Lesson
3-4A
Action Fractions
Math+CT

Ambling Animals

Math Connections: Students compare fractions represented on number lines.
CS Connections: Students work with variables and conditionals to store and react to user input in a program.

everydaycomputing.org

Computational Thinking

- **CONDITIONALS:** Conditional statements are computer commands to evaluate conditions and complete connected actions.
- **VARIABLES:** It can be helpful to use variables within programs.
- **VARIABLES:** Many programs accept user input.
- **VARIABLES:** Variables can hold user input.

1 Warm Up 5–10 min

	Materials	
Conditionals and Variables Students consider a scenario in which variables and conditionals may be helpful.		
"I Can ..." Statements Students read the explicit math and CS goals.		

2 Focus 35–40 min

Equivalent Fractions Students consider how to compare fractions represented on two number lines.	*Math Journal 1,* p. 73 (optional)	4.NF.2
Ambling Animals TIPP&SEE Students explore the Scratch project to discover how it uses conditionals and variables.	*Ambling Animals* project; *Ambling Animals TIPP&SEE* journal pages	
Modify and Test Students modify scripts to react to user input.	*Ambling Animals* project; *Ambling Animals* journal page	

"I Can ..." statements

- *I can use variables to store user input.*
- *I can use inequality symbols to compare two fractions.*
- *I can write a conditional statement to decide which fraction is bigger.*
- *I can use an operator block for the condition in an If-Then block.*

Anticipated Barriers

- The layout of the number lines in the Scratch project may confuse some students when comparing fractions.
- Students may have difficulty in recognizing how conditionals and variables might be used in the Scratch project.
- Students may think variables can store multiple values at once.

Student Options

Consider these options for adapting the lesson to your students' preferences:

- Some students may need extra paper to redraw the number lines.

1 Warm Up 5–10 min

Conditionals and Variables

Start with a discussion of conditionals and variables. Remind students about the definition of each term. Ask:

- *What do you remember about Conditional statements?* Sample answer: A conditional or If-Then statement tells you what to do If a certain condition is met.
- *What do you remember about Variables?* Sample answer: A variable holds a value that we might use multiple times in a program. This value can change.

Explain that conditionals and variables are very useful when making decisions or handling different scenarios. As an example, ask students to consider the following scenario: organizing a check-in table at a large school event. There will be so many students coming that you would need to break them into smaller groups, use multiple check-in tables, and have different events or rooms for different groups of students. Ask students to make suggestions for easy ways to sort a large number of students into smaller groups. Sample answers: by age; by grade; alphabetically (by first or last name) Ask: *What might you do as each person came in the door?* Sample answer: Say hello, and ask them what grade they are in or for the first letter of their last name. Then, based on that grade/letter, direct them to the appropriate table. *Once a person gets to the table matching their grade level or last name, what might happen then?* Sample answer: The volunteer at the table would ask for their name, check to find it on their list of names, greet them by name, then direct them to their first activity/room.

Now have students consider how they might direct a computer to handle a similar situation. Ask them to imagine that students check in on a computer instead of with a human. Ask: *If you wanted the computer to remember the student's name and use it again, how could the computer do that?* Sample answer: store it in a variable *What kind of statement could you use to tell the computer how to make decisions about what to do based on what the student enters?* a conditional statement

I Can ...

Display the "I Can ..." statements and remind students that these statements express the goals for today's lesson and can give them clues about what to expect. Carefully read each statement and ask them to use their thumbs to show how true they feel each statement is for them right now.

"I Can ..." statements

- *I can use variables to store user input.*
- *I can use inequality symbols to compare two fractions.*
- *I can write a conditional statement to decide which fraction is bigger.*
- *I can use an operator block for the condition in an If-Then block.*

2 Focus 35–40 min

Equivalent Fractions

WHOLE CLASS | SMALL GROUP | PARTNER | INDEPENDENT

Remind students of their work with equivalent fractions in Lesson 3-3: Number Lines and Equivalence. You may wish to have them look back at their completed Fraction Number Lines on journal page 73. Ask:

- *If we have two fractions represented by points on number lines, how might we tell that they are equal?* Sample answer: We could use a straight edge to vertically line up the fractions.
- *How could we tell if one fraction is greater than the other?* Sample answers: We could see which fraction is closer to 0 or closer to 1. We could use a straight edge to vertically line them up and see which point is further to the right.

