LESSON 2

INTRODUCTION TO SCRATCH & COCO

GRADES 5 & 6

CS FOR ALL

NSF
Award #1837380

Welcome back to learning about computer science!

SUMMARY AND STANDARDS

Summary:

In this lesson, students will review patterns and sequencing, and be introduced to algorithms, decomposition and abstraction. They will learn about the Compose and Code (CoCo) platform and use it to compose a set of written instructions ("algorithm") to program a sprite to walk in a square.

ELA Standards:

The student will use effective communication skills in group activities.
a) Listen attentively by making eye contact, facing the speaker, asking questions, and summarizing what is said.
b) Ask and respond to questions from teachers and other group members.
c) Explain what has been learned.
d) Use language appropriate for context.
e) Increase listening and speaking vocabularies.

CS Standards:

The student will construct sets of step-by-step instructions (algorithms), both independently and collaboratively
a) using sequencing;
b) using events.

Optional: Introduce lesson, learning goals, and resources (Slides # 2-4)

 In today's lesson, we will continue to work in Scratch. We will learn new computational thinking skills, and be introduced to an amazing graphic organizer that supports our writing AND our coding.

MATERIALS AND RESOURCES NEEDED FOR THIS LESSON:

- Chromebook/Laptop
- Internet Access
- Scratch Offline Editor (app)
- Teacher slides
- CoCo Link
- Optional: Explanatory text graphic organizer & storyboard

Optional: Introduce lesson, learning goals, and resources (Slides # 2-4)

Does everyone have what they need today?

LESSON OBJECTIVES: I CAN...

❏ Review the definitions of patterns, sequencing, and Scratch vocab

❏ Define algorithms, decomposition and abstraction

❏ Identify the purpose of CoCo

❏ Write an algorithm for how to walk in a square

❏ Identify the Scratch blocks used to program a sprite to walk in a square

Optional: Introduce lesson, learning goals, and resources (Slides # 2-4)

As we go through the lesson, we will be able to....[read slide]

VOCABULARY

★ Algorithm

★ Commands

★ Code

★ Decomposition

★ Abstraction

And we're going to learn all these new words! [read vocab words]

Let's review: match each CT skill to its definition

Pattern	An ordered set of instructions
Sequence	Background in Scratch project
Scratch Block(s)	Something that repeats
Sprite	Commands in Scratch
Backdrop	Characters in Scratch

Before we begin let's review some important vocabulary from our previous lesson. Work together to match the vocabulary term to its definition. We will share the answers next.

Let's review: match each CT skill to its definition

Pattern

Sequence

Scratch Block(s)

Sprite

Backdrop

An ordered set of instructions

Background in Scratch project

Something that repeats

Commands in Scratch

Characters in Scratch

Before we begin let's review some important vocabulary from our previous lesson. Work together to match the vocabulary term to its definition. We will share the answers next.

TODAY'S CAREER IN TECH: MEET TESS

Play video

LESSON OBJECTIVES: I CAN...

✓ Review the definitions of patterns, sequencing, and Scratch vocab

❏ Define algorithms, decomposition and abstraction

❏ Identify the purpose of CoCo

❏ Write an algorithm for how to walk in a square

❏ Identify the Scratch blocks used to program a sprite to walk in a square

Check off the objectives as the lesson proceeds.

HOW DO COMPUTER
SCIENTISTS THINK?

Does anyone remember the word we use to describe how computer scientists think?

COMPUTER SCIENTISTS USE COMPUTATIONAL THINKING SKILLS

1. Patterns
2. Sequencing
3. Algorithms
4. Decomposition
5. Abstraction

We are going to learn how to do **all of these!**

That's right–computational thinking skills! Last time, we talked about patterns and sequencing.

COMPUTER SCIENTISTS USE COMPUTATIONAL THINKING SKILLS

Today our focus is on

1. Patterns
2. Sequencing
3. **Algorithms**
4. **Decomposition**
5. **Abstraction**

Today, we're going to discuss creating algorithms, decomposing, and abstracting.

ALGORITHM: A LIST OF STEPS TO FINISH A TASK

An Algorithm is a list of steps or commands to finish a task.

TURN & TALK

Think about the algorithms in your life.

What are steps you complete every **day, week**, or even **month**?

Are there certain activities you do, for example, every **day** or every **weekend**?

[read slide]

CODING AN ALGORITHM

when ⚑ clicked

say Hi! for 2 secs
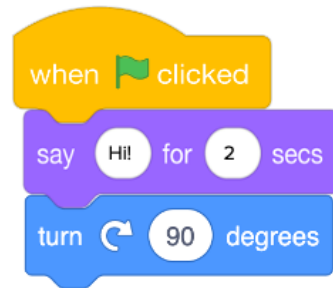
turn ↻ 90 degrees

This algorithm tells the computer to make the sprite say "Hi!" for 2 seconds then turn right 90 degrees.

Here's an example of an algorithm. We've taken Scratch commands and put them in the correct order from top to bottom you have created a sequence of instructions. This would tell the computer to say Hi for 2 seconds then turn right 90 degrees.

COMMANDS:
TELL A PERSON OR COMPUTER WHAT TO DO

Introduce vocabulary slides: Commands

Commands tell a person or a computer what to do. "I command you to give yourself a pat on the back" (pat my own back).

An Algorithm is a list of steps or commands to finish a task. So if I told you to pat yourself on the back, then wiggle your ears, I am putting two commands together in a particular **sequence** I want you to follow. This is a really simple example of an **algorithm**. (Model, pat back, wiggle ears)

CODE:
THE LANGUAGE THAT COMPUTER SCIENTISTS CREATE AND USE TO TELL A COMPUTER WHAT TO DO.

Introduce vocabulary slides: Code
So a command is an instruction and we can create commands for the computer.
When you put commands together you create an algorithm or a list of steps. But how do you give computers instructions?

We use code! Code is the instruction that computer scientist create and use to tell a computer what to do.
Writing code is like writing the commands for a computer. When you know how to write code, you can tell computers what to do. We'll learn how to write code in Scratch.

COMMANDS MUST BE IN THE CORRECT SEQUENCE

So remember: When we write instructions or when we write code, the **order** in which we put the commands is called the **sequence**. It is important that you put your commands in the correct order. Just like when you get ready for school. It is important that you put on your socks BEFORE you put on your shoes. It would not look quite right (and be very uncomfortable) if you put your shoes on THEN put on your socks! The same is true when we give computers commands in code. Putting commands in the correct sequence is VERY IMPORTANT!"

