Day 1:
An Introduction to Snap!

Infusing Computing Professional Development 2019

Welcome!

Activities for Today's Coding Session

Activity 1: Draw Square

Refresh your Snap*!* knowledge by creating simple shapes

Activity 2: Introduce Yourself

Combine algorithms and events to tell a story in Snap*!*

Goals for today's coding activities!

- **Learn about Pair Programming!**
- **Get comfortable with Snap*!*** and do some coding! Both of today's activities are general and non-discipline specific, though you can take each of the activities and make them specific to the field you teach.
- The first activity will help you **experience the Snap*!* interface**. In fact, this is a typical first lab I teach students when they are learning how Snap*!* and programming works.
- The second activity is **an Interactive Narrative which you can modify to introduce yourself** to your students! This activity could be used in any discipline course.

What is Computational Thinking

Computational Thinking (CT) is a problem solving process.

CT is essential for developing programs, but it can also be used to support problem solving across all disciplines, including the humanities, math, and science.

In this PD we will highlight 4 essential elements of CT:

PAIR PROGRAMMING - Do

Do:

- Be respectful
- Talk to one another about the work
- Explain what you are doing
- Think ahead and make suggestions
- Switch roles often

Don’t:

- Be a bossy navigator
- Grab the driver’s mouse/keyboard
- Be disrespectful to your partner
- Have poor physical hygiene

Remember to Pair Program

During activities, we expect you to switch who is **driving** (coding) and who is **navigating** (reading the instructions) with your partner.

This way, you get a chance to program and have the benefit of working with someone.

Designate your computers: one as the **Driving Machine**, the other as the **Navigator**.

The slides will tell you when to switch.

Now, **choose who will Teacher A (first driver) and Teacher B (first navigator).**

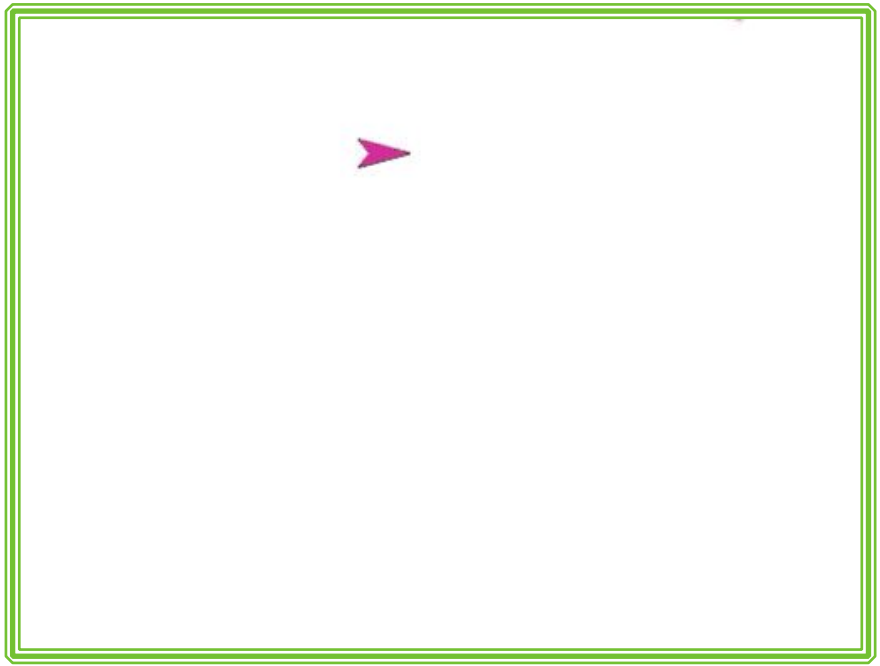
Pair Programming!

Teacher A: Drive!
Teacher B: Navigate!

Activity 1: Draw a Square

Goals & Project Demo

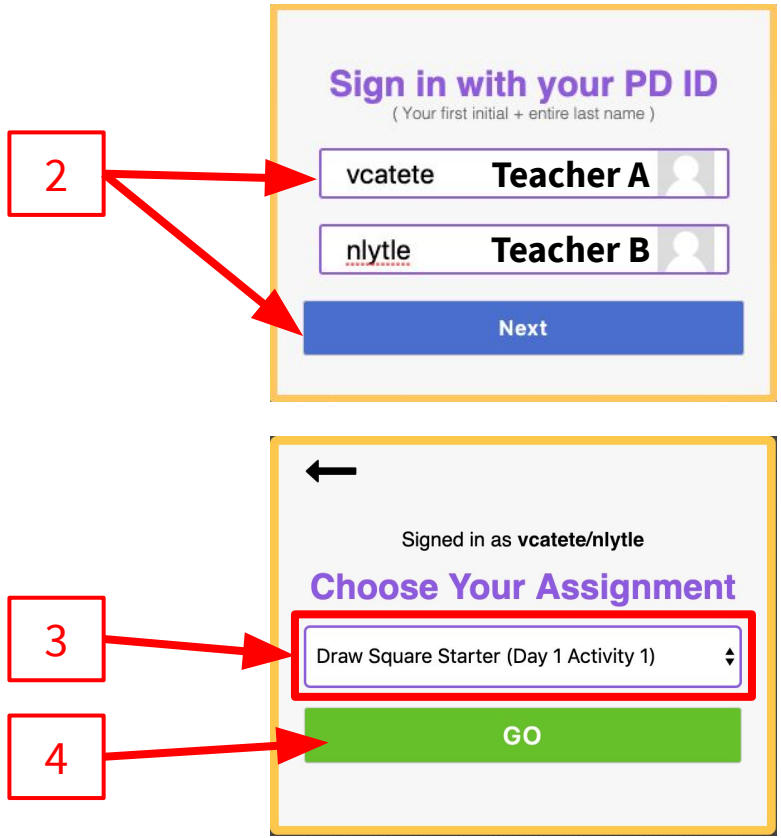
- Learn about the Snap! Programming Interfaces
- Learn about programming Snap! Commands by drawing
- Learn about Computational Thinking Terms



Sign in to Snap!

1. Go to stemc.csc.ncsu.edu/2019_pd_snap to "Sign in with PD ID"
2. Type the PD ID of Teacher A & Teacher B and click "**Next**"
3. Choose "**Draw Square Starter**" from the "Choose Your Assignment" window
4. Click "Go"

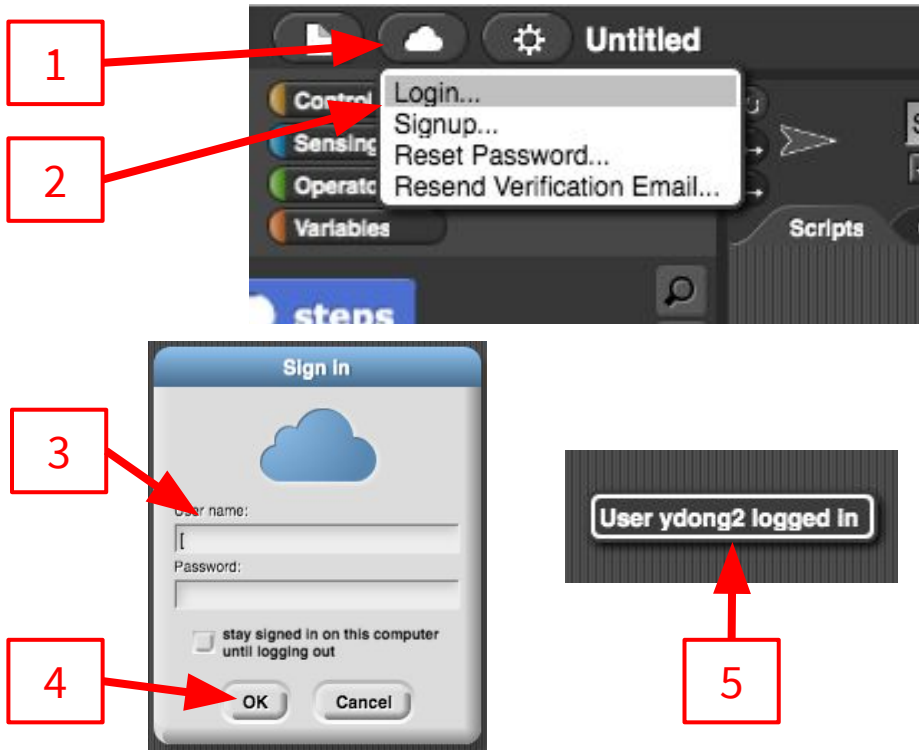
If you have trouble logging in, message the **#code-session** channel on Slack with your name



Step 1: Login to Snap Cloud

1. Click on the "**Cloud**" Icon
2. Select "**Login...**"
3. Input **Teacher A**'s login information in the pop up window
4. Click on **Okay** button
5. Look for confirmation information

(If neither partner has a Snap! account, click here)



Step 0: Revisit the Snap Interface

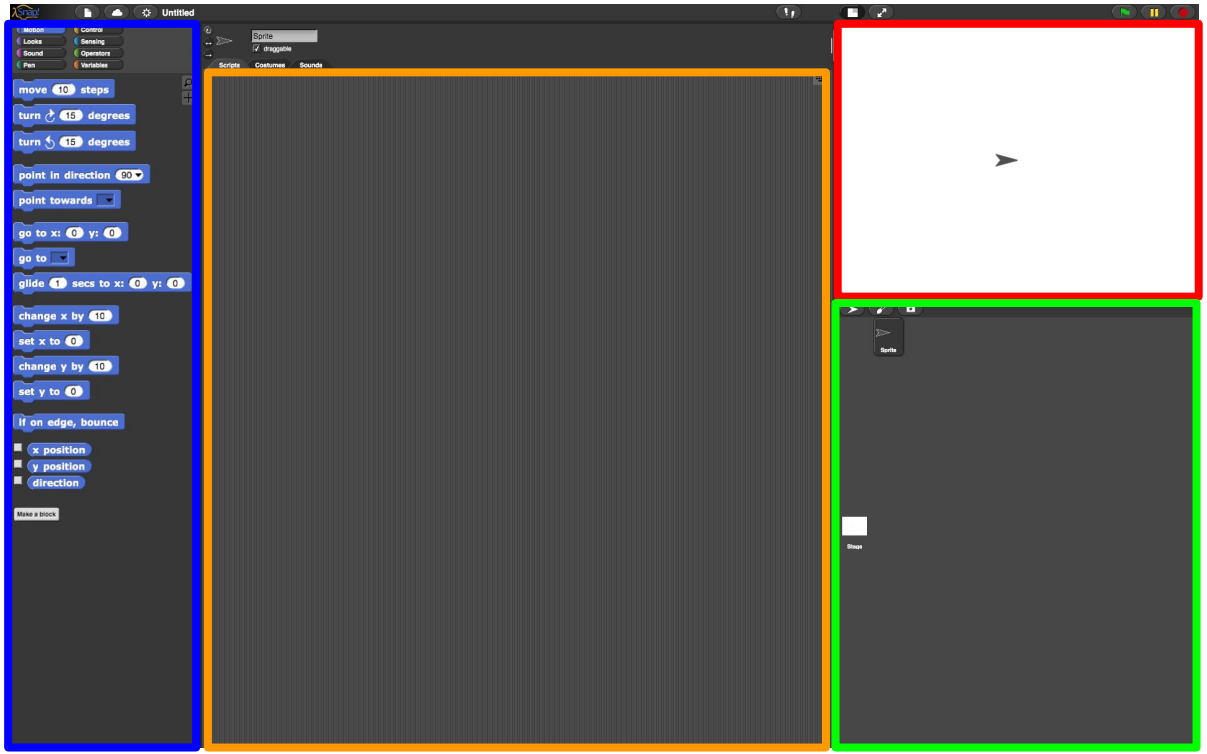
Palette - Where we find our command blocks

Scripting Area - Where we build programs.

Stage - Where we see our program's output.

Sprite Corral - Where we see our sprite objects.

