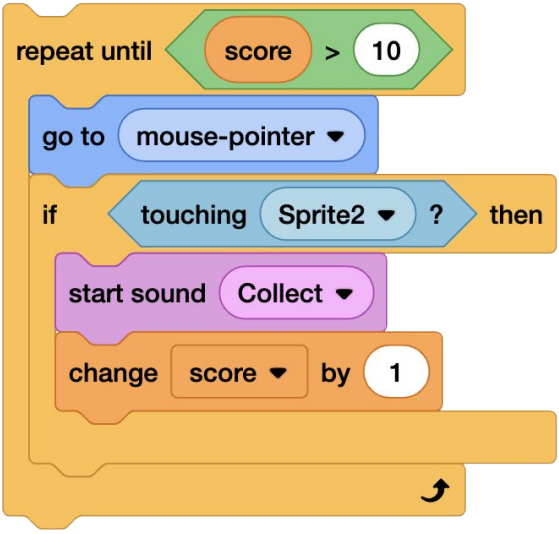
Conditional Statements

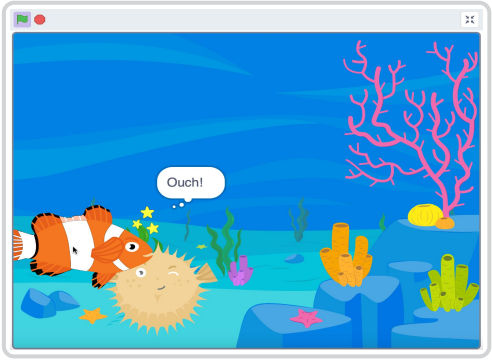


Create dynamic programs that are interactive or offer multiple outcomes

Cards in This Pack

- Fish Game: Code the Fish
- Different Code, Similar Results
- Conditional Statements: “If Then”
- Conditional Statements: “Until”
- Operators in Conditional Statements
- Fish Game: Nested Condtional Statements
- Nested Conditional Statements
- Create a Maze Game
- Create a Math Game

Fish Game: Code the Fish



- Have you ever wanted to create a Scratch program that is interactive or offers multiple outcomes? Let's create a fish game that the user can interact with, and code different animations triggered by conditional statements.
- First, let's code a sprite to be controlled by the user's mouse. Then, use a conditional statement to trigger costume changes when sprites touch.
- See additional cards to code other sprites and trigger additional animations.

Fish Game: Code the Fish

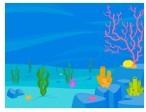
scratch.mit.edu

GET READY

Choose two sprites.

Fish and Pufferfish

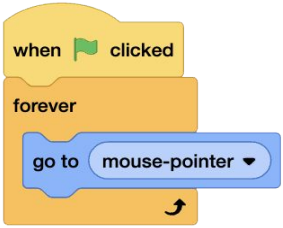
Choose any backdrop.



Underwater

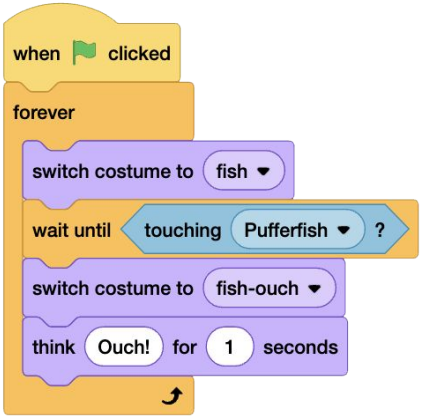
ADD CODE

1. Add code to make the fish continuously follow your mouse.

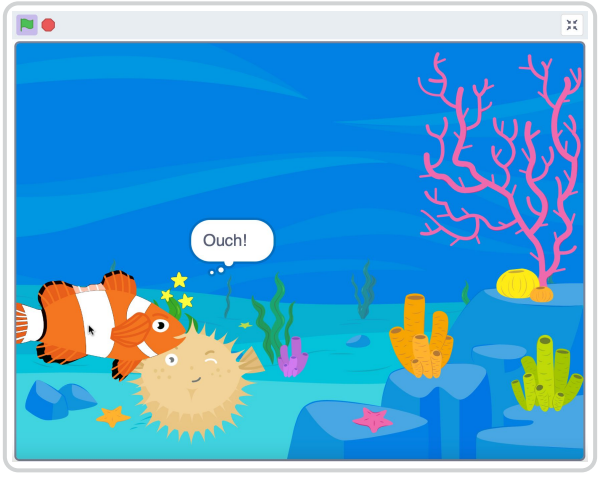


2. Create an additional costume for the fish that will show when it touches the pufferfish. You can duplicate and edit the fish costume to show stars or a funny face, etc.

