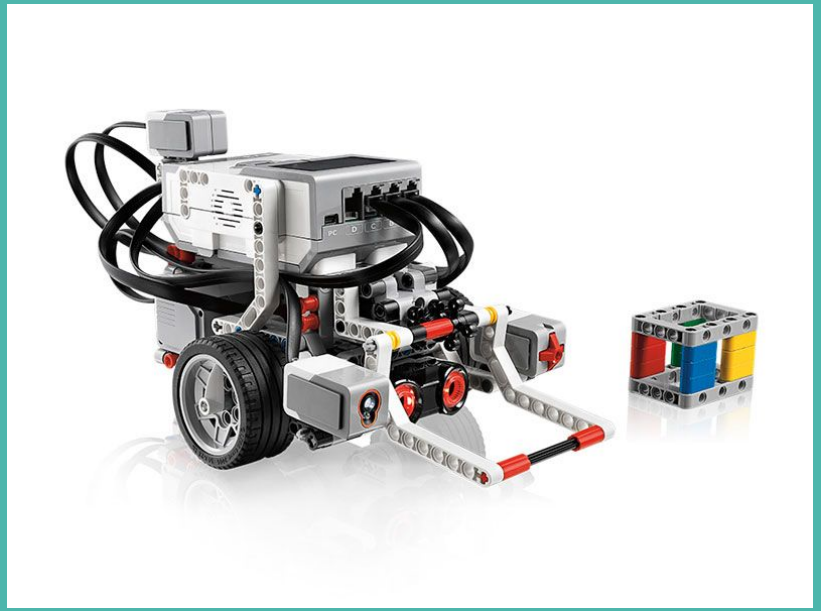
R.I.S.E. Summer Camp

Raptors Inspiring Science and Engineering

Before We Start...

Be sure to write everything that sounds interesting to you in your ***ENGINEERING NOTEBOOK***

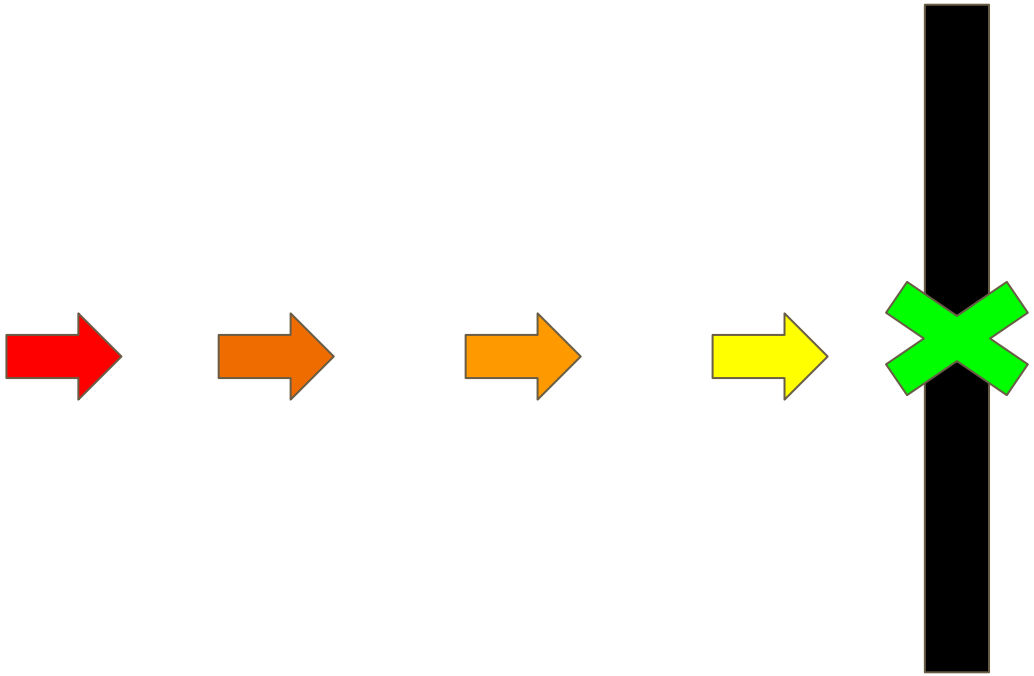
Lets Go Over the Challenges Again!



Land Here!

This is the Land Here Challenge! The point of the Challenge is to stop at the black line!