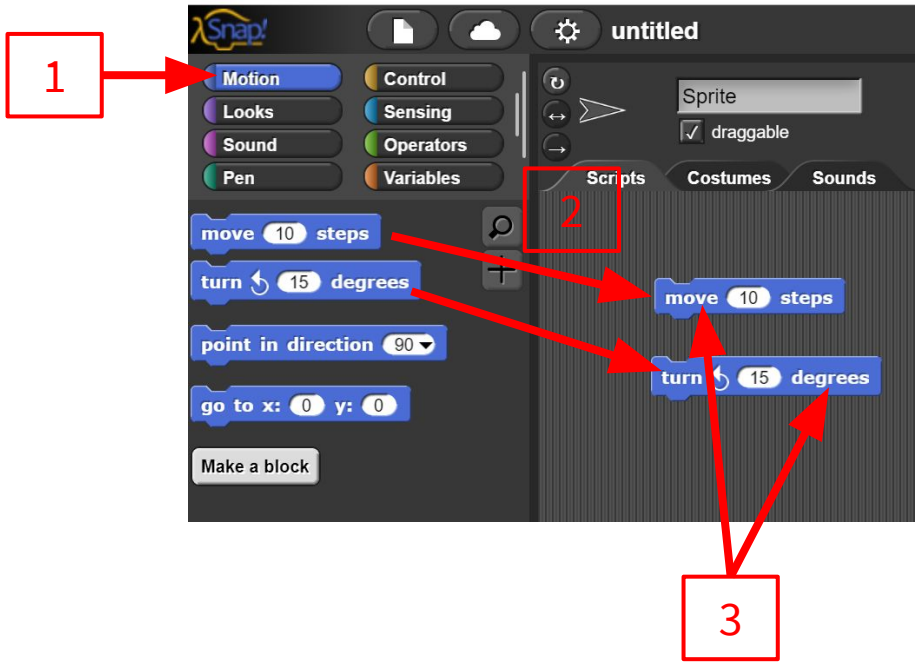
[we write code for a sprite]



Step 2: Use 'Move' and 'Turn' Blocks

Use motion blocks to transform a Sprite

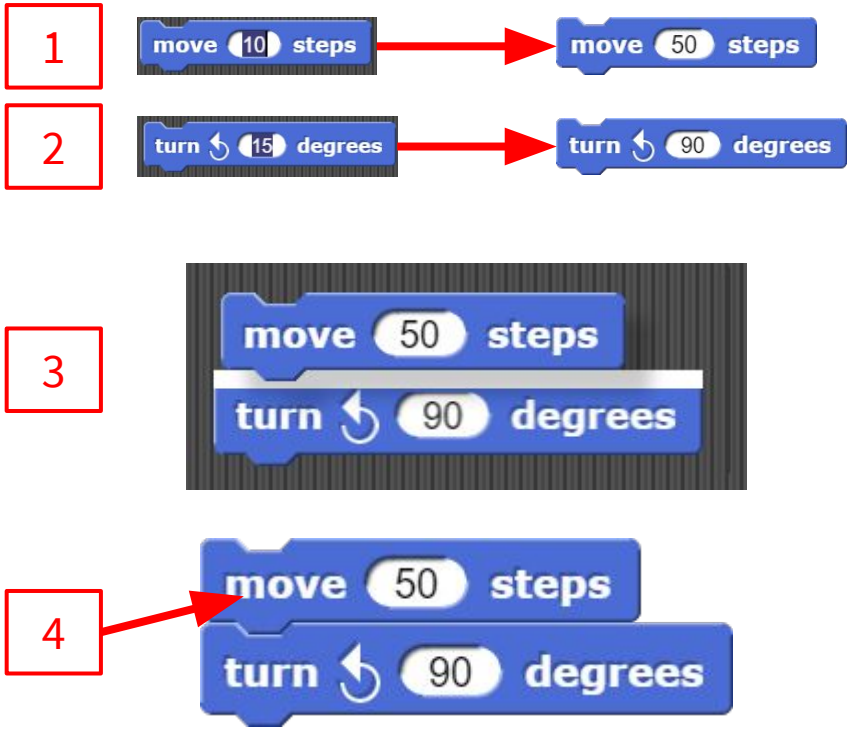
1. Select the Motion category from the Palette
2. Click and Drag the "Move 10 steps" and "Turn 15 degrees" blocks to the Scripting Area.
3. Click on each block to see how they affect the Sprite in the Stage.



Step 2: Move *then* Turn

1. Click to Type the number 50 in the Move block to make the Sprite move 50 steps
2. Now, Update the Turn block to make the Sprite rotate 90 degrees
3. Click and Drag, to Snap the Move block and Turn block together
4. Click the new block pair. What happens?

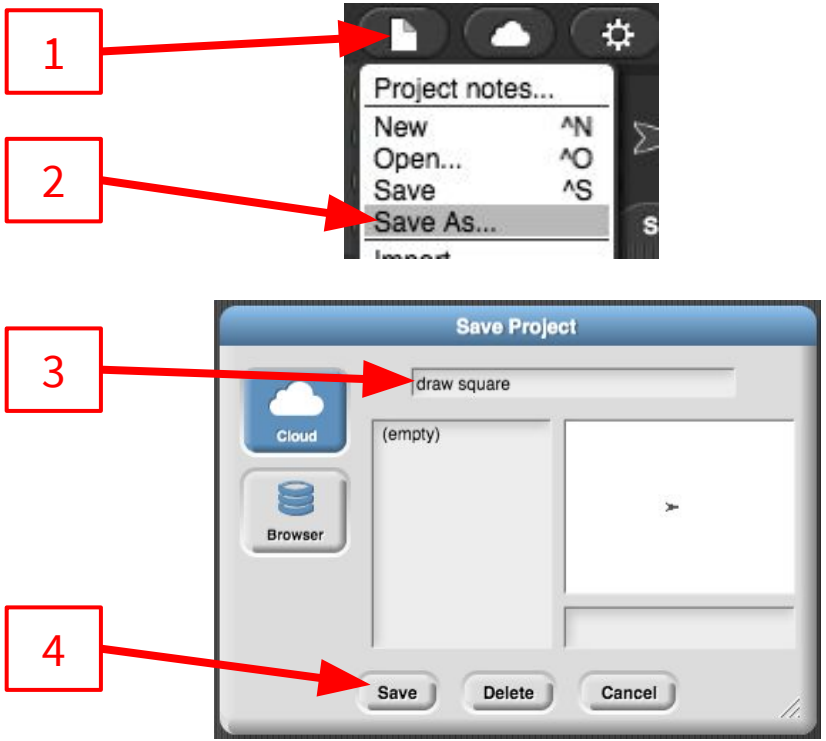
Now the sprite can move and turn! Yay!



Save Your Code

It's a good practice to Save your code every time you finish something!

1. Click the **File Icon**
2. Click "**Save as...**"
3. Type in "**draw square**"
4. Click on the **Save** button

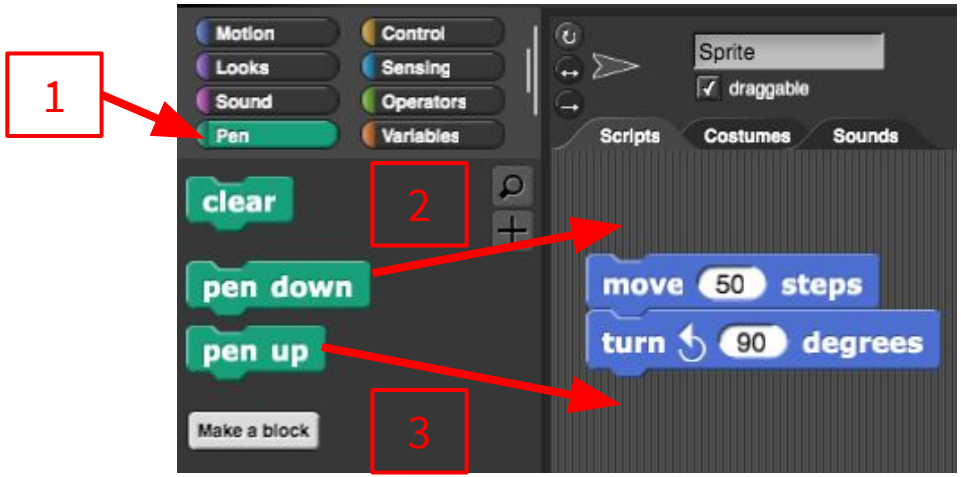


Save your code!

Step 3: Draw Something

Now, let's use the pen commands so the Sprite draws a line as it moves.

1. Click on the Pen tab in the Palette.
2. Drag and Snap the Pen Down Block above the Move Block.
3. Drag and Snap the Pen Up Block below the Turn Block.



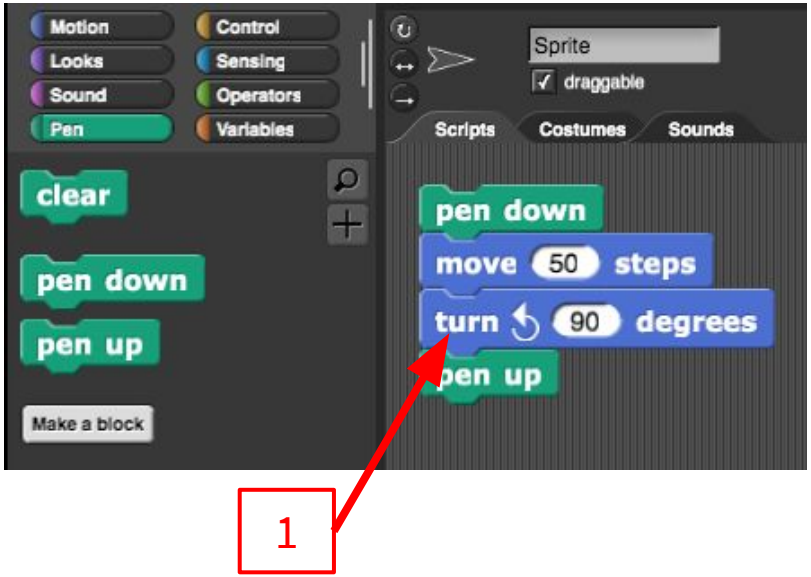
Check your answer

Step 3: Draw Something

1. Click on the script to run it.

Does your sprite draw a line and turn 90 degrees counterclockwise?

Yay! Your sprite can **draw something** now!



Save your code!

Switch for the next activity!

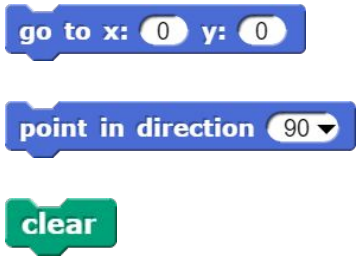
Teacher A: Navigate!
Teacher B: Drive!

Step 4: Make code to Reset the Stage

Let's make an **algorithm** to reset the stage!

Here are the sequence of steps:

1. go to the center of the Stage (0, 0)
2. point in direction 90 degrees (facing right)
3. clear the drawings on the Stage



Find the above blocks in the Motion and Pen tab and then **snap them together** in the right order.

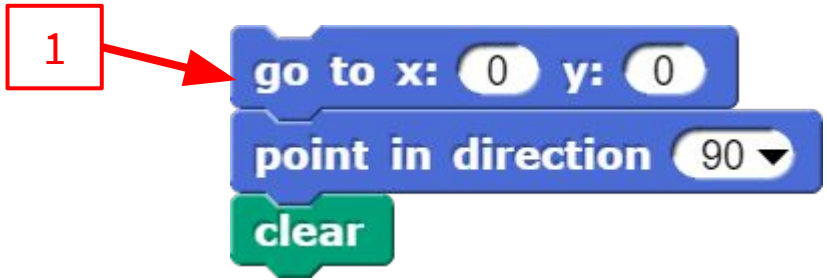
Check your answer

Step 4: Make code to Reset the Stage

Your code could look like this.

1. Click on the script to see if it clears the drawing and reset the sprite on the Stage.

Reset Algorithm



Algorithms

You just created an **Algorithm** to reset and clear the Stage!

Save your code!

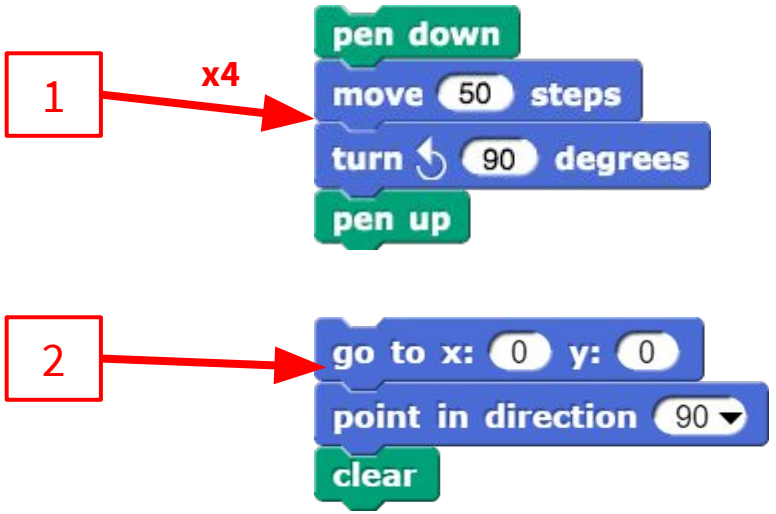
Step 5: Draw a square with duplicate code

Now, let's go back and finish the algorithm to draw a square.