3. Create a second script that uses a conditional statement to make the fish change costumes when it touches the pufferfish. Test and debug!



Different Code, Similar Results



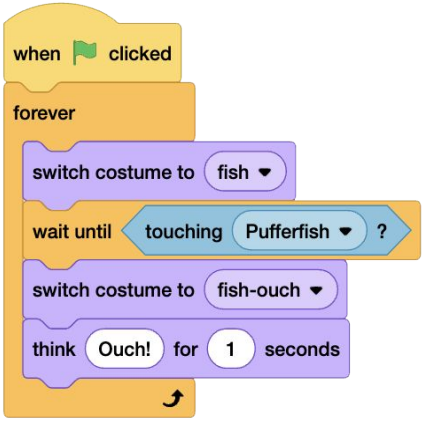
- There is often more than one solution/more than one way to code a program to get a similar result.
- Experiment with using different types of conditional statements (like “wait until” versus “if then else”). What differences, if any, do you notice?

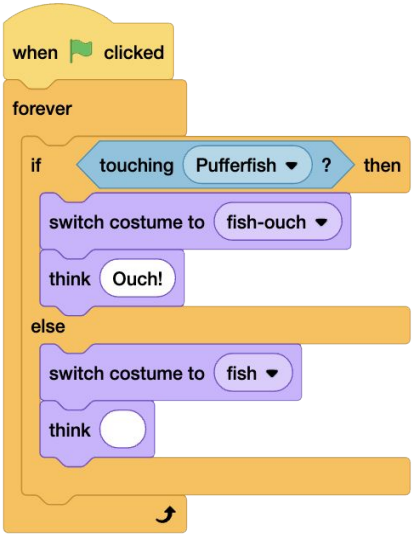
Different Code, Similar Results

scratch.mit.edu

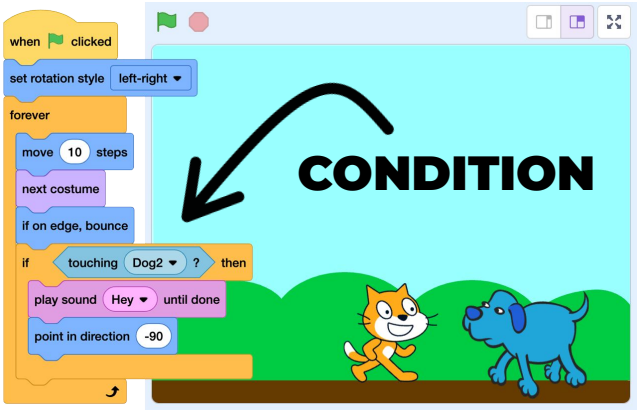
COMPARE CODE

1. The first code stack on the right is the code tried on the “Fish Game: Code the Fish” card.
2. Compare it to the second code stack below that uses an “if than else” statement instead of a “wait until” conditional statement. What is the same and what is different?
3. Test each code stack. Make the fish touch the pufferfish and then move it away quickly, what do you observe happening?
4. Experiment and customize! What solution works best for your game?





Conditional Statements: “If Then”



Boolean blocks that report "true" or "false" are used in conditional statement blocks. Try using:

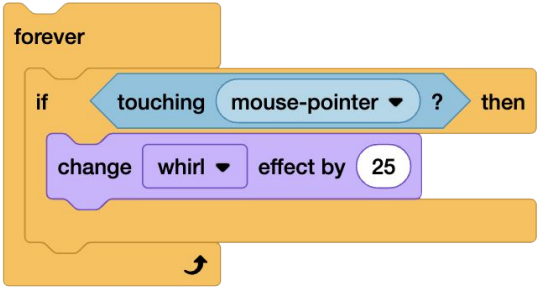
- user actions, such as pressing keyboard keys or mouse positioning or clicking
- sprite interactions (touching another sprite), comparisons (distance between sprites), and touching colors of sprites or backdrops
- data input by users, data stored in variables and lists, or data stored in reporter blocks

Conditional Statements: “If Then”