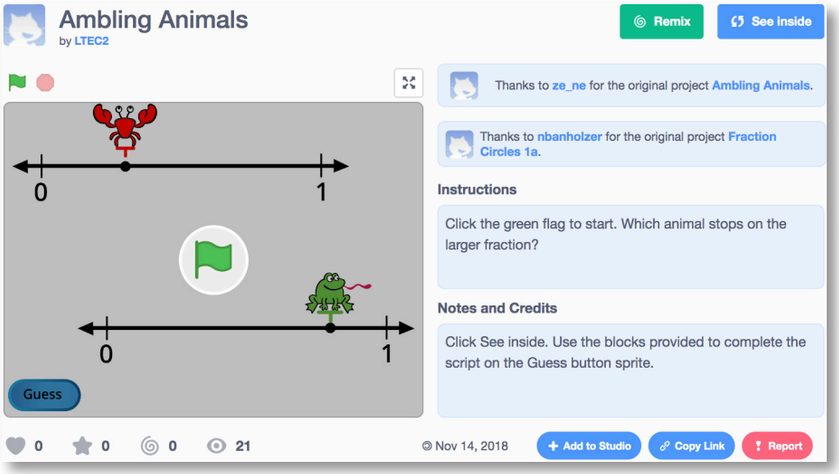
Point out to students that they are able to use straightedges to compare fractions on different number lines (as in Lesson 3-3) when the zero point on each number line is vertically lined up and the number lines have the same scale. Ask:

- *What could we do if we could not physically line up the fraction number lines?* Sample answer: We could use the Equivalent Fractions Rule, or sketch another fraction representation.

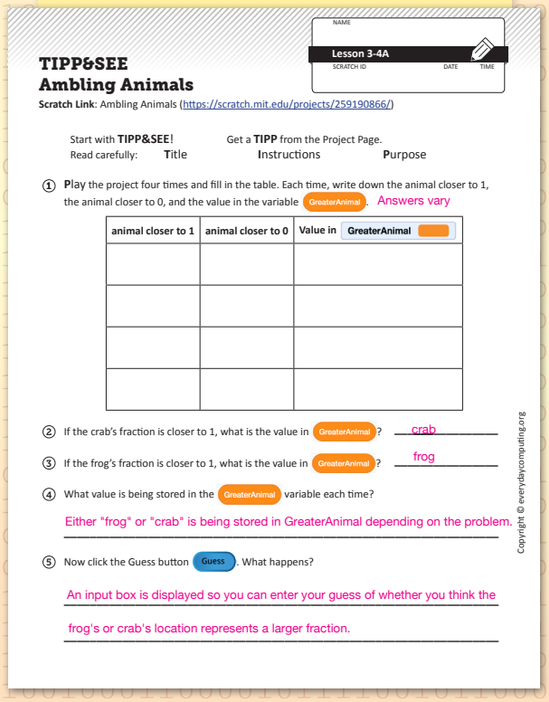
Ambling Animals *TIPP&SEE*

WHOLE CLASS | SMALL GROUP | **PARTNER** | **INDEPENDENT**

Tell students that today's Scratch project will show an example of how to use variables and conditionals to compare fractions. Open today's class project: Ambling Animals (https://scratch.mit.edu/projects/259190866/). Use the TIPP&SEE strategy to explore the project. Tell students you will begin to explore the project with the whole class, then they will further explore the project on their own with their journal pages.



Ambling Animals TIPP&SEE p. 1



NAME

Lesson 3-4A

SCRATCH ID DATE TIME

TIPP&SEE
Ambling Animals

Scratch Link: Ambling Animals (https://scratch.mit.edu/projects/259190866/)

Start with **TIPP&SEE**! Get a **TIPP** from the Project Page.
Read carefully: **T**itle **I**nstructions **P**urpose

① **P**lay the project four times and fill in the table. Each time, write down the animal closer to 1, the animal closer to 0, and the value in the variable GreaterAnimal. Answers vary

animal closer to 1	animal closer to 0	Value in GreaterAnimal

② If the crab's fraction is closer to 1, what is the value in GreaterAnimal? crab

③ If the frog's fraction is closer to 1, what is the value in GreaterAnimal? frog

④ What value is being stored in the GreaterAnimal variable each time?