1. Click on the **Draw Something** script 4 times

See how it draws a square? It seems we just need to duplicate the move and turn blocks 4 times to draw a square!

2. Click on the **Reset script** to clear the stage

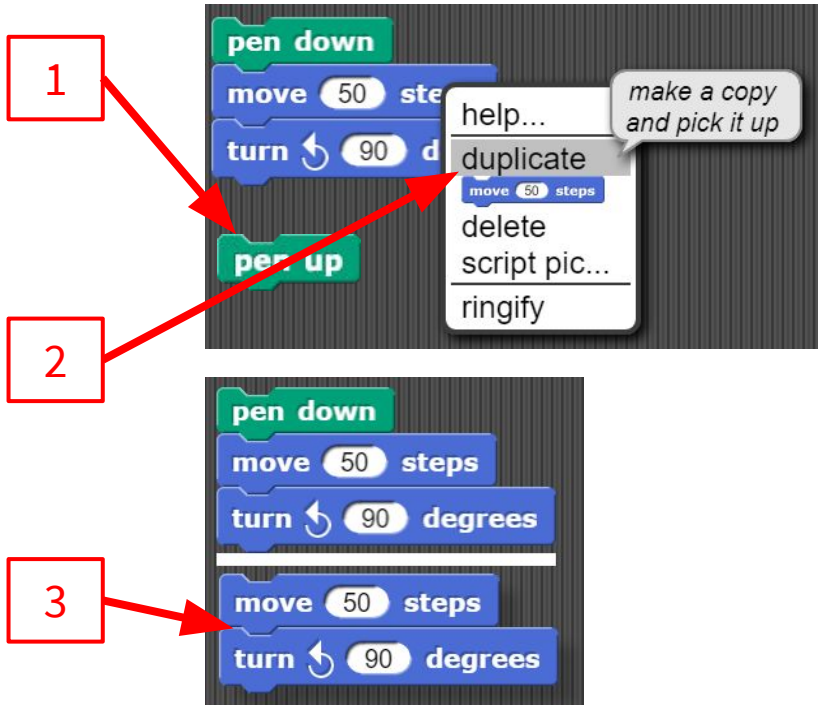


Step 5: Draw a square with duplicate code

Let's duplicate the move and turn blocks!

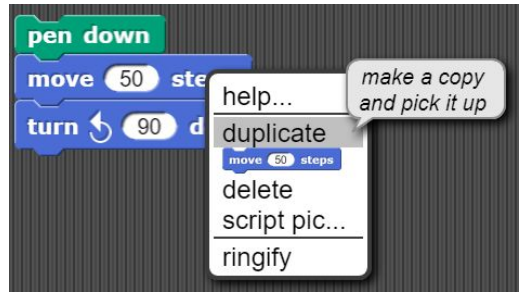
1. Unsnap the pen up block and put it aside
2. Right-Click on the Move block and select Duplicate.
3. Snap the blocks together.

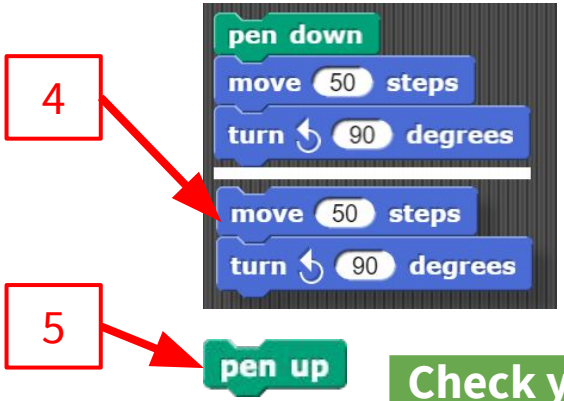
Does it draw half a square?



Step 5: Draw a square with duplicate code

4. Repeat duplicating the blocks until it draws a whole square.
5. Snap the pen up block at the end



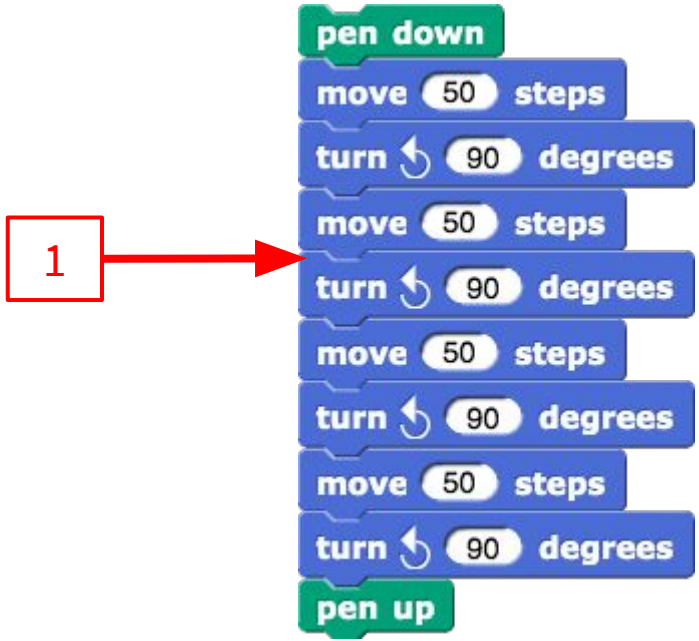


Check your answer

Step 5: Draw a square with duplicate code

Does your code look like this?

1. Click the script and see if it **draws a square**



Algorithms

You just created an **Algorithm** to draw a square!

Save your code!

Switch for the next activity!

Teacher A: Drive!
Teacher B: Navigate!

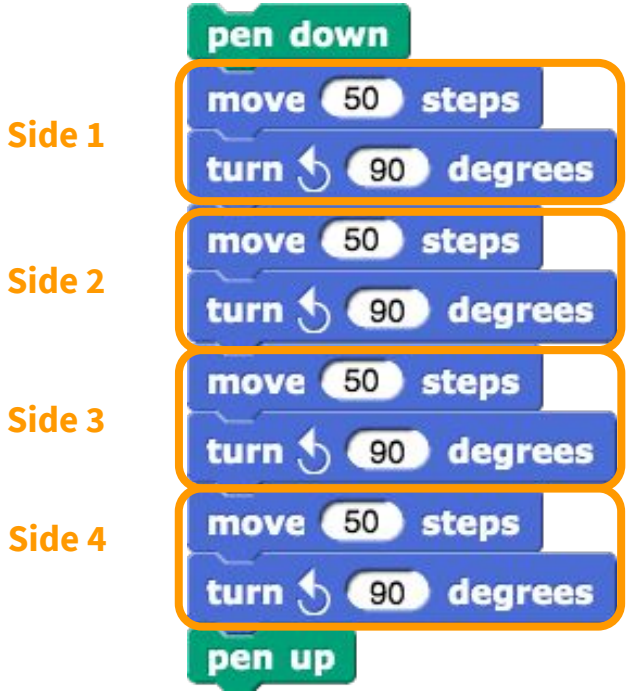
Step 6: Draw a Square with Repeat Block

Do you recognize the repetitive pattern?

Does it seem like there should be a more efficient way to do this?

Pattern Recognition

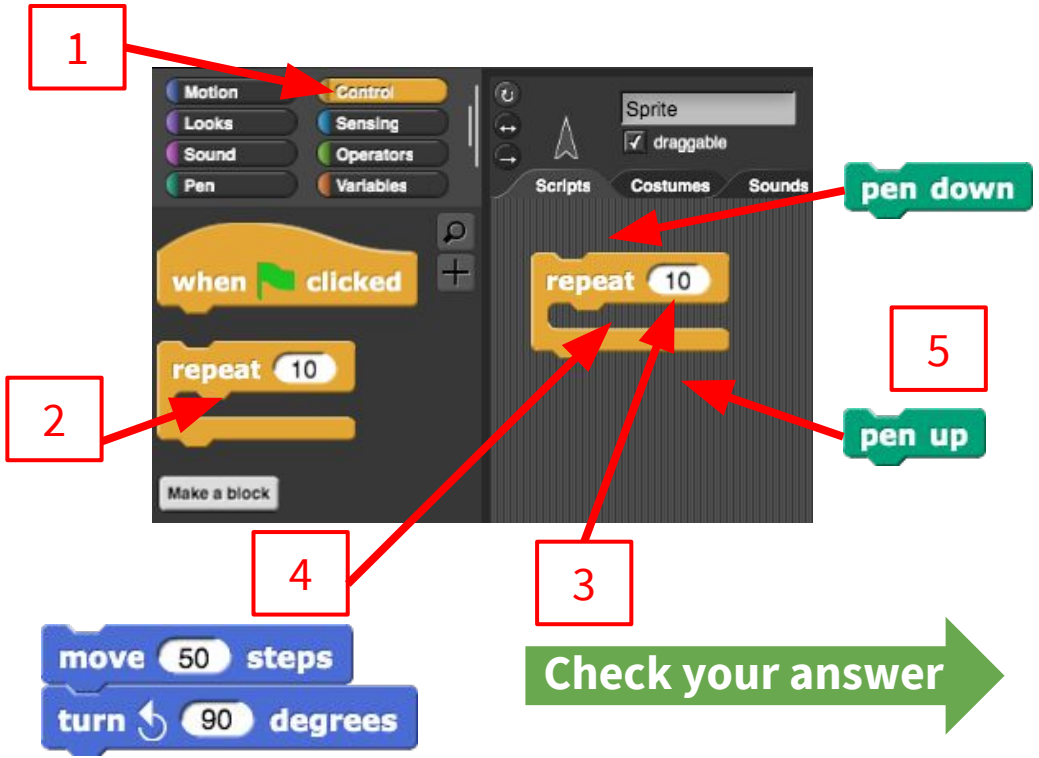
Notice the **pattern** of repeating code blocks.



Step 6: Draw a Square with Repeat Block

Blocks snapped inside a repeat loop will *repeat* however many times the loop says.

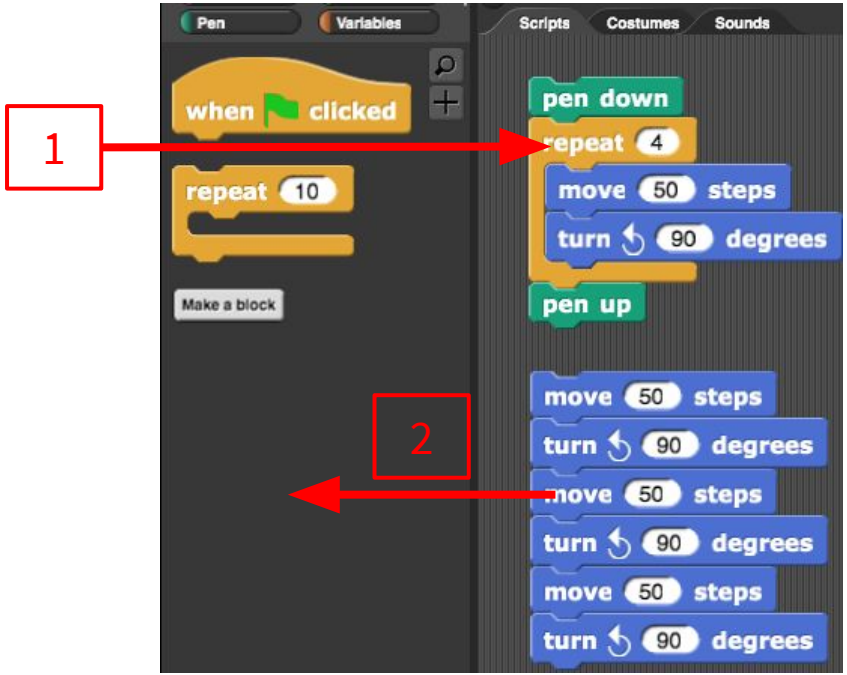
1. Click on the Control category
2. Add a repeat block to the Scripts area.
3. Change repeat to 4 times
4. Snap a move and turn block into the repeat loop
5. Snap a pen down and pen up block around the repeat block



Step 6: Draw a Square with Repeat Block

Does your code look like this?

1. Click on the script to see if it draws a square!
2. You may drag and drop unwanted scripts into the palette to delete them.



Pattern Recognition

Notice how the repeat block uses Pattern recognition to simplify the code!

Save your code!

