



Conditionals

Robotics Curriculum
IT Adventures



Lesson Overview

- Learn what conditionals are and how to use Boolean logic
- Learn how to use conditionals while programming the micro:bit
- Create a series of programs for the RVR using conditional statements



Conditional Overview

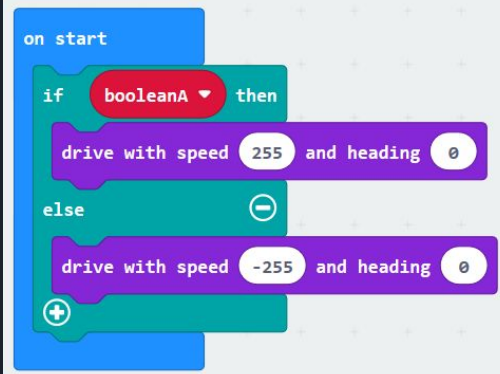
From a high-level standpoint, conditionals are any statement that rely on some *condition* being met. One example from real life would be grocery shopping, where you may be asked to buy bread *if* there is a sale on bread. The conditional in the previous scenario would be the existence of the sale. The condition of the sale existing or not existing determines the result of whether or not you buy bread.

Conditionals are the same in code as well, with a variable either meeting or not meeting the condition (a True or False value). Usually, Boolean logic in code is represented by a 0 for a False condition, and a 1 (depending on the language) as a True condition.

Code Example: Using Conditionals in If/Else statements

In the code example to the right, the variable `booleanA` is used to store a True or False value. In this example, we don't know what `booleanA` is set to, but we can see that it is used to decide which code to run. In an If/Else statement, the first block of code will run *if* the condition is true, and the second block will run in any other statement. Note that there are several variants of this statement, including an If statement and an If/Else If/Else statement (with each Else-If only being evaluated if the previous statement is false).

In the example code, it is declared that the RVR will run forward at maximum speed if `booleanA` evaluates to True, and will run backwards at maximum speed if `booleanA` evaluates to False.



```
on start
  if booleanA then
    drive with speed 255 and heading 0
  else
    drive with speed -255 and heading 0
```

The image shows a Scratch code block starting with 'on start'. It contains an 'if' statement with a dropdown menu set to 'booleanA'. The 'then' block is a purple 'drive with speed 255 and heading 0' block. The 'else' block is a purple 'drive with speed -255 and heading 0' block. The code is set against a light blue grid background.



Expanding on Boolean Logic

So far, the examples have contained only one case, but multiple cases can be combined to make more complex statements. There are four basic Boolean operators to know about to create these more complex statements:

- **AND:** The AND case only returns True when both statements are True.
- **OR:** The OR case returns True when either statement, or both statements, are True.
- **XOR:** The XOR (exclusive-or) case is similar to the OR case, except that it will return False if both statements are True.
- **NOT:** The NOT case flips the logic around; essentially, if a statement is True, the NOT operator would change it to be False, and vice versa.

A	B	AND	OR	XOR
0	0	0	0	0
1	0	0	1	1
0	1	0	1	1
1	1	1	1	0

A	NOT A
0	1
1	0

Code Example: Complex Logic Statements

```
on start
  if A and B then
    drive with speed 255 and heading 0
  else if A or B then
    drive with speed -255 and heading 0
  else
    stop with heading 0
```

In the code example to the left, the same code is presented, but this time with a more complex set of requirements. One could easily imagine the variables A and B being linked to some button inputs or some other thing. However, in this example, we can see that the RVR will run forwards if both buttons are pushed, and backwards if either button is pressed.

These kits can be found under the “Logic” tab, where there are also other options that can be explored (such as comparing equality, or greater than/less than). Try them out!




Primary Learning Challenge: If This, Then That

As you've learned about how conditionals work, you'll now use them to control the RVR. The micro:bit has two main buttons built onto the board. Use them to create a program that controls the RVR, meeting the following conditions:

- If both buttons are/were pressed, go forward
- If only A was pressed, go to the right
- If only B was pressed, go to the left
- Otherwise, go backwards

Remember that only one of these conditions should be met at any given point! Good luck!



Secondary Learning Challenge: Learn to Drive

For the extra challenge, you'll be simulating starting and driving your car! This set of conditionals must be met in exactly the order specified - consider how to order your if statements to make it work.

On start, pause for X seconds to allow for any input.

If a clap is heard during this break, "unlock" the RVR by playing a sound and then pausing again for X seconds.

If the logo is pressed during that pause, play a sound for starting the car and pause again.

If both A and B are pressed during that pause, drive the RVR in reverse and then turn 90 degrees to back out of a parking spot. Congratulations, you've started your car!