

Tutorial

Natsiq Hunter

Level: Grade 5–8 • **Topic:** Game Design • **Subjects:** Science & Technology

If you are looking for an accessible online version of this content, please visit the [Pinnguaq website](https://pinnguaq.com/learn/natsiq-hunter) (link: <https://pinnguaq.com/learn/natsiq-hunter>).

About the Author

Mac Pavia

Mac is an educator at the Iqaluit Makerspace. Having worked with the many Iqalummiut youths located in Iqaluit, during his time at the space Mac has developed a passion for getting youth excited about STEAM. Recently taking a major step in his own life Mac attended his first year of college at Mohawk's Fennell campus located in Hamilton, Ontario. There he is studying in a 3-year graphic design program.



Introduction

In this tutorial, participants will learn how to use if/else statements to develop a random position algorithm for a game.

Vocabulary

- **Variable:** a way for a computer to store data for later use
- **Vector:** graphics that use math and straight lines to create graphics that can be scaled to any resolution without loss of detail
- **Natsiq:** Ringed seal
- **Sprite:** an object that is controlled by the code block
- **Algorithm:** a set of rules or steps to be followed in problem solving

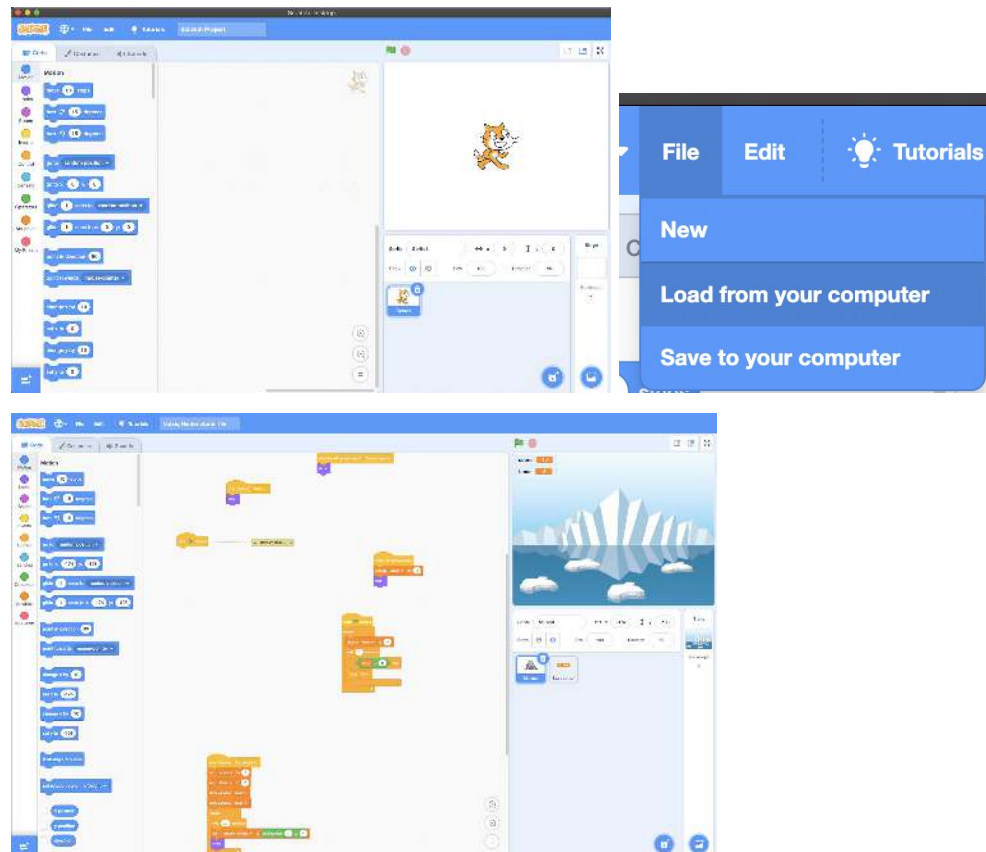
Materials

- Download the **Natsiq Hunter asset folder** (link: <https://pinnguaq.com/app/uploads/2020/06/pinnguaq-natsiq-hunter-assets.zip>)

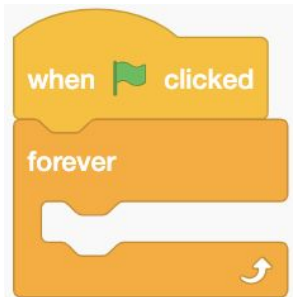
Computer Activity

In this tutorial, we are going to be building a random placement for Mr. Seal in Natsiq Hunter.