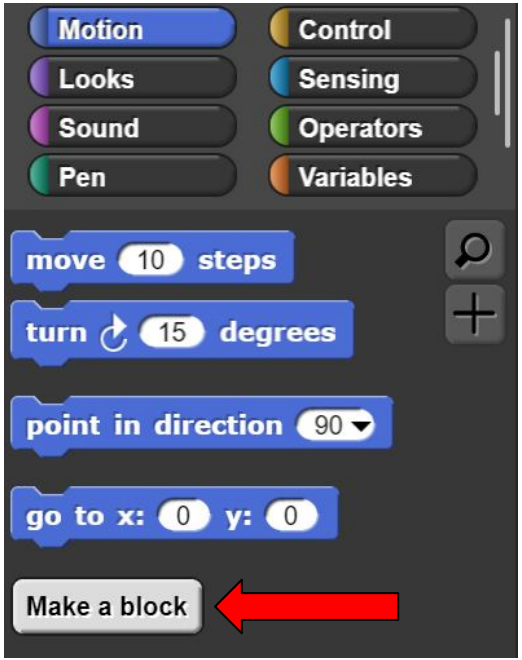
Step 7: Make your own Draw a Square Block

What if, just like the move and the turn blocks,
I want to have my own block to draw a square?
Can I do that?

Of course!

Snap offers a way to hide details of your code
in your own Custom Blocks!

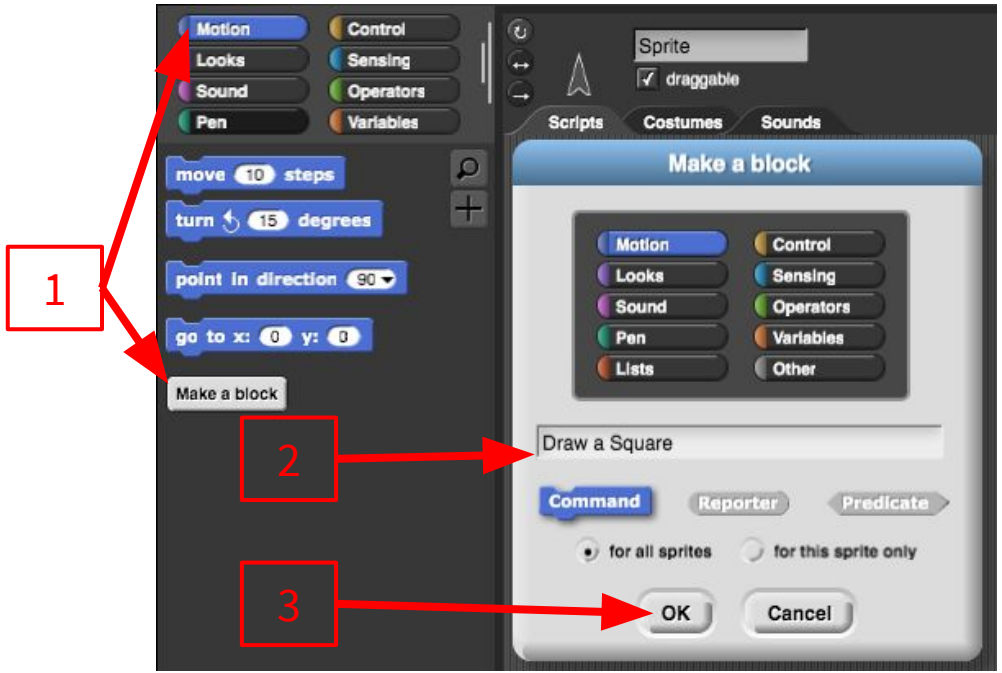
This way, you can give your script a
meaningful name, in this case, draw a square.



Step 7: Make your own Draw a Square Block

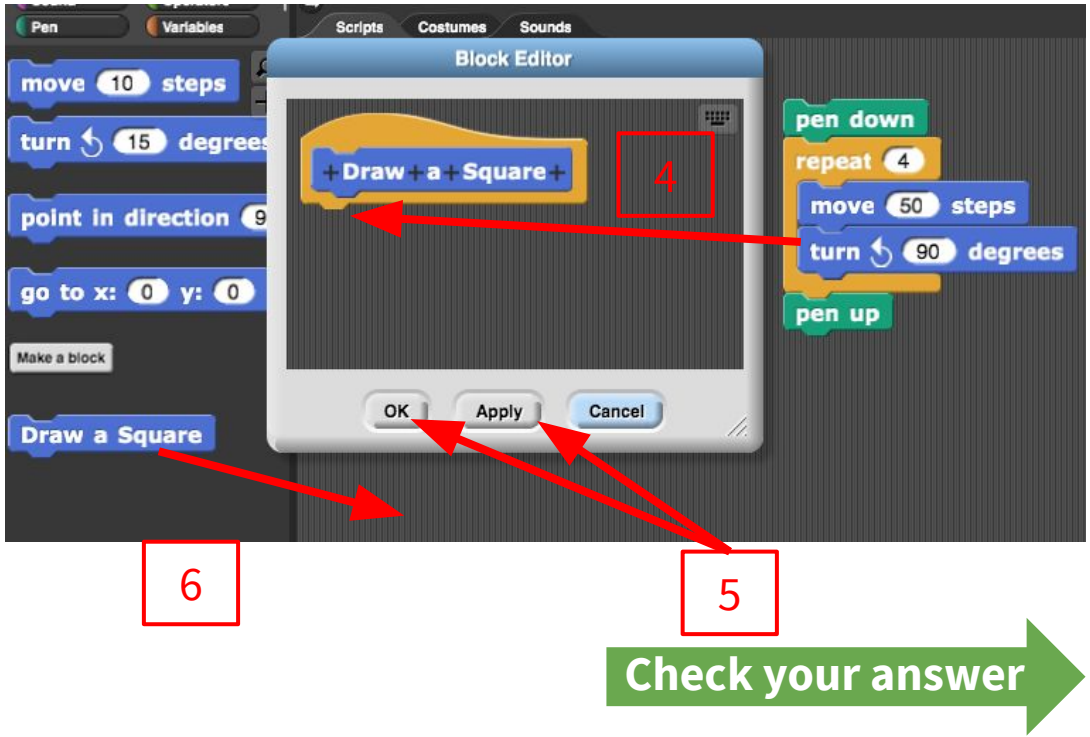
Let's create a custom block to draw a square.

1. In the Motion tab, click on the "Make a block" button
2. Give the custom block a meaningful name, Draw a Square
3. Click on the OK button



Step 7: Make your own Draw a Square Block

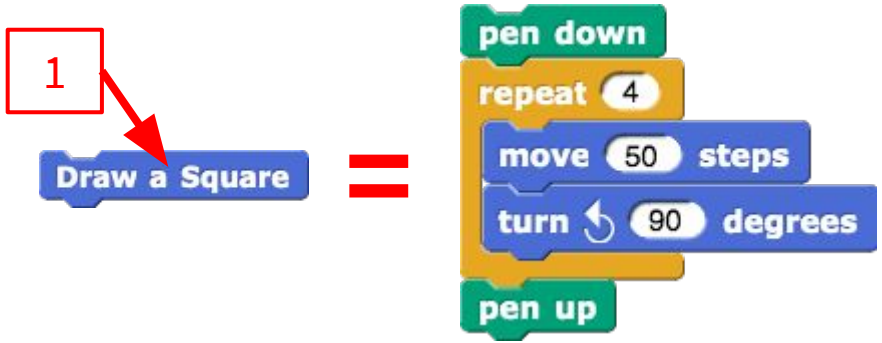
4. Drag and snap the draw a square script into the Block Editor below Draw a Square
5. Click Apply and then click OK to close the Block Editor
6. Drag and drop the Draw a Square block from the Palette into Script Area



Step 7: Make your own Draw a Square Block

1. Click on the Draw a Square block.
   Does it draw a square?

You've created your own Draw a Square custom block! It behaves exactly like the draw a square script!



Abstractions

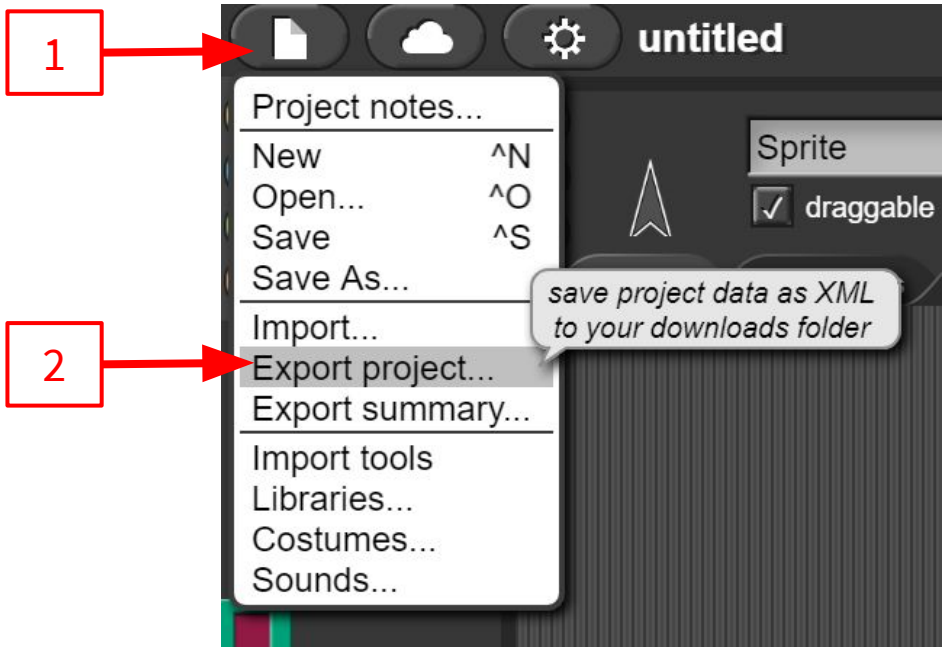
Custom Blocks uses Abstraction to hide a script and give it a meaningful name.

Save your code!

You've Completed Activity 1!

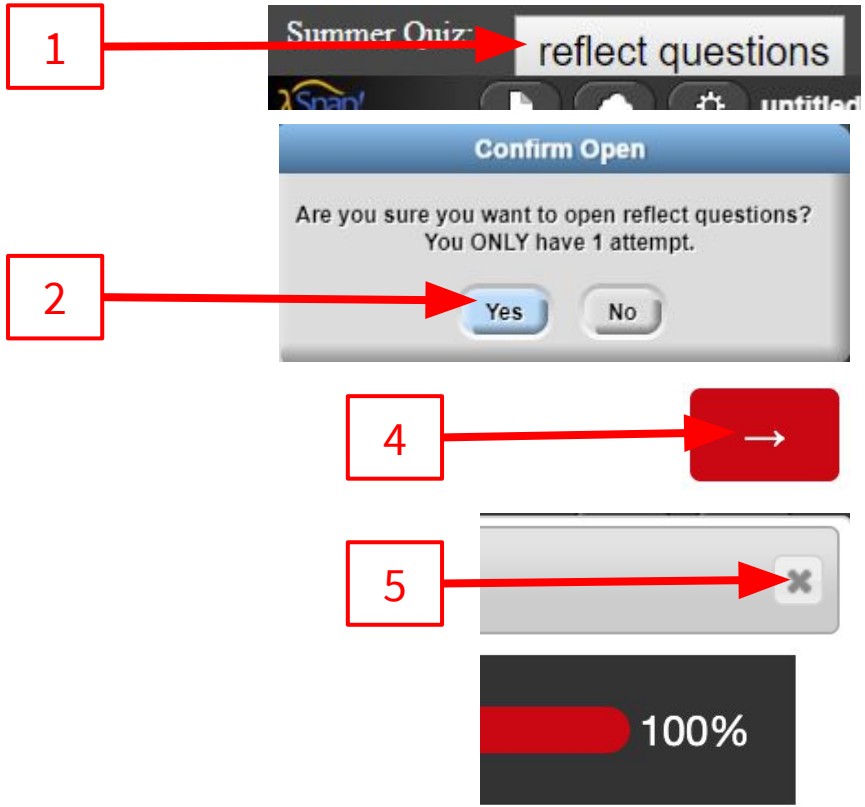
If you want to export and download the project to your computer, here is how:

1. Click on the File Menu
2. Click on Export project
3. Choose where you want to save the project file and click save



Let's Reflect!

1. Click "Reflect Questions" in the top right.
2. Press **Yes** to begin the reflection.
3. Use the **red arrow** button to go to the next question.
4. Finish out the reflection quiz.
5. After finishing the quiz, use the **cross button** on the top right corner of the survey window to close.