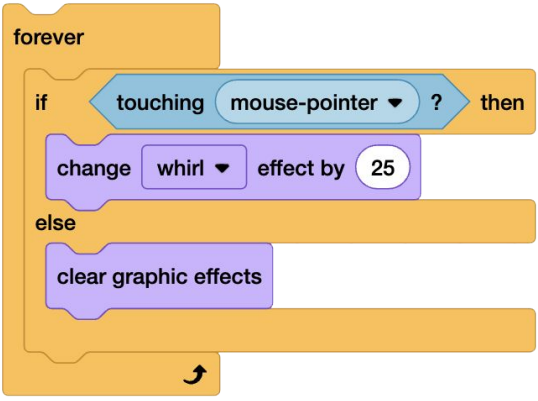
scratch.mit.edu

Experiment with “if (something) is true or false then” conditional statement blocks. What is the difference between these scripts below?

IF THEN

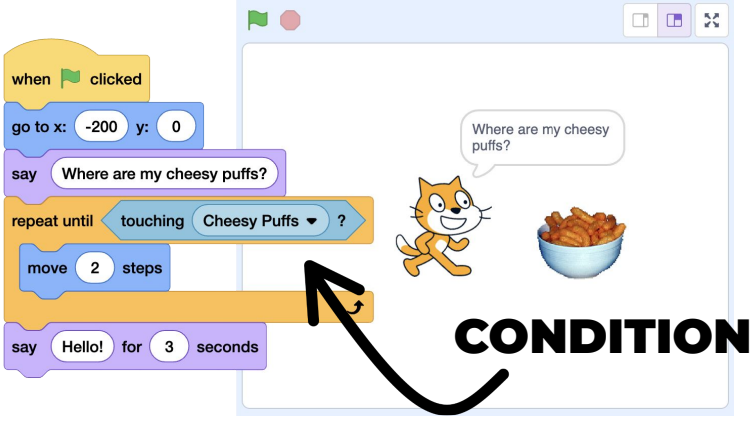


IF THEN ELSE



Try different conditions and customize the results. What is the difference if the condition block is not in a forever loop?

Conditional Statements: “Until”



Boolean blocks that report "true" or "false" are used in conditional statement blocks. Try using:

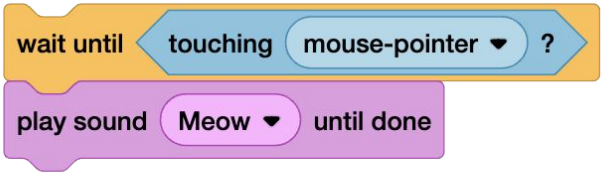
- user actions, such as pressing keyboard keys or mouse positioning or clicking
- sprite interactions (touching another sprite), comparisons (distance between sprites), and touching colors of sprites or backdrops
- data input by users, data stored in variables and lists, or data stored in reporter blocks

Conditional Statements: “Until”

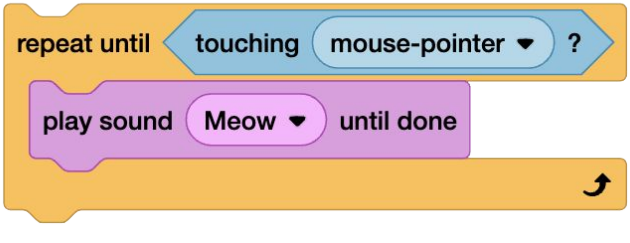
scratch.mit.edu

Experiment with “until (something) is true or false” conditional statement blocks. What is the difference between these scripts below?

WAIT UNTIL

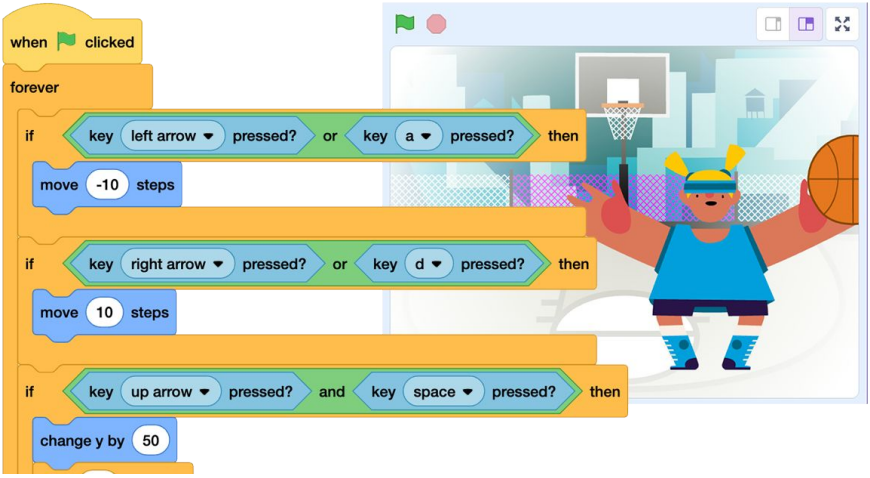


REPEAT UNTIL



Try different conditions and customize the results. What is the difference if the condition block is inside a forever loop?

