

RECAP