Switch for the next activity!

Teacher A: Navigate!
Teacher B: Drive!

Activity 2: Introduce Yourself

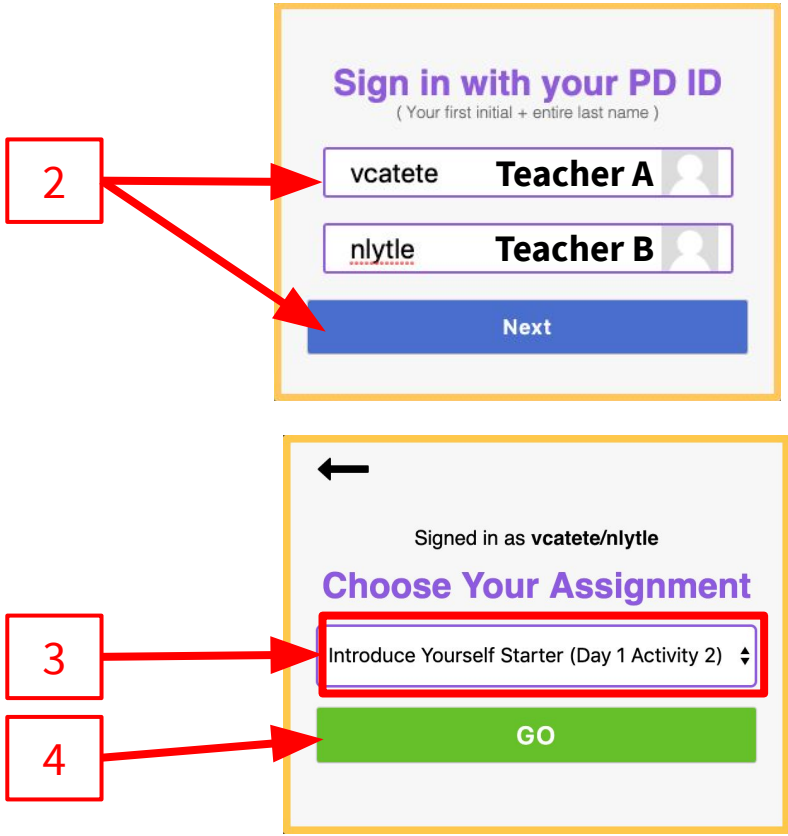
Goals & Project Demo

- Learn some new Code Blocks
- Explore Different Types of Programming Tasks
- Make an Interactive Narrative - a program that introduces a concept and users can interact with

Open Activity 2 in Snap!

1. Go to stemc.csc.ncsu.edu/2019_pd_snap to "Sign in with PD ID"
2. Type the PD ID of Teacher A & Teacher B and click "**Next**"
3. Choose "**Introduce Yourself Starter**" from the "Choose Your Assignment" window
4. Click "**Go**"

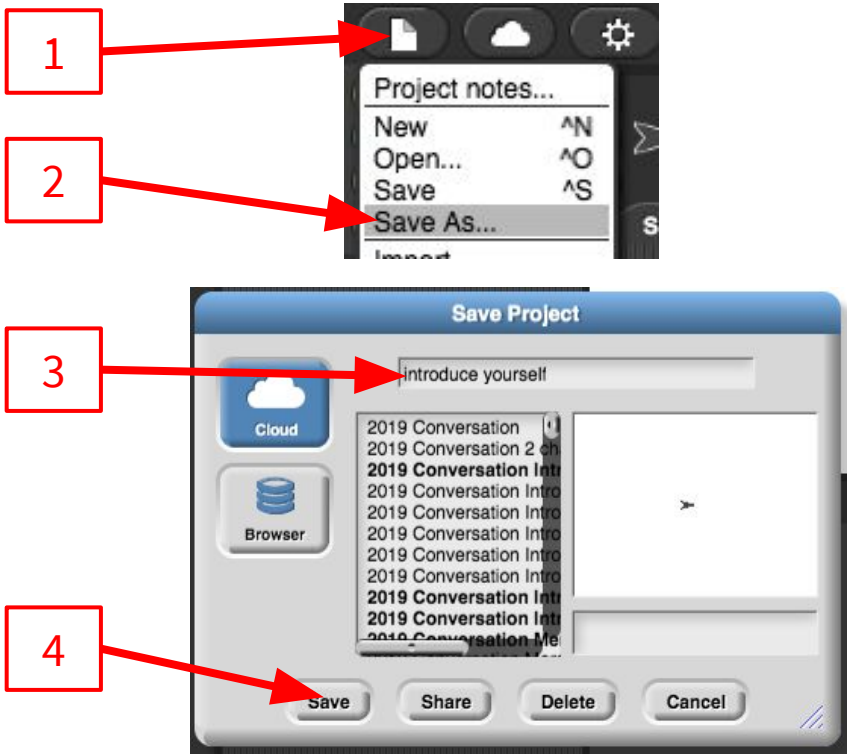
(Need help finding your PD ID? Click here)



Save Your Code

It's a good practice to Save your code every time you finish something!

1. Click the **File Icon**
2. Click "**Save as...**"
3. Type in "**Introduce Yourself**"
4. Click on the **Save** button

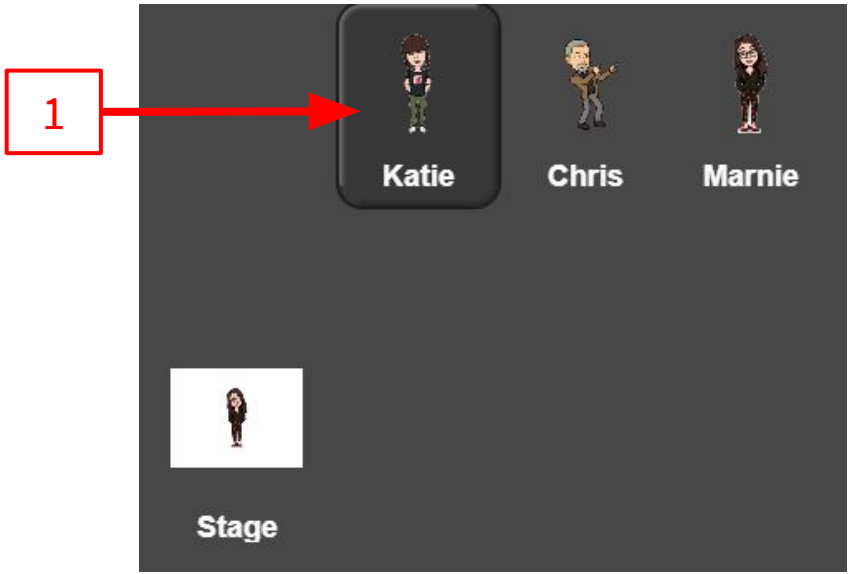


Explore Starter Code

You'll notice we already have a lot of code and sprites in the environment.

Sometimes we can use **starter code** to make the assignment easier to learn instead of coding everything from scratch! We'll explore this today!

1. Click on the Katie sprite to begin!



Step 1: Katie's Worked Example

Katie's Code is already written out:

1. Click on each of the code blocks to trigger each event to see what happens

An event is code that runs when something else triggers it (like clicking the Flag, or pressing the space key)

Decomposition

We use Decomposition to breakdown Kaitie's actions into 3 events.

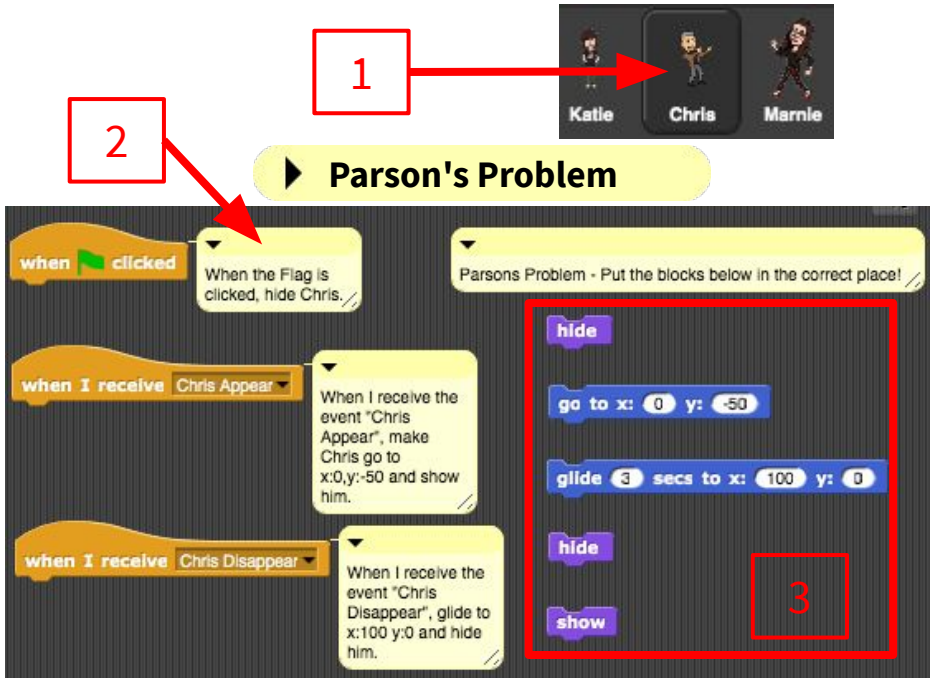
Worked Example

Step 2: Complete the Chris Code

1. Click on the **Chris** Sprite to look at his code.

Chris should have very similar behavior to Katie. The blocks we need to complete this code are all here, but they aren't snapped in place yet.

2. Read the comments in Chris' Sprite
3. Snap the Move and Look commands into the correct place



Check your answer

Step 2: Complete the Chris Code

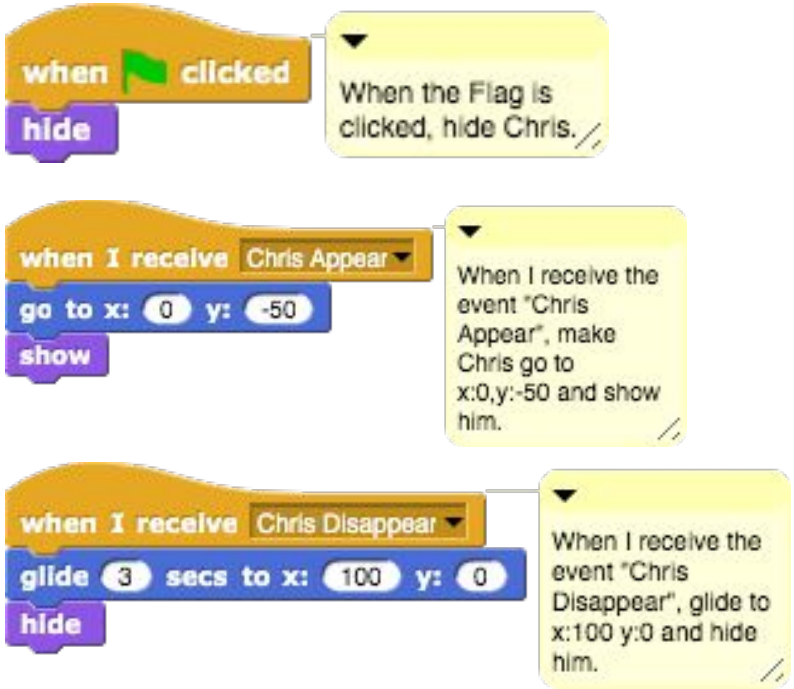
How does your code compare?

If it works congrats! You've added behaviors to the Chris Sprite!

Good job!

Pattern Recognition

Do you **recognize a pattern** between the Chris blocks and the Katie blocks?