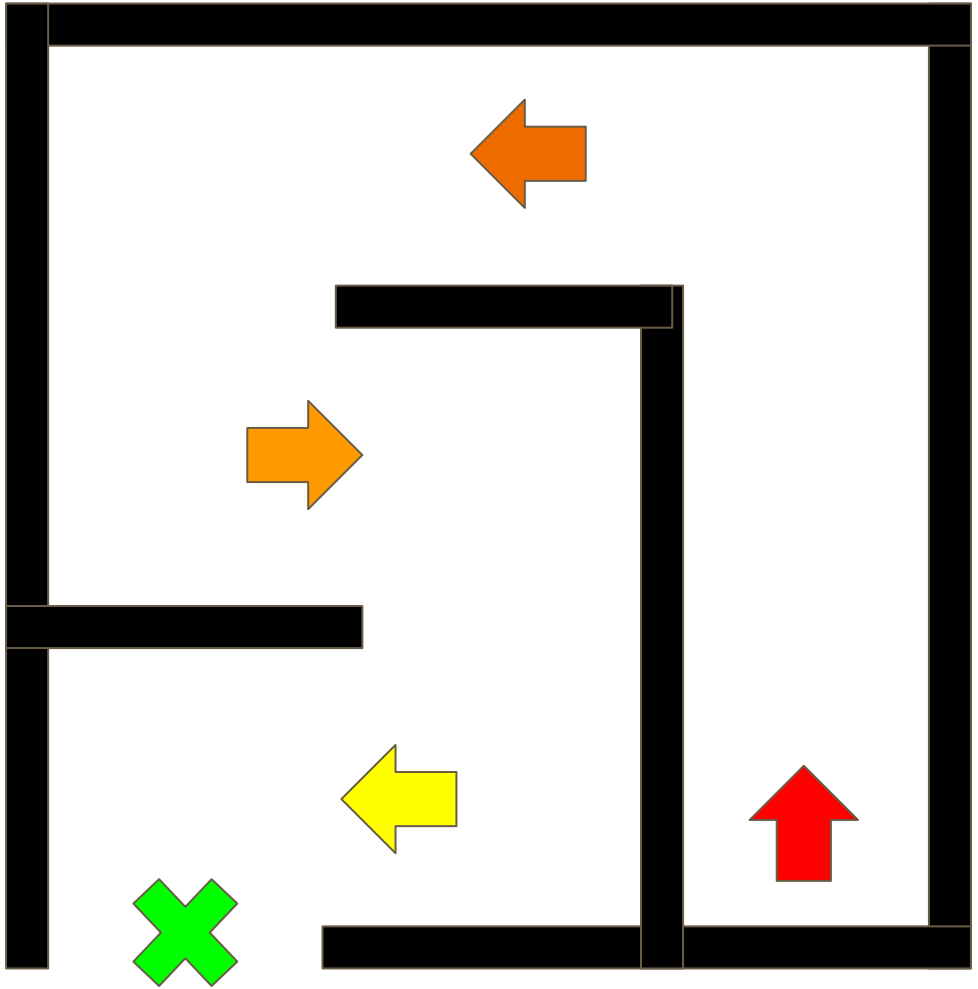
The robot will start at a random position directly in front of the black line. The robot must stop around the black line.



Labyrinth!

This is the Labyrinth Challenge!

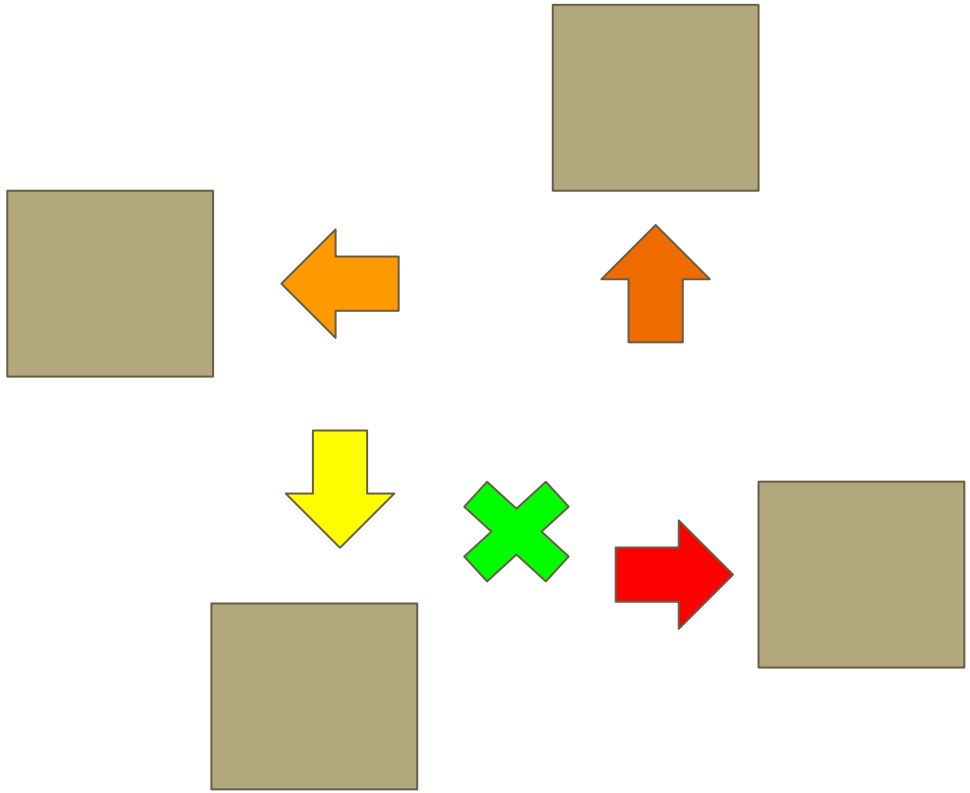
In Labyrinth challenge, your robot must navigate out of the labyrinth without touching any of the walls.



Touch Mad!

This is the Touch Mad Challenge!