WHY DO PATTERNS AND SEQUENCES MATTER WHEN CODING?
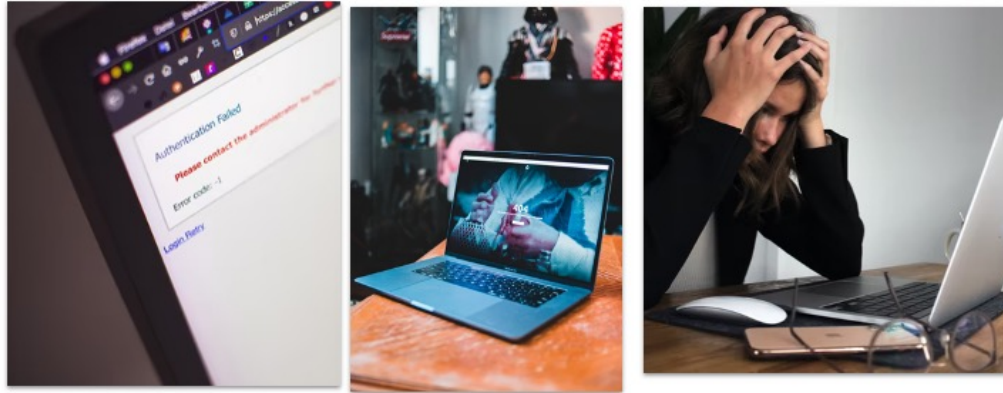
Ask the class to respond: "Why do patterns and sequences matter when coding?"

"Let's think of a cooking example. If you do not follow the correct sequence of commands for making a cake, your recipe will not turn out correct.
The same thing happens when you don't identify patterns and follow the correct sequence when writing a code...things will not turn out like you wanted them to!

When you write a set of instructions or a procedure, you have to be very clear so that others can follow your instructions without getting confused. This is a lot like what computer scientists do: they write instructions for computers. They have to be very clear and careful, because a computer is just a machine and will do exactly what it is told. No matter if you're writing for a human or a computer, the instructions have to be in the right order, or the right **sequence**, or else they would make no sense!

DECOMPOSE

Have you ever heard the word decompose? Or even decomposition? In science you may have learned about decomposers, those organisms that break things down. Well, today we are going to learn about how we can decompose in both computer science and in our writing and literacy!

DECOMPOSE: TO BREAK A PROBLEM DOWN INTO SMALLER PIECES

DECOMPOSITION (DEE-COMP-UH-ZI-SHUN): BREAKING A LARGE PROBLEM INTO SMALLER PARTS

Computer scientists have to solve coding problems and need to make think about what steps they need to take and what order to do them in.

Sometimes the problem is so big or complex, it can be hard to know where to start.

Decomposition is when we break a problem down into smaller parts to make it easier to tackle.

It has many advantages. It helps us manage large projects and makes the process of solving a complex problem less scary and much easier to take on.

This Unit, we're going to learn more about decomposition, our final computational thinking concept. Decomposition is going to be key in helping us to animate our new stories in Scratch with more Scratch blocks."

$$\begin{cases} 2x_1 + x_1 = 7 \\ x_1 + x_2 - 3x_3 = -10 \\ 6x_2 - 2x_3 + x_4 = 7 \\ 2x_3 - 3x_4 = 13 \end{cases}$$

WE CAN DECOMPOSE MORE THAN JUST COMPUTER SCIENCE PROBLEMS!

2020 12 . December

HOW TO BUILD THE PERFECT BURGER

According to code.org: "Decompose—To break a hard problem up into smaller, easier ones" *Decomposition is going to be key in helping us to animate our new stories in Scratch with more Scratch blocks."*

CAN YOU THINK OF MORE EXAMPLES?

PAUSE HERE

When do you break things down?

Break down a recipe, a workout, picking out your outfit, ex

DECOMPOSITION

In literacy, we use decomposition quite often:

- When we sound words out (mis-com-mu-ni-cate)
- When we edit and fix mistakes
- When we create a set of instructions
- When we share the main idea
- When we summarize of a story

We actually use the skill of decomposition quite often in literacy. We decompose when we break down a word or sentence into small parts, like sounding a word we don't know in a sentence. We decompose when we break down our writing to look for errors, like debugging. We decompose and abstract when we break down a series of steps, like a set of instructions, into the most important parts. We also decompose when we break a story down into the main idea or try to provide a summary.

Abstraction

Now that we have reviewed sequencing and coding, let's learn about a new word in computer science. Abstraction! We can use abstraction to "work smarter" when writing code.

ABSTRACTION (AB-STRAC-SHUN) IS IDENTIFYING WHAT IS IMPORTANT AND LEAVING OUT INFORMATION WE DO NOT NEED.

Abstraction sounds like a really big word, but it just means identifying what is important and leaving out information that we do not need. Computer scientists are efficient, which means they look for the simplest solutions to problems.

Title: How to build a sandcastle

First	
Then	
Next	
Last	
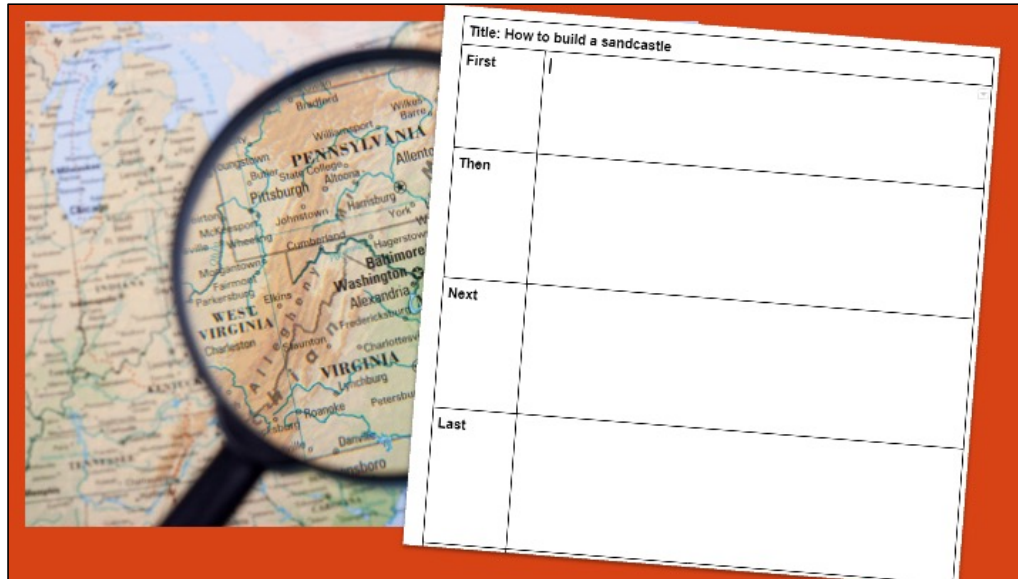
Another common example of abstraction is in our writing when we plan our writing with an outline or a graphic organizer. In order to write clearly, we have to identify the most important details BEFORE we start writing.

