**Basic Challenge Requirements (Level 1 to Level 2)**

1. Write an Arduino program that contains the required “Setup” and “Loop” procedures.
2. Define an integer variable to hold the pin position for an external “red” LED
3. Define an integer variable to hold the pin position for an external “green” LED
4. Add code to blink the “red” LED for one second and then blink the “green” LED for one second.
5. Cut and copy your program code below this line and submit to your GitHub repository.

1.        /\* Blinking LED  
\* ------------

2.        \* Turns on and off a dual color light emitting diode (LED) with its color-specific leads connected to digital   
\* pins 12 and 13. The LED blinks alternating from red to green in a sycopated interval of 1 s red, 700 ms green.

3.        \* To limit current flow through the circuit, we added a 220 ohm resistor on the middle lead,

4.        which then connects to the ground (GND) pin on the Arduino.

5. 

6.        \*  
\* Created 8 Sept. 2009 by Chung-Hay  
\* [**http://arduino.berlios.de**](http://arduino.berlios.de/)  
\*  
\* based on an orginal by H. Barragan for the Wiring i/o board  
\*/

7. 

int ledred = 13;                // red LED connected to digital pin 13

int ledgreen = 12;

void setup()                    // run once, when the sketch starts

{

  pinMode(ledred, OUTPUT);      // sets the digital pin as output

  pinMode(ledgreen, OUTPUT);

}

void loop()                     // run over and over again

{

  digitalWrite(ledred, HIGH);   // sets the red LED on

  digitalWrite(ledgreen, LOW);

  delay(1000);                  // waits for a second

  digitalWrite(ledred, LOW);    // sets the red LED off

  digitalWrite(ledgreen, HIGH);

  delay(700);                  // waits for a second

}

**Standard Challenge Requirements (Level 3)**

1. Modify your program to read and write character strings from the serial monitor.
2. If the user types “red” then run the code to turn on the “red” LED.
   1. Also print “Red LED is On” to the serial monitor.
   2. Also make sure the “green” LED is off.
3. If the user types “green” then run the code to turn on the “green” LED.
   1. Also print “Green LED is On” to the serial monitor.
   2. Also make sure the “red” LED is off.
4. If the user types something other than “red” or “green” then run the code to turn on both LEDs off.
   1. Also print “Both LEDs are Off” to the serial monitor.
5. Cut and copy your program code below this line and submit to your GitHub repository.

void setup() {

Serial.begin(9600);

pinMode(LED\_BUILTIN, OUTPUT);

}

void loop(){

Serial.write(45) ;

// initialize digital pin LED\_BUILTIN as an output.

digitalWrite(LED\_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)

int bytesSent = Serial .write("Red LED is ON");

Serial.println("");

delay(1000); // wait for a second

digitalWrite(LED\_BUILTIN, LOW); // turn the LED off by making the voltage LOW

bytesSent = Serial .write("Red LED is OFF");

Serial.println("");

delay(1000); // wait for a second

gitalWrite(LED\_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)

int bytesSent = Serial .write("Green LED is ON");

Serial.println("");

delay(1000); // wait for a second

digitalWrite(LED\_BUILTIN, LOW); // turn the LED off by making the voltage LOW

bytesSent = Serial .write("Green LED is OFF");

Serial.println("");

delay(1000);

digitalWrite(LED\_BUILTIN, LOW); // turn the LED off by making the voltage LOW

bytesSent = Serial .write("Both LED'S are OFF");

Serial.println("");

delay(1000);

}

**Enhanced Challenge Requirements (Level 4)**

1. Modify your program to read numbers from the serial monitor.
2. If the number is even then blink the “green” LED the number of times.
3. If the number is odd then blink the “red” LED the number of times.
4. Cut and copy your program code below this line and submit to your GitHub repository.