Operators in Conditional Statements



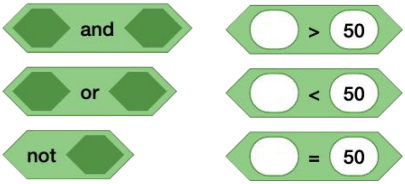
Using operators in conditional statements with sensing blocks or reporter blocks can give you a wider range of conditions to choose from.

Scratch comes with some built-in reporter blocks that store information, like the x or y position of a sprite, the volume in the project, the sprite's size, etc. Looks for these oval blocks under a number of the block categories.

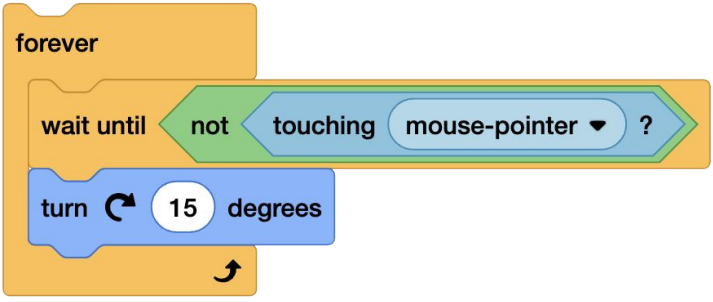
Operators in Conditions

scratch.mit.edu

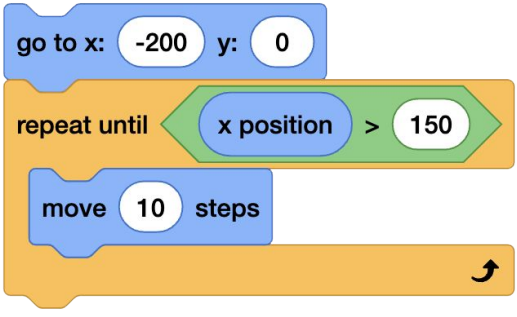
Experiment with different operators in your condition, such as:



"NOT" OPERATOR

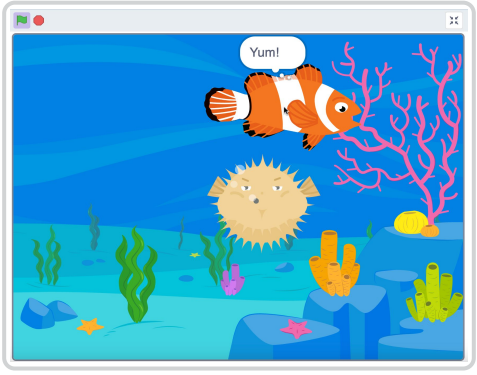


COMPARISON OPERATORS



Try using "and" or "or." What is the effect of two conditions needing to be true, or multiple options that make it true?

Fish Game: Nested Conditional Statements



- Imagine a project where multiple effects can be triggered based on the distance between sprites, colors sprites are touching, or other conditions.
- Nested conditional statements are where one conditional statement is placed inside another. Use them to add additional complexity to your program.
- You could use nested statements to add scoring, or use color as another condition.

Fish Game: Nested Condition

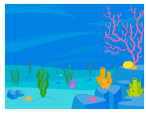
scratch.mit.edu

GET READY

Choose two sprites.