Video link: https://www.dropbox.com/s/6nkvpaowwwok268/Abstraction.mp4?dl=0
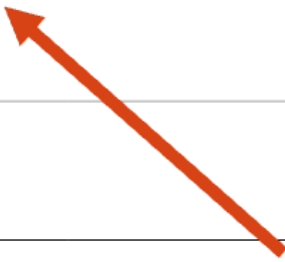
Title: How to build a sandcastle	
First		
Then	
Next	
Last	

Remember that graphic organizers are examples of abstraction. Graphic organizers are templates we often use to help us organize our writing. This helps us to plan our writing more efficiently. We can use these words for a variety of different types of writing and purposes. This helps us be more efficient.

For example, even though we filled in this graphic organizer for "how to build a sandcastle" we could use it to write a set of instructions for…..

Title: How to build a	Lego starship	
First		
Then		
Next		
Last		

How to build something OTHER than a sandcastle. Like a Lego star ship! Nothing else about the graphic organizer needs to change but your instructions would be very different.

My Ideas:	What I Want to Do:		
First First, get all your materials. You will need - list your materials here.	Do I need to add a start block?		No ⦿ Yes
	Do I need to add a character or object?	No ⦿ Yes	
	Do I need to add or change a background?		No ⦿ Yes
	Do I need to make my character talk?	No ⦿ Yes	
Then Then, list the first step in your process.	Do I need to add or change a background?		No ⦿ Yes
	Do I need to make my character talk?	No ⦿ Yes	
	Do I need to make my character think something?		No ⦿ Yes
	Do I need to create a costume?	No ⦿ Yes	
	Do I need to switch a costume?		No ⦿ Yes
Next Next, list the next step in your process.	Do I need to add or change a background?		No ⦿ Yes
	Do I need to make my character talk?	No ⦿ Yes	
	Do I need to make my character think something?		No ⦿ Yes
	Do I need to create a costume?	No ⦿ Yes	
	Do I need to switch a costume?		No ⦿ Yes
Last Last, list the last step in your process.	Do I need to add or change a background?		No ⦿ Yes
	Do I need to make my character talk?	No ⦿ Yes	
	Do I need to make my character think something?		No ⦿ Yes
	Do I need to create a costume?	No ⦿ Yes	
	Do I need to switch a costume?		No ⦿ Yes
Finally Great job!	Do I need to create a costume?		No ⦿ Yes
	Do I need to switch a costume?	No ⦿ Yes	
	Do I need to make my character think something?		No ⦿ Yes

Soon, you will be introduced to another graphic organizer that is also abstraction in action!

LESSON OBJECTIVES: I CAN...

✓ Review the definitions of patterns, sequencing, and Scratch vocab

✓ Define algorithms, decomposition and abstraction

❏ Identify the purpose of CoCo

❏ Write an algorithm for how to walk in a square

❏ Identify the Scratch blocks used to program a sprite to walk in a square

Check off the objectives as the lesson proceeds.

GRAPHIC ORGANIZERS FOR WRITING

Explain that when you are planning to code or to write, there are certain pattern of steps that we may follow each time.
We can use a graphic organizer to help us.

The graphic organizers you have used before helped you brainstorm and eventually write. But now you are going to use a graphic organizer to help take your writing one step further to the final step of the writing process: to present and publish your writing with Scratch! Using Scratch to animate your writing is one way to publish, or share, your work.

The name of our graphic organizer is CoCo, which stands for Compose and Code! Today you are going to learn how to use Coco, a new graphic organizer to organize your thoughts, write instructions, and present your ideas in a coding project.
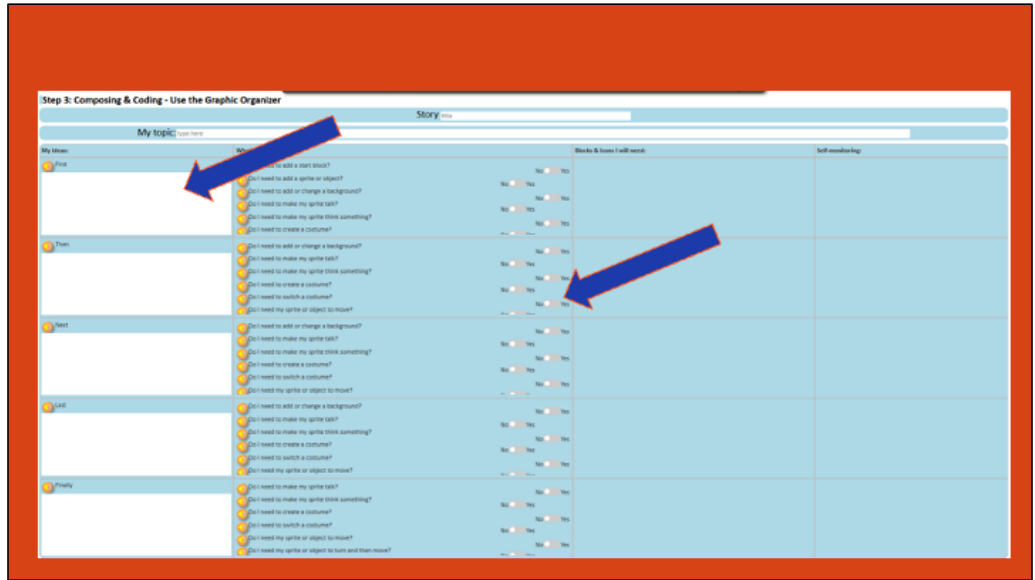
Coco looks a lot like other graphic organizers we've used. Before Coco is filled in with your writing and ideas, it will look like this.

My ideas is where your writing will go. Remember to always include a topic sentence. Including a topic sentence will help orient your reader towards your topic and it will help you set the scene for your Scratch animation.