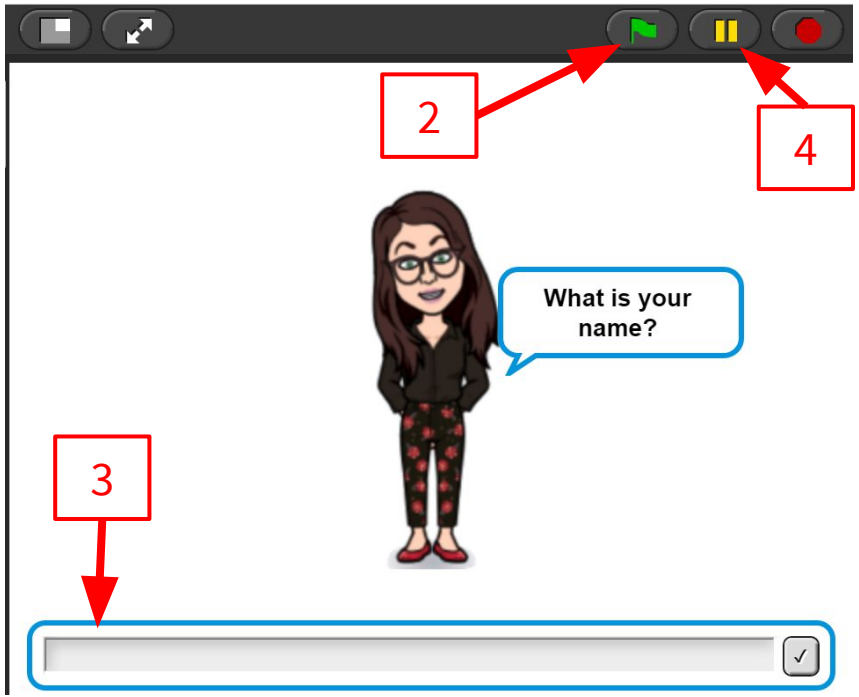


Save your code!

Step 3 - Mrs. Hill's Worked Example

Let's look at another worked example from Mrs. Hill's Sprite.

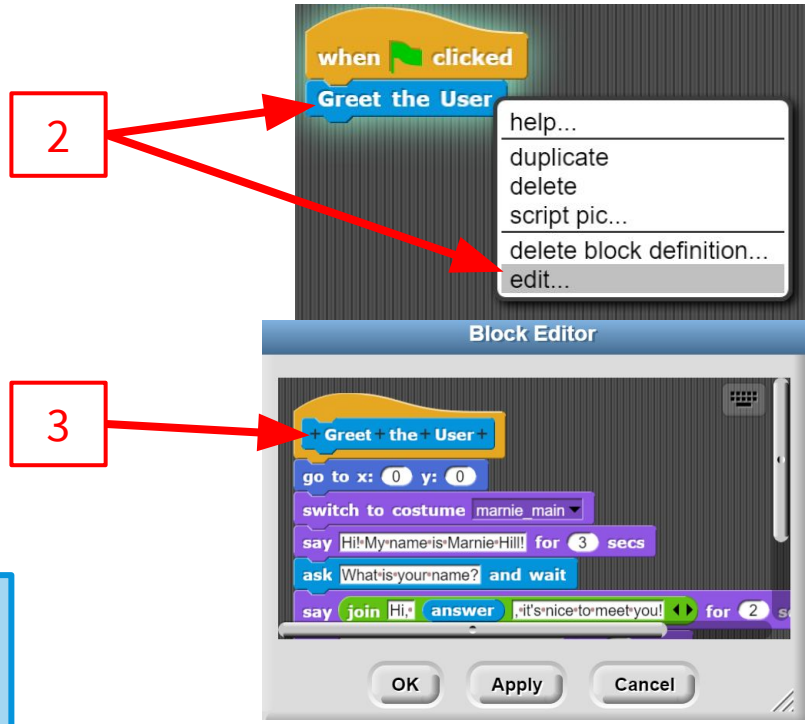
1. Click on the "Marnie" Sprite
2. Click on the Green Flag icon to run the "When Flag Clicked" code
3. Type in your name when asked
4. Click the Yellow pause button when she asks you to press the "Up arrow".



Step 3 - Mrs. Hill's Worked Example

First, let's see what Mrs. Hill's code looks like to greet the user.

1. Click on Marnie Sprite.
2. Right Click on "Greet the User" Block and then click **edit** to look inside.
3. Read through the code line by line and figure out how it works with your partner.



Abstractions

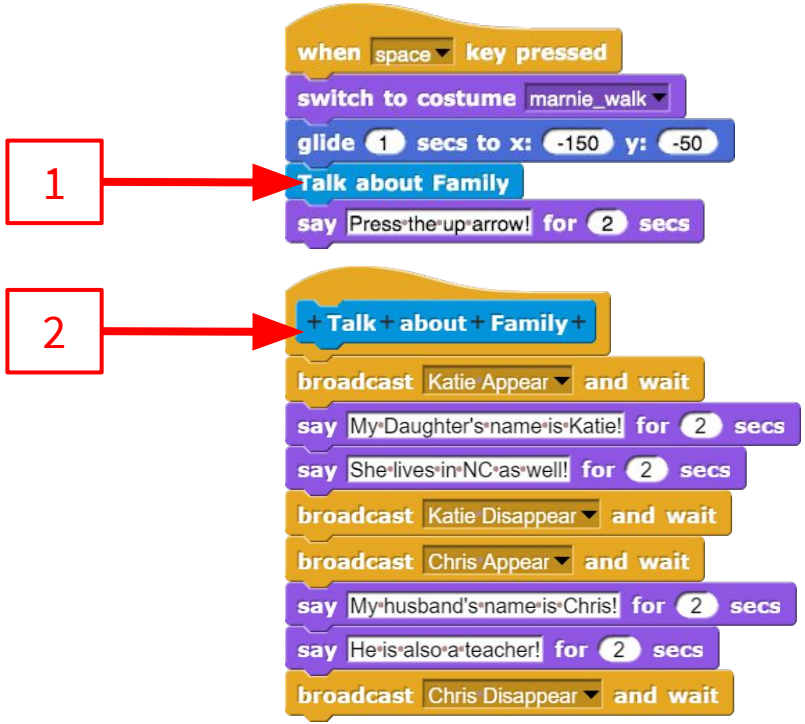
Notice how greet the user script is grouped and **abstracted** away into a custom block with a **meaningful** name

Step 3 - Mrs. Hill's Worked Example

Now, let's look into "Talk about Family" Block.

1. Right Click and Edit the "Talk about my Family" Block.
2. Read over the code and break down the code into two sections.

Which section of the code relates to Katie?
Which relates to Chris?



Abstractions

"Talk about Family" is another instance of Abstraction using custom block.

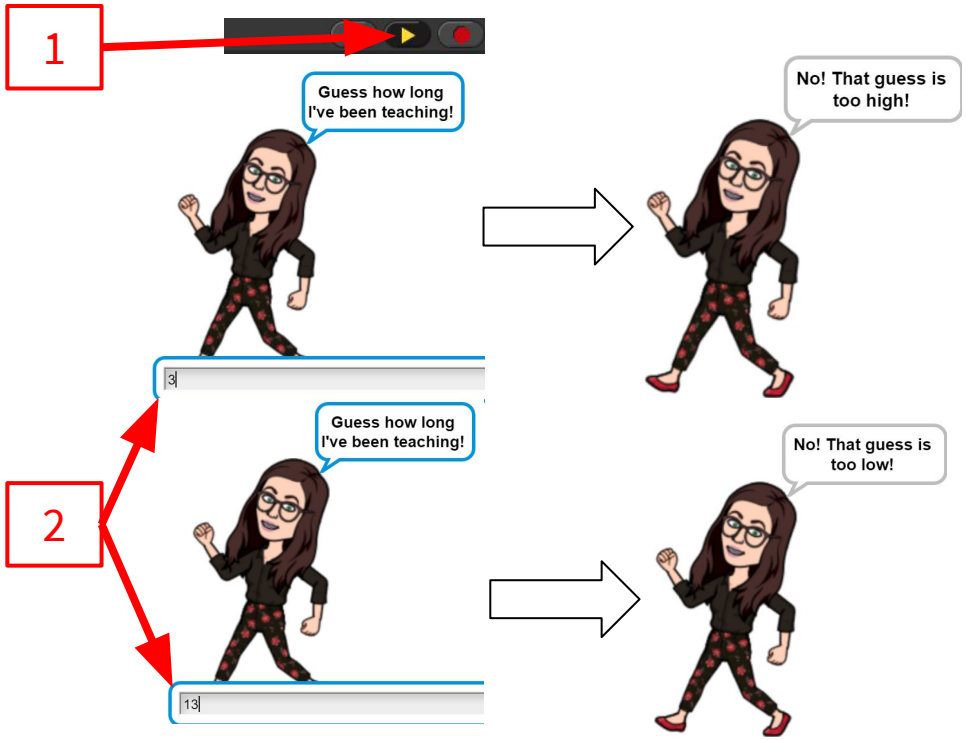
Step 4 - Fix a "Bug" in the Guessing Game

Let's continue to run the code.

1. Unpause the code by Clicking the Pause button again.
2. Try to guess how many years she has been teaching.

You'll notice the logic in Mrs. Hill's feedback for your guess seems to be reversed. (The answer is 7, shhh.....)

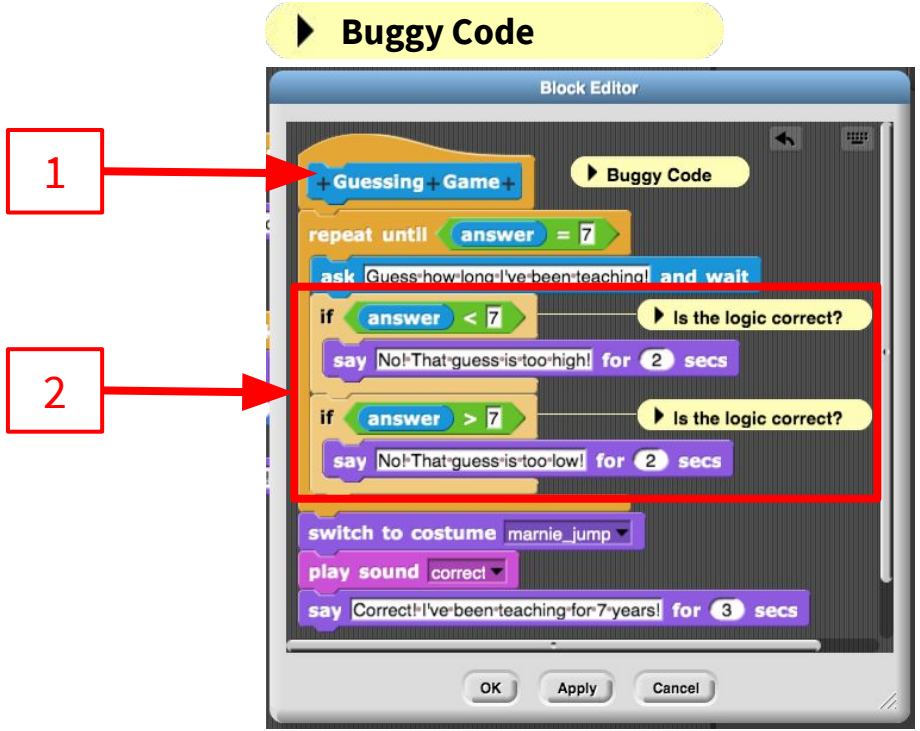
We are going to fix this "Bug".



Step 4 - Fix a "Bug" in Guessing Game

Let's look into the Guessing Game block and find out what went wrong.

1. Find the Guessing Game block and **edit** it in the Block Editor.
2. Read the code, focus on the two if blocks and see if you see any buggy logic in there.



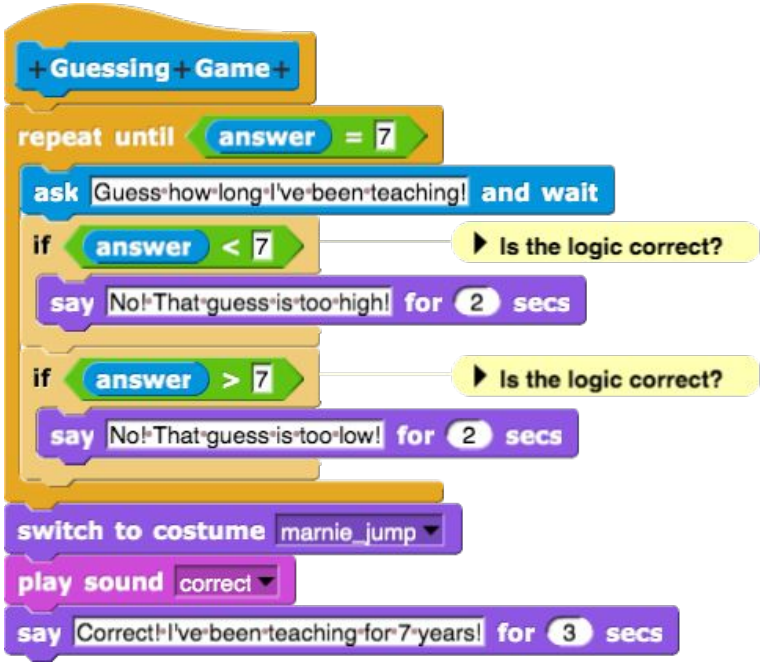
Step 4 - Fix a "Bug" in Guessing Game

The feedback logic is reversed!

if answer < 7, Mrs. Hill should say too low instead of too high!

if answer > 7, Mrs. Hill should say too high instead of too low!