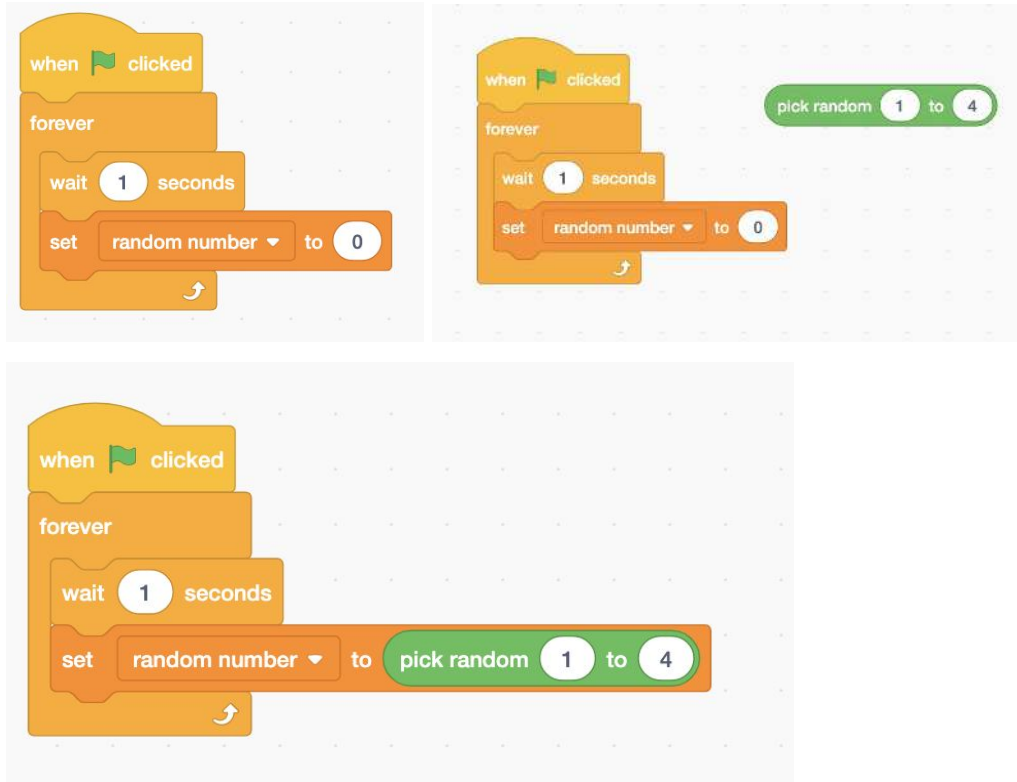
1. To start,
 - Open Scratch.
 - Navigate to the File drop-down menu in the top left corner.
 - Click on the 'load from your computer' option, and load the Natsiq Hunter starter file. In this file, you will find all the backdrops, sprites, and most of the code already placed. But you will need to code to the random position for Mr. Seal.



2. Now let's place an Event block; the Event block will define what triggers the code to run. In this case, we're using a *when the green flag is clicked run the code underneath*; this tells Scratch when to begin something. Attached to that Event block, we're also going to place a forever loop. We want Scratch to run this script continually. If we were to only put the rest of the script beneath it, Mr. Seal would only randomly go to one spot once. That wouldn't make for a hard game, so let's place it in a forever loop.



3. Now that we have our basic structure defined, let's start building the logic behind Mr. Seal's random placement. For this, we are going to use a variable; this variable will let us use the *pick random function* in the operator's tab. We are going to have Scratch pick a random number from one to four; this random number is going to be used to check the conditions of an if/else statement. We will also add a *wait 1 second*; this will slow the script down, so it isn't super difficult to play the game, if you want to adjust the difficulty of the game, you will adjust the *wait* duration to speed up the number of times Mr. Seal changes places.



4. Now that we have our random number generator set up, we need to define the logic behind the random numbers. Basically, what is Scratch going to do with the random numbers now? For this, we are going to be using *if/else* statements. These statements use a set of conditions to evaluate and execute specific code. An example of an *if* statement would be:

if I finish my homework, then I get to play videogames.