In the column, "What i want to do" You will be asked a number of questions about what you want to do in scratch based on your writing. You can respond to each question with either "yes" or "no"
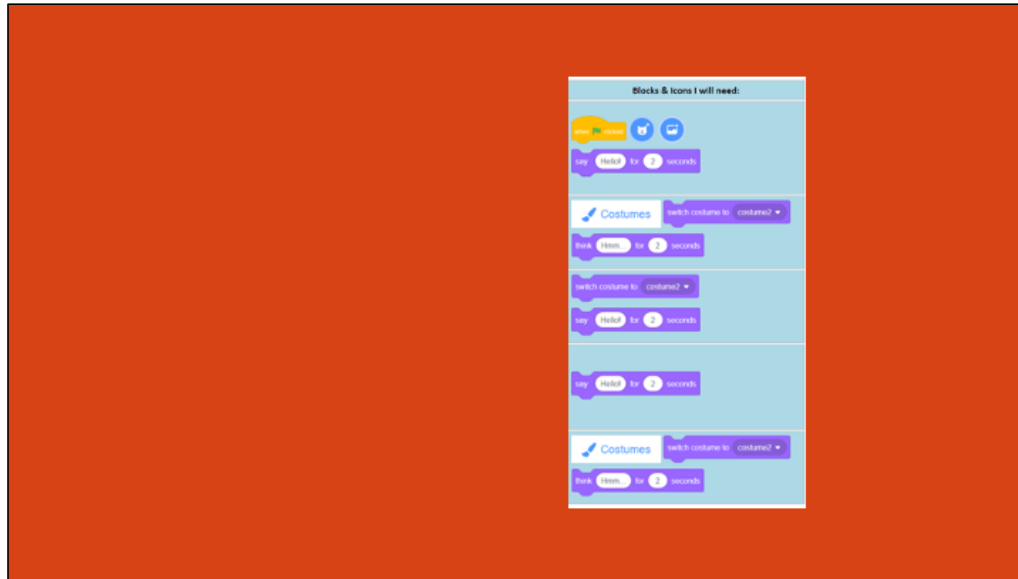
In column 3, called blocks and icons I will need, you will see the blocks you need in scratch based on the decisions you made in column 2.

Self-monitoring:

- [ ] I found the start block.
- [ ] I created a character (sprite).
- [ ] I added a background.
- [ ] I made my character talk.

- [ ] I created a new costume.
- [ ] I changed a costume.
- [ ] I made my character (sprite) think something.

- [ ] I switched a costume.
- [ ] I made my character (sprite) say something.

- [ ] I made my character (sprite) say something.

- [ ] I created a costume.
- [ ] I changed a costume.
- [ ] I made my sprite think something.

In the final column called self-monitoring, you will see a list of statements that you need to check before you move on. These statements will help you check and make sure that you have found everything you need to code in scratch.

My Ideas:	What I Want to Do:	Blocks & Icons I will need:	Self-monitoring:
First First, get all of your materials. You will need a mug, hot chocolate mix, milk, and a spoon.	Do I need to add a start block? No◯Yes Do I need to add a character or object? No◯Yes Do I need to add or change a background? No◯Yes Do I need to make my character talk? No◯Yes	say Hello! for 2 seconds	☐ I found the start block. ☐ I created a character (sprite). ☐ I added a background. ☐ I made my character talk.
Then Then, pour the milk into the mug and warm it in the microwave.	Do I need to create a costume? No◯Yes Do I need to switch a costume? No◯Yes Do I need to make my character think something? No◯Yes	Costumes switch costume to costume2 ▾ think Hmm... for 2 seconds	☐ I created a new costume. ☐ I changed a costume. ☐ I made my character (sprite) think something.
Next Next, pour the hot chocolate mix into the mug.	Do I need to switch a costume? No◯Yes Do I need to make my character talk? No◯Yes	switch costume to costume2 ▾ say Hello! for 2 seconds	☐ I switched a costume. ☐ I made my character (sprite) say something.
Last Last, stir the ingredients with a spoon until everything is well-mixed and enjoy your hot chocolate!	Do I need to make my character talk? No◯Yes	say Hello! for 2 seconds	☐ I made my character (sprite) say something.
Finally Great job!	Do I need to create a costume? No◯Yes Do I need to switch a costume? No◯Yes Do I need to make my character think something? No◯Yes	Costumes switch costume to costume2 ▾ think Hmm... for 2 seconds	☐ I created a costume. ☐ I changed a costume. ☐ I made my sprite think something.

And when you're all done, you will have a filled out graphic organizer that looks something like this.

Let's look at an example to see how we would publish our writing as an animation. This one is called "How to Make Hot Chocolate."
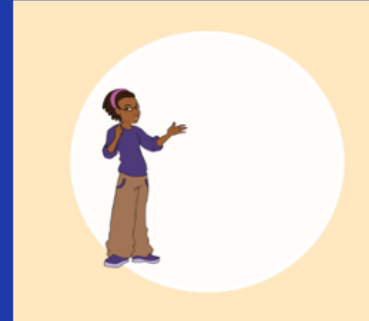
How to make hot chocolate story:

Simple example:
https://www.dropbox.com/s/jlist1zaujjpc8e/Screen%20Recording%202023-09-18%20at%209.21.46%20PM.mov?dl=0

More complex example:
https://www.dropbox.com/s/nh93dmdkphd4n0z/Screen%20Recording%202023-09-19%20at%201.45.13%20AM.mov?dl=0

Here is an example of a recipe written using CoCo and animated in Scratch! Two examples, both are great. One is simple: https://www.dropbox.com/s/jlist1zaujjpc8e/Screen%20Recording%202023-09-18%20at%209.21.46%20PM.mov?dl=0 and one is complex: https://www.dropbox.com/s/nh93dmdkphd4n0z/Screen%20Recording%202023-09-19%20at%201.45.13%20AM.mov?dl=0

LESSON OBJECTIVES: I CAN...

✓ Review the definitions of patterns, sequencing, and Scratch vocab

✓ Define algorithms, decomposition and abstraction

✓ Identify the purpose of CoCo

❏ Write an algorithm for how to walk in a square

❏ Identify the Scratch blocks used to program a sprite to walk in a square

Check off the objectives as the lesson proceeds.

WRITING (& CODING) ALGORITHMS

Next, we're going to use CoCo to write and code algorithms!

WE CAN WRITE ALGORITHMS IN CODE USING SCRATCH

Remember Scratch? Re-introduce Scratch - Explain that Scratch is a program that you can use to code and create interactive stories, games, and animations.