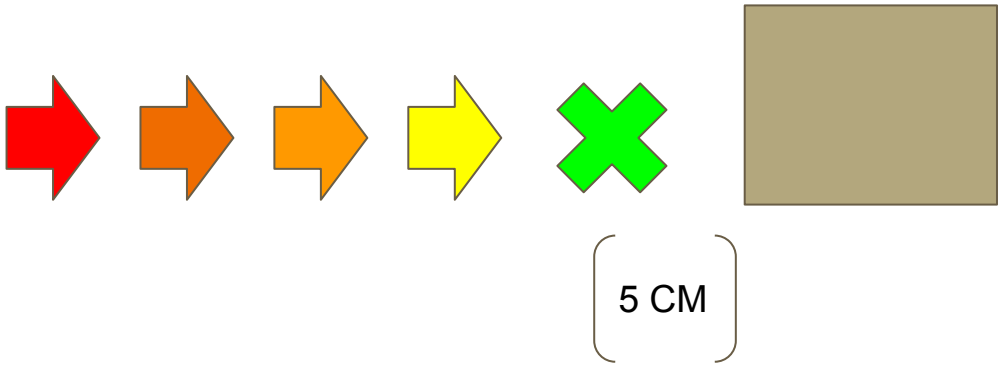
The robot must touch each of the boxes, move backwards, and turn to go to the other box.



Stop at 5!

This is the Stop at 5 Challenge!

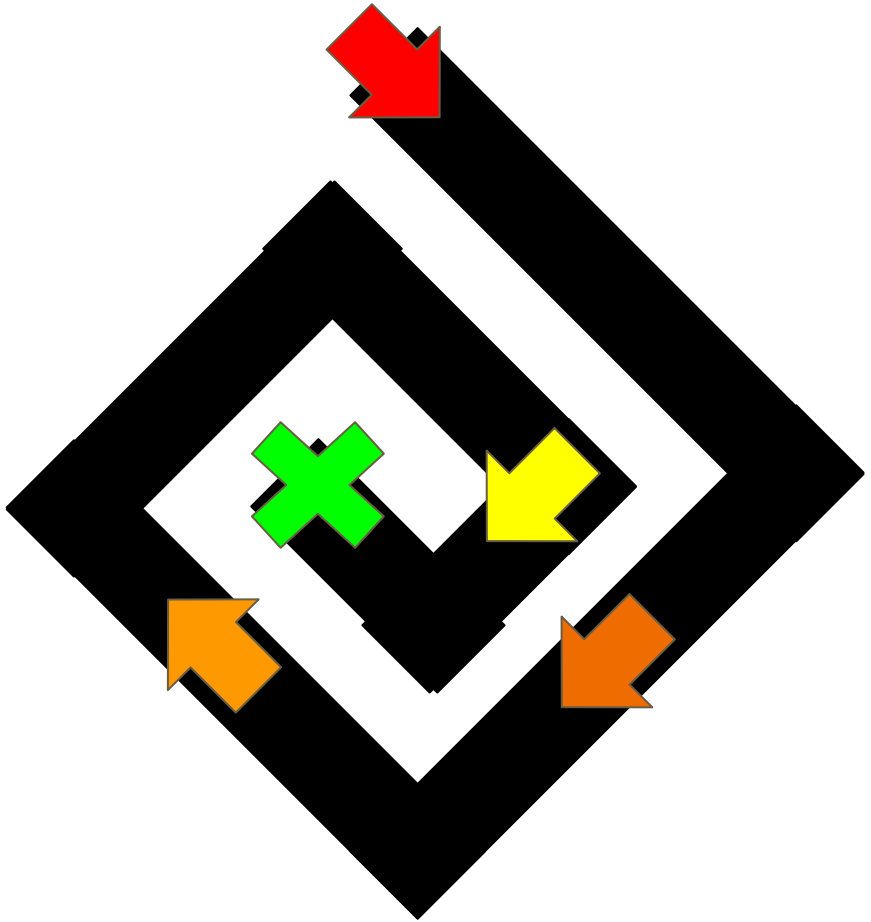
For this challenge, your robot must stop 5 Centimeters away from the object, however, the robot will start at a random distance in front of the box.



Follow the Line!

This is the follow the line challenge!

Your robot must follow the black line into the center.



Let's Continue With Programming!

Programming Part 2

Yesterday, we learned how to program motors.

Today, we'll learn how to program Sensors!

But Before we can do that, we need to learn how to program **Statements**!

Statements and Logic

Statements are pieces of code that test **Logic**.

Here is a Verbal Example of a Statement!

If it is raining outside, I will bring my umbrella.

If if it is Not Raining outside, I will leave my umbrella behind.

Lets see some examples of this this in Pseudo Code!

```
IF IT IS RAINING
        BRING UMBRELLA
ELSE
        DON'T BRING UMBRELLA
```

```
IF IT IS COLD OUTSIDE
        WEAR A SWEATER
IF IT IS HOT OUTSIDE
        WEAR A T-SHIRT
ELSE
        WEAR NORMAL CLOTHES
```

Sensor Logic

Sensors Work in a similar way! They gather data and send it to the **CPU** (EV3) where it is **Computed**.

