**Blink**

/\*Blink - Turns on an LED on for one second, then off for one second, repeatedly. \*/

// Pin 13 has an LED connected; Here we declare it as a variable

// int refers to the type of variable, ie. integer

**int led = 13;**

// the setup function runs ONCE the board is powered/restarted:

**void setup() {**

// initialize the pin to be used for output signals

**pinMode(led, OUTPUT);**

}

// the loop routine runs over and over again forever:

**void loop() {**

// HIGH, LOW are terms used to describe a pin's state

// HIGH means the pin is given 5V and is being used

**digitalWrite(led, HIGH); // turn the LED**

// the delay function tell's the board to pause all processes

// for the given time in milliseconds

// 1000ms = 1s

**delay(1000);**

// turn the LED off by making the voltage LOW

// LOW means the pin receives 0V

**digitalWrite(led, LOW);**

**delay(1000);** // wait for a second

}

**oneMotorIntro**

//Motor pin configuration

**int directionPin = 4;** //Pin controlling Left motor direction

**int speedPin = 5;** //Pin controlling Left motor speed

void setup() {

//Declaring the directin pin as an output

**pinMode(directionPin, OUTPUT);**

//Declaring the directin pin as an output

**pinMode(speedPin, OUTPUT);**

//The direction pin takes only 2 values, HIGH or LOW

//This pin controls the rotation of the motor

//HIGH can mean clockwise, while LOW is counter-clockwise

//Or vise versa, depending on how your motor is wired

//Assigning HIGH to the direction pin

**digitalWrite(directionPin, HIGH);**

//Assigning LOW to the speed pin cuts off the power

//and without power, the motor cannot turn

//Toggling between HIGH/LOW is how we can control

//when we want a motor should spin and when not to.

**digitalWrite(speedPin, LOW);**

}

**void loop() {**

//until now, we've used digitalWrite()

//digitalWrite() is used for digital pins

//digital pins have only two states, either HIGH or LOW

//analog pins have a range of values (0-255 or 0-1024)

//the speedPin ranges from 0-255

**analogWrite(speedPin, 100);**

}