INTRODUCING MOVE, TURN, & WAIT BLOCKS

Today, we're going to learn about some new blocks in Scratch!
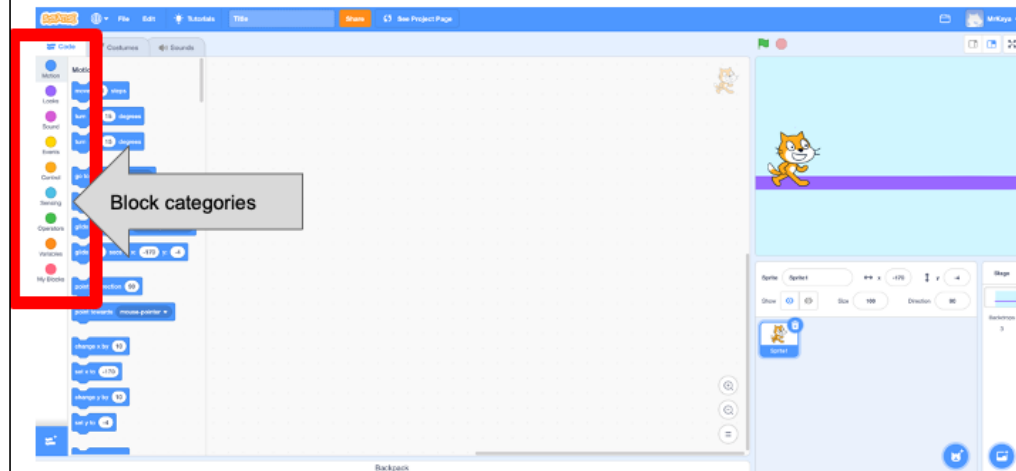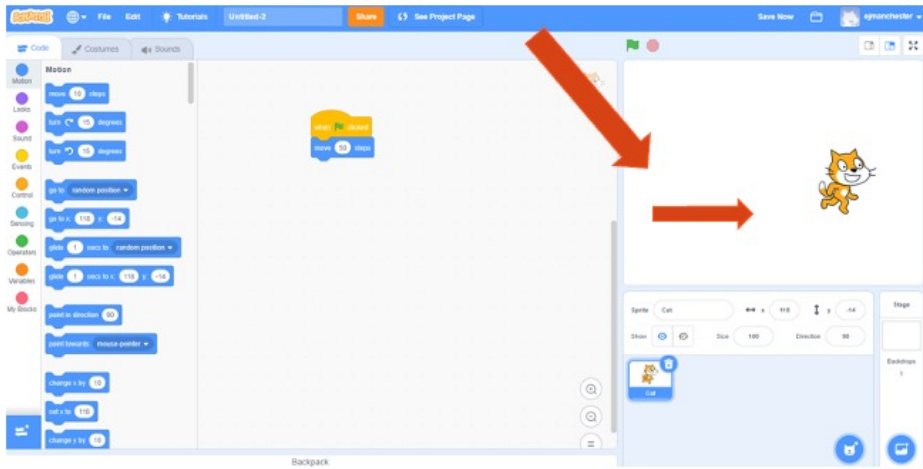
Remember that on the far left hand side of our page, we see a color palette. Each color corresponds to a different type of block. Today, we'll focus on motion blocks, which are BLUE, and control blocks, which are YELLOW.
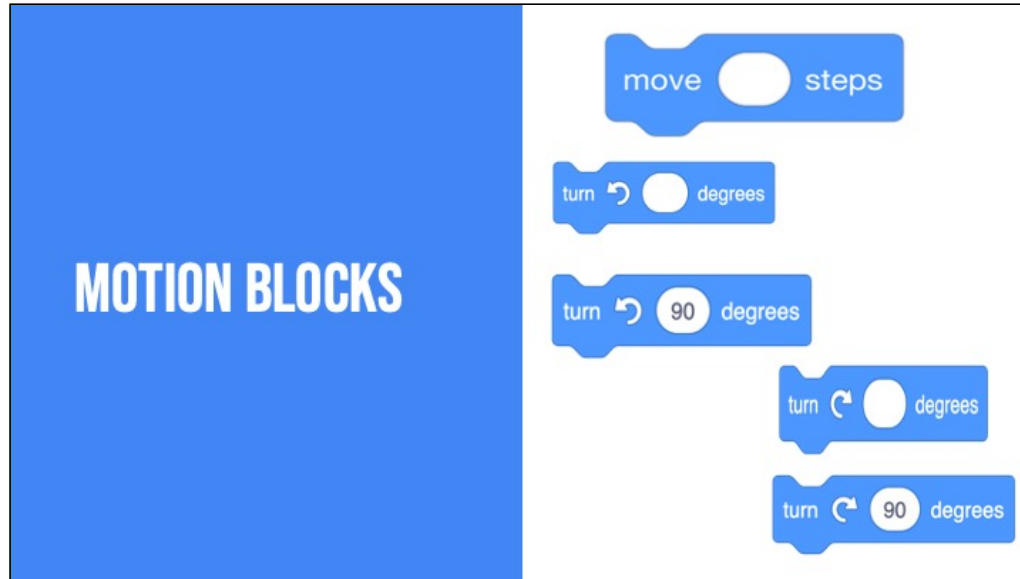
There are lots of motion blocks. We're going to learn about two of them today. One you've already seen in lesson 1 and one is new.
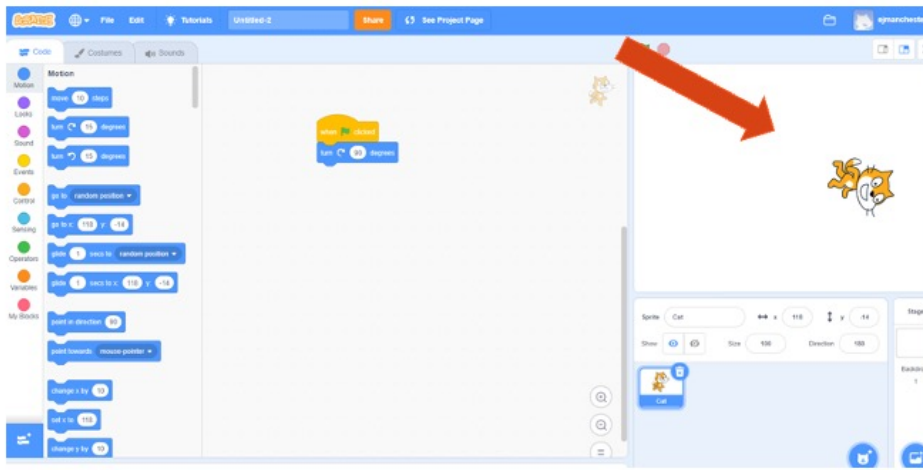
Moves your sprite a certain number of steps.

It may not be obvious but this cat looks different from the one automatically in scratch.
This one looks like it's running.

TURN BLOCK (EXPLAINER VIDEO)

Turns your sprite right or left by a matter of degrees.

It may not be obvious but this cat looks different from the one automatically in scratch.
This one looks like it's running.

The yellow blocks are our control blocks. You already know one of them: the start block. Today you will learn about the wait block.

The wait block simply codes the sprite to wait a set number of seconds before executing the next command.