When something is computed, some logic or operator is being tested on it!

Let's see how a sensor would work in Pseudo Code:

The top pseudocode is to have a robot turn to look for a black line! Once the robot sees the black line, it will stop

```
IF COLOR SENSOR REGISTERS 0
        STOP ALL MOTORS
ELSE
        TURN TO THE RIGHT
```

Can you figure out what the following pseudo codes do?

```
IF ULTRASONIC SENSOR REGISTERS 10
        TURN TO THE LEFT
ELSE
        GO FORWARD
```

```
IF TOUCH SENSOR REGISTERS 1
        MOVE BACKWARD
ELSE
        MOVE FORWARD
```

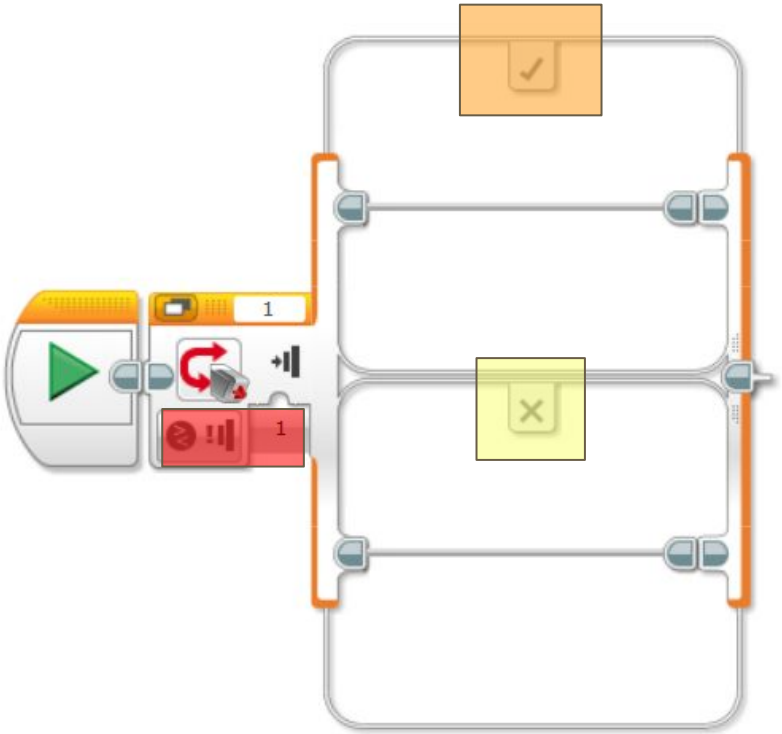
```
IF GYRO SENSOR REGISTERS 20
        MOVE BACKWARD
ELSE
        MOVE FORWARD
```

Sensor Programming

Lets see some examples of Logic in Mindstorms. We will be putting the **Switch Statement** in our IDE.

Let's look at this Statement more carefully!

- The **Condition**
  - What gets Tested
- **IF Statement**
  - What gets run if the condition is true
- **ELSE Statement**
  - What gets run if the condition is false

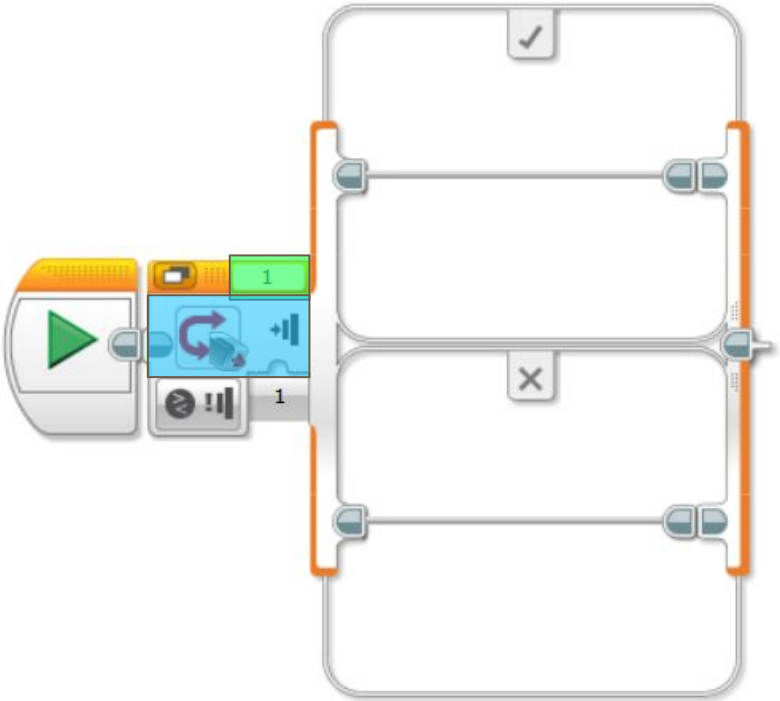


Sensor Programming

Let's look at the statement a little more carefully!