1. Figure out how to fix the bug in the code with your partner.



Check your answer

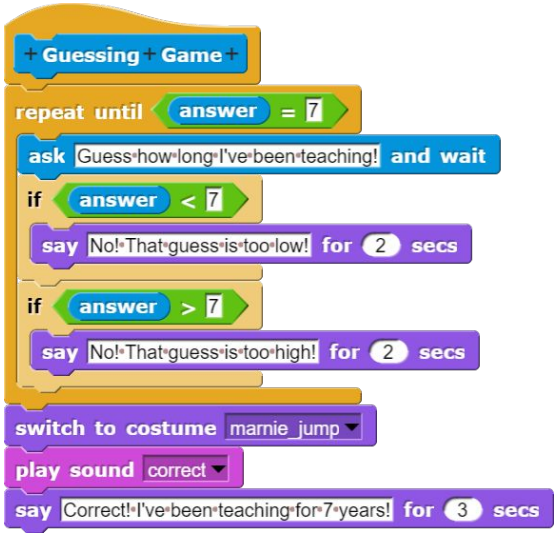
Step 4 - Fix a "Bug" in Guessing Game

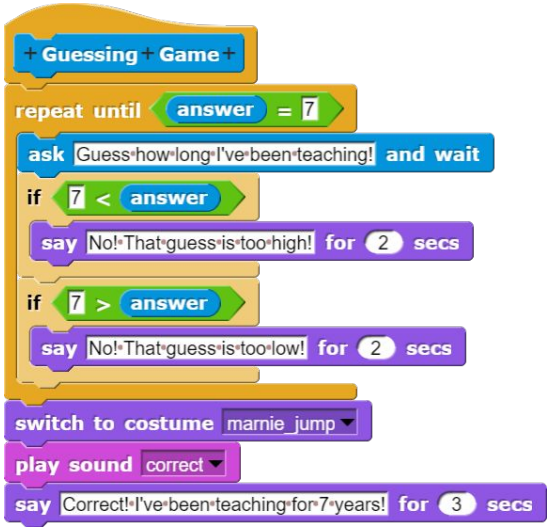
Does your code look like any of these?

If yes, that's great!

If not, as long as it works, it okay!

In computing, you can have **multiple solutions** for the same problem!

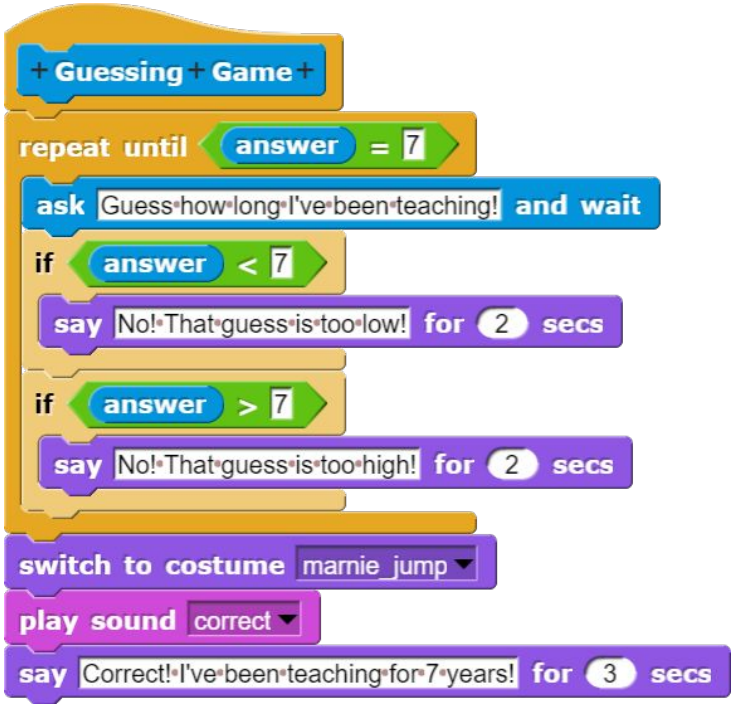




Step 4 - Fix a "Bug" in Guessing Game

Let's take a pause and think about Computational Thinking.

What Computational Thinking can you identify in this code?



Step 4 - Fix a "Bug" in Guessing Game

Pattern Recognition

The guessing game **repetitively** asks user to guess the number of years of teaching.

Abstractions

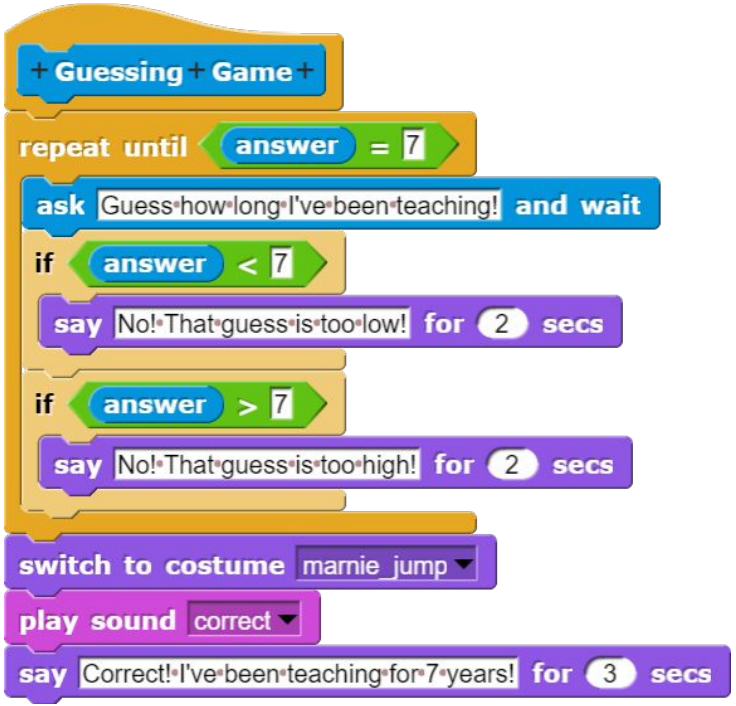
The Guessing Game **abstracted** away its code inside of a custom block.

Decomposition

The if blocks **decompose** the user's answers into different conditions and handles them accordingly.

Algorithms

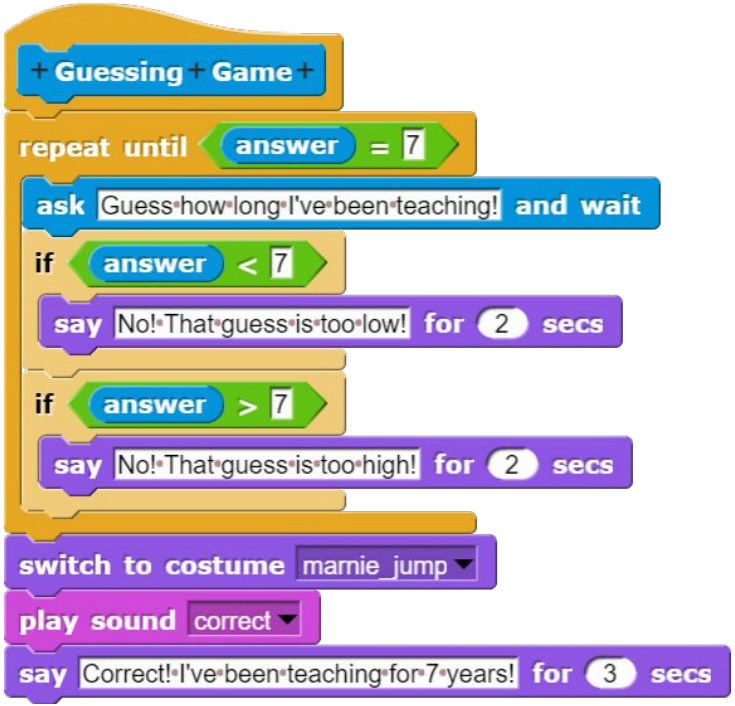
The whole script is **step-by-step** instructions on how to carry out a guessing game



Step 4 - Fix a "Bug" in Guessing Game

WOW!

That was a lot of Computational Thinking in one script!!!



Save your code!

Switch for the next activity!

Teacher A: Drive!
Teacher B: Navigate!

Step 5: Modify "Age in Days"

What if we want to, instead of saying "Age in Days", say "Age in Months"? How do we do it?

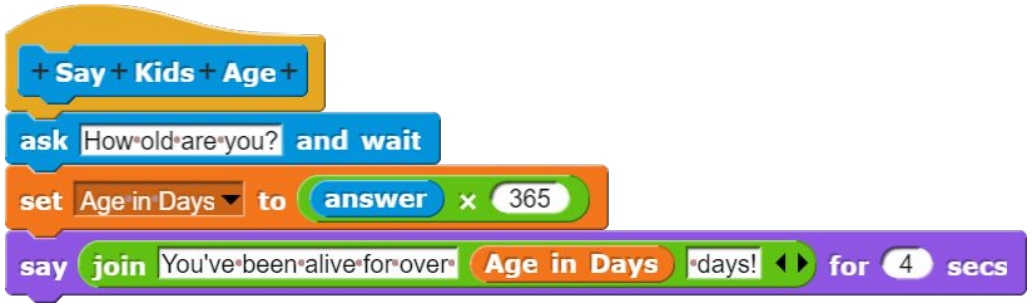
There is a custom block that handles "Say Kids Age".

Guess what is the name of the custom block?

Step 5: Modify "Age in Days"

The custom block is "Say Kids Age"!

1. Find the Say Kids Age block, then right click and edit it.
2. Find the set Age in Days block



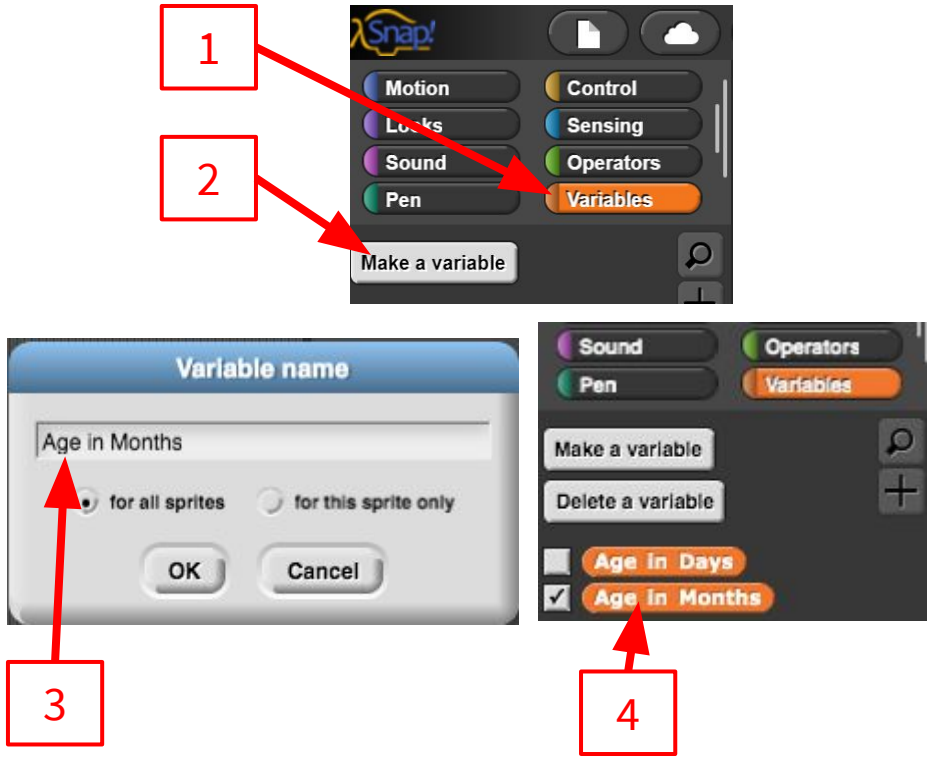
Here, we set a **variable** called Age in Days to be the answer of our question, "How old are you?" multiplied by 365.