WAIT BLOCK (EXPLAINER VIDEO)

forever
wait 60 seconds
next costume

repeat 10
wait 0.05 seconds
next costume
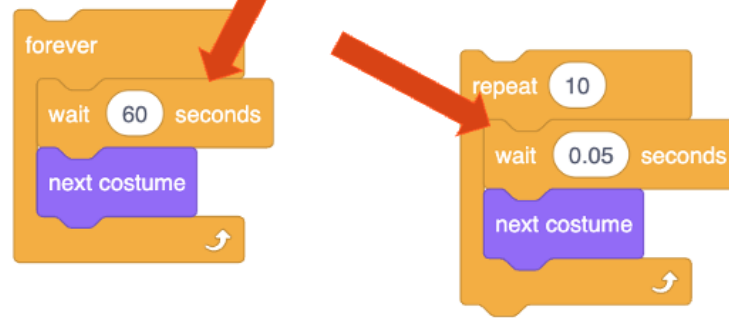
Your Sprite will PAUSE or wait for a specified amount of time.

Play video and explain:

NEW WAYS TO CHANGE COSTUME

OTHER WAYS OF CHANGING COSTUME

➡ **Upload:** a file saved on your computer

➡ **"Surprise"** : the computer chooses a random costume for you

➡ **Paint:** draw your own sprite!

Scratch comes with a preset sprite and backdrop. However, it is also possible to upload your OWN costume by using a photo you already have saved on your computer.

INDEPENDENT PRACTICE

WRITE AN "ALGORITHM" IN COCO

1. Pretend you have a robot.
2. Write a set of instructions (also know as an algorithm) instructing your robot to walk in a square.
3. Use CoCo to guide you:
   a. First
   b. Next
   c. Then
   d. Last
   e. Finally
4. Read aloud your algorithm to a partner. See if they can follow your instructions!
5. Reflect: did they walk in a square? If not, where did your algorithm go wrong?

Let's practice using CoCo and Scratch with an activity. [read instructions on slide]

- Option to use the explanatory text graphic organizer template as additional scaffolding for the initial writing prompt.

OPTIONAL EXTENSION ACTIVITY- ONLY USE IF ADDING EXTENSION

➢ Use CoCo columns 2-3 to begin planning your Scratch animation!

➢ Code your sprite to walk in a square in Scratch

Option for early finishers: have them begin planning an animation in Scratch using CoCo and THEN move into Scratch
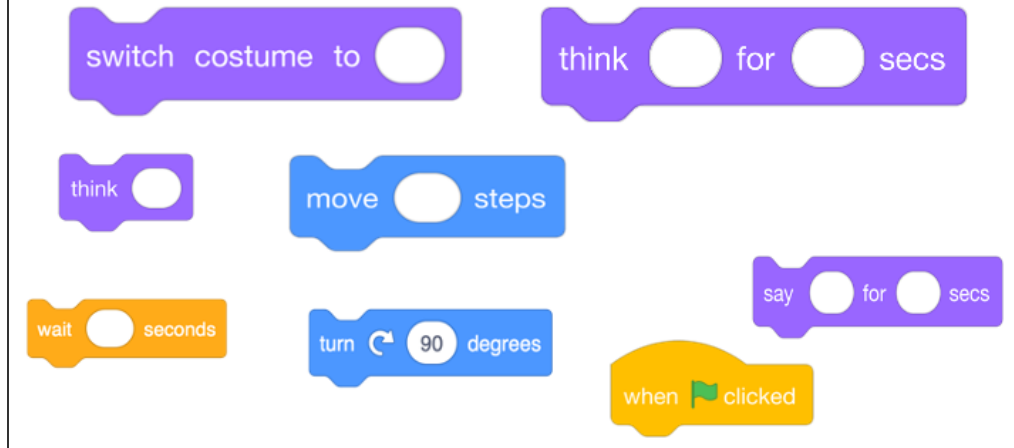
Optional support videos

LESSON OBJECTIVES: I CAN...

✓ Review the definitions of patterns, sequencing, and Scratch vocab

✓ Define algorithms, decomposition and abstraction

✓ Identify the purpose of CoCo

✓ Write an algorithm for how to walk in a square

❑ Identify the Scratch blocks used to program a sprite to walk in a square

Check off the objectives as the lesson proceeds.

HOW WOULD OUR INSTRUCTIONS LOOK IN CODE?

Okay, did the "algorithms" we wrote using words make sense? How would they look in code?

COMMANDS FOR OUR COMPUTER (ROBOT)

switch costume to ◯

think ◯ for ◯ secs

think ◯

move ◯ steps

say ◯ for ◯ secs

wait ◯ seconds

turn ↻ 90 degrees

when ⚑ clicked

I want to use my scratch blocks to instruct our robot to walk in a square. But wait! I can't find a block with the command that says *walk in a square. What do you think I should do? I think I first need to make my robot walk forward. Can you find a block that would* command *the robot walk forward? Hmmm, which of these blocks might be the correct command?* When you think you know, tell your teacher.
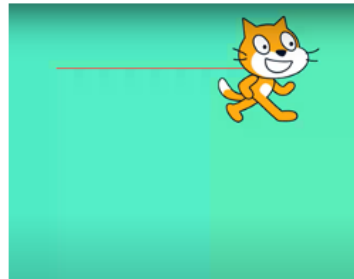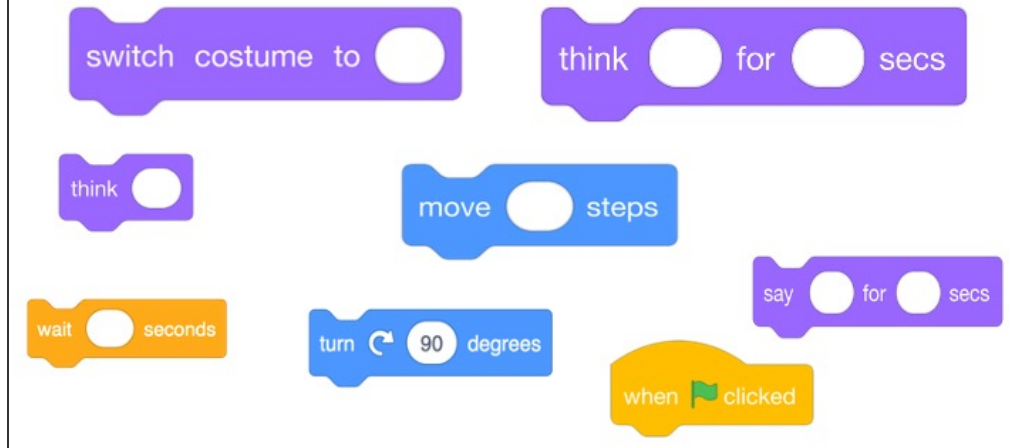
If you choose this block, you would be correct. This block says "Move ___ steps".
I want my robot to move 2 steps, so that would go in this blank. CLICK
This means our robot should move directly forward 2 steps. If you are the robot, move forward 2 steps.