- Tells the IDE what Port the Sensor is in
- Tells the IDE what Sensor to expect

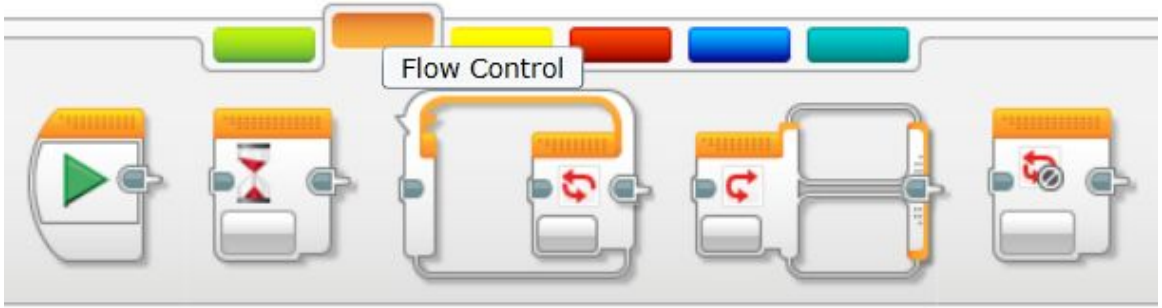
Let’s Practice writing logic with our Switch Statement!



Where to Find Statements

Statements can be found under the Flow Control Tab.

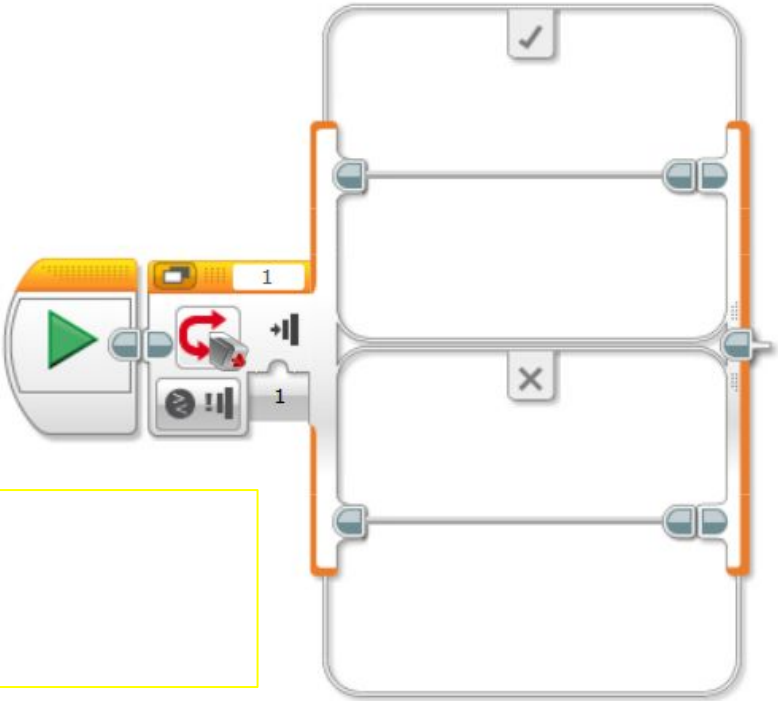
The Statement we are after is called the Switch Statement, and it is the second to last statement.



Sensor Logic

Let's Practice!

If we wanted our robot to run the following Pseudocode, what should we put in the True and False Boxes?

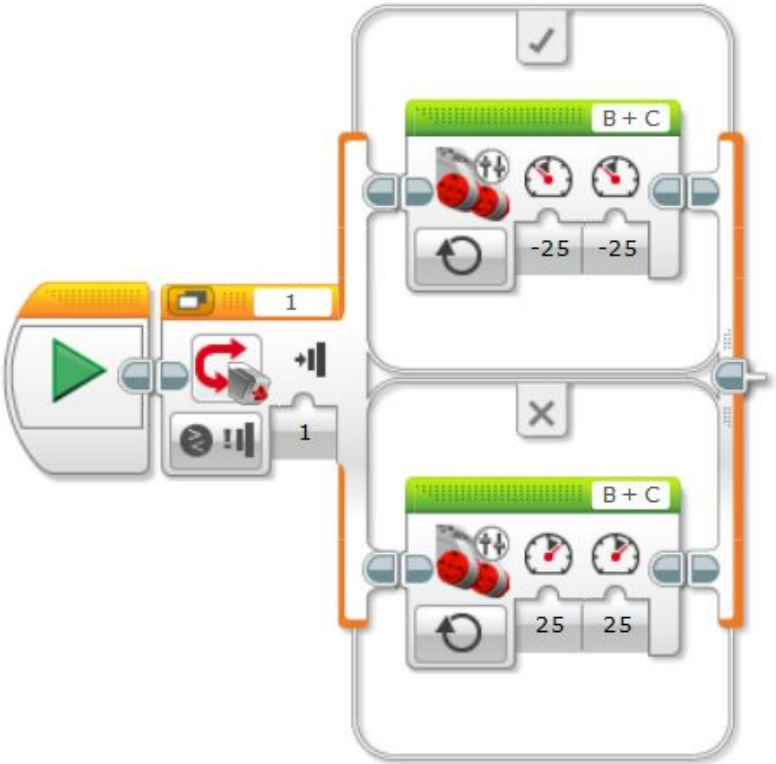


IF TOUCH SENSOR REGISTERS 1
MOVE BACKWARD
ELSE
MOVE FORWARD

Sensor Logic

Here is what we would put if we wanted to run the Pseudo Code from the last slide!