A **Variable** is a block that stores a value for later use

Step 5: Modify "Age in Days"

To modify the code, let's first create a variable called "Age in Months"

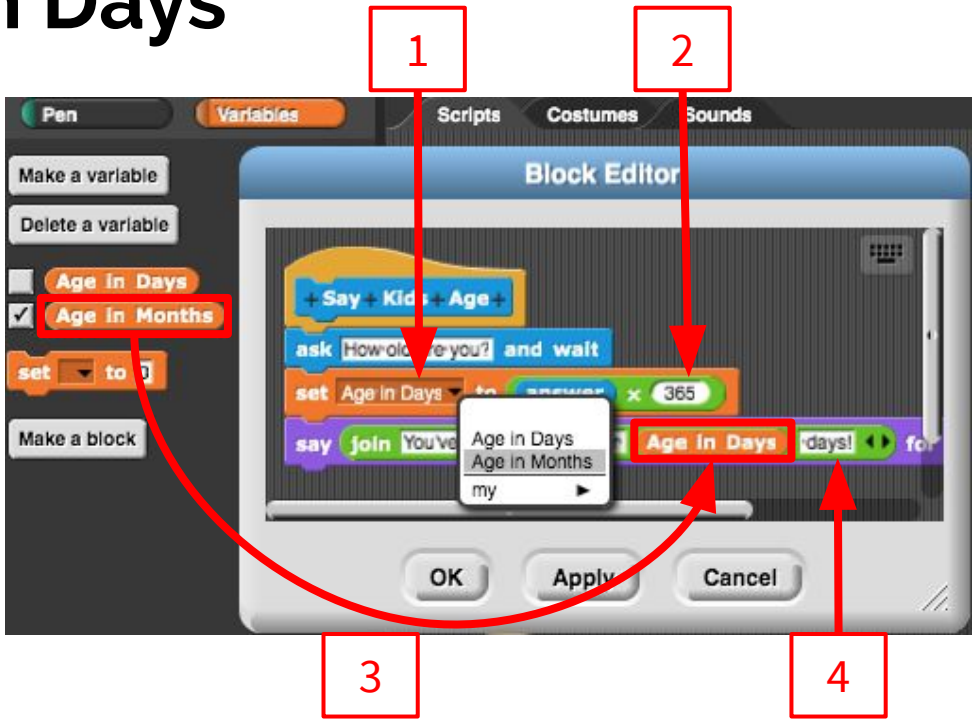
1. Switch to the Variables tab
2. Click the **Make a variable** button
3. Type in "Age in Months" as the Variable name and click OK
4. You should see a "Age in Months" variable created below "Age in Days" in the palette



Step 5: Modify "Age in Days"

There are three changes we want to make to the code:

1. Change the set block to set "Age in Months"
2. Change "answer *365" to "answer *12".
3. Replace the Age in Days variable with Age in Months
4. Change the final textbox from " days" to " months"



Check your answer

Step 5: Modify "Age in Days"

What does your code look like?

1. Run your code to see what happens

Does Mrs. Hill say kids' Age in Month now?

Even a small modification to the code can produce very different results!

Abstractions

Variables allow us to **ABSTRACT** some details away so we don't need to write "Answer x 12" every time

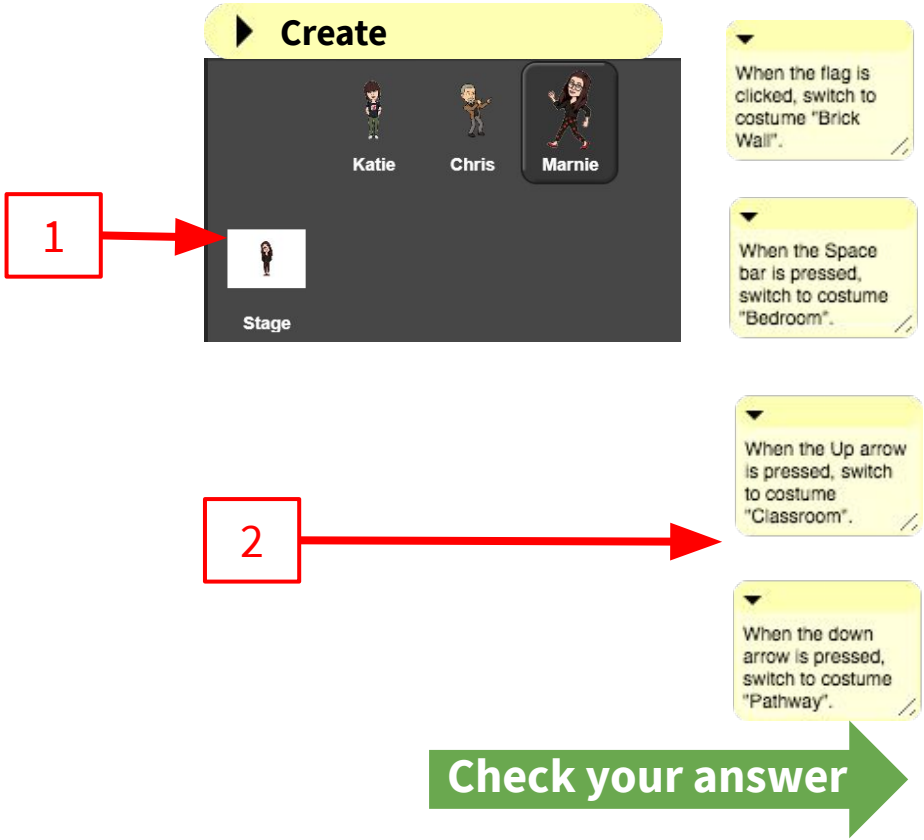
Save your code!

Step 6: Create behavior for Stage

Let's create some code to make the Stage change costumes (background)

1. Click on the Stage icon. You'll see comments describing what code should be written.
2. Create code blocks based off of the comments described.

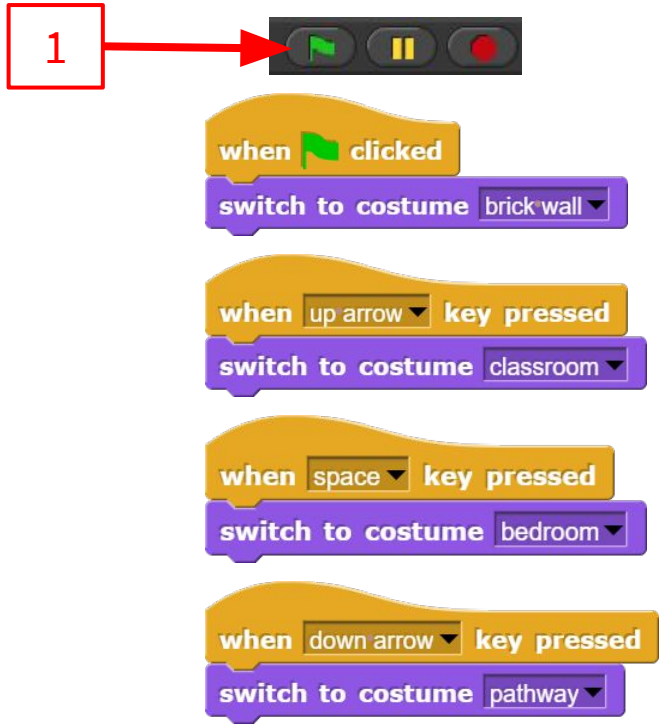
Hint: you may look at Katie, Chrise, or Marnie sprite for reference.



Step 6: Create behavior for Stage

Does your code look like this?

1. Run you script from the beginning by clicking on the Green Flag button.
2. See if the stage changes with different key presses.



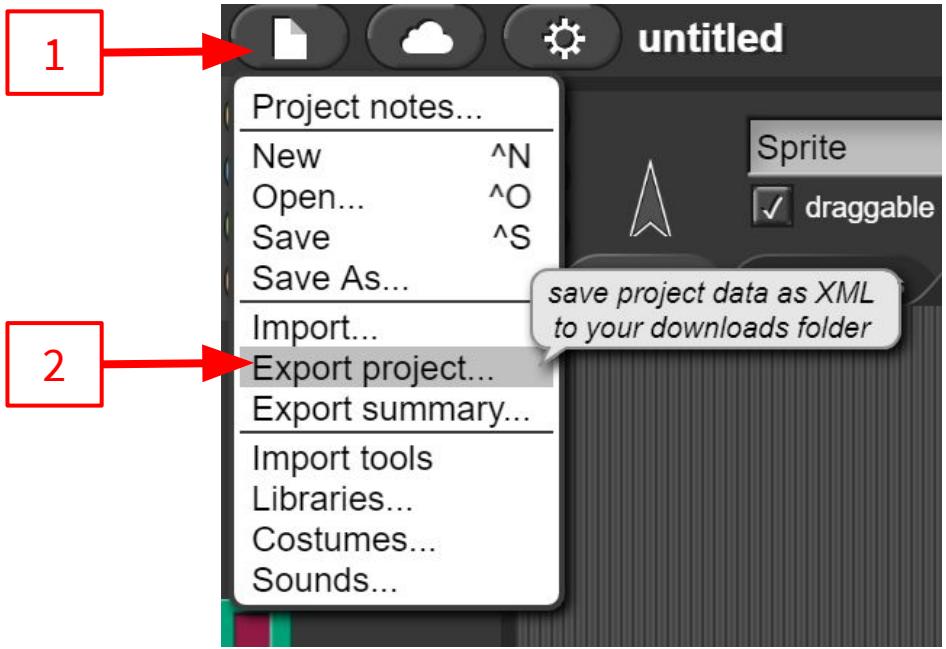
Decomposition

You just **decomposed** the stage's behavior based on different events!

You've Completed Activity 2!

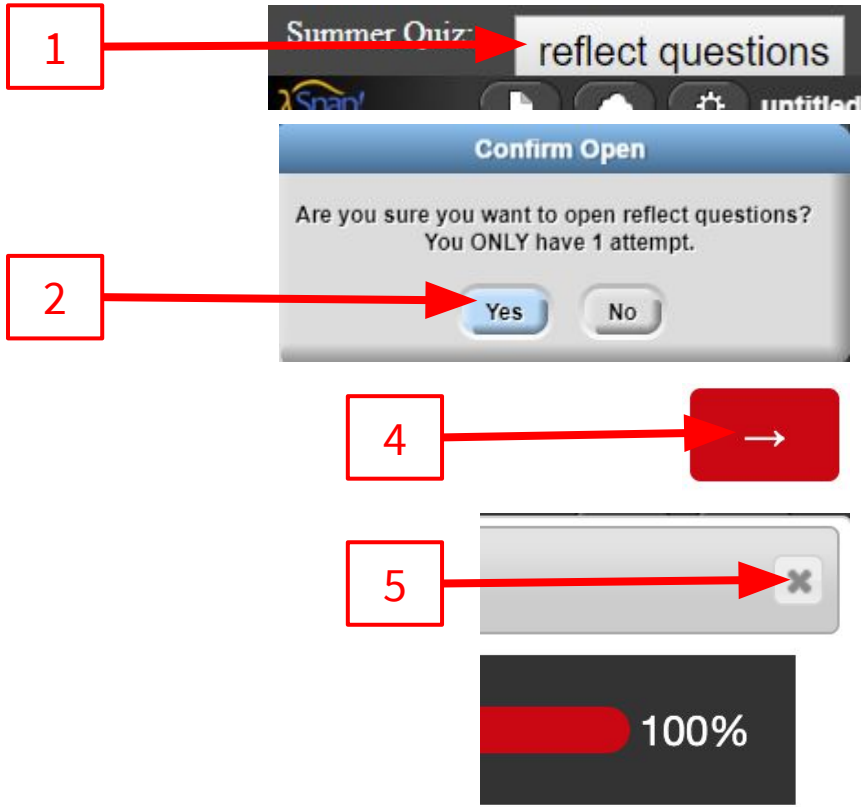
If you want to export and download the project to your computer, here is how:

1. Click on the **File** Menu
2. Select **Export project**
3. Choose where you want to save the project file and click **save**



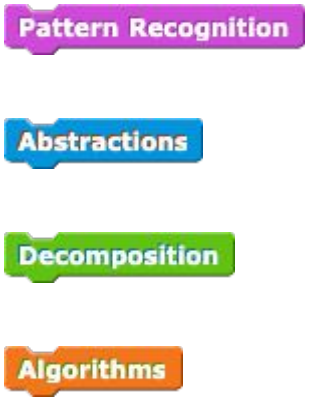
Let's Reflect!

1. Click "Reflect Questions" in the top right.
2. Press **Yes** to begin the reflection.
3. Use the **red arrow** button to go to the next question.
4. Finish out the reflection quiz.
5. After finishing the quiz, use the **cross button** on the top right corner of the survey window to close.



Reflection

We saw PRADA Concepts in action



We also learned Snap! Concepts

How to use the interface

How to read, edit, make code

Make and edit custom blocks

Make and use variables

Congratulations on your coding conquest!

3C