Fish and Pufferfish

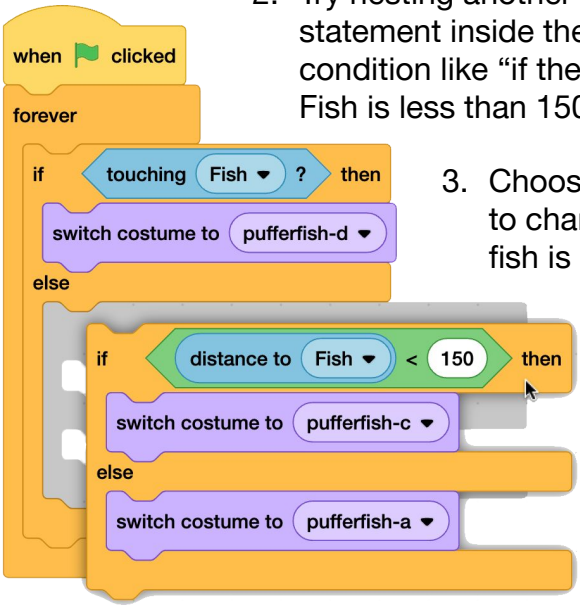
Choose any backdrop.



Underwater

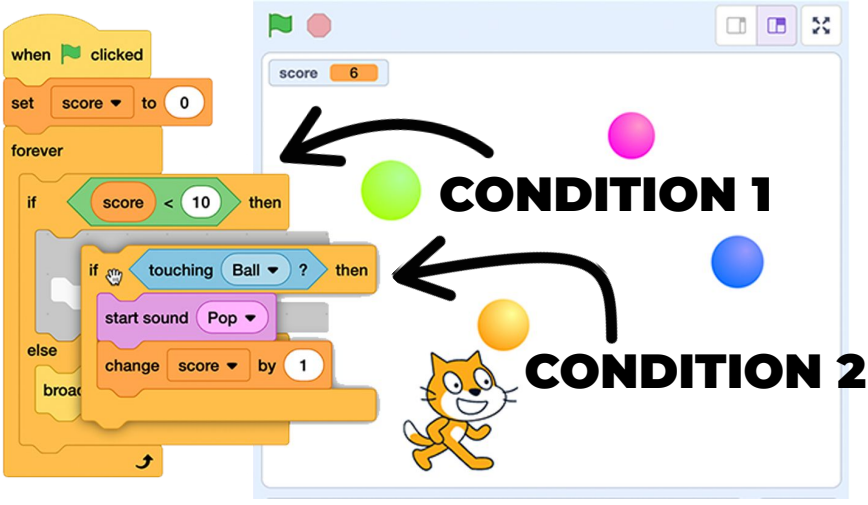
ADD CODE

1. Add an “if then else” block to the pufferfish so it changes costumes when it touches the fish.
2. Try nesting another “if then else” statement inside the first. Add a condition like “if the distance to the Fish is less than 150 pixels then…”
3. Choose a third costume to change to when the fish is close.



Test the code. What is the difference if the sequence is reversed and it checks the distance first?

Nested Conditional Statements



Nested conditional statements are where one conditional statement is placed inside another. The first condition is checked and, depending on if it is true or false, the program may then move forward to check the nested condition inside.

This means that the sequence of the nested statements is important.

Nested Conditional Statements

scratch.mit.edu

Experiment with nested conditional statements using two or more “until (something) is true or false” or “if (something) is true or false then” conditional statement blocks.

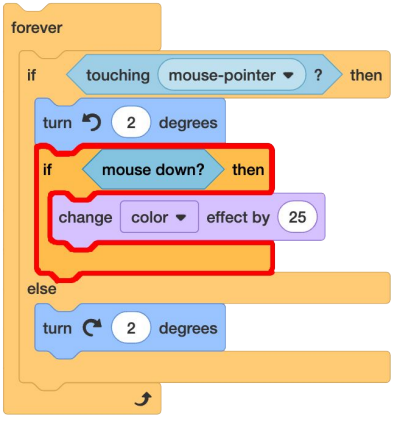
GET READY

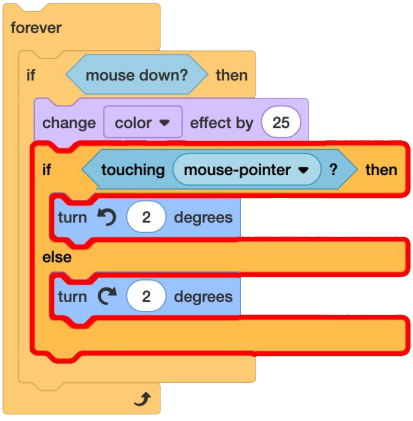
Choose two sprites.

