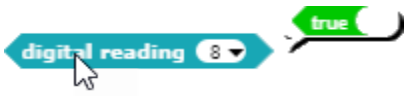
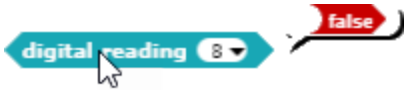
Reading an Input

The Arduino can control the output of the digital pins. It uses this capability to turn LEDs connected to the pins on and off. It can also read the status of the pins. If a wire is connected between one of the digital pins and a voltage source, the Digital Reading code block generates an output of *True*.

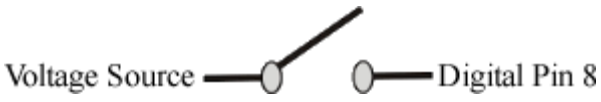


On the other hand, if a wire is connected between one of the digital pins and ground, the Digital Reading code block generates an output of *False*. In the illustration, a wire has connected Digital Pin 8 (at the top) to a ground pin (at the bottom).

Therefore the Digital Reading code block generates an output of *False*.



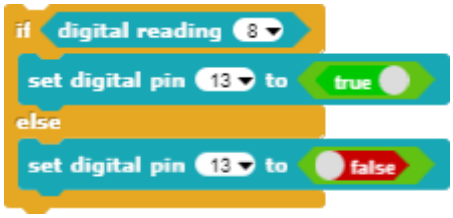
A switch provides a convenient means of closing a circuit. If one side of a switch is connected to Digital Pin 8 and the other side is connected to a voltage source, the output of the Digital Reading 8 code block will read *True* when the switch is closed and *False* when the switch is open.



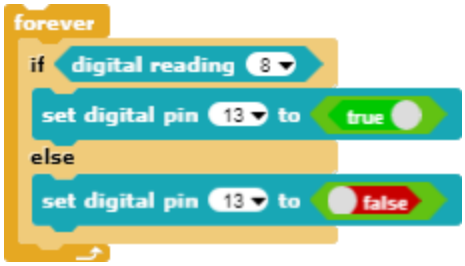
A push button is a specific type of switch that typically completes a circuit when the button is pushed down and opens the circuit when the button is released.

Controlling an LED with a Button

The ability to monitor the status of an input on the Arduino makes it possible to write a program that turns on an LED when a button is press. The sample program reads the status of Digital Pin 8. If the output of Digital Pin 8 is *True*, the program sets Digital Pin 13 to *True*, turning on the LED. If the output of Digital Pin 8 is *False*, the program sets Digital Pin 13 to *False*, turning off the LED.



This program could be placed within a Forever loop to constantly monitor the status of Digital Pin 8.



A When code block (found under the *Control* palette) can be used to accomplish the same result. The When code block monitors the output of Digital Pin 8 when the *Green Flag* is clicked, taking action based on the output of the pin.