First, plug the Finch into your computer. Use the port above the tail.

Here's a tip! Hold the cord out of the way when the Finch is moving, so it doesn't get tangled!

On a Mac or PC, open "BirdBrain Robot Server."

On a Chromebook, open "Finch Connection App."

A window will appear, letting you know if your Finch is connected.

Click "Open Scratch"
After you've made your choice, your computer should open Scratch 2.0.

The big open area on the right is called the **scripts area**...any commands that you want your robot to do will wind up here.

To the left are all the blocks. These are the pieces that you will drag out to the scripts area. Each one has a specific purpose.

Above the blocks, there are different menus, or categories. All of the blocks that are specific to Finch will be in the:  

"**more blocks**" category

To test any of these blocks out (for now), drag them into the scripts area and click on them. Drag them back to delete them.

If you forget where the Finch blocks are, there’s a note to tell you.
To use the motion blocks, we want to write programs that Scratch will understand. For example, let's tell it:

"Move forward at medium speed for one second, then stop."

But in order to do that, we need more than just the motion blocks. We'll also need the dark yellow event blocks and the light yellow control blocks.

In the Events menu, you'll find the "When space key pressed" block. Click on the black triangle in the block to choose a different key.

In the Control menu, you'll find the "Wait 1 second" block. You can type any number into the white circle.

There are four main types of Finch-specific blocks in Scratch. They are all in the More Blocks menu.

Motion Block

Looks Block

Sound Block

Sensor Blocks

First, we'll try out the motion blocks and get the Finch's wheels rolling!
Next, we need to tell the finch how long it should go. We can do this by using the "wait" block. You can make the finch move as long as you'd like.

Next, we need to set the left and right wheel speeds. They range from 0 to 100. We'll go with 50.

However, we need to add another "move finch" block to really end the program. By setting the motors to zero, we tell the finch to stop moving after the "wait" time has passed.

What happens if you insert negative values into the "move" block?

Moving forward (or backward) is helpful, but you may need to turn the finch. To do this, the wheels need to move at different speeds.

To have the finch spin in place, one wheel needs to move backwards (have a negative value).

To have the finch move in a curved line, one wheel needs to move slower than the other.
Now we can use other features of the finch, such as beak color and the buzzer. Remove the previous program by dragging the motion blocks back over to the menu.

The block that controls beak color is the "finch led" block. Pull it out and snap it to the control block.

Try entering different values (0-100) in the red, green, and blue slots. What happens when you mix colors?

To turn the beak off, you'll need to set all the values to zero.

To make the finch buzz, head to the sound menu and pull out the "buzzer" block.

Try messing around with the values. The first controls pitch, which can be set from 50-20,000Hz. The second is the length of the sound in milliseconds.

Try making a program with "buzzer" blocks separated by "wait" blocks. Can you make the finch sing?

Now you can make your finch move, alter its appearance, and even sing a song! But what if you want the finch to react to its environment? The finch is equipped with sensors that allow you to do just that.

The acceleration sensor reports on the finch's orientation--where it is in space.

The light sensors detect light, allowing the finch to determine whether it is dark or bright.

The temperature sensor detects the temperature around the finch.
The obstacle sensors detect whether there are objects in front of the finch.

The orientation sensor detects the position of the finch - beak up, upside down, etc.

The obstacle blocks and the orientation blocks are both a little different! Obstacle blocks give results in **boolean** (true or false), and the orientation block delivers a **string** (words). All other sensors give **numbers**.

Let's build a program using the sensors.

If the room is dark, turn the finch's beak on. If not, keep the beak off.

This program is a little more complex than the last one. We will need to use a few more of the control blocks, specifically the "if, else" block and the "forever loop" block.

The "if, else" block is like a little sentence:

**If** (some condition is met), then do this **action**, or **else** do this other **action**.

But what goes in the first part of the "if, else" block?

With the green **operator** blocks, we're going to look at two symbols you might remember from math class:

- `<` (less than)
- `>` (greater than)
IF you put in something **FALSE**...like that 9 is more than 10, it will tell you when you click on it!

**WE CAN PUT OUR SENSORS IN THESE BLANKS!** For example, this is saying “THE LIGHT HITTING OUR SENSOR IS LESS THAN 20”!

**IS THAT TRUE OR FALSE? DEPENDS ON HOW BRIGHT THE ROOM IS!**

**NOW, WE NEED TO SET THE FIRST ACTION FOR THE "IF, ELSE" BLOCK. DRAG THE BEAK LED BLOCK INTO THE FIRST SECTION OF THE "IF, ELSE" BLOCK.**

**WE CAN HAVE THE SENSOR CHECK THE LIGHT IN THE ROOM - TO DO THAT, WE’LL PUT OUR OPERATOR INSIDE THE "IF, ELSE" BLOCK DIAMOND.**

**TIP: SNAP ON A “WHEN SPACE KEY PRESSED” AT THE TOP TO BE AN ON SWITCH!**

**NOW YOU HAVE THIS PROGRAM:**

**IF** THE LIGHT HITTING OUR SENSOR IS LESS THAN **20**, TURN THE BEAK RED!

**DOES IT WORK? IF NOTHING CHANGES, TRY TURNING OFF THE LIGHTS IN THE ROOM AND TRYING AGAIN!**

**BUT, WHAT IF YOU TURN THE LIGHTS BACK ON? HOW DO YOU GET THE LED TO TURN OFF WHEN IT’S NOT NEEDED?**

**WE WANT TO CREATE A PROGRAM THAT SAYS: IF THE LIGHT HITTING OUR SENSOR IS LESS THAN **20**, TURN THE LED TO **100**.**

**OTHERWISE, TURN THE LED TO ZERO.**

**THE ELSE IS THE SECOND PART OF THIS STATEMENT. ELSE APPLIES WHEN THE FIRST IF CONDITION IS NOT MET.**
As it stands, that statement will only check the room once. To have it constantly check, put everything inside a "Forever" block. The "Forever" block can be found in the control menu.

Adding the **else** action is easy — it's just like adding the first **if** action.

This time, we'll set the LED intensity to zero — completely off.

Let's test it again! Try the flipping lights on and off.

**Be sure to save your work!**

**Remember to make a note of the location where you are saving the file.** Try to name it something related to the project at hand.

**Go to the File menu and choose "Save As".**

These are just a few of the things you can do with the Finch robot! Check [www.birdbraintechnologies.com](http://www.birdbraintechnologies.com) for more ideas, lessons, and activities.