Cat

Dog2

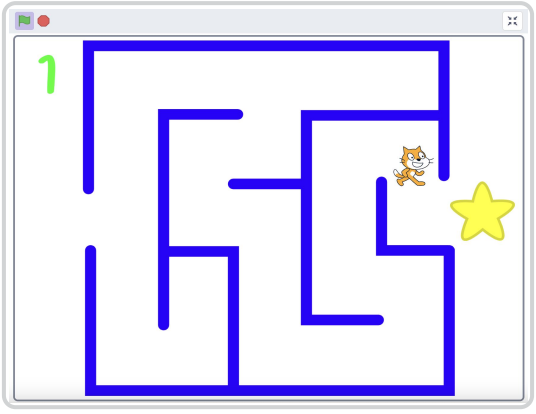
ADD CODE





These code stacks use the same blocks. The difference is the sequence (which conditional statement is nested). Code each sprite with a different stack. Click the mouse when touching each sprite and when not touching each sprite. What differences do you notice? Customize and experiment!

Create a Maze Game



- Use the line tool to create maze walls with color. Use the “touching color” condition to stop sprite movement through the walls. To select the color, use the eyedropper tool.

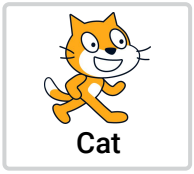
- Make sure your sprite can fit through all passages and around all corners.
- Bonus: Try using another conditional statement to code a winning animation when the end is reached!

Create a Maze Game

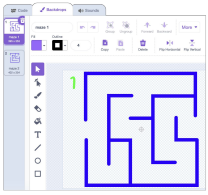
scratch.mit.edu

GET READY

Choose any sprite.

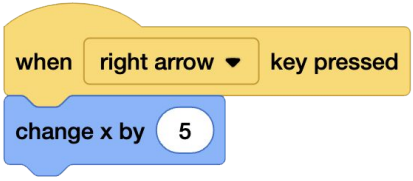


Create maze backdrops using the paint editor line tool.



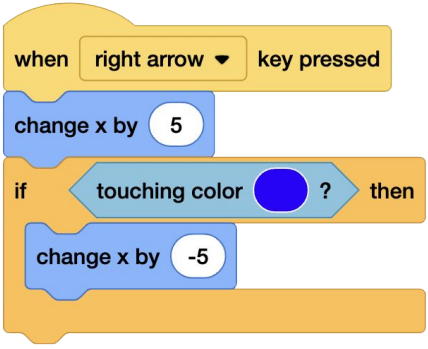
ADD CODE

1. Add code to move the sprite up and down when arrow keys are pressed by changing y, and move left and right by changing x. Use positive and negative numbers.



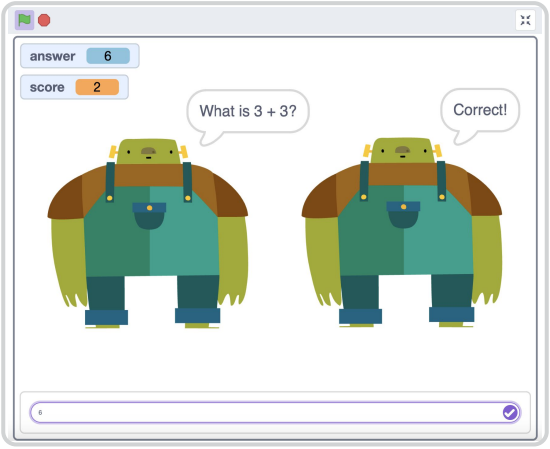
2. Add an “if then” conditional statement to reverse the move if touching the maze wall color.

3. If necessary, add code to change maze backdrops.



You could add a sprite at maze end and use a conditional statement to change the backdrop when touched.

Create a Math Game



- Choose addition, subtraction, division, or multiplication questions to ask a user.
- Use a conditional statement to check the answer given, and have the program respond differently if the answer is correct or incorrect.
- Use a score variable to record points.
- Bonus: Try using My Blocks to make your program more efficient to write and edit.

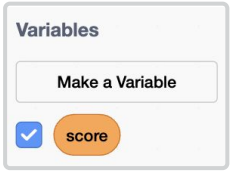
Create a Math Game

scratch.mit.edu

GET READY

Choose any sprite.

Create a "score" variable



ADD CODE

1. Use the ask block to ask a math question.
2. Add an "if then else" conditional statement block. Then, use an equals operator block to set the condition to "answer" equals the correct answer.
3. Add blocks to say "Correct" if the answer is correct, else "Wrong." Customize with sounds or other effects.
4. Use variable blocks to set and change the score.