

BOTS FOR KIDS

Programming | The First Block | Getting Loopy | Which Way?

Programming

Last week you built your robot, and some started their first program. This week you will learn to make the mBot do more interesting things. We will get you started with some fun programs you can modify and take ideas from.

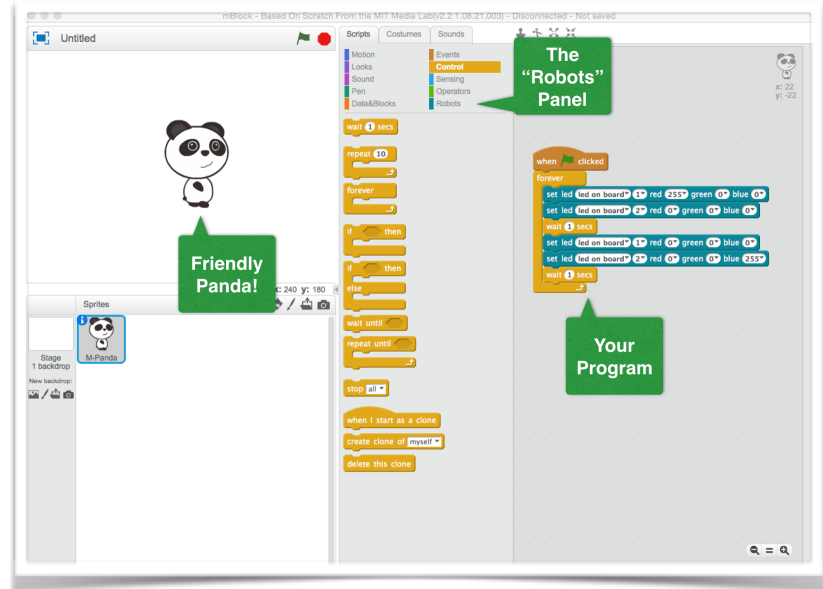
To the right is mBlock. The Panda bear is the default character, and it can talk with your robot by giving it a program on the right-hand side of the screen.

In the program shown above, the Panda is telling the robot to blink its lights like a police car. (The blue-green robot commands are under the “Robots” panel in the “Scripts” tab.) Robot commands can tell the robot to move, turn on lights, play notes, or take any other action it can do. You can also receive information from the robot’s sensors.

If you want the robot to control the Panda instead, look in the “Motion” and “Looks” panels, where the Panda can be told to say, think, go, or even look like anything you want.

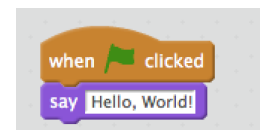
To run the program, first connect to your robot using the “Connect” menu, then click the green flag.

We went pretty quickly. Let’s break it down...

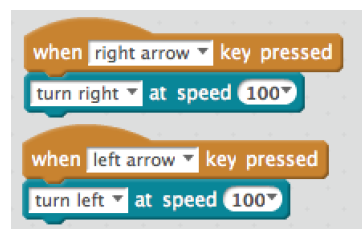


The First Block

All programs start with an event, found in the “Events” panel. A common way to start a scratch program is the “Flag” event, which happens when the green flag at the top of the screen is pressed.



Another event is a keypress. If you want something to happen in response to pressing a key, use the “When [space] key pressed” event. You can change “space” to any key. You can also



create programs assigned to different keys, and have each key do a different thing.

Getting Loopy

A program runs each command one at a time, and when there are no more commands, it ends. If you don't want your program to end right away, create a loop. Loops can run forever, or stop when something happens.

Loops can be dragged in from the “Control” panel. It is common for a program to consist of two major sections: at the beginning things get set to their starting values, then inside a “forever” loop the robot continues to take action.

The program to the right makes sure the robot is stopped, then has it squiggle forward five times before stopping the motors again.



Note: When a program ends, the robot will continue doing the last thing it was told for each component, like running motors, making noise, keeping lights lit, etc. So be sure to stop them at the end of your program!

Project Ideas

- Light & Sound Show
- Light Detector
- Dance Bot
- “Sonar” alarm which spins and detects things coming close.
- Police Car (including Lights and Sirens)

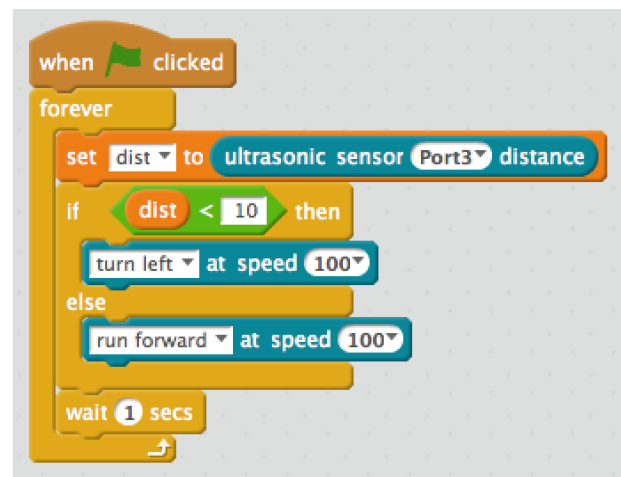
Which Way?

Sometimes you want the robot to do different things depending on what it “sees” with its sensors. You can use a block container with an “if <> then” to choose. But what do you put in the space?

Blocks with triangular edges give the program “true” or “false” inputs. For example, in the “Robots” panel you have the “<button pressed>” block which returns “true” when the button on the robot is pressed.

Another way to decide something is to compare it to something else. Under the “Operators” panel, you will see the less-than, equals, and greater-than operators.

Warning: mBlock has a bug where you can't use the “Robots” panel sensors directly in an “if” block. Until it is fixed, you must use a “Variable” under the “Data&Blocks”. Click “Make a Variable”, type a name, then set the variable. This technique is shown below.



Above we have a program that will move forward until it gets close to something, then turn left until it sees an open path. The “dist” variable was defined in “Data&Blocks”.