Remember, we want our robot to move backwards if the touch sensor is at 1 (pressed), and move forward otherwise.



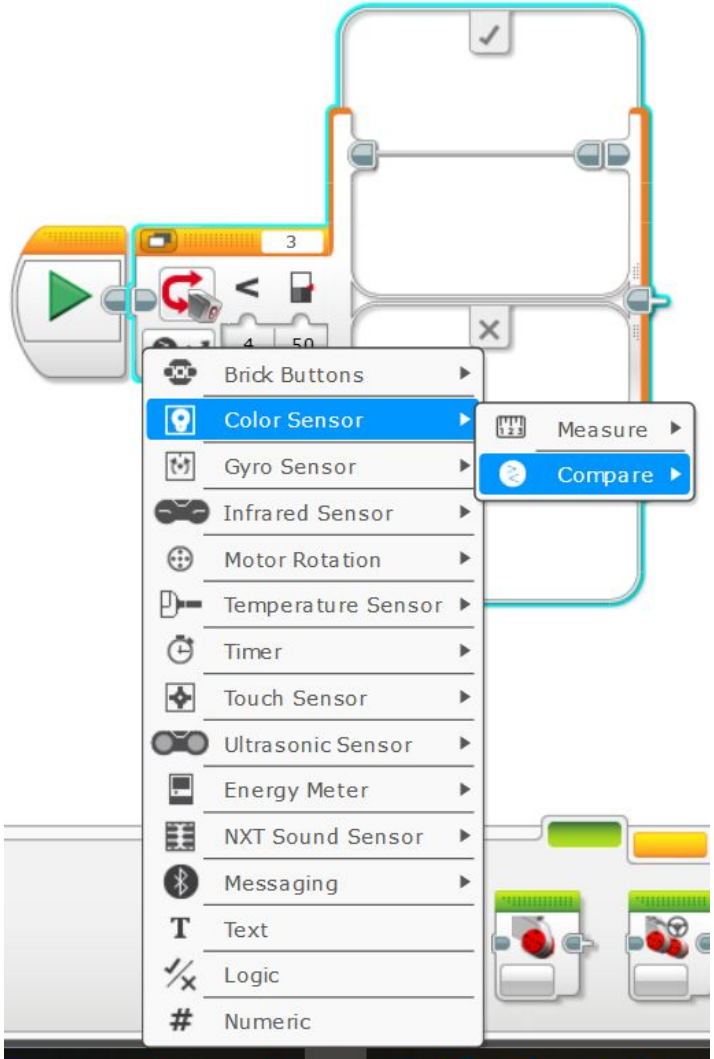
Let's Look at all the things we can test!

We can test a variety of sensors and data sets!

Let's look closely at the Color Sensor settings!

There are two options: **Measure** and **Compare**. For the purpose of testing Logic, we will use **Compare**!

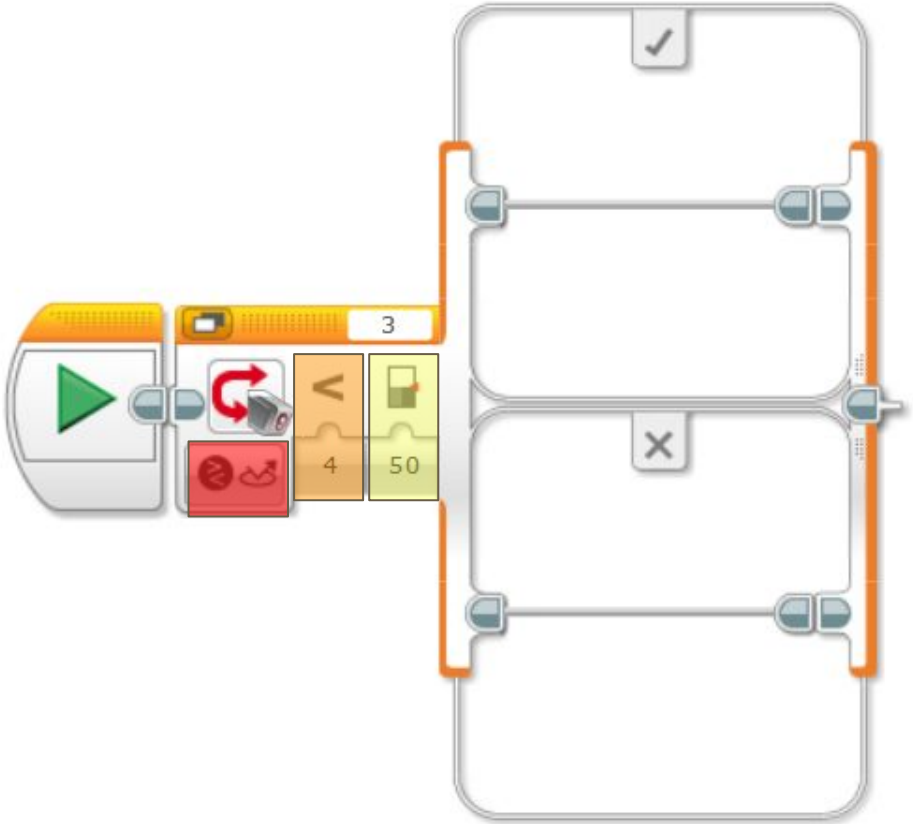
Compare allows for values to be tested against one another!