Another type of *if* statement, and that is the *if/else* statement. An example of this is:

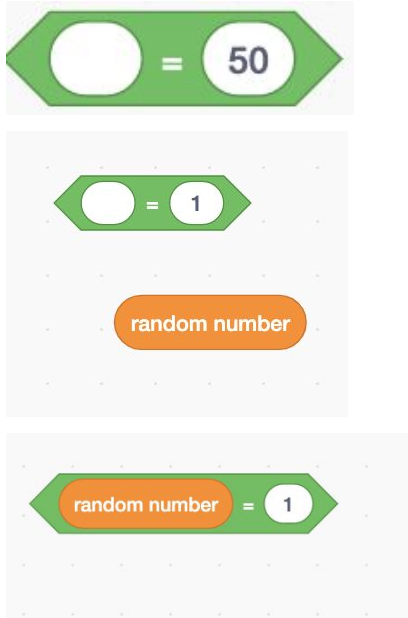
if it's the weekend, then I can play video games, else I need to do my homework.

These statements help programs like Scratch to figure out what to do when given a set of conditions.

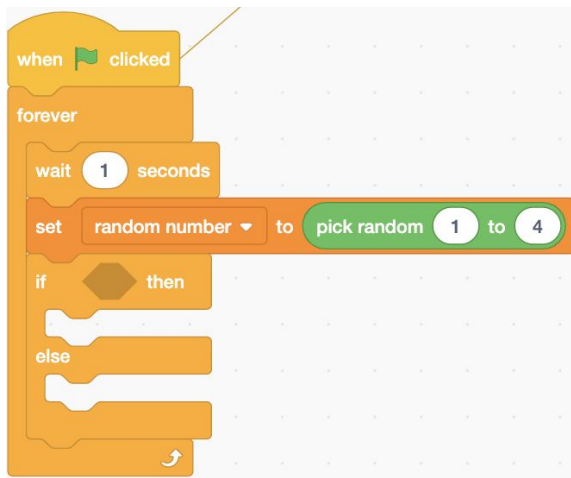
Let's build these statements now, first navigate to the Operator's tab on the far left side of the screen, grab an equal statement and change the number 50 to a one. Inside the equal statement place, the variable *Random number*. This variable will let us check its value against the condition's assigned value. An example would be:

if random number is equal to one, then run the code below.

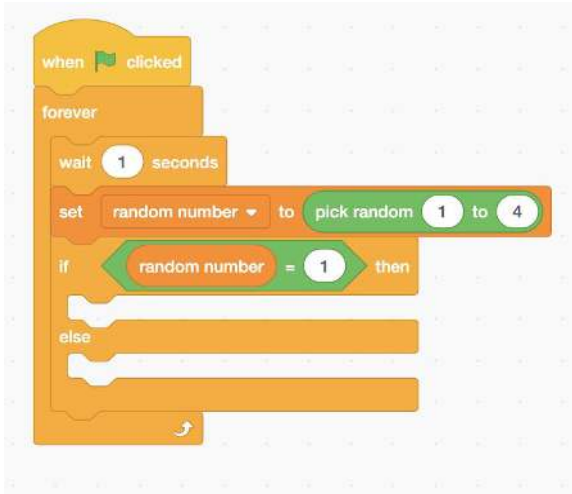
This is how Mr. Seal knows where to go on the stage.



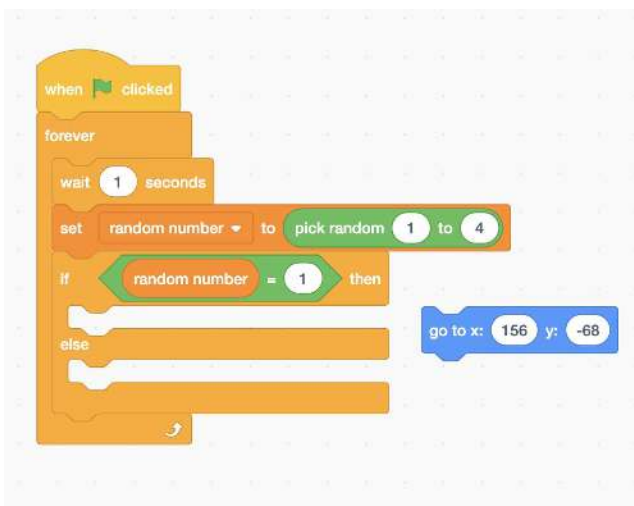
- Now that we have defined our condition, we need to place it in an if/else statement. Navigate to the far left again and find the Control tab. There, you will find an if/else statement. Place it inside the forever loop.

