? Which sprite receives the broadcast?
- This activity has a new sprite. Discuss how this sprite was created. (It is hard to see because the words, Cool Circles, use a white pen so that they show on the black backdrop.)
- This activity has a new backdrop. Discuss how this new backdrop was created.
- Make changes to the learner's Silver Scratch activity so that it **forever**:
  - ⇒ Creates a design of seven different size concentric circles that are each divided into fractional parts.
  - ⇒ After the script completes three groups of seven circles in random places on the stage, it should pause and the words "Cool Circles" should appear for a short time.
- Be sure to use **sequence**, **iteration**, **modularization,** and **conditionals** as you help the learner make these changes.
- Allow time for the learner to share the activity with others.


What will you do differently when you teach this lesson again?

Self-Evaluate

Remember to self-evaluate. Review the questions on page 5.



Congratulations! You have finished.

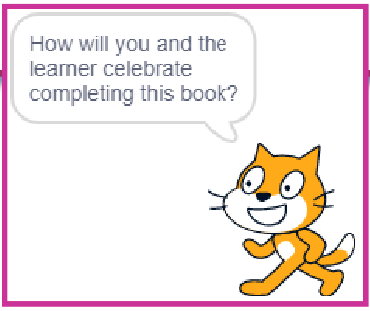
Read this quote from Malala Yousafzai's book:

"One child, one teacher, one book, one pen can change the world."

— **Malala Yousafzai**

<u>I Am Malala: The Story of the Girl Who Stood Up for Education and Was Shot by the Taliban</u>

By helping someone learn to code, you have changed the world.


How will you and the learner celebrate completing this book?

Thanks!

Judy Rocke