Comparing Logic

Let's look at how we compare logic in Mindstorms!

- Tells What is being Compared
- **Comparison Operator**
  - Greater Than >
  - Less Than <
  - Greater Than or Equal ≥
  - Less Than or Equal ≤
  - Equal =
  - Not Equal ≠
- **Threshold**
  - The value at which the statement determines if it is true or false

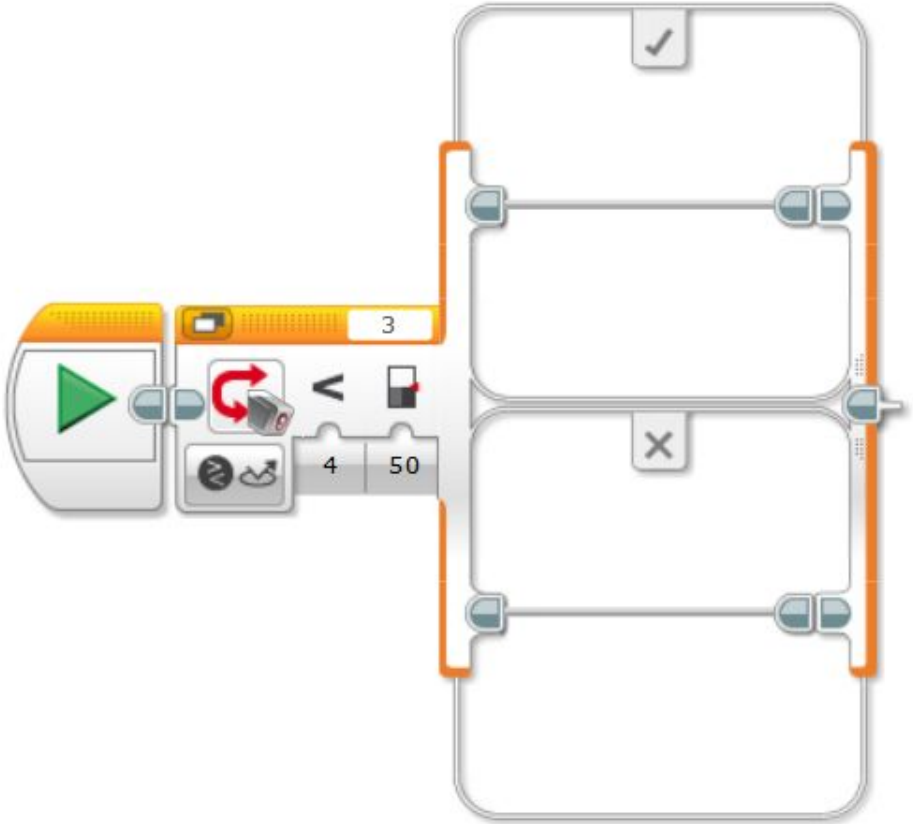


Thresholds

Let's look at the Comparison Operator and Threshold.

Since we have our comparison operator to be *Less Than* the Value must be *Less Than the Threshold Value* in order for the statement to be true.

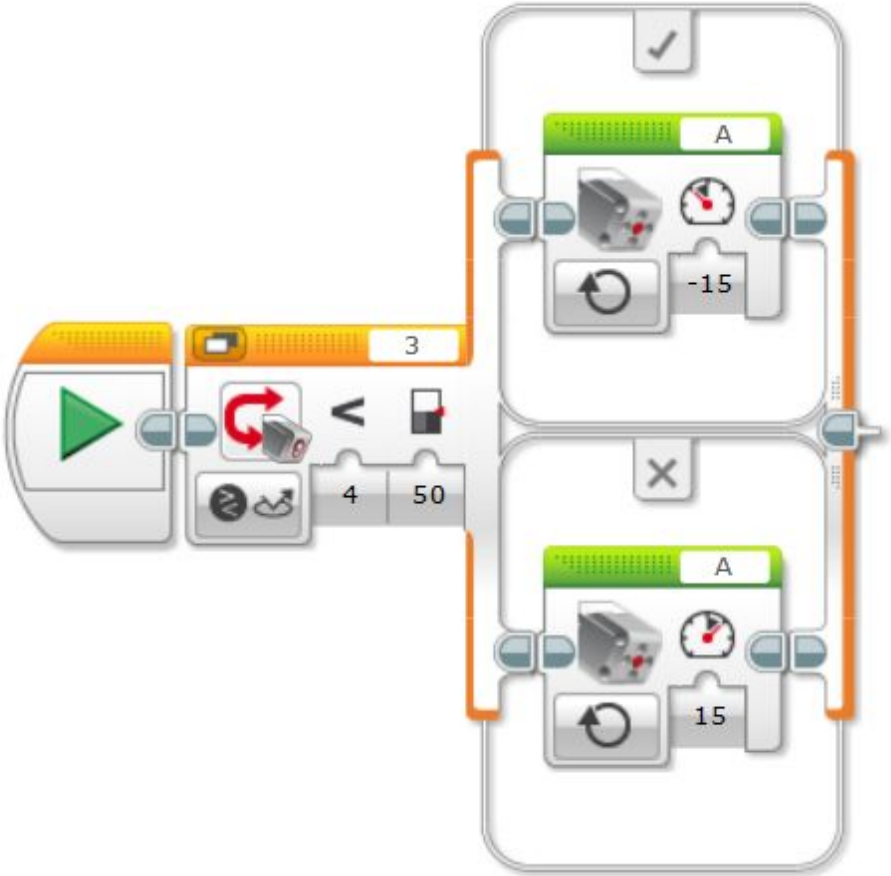
To Calculate threshold, you must add the darkest color value and the lightest color value and divide the sum by 2