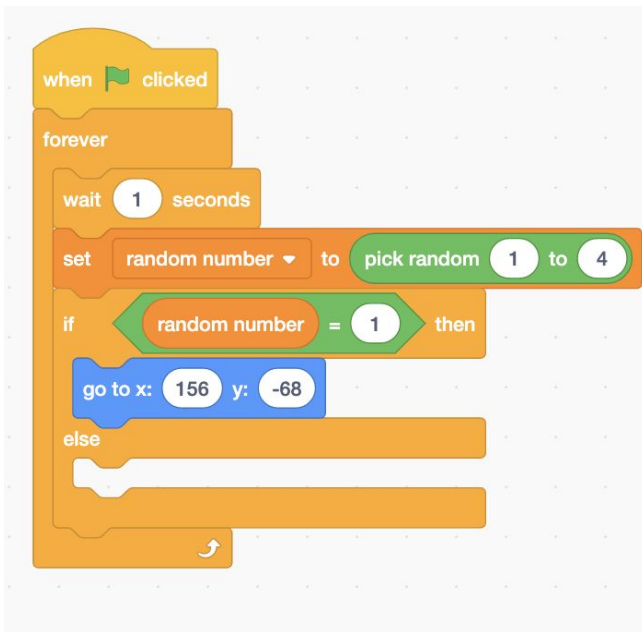


Place the condition you made in step four inside the if/else statement.

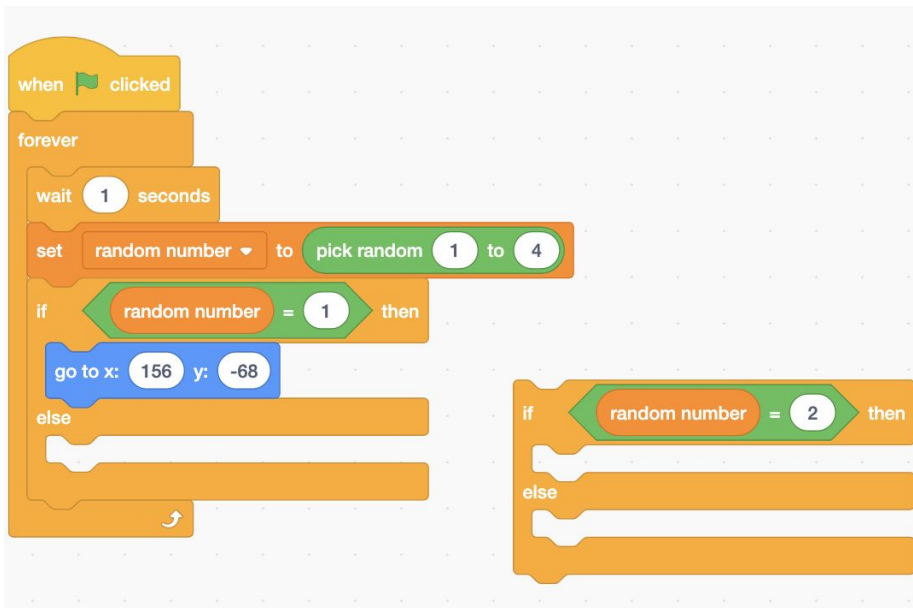


Now we need to define what happens when the condition evaluates to true, for this we are going to use a go to X: Y: Place it inside the if statement. Also, make sure to change the values inside it to the ones in the picture below.