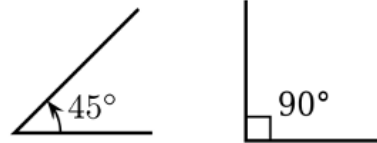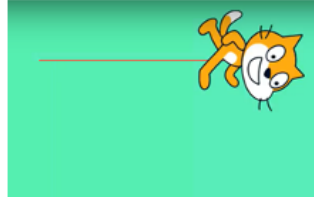## COMMANDS FOR OUR COMPUTER (ROBOT)

switch costume to ( )

think ( ) for ( ) secs

think ( )

move ( ) steps

say ( ) for ( ) secs

wait ( ) seconds

turn ↻ 90 degrees

when 🚩 clicked

Ok, we want our robot to move in a square, What should I do next? Which way does my robot need to go and which command do I need to use to make the robot go that way? Pause for a moment to choose the next command block we need.
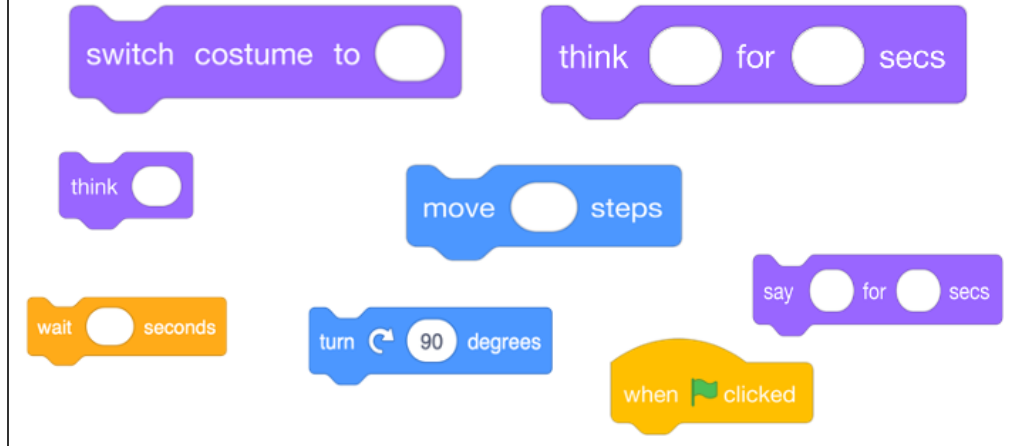
COMMANDS FOR WALKING IN A SQUARE

Step 2 (Then): Turn right

turn ↻ 90 degrees

45°   90°

We need our robot to TURN. This common block commands the robot to turn right 90 degrees, which is like turning one corner of our square. Go ahead robot, turn right 90 degrees or one of our corners.

COMMANDS FOR OUR COMPUTER (ROBOT)

switch costume to ◯

think ◯ for ◯ secs

think ◯

move ◯ steps

say ◯ for ◯ secs

wait ◯ seconds

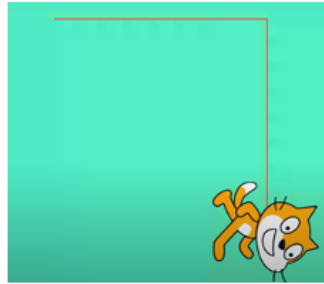turn ↻ 90 degrees

when 🚩 clicked

My robot has turned, but now what does the robot need to do? Take a moment to think and choose. What should we command the robot to do next?

(PAUSE)

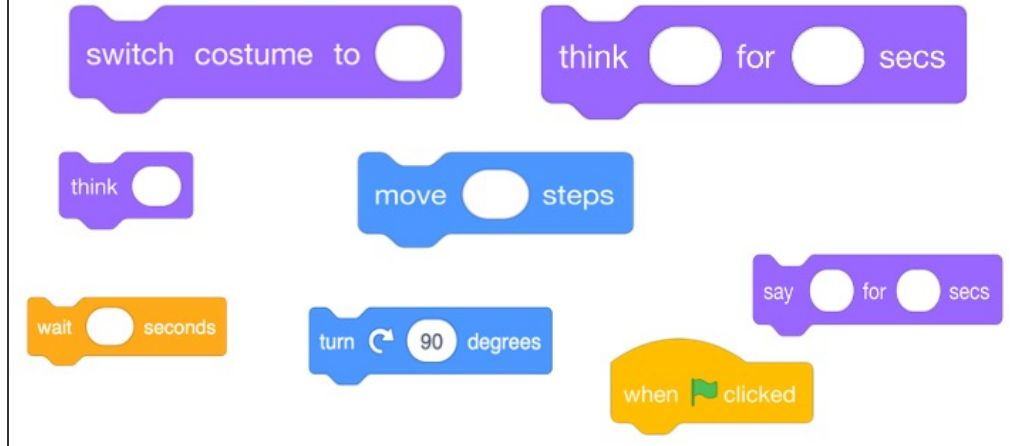*Think about the shape of a square. Are the sides all the same length or different?*

The sides of a square are equal, so we need our robot to walk to same number of steps as it did the first time, 2! So now our robot needs to take two steps. Things should start to get a little easier now.
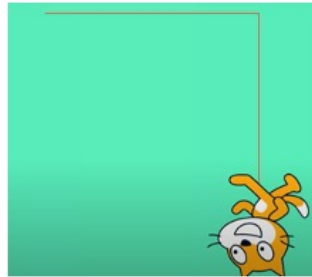
COMMANDS FOR OUR COMPUTER (ROBOT)

switch costume to ◯

think ◯ for ◯ secs

think ◯

move ◯ steps

say ◯ for ◯ secs

wait ◯ seconds

turn ↻ 90 degrees

when ⚑ clicked

What next? We need our robot to complete the square.

COMMANDS FOR WALKING IN A SQUARE

Step 4 (Then): Turn right

turn ↻ 90 degrees

You got it! The robot turns right again 90 degrees! Go ahead robot, turn.

COMMANDS FOR OUR COMPUTER (ROBOT)

switch costume to ◯
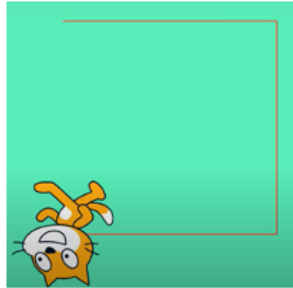
think ◯ for ◯ secs

think ◯

move ◯ steps

say ◯ for ◯ secs

wait ◯ seconds

turn ↻ 90 degrees

when ⚑ clicked

Ok, the robot now needs to….

Move 2 steps! Go ahead robot, move two steps again!