Practicing Logic

What would happen if the color sensor registers the following Values?

- 75
- 25
- 50



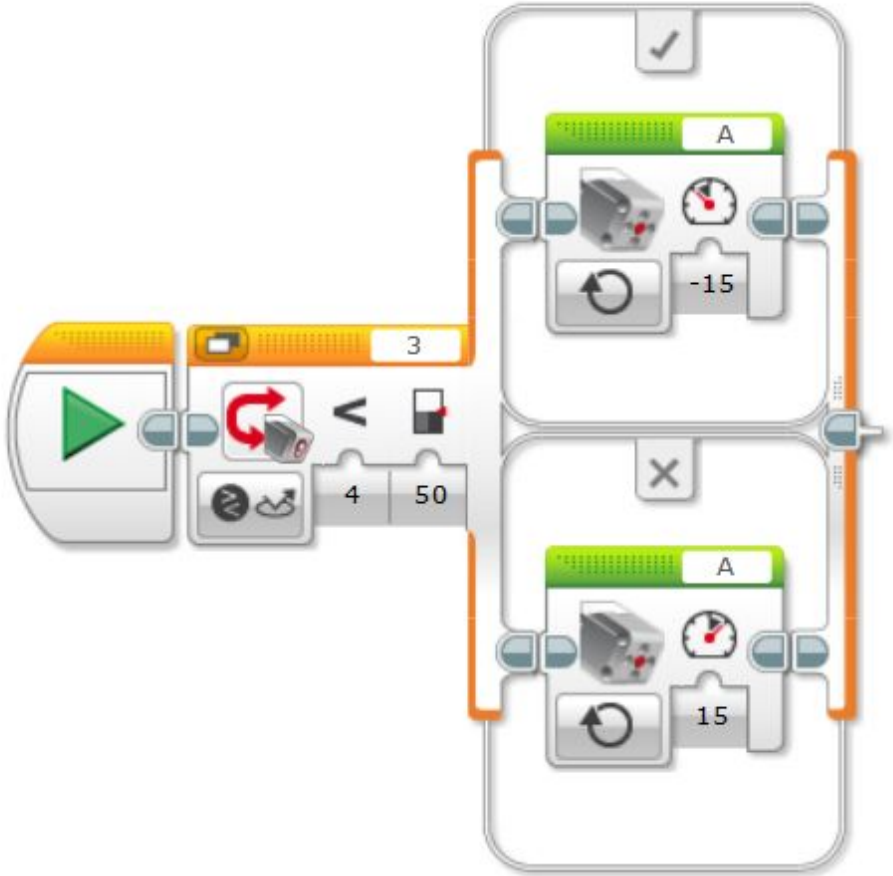
Practicing Logic

Here's what would happen!

At a Sensor Value of 75, the motor would spin Forward, because 75 is greater than 40

At a Sensor Value of 25, the motor would spin backwards, because 25 is NOT greater than 40

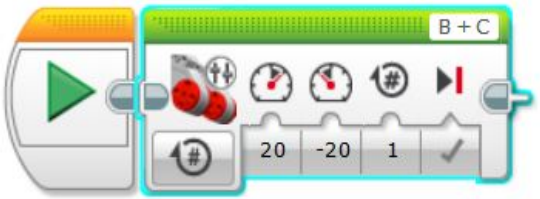
At a sensor Value of 50, the motor would spin Backwards, because 50 is NOT greater than 50

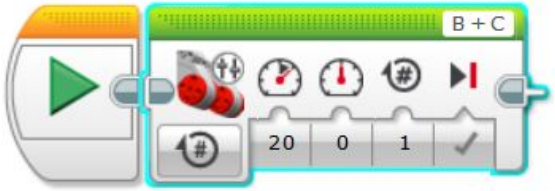


A Few More Things Before We Begin With the Robots!

Types of Turns

- **Point Turn**
  - A precise turn that covers little space
  - One Drive Motor is Positive, the Other Drive Motor is Negative
- **Swing Turn**
  - A Wide Turn that covers a lot of space
  - One Drive Motor is Positive, the other is turned off
- **Modified Turn**
  - A Turn that is modified by having one motor at a higher power than the other motor.
  - It is very hard to program, and isn't very accurate





Let’s Begin!