Analog and Digital Review

The next step is to now integrate components that send a range of analog input *into* the Arduino. Remember that a primary distinction between analog and digital output is that analog output resulted in a *range* of output. You tested this in your project by programming a range of brightness that the LED cycled through, much like what would result from a dimmer switch. This is contrasted with digital output, which would result in only two states: on or off. The digital output is the equivalent of a light switch.

You may have noticed previously that the Blink LED project used **digitalWrite**, which sent either a high or low value to the LED pin, resulting in the LED turning on or off. The analog output project with the fading LED introduced **analogWrite**, which was used to send the range of output to the LED via the pin. The modes of output that you've used so far are **digitalWrite** and **analogWrite**.

Analog Input

The Arduino supports analog input as well. To parallel the previous example, digital input in the Arduino has two values: high and low and the associated method is **digitalRead**. Just like in the output example, digital input has either a high or low value. The digitalRead method is used to read the value from a specified pin, but only reads high or low. An example of this would be if you connected an on-off button to your Arduino board. The button would have two states: on or off. This would be an example of a digital input since the button press would send either the on state or the off state. However, several projects require an input range. An example would be a volume knob on a stereo. This sends a range of input to the device.

The command to accomplish this is **analogRead**, which will read the input from a pin and return a value between 0 and 1023. This will greatly expand the possibilities of your projects since you'll now be able to work with another type of input. The next component that you'll be working with, the *Potentiometer*, sends a range of input into the Arduino.

Once you're familiar with the difference between the two types of input, move on to the project!