Either "frog" or "crab" is being stored in GreaterAnimal depending on the problem.

⑤ Now click the Guess button Guess. What happens?

An input box is displayed so you can enter your guess of whether you think the frog's or crab's location represents a larger fraction.

Copyright © everydaycomputing.org

NOTE You may wish to play the project at least twice, so students can see that there is a new fraction comparison problem each time.

There are "Do Not Disturb" scripts to make this part of the project work.

When you **P**lay and press the Guess button, note that the project stops and asks a question. When it stops, ask students: *What just happened? Do you think the project is done?* Sample answer: No it is not done. It asked a question and is waiting for us to answer. Point out the connection between this situation — where the computer is waiting for user input — and the scenario you discussed at the beginning of class, where each person was asked a question at the door. With computers, sometimes a project cannot proceed without user input. Follow the prompt and enter the name of one of the animals. Be sure to draw attention to how the project utilizes the user input later to say something (an animal sound) and give feedback (show the correct answer). Ask:

- *How do you think this project might use conditionals?* Answers vary.
- *How do you think this project might use variables?* Answers vary.
- *Where have we seen variables in Scratch before?* in the Math Chat project *What did they look like?* Sample answer: Variable blocks are orange elongated ovals.

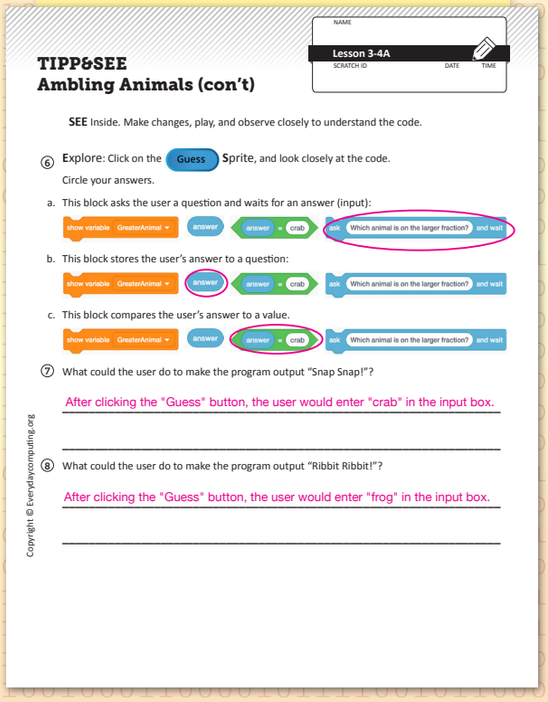
Tell students that the text in the orange box displayed on the bottom right corner of the stage is showing them the value of a variable called GreaterAnimal that is used in this program. Then ask:

- *What user input did this project take?* the name of the animal we thought was on the larger fraction
- *What output did this project give?* an animal sound that matched our input, the correct answer

Distribute and display the Ambling Animals TIPP&SEE journal pages. Have students complete the pages to explore the use of variables and input boxes in this Scratch project. When students are done with the pages, bring them together for a whole-class discussion. Ask:

- *What new blocks did you see in this project?* Sample answer: the "If-Then" conditional block; the comparison block; a blue variable block.
- *What information is being stored in variables in this program?* the user input, the animal with the larger fraction
- *How did you recognize the conditional block?* Answers vary. *Where does the condition go?* in the top part of the block after the if *Where does the action go?* in the middle of the block where there is space *Can we place multiple actions in the condition block?* yes *How?* by placing multiple blocks in the middle
- *Do we know what the user will guess?* no *How can we figure out if the guess is correct or not?* Answers vary. Sample answer: We can have the computer figure out the correct answer and compare it to what the user enters.

Ambling Animals TIPP&SEE p. 2



NAME

Lesson 3-4A

SCRATCH ID DATE TIME

TIPP&SEE
Ambling Animals (con't)

SEE Inside. Make changes, play, and observe closely to understand the code.

⑥ Explore: Click on the Guess Sprite, and look closely at the code.
Circle your answers.

a. This block asks the user a question and waits for an answer (input):

b. This block stores the user's answer to a question:

c. This block compares the user's answer to a value.

⑦ What could the user do to make the program output "Snap Snap!"?

After clicking the "Guess" button, the user would enter "crab" in the input box.