COMMANDS FOR OUR COMPUTER (ROBOT)

switch costume to ( )

think ( ) for ( ) secs

think ( )

move ( ) steps

say ( ) for ( ) secs

wait ( ) seconds

turn ↻ 90 degrees

when 🏳 clicked

Now our robot needs to…..
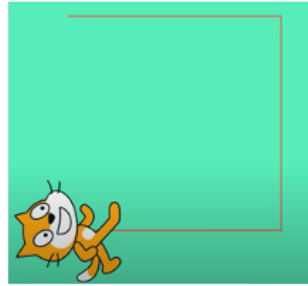
Turn right 90 degrees again. Go ahead robot!

COMMANDS FOR OUR COMPUTER (ROBOT)

switch costume to ◯   think ◯ for ◯ secs

think ◯   move ◯ steps   turn ↺ 90 degrees

wait ◯ seconds   turn ↻ 90 degrees   say ◯ for ◯ secs

when 🚩 clicked
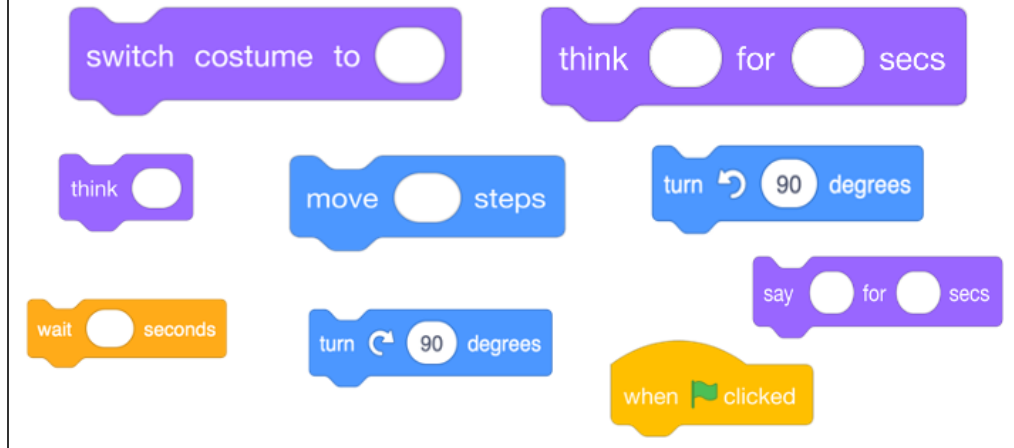
We are almost done with our square, what is last?

COMMANDS FOR WALKING IN A SQUARE

Step 7 (Finally): Walk 2 steps
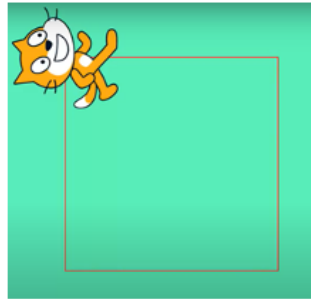
Finally, our last step is to take 2 steps and complete our square. How did your robot do?

COMMANDS FOR WALKING IN A SQUARE

Written commands	Computer commands (code)
Walk forward 2 steps	move **2** steps
Turn right	turn ↻ **90** degrees
Walk 2 steps	move **2** steps
Turn right	turn ↻ **90** degrees
Walk 2 steps	move **2** steps
Turn right	turn ↻ **90** degrees
Walk 2 steps	move **2** steps

Let's review the commands we gave the "robot" using code. We told our robot to walk forward 2 steps, turn right, forward 2 steps, turn right, forward 2 steps, turn right...until we completed our square. We had to put our commands in the correct sequence or our robot would not create a square! Think if we had put them out of order, what would our robot had done?

When you look at the code you created, you can see how these blocks would fit together. This is your algorithm, or your list of commands!
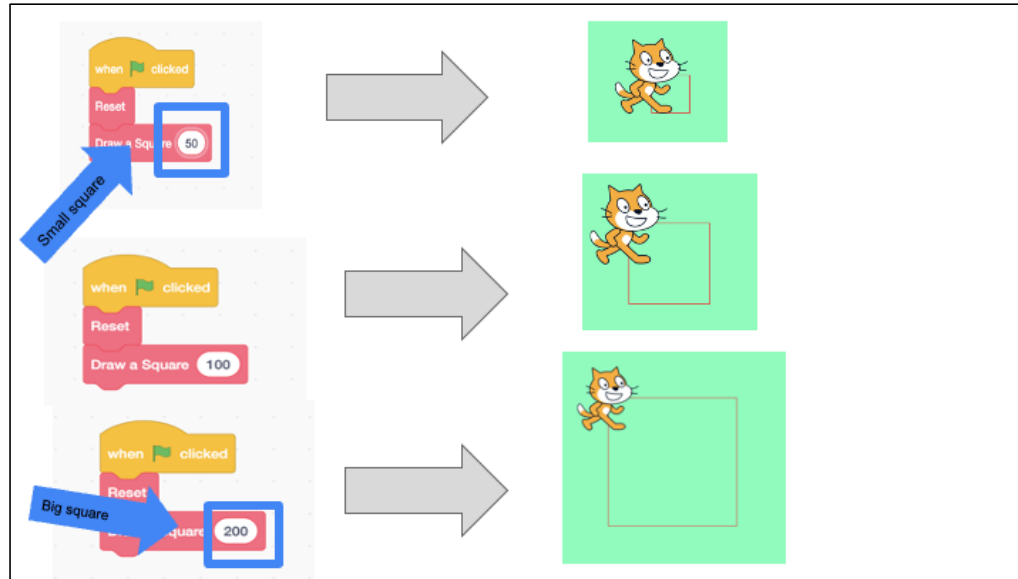
We could've used abstraction here!

In computer science, we use abstraction when we create code that we can reuse and when that code is written in the most efficient way possible. Remember when we tried to draw a square? We had your classmate take two steps each time. Well, we can use abstraction to reuse that pattern efficiently. We can just change the number of steps and create different size squares.
The same is true in Scratch. Do you see how the first the first square says 50, 100, and then 200. This would create different size squares using the same sequence. We only have to change the number of steps and our square can change size!

Scratch link: https://scratch.mit.edu/projects/599143669

LESSON OBJECTIVES: I CAN...

✓ Review the definitions of patterns, sequencing, and Scratch vocab

✓ Define algorithms, decomposition and abstraction

✓ Identify the purpose of CoCo

✓ Write an algorithm for how to walk in a square

✓ Identify the Scratch blocks used to program a sprite to walk in a square

Check off the objectives as the lesson proceeds.

AND REMEMBER....ANYONE IN THE WORLD
CAN BE A COMPUTER SCIENTIST!