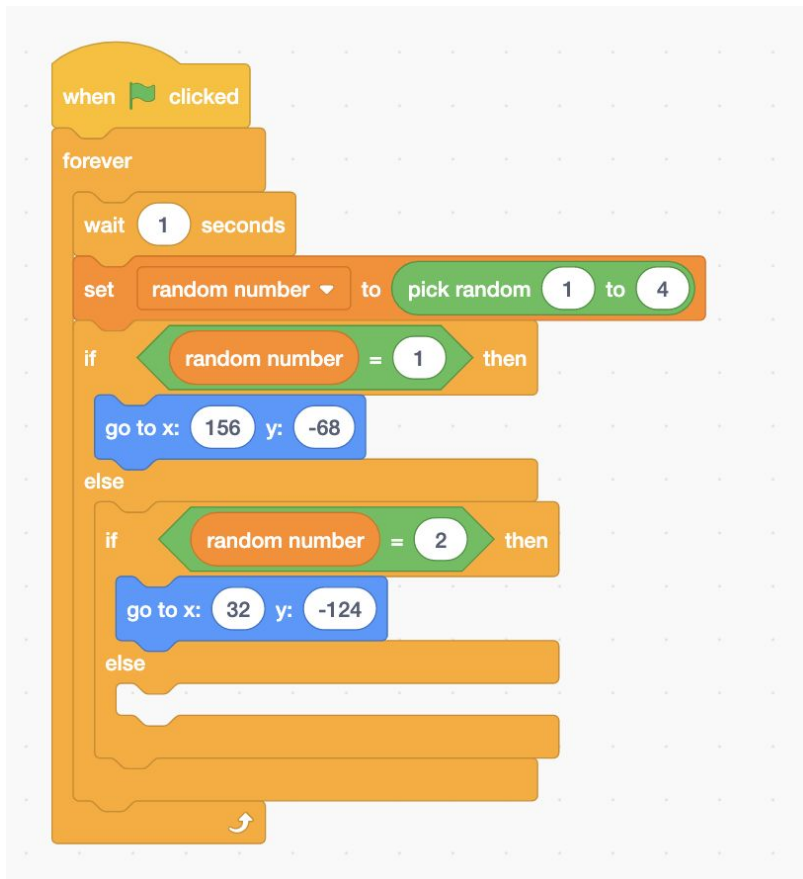


- Next, we are going to repeat the same steps above two more times but this time we are changing some of the values.

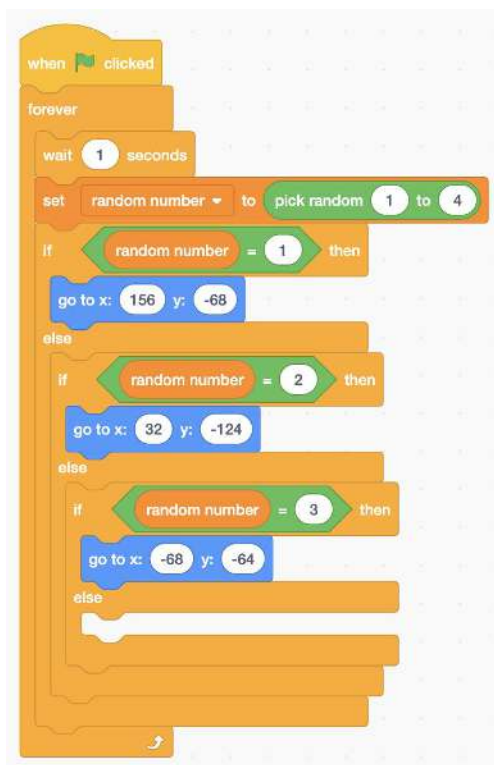
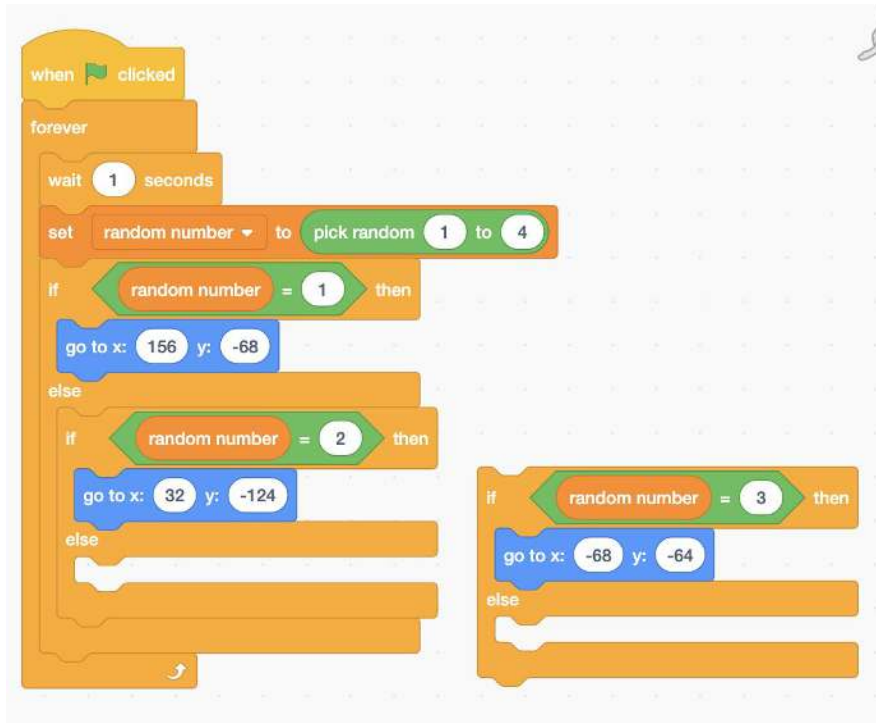


```
when clicked
  forever
    wait 1 seconds
    set random number to pick random 1 to 4
    if random number = 1 then
      go to x: 156 y: -68
    else
      if random number = 2 then
      else
```

```
when clicked
  forever
    wait 1 seconds
    set random number to pick random 1 to 4
    if random number = 1 then
      go to x: 156 y: -68
    else
      if random number = 2 then
        go to x: 32 y: -124
      else
```

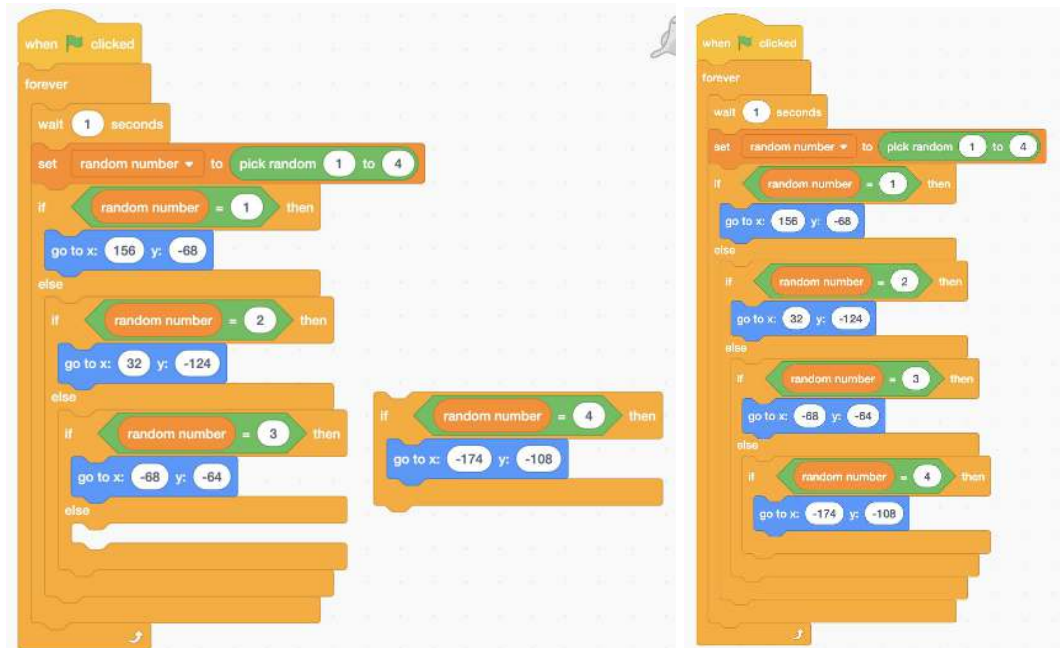
7. Let's build the third if/else statement. Make sure to copy all of the values in the pictures below. Also make sure to double-check your code against the pictures in this tutorial.



8. **Checkpoint:** make sure to double-check all of your code, are the values inside the code blocks correct? Does your code look like the pictures so far? *If your code is placed correctly Then proceed to the next step Else look back at the previous steps to correct your*

bugs. This is another example of an *if/else* statement, in this case the condition you were checking is whether or not your code is placed correctly, if that condition evaluates to true then you proceed to the next step, if that condition evaluates to false then you look back at the previous steps to correct your bugs and then reevaluate this *if/else* statement.

- To finish the code off, we need to place a final *if* statement, this is the last statement for the fourth position.



Conclusion

Using this simple algorithm, what are other ways you can control Mr. Seal? What are some other uses of if/else statements? Maybe you can think of some if/else statements that you might use around your home.

We want to see the awesome things you're creating! Take a photo or video and share your work with us by emailing media@pinnguaq.com or tagging @pinnguaq on [Facebook](#), [Twitter](#), or [Instagram](#). Don't forget to include the hashtag #LearnWithPinnguaq! You can also upload your project to the Pinnguaq Studio (link: <https://scratch.mit.edu/studios/26567463/>).

Resources

- **Inuktut Tusaalanga:** use this website to learn more about Inuktitut (link: <https://tusaalanga.ca/node/2502>)