⑧ What could the user do to make the program output "Ribbit Ribbit!"?

After clicking the "Guess" button, the user would enter "frog" in the input box.

Modify and Test

WHOLE CLASS | SMALL GROUP | **PARTNER** | **INDEPENDENT**

Distribute the Ambling Animals journal page. Have students remix the project and work individually to modify their project to complete the challenges on the journal page. Circulate as students work. When students think they are done, have partners test each other's programs. If students' programs do not work as expected, encourage them to work together to debug their projects until they have completed the challenges to their satisfaction.

Wrap Up

WHOLE CLASS | **SMALL GROUP** | PARTNER | INDEPENDENT

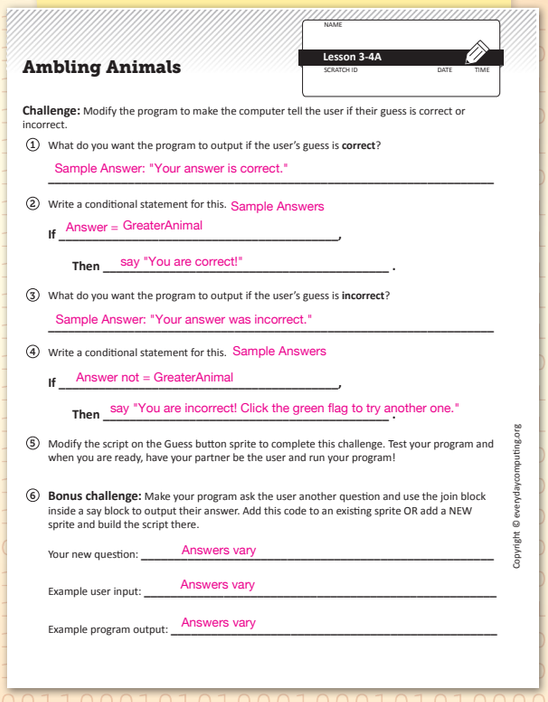
When students have had sufficient time to work, bring them together for a whole class discussion.

Suggested questions:

- *What question did your program ask the user?* Answers vary.
- *How did you use the user input in your remixed project?* Answers vary.
- *What variables did you see and use in this project?* answer; GreaterAnimal
- *What types of values did the variables hold?* Sample answers: Both variables held a word. The answer variable held whatever the user typed; the GreaterAnimal variable held either "frog" or "crab" depending on the problem.
- *What did you learn about conditional blocks in Scratch?*
- *What did you notice about the comparison block(s)?* Sample answer: The comparison block is bright green and is a hexagon.
  How do you think it works? The block checks if the first value is equal to the other value. If the values are equal, the block decides the condition is True; if the values are not equal, it decides the condition is False.
  Do you think there may be other blocks like this that we did not use? There is a block like this for each of the comparison symbols: =, <, >.
- *What else did you discover about Scratch today that we have not discussed?* Answers vary.

Now "I Can ..." Review today's "I Can ..." statements and ask students to use their thumbs to show their opinion of each statement.

Ambling Animals



NAME

Lesson 3-4A

SCRATCH ID DATE TIME

Ambling Animals

Challenge: Modify the program to make the computer tell the user if their guess is correct or incorrect.

1. What do you want the program to output if the user's guess is **correct**?
   Sample Answer: "Your answer is correct."
2. Write a conditional statement for this. Sample Answers
   If Answer = GreaterAnimal,
   Then say "You are correct!".
3. What do you want the program to output if the user's guess is **incorrect**?
   Sample Answer: "Your answer was incorrect."
4. Write a conditional statement for this. Sample Answers
   If Answer not = GreaterAnimal,
   Then say "You are incorrect! Click the green flag to try another one."
5. Modify the script on the Guess button sprite to complete this challenge. Test your program and when you are ready, have your partner be the user and run your program!
6. **Bonus challenge:** Make your program ask the user another question and use the join block inside a say block to output their answer. Add this code to an existing sprite OR add a NEW sprite and build the script there.

   Your new question: Answers vary

   Example user input: Answers vary

   Example program output: Answers vary

"I Can ..." statements

- *I can use variables to store user input.*
- *I can use inequality symbols to compare two fractions.*
- *I can write a conditional statement to decide which fraction is bigger.*
- *I can use an operator block for the condition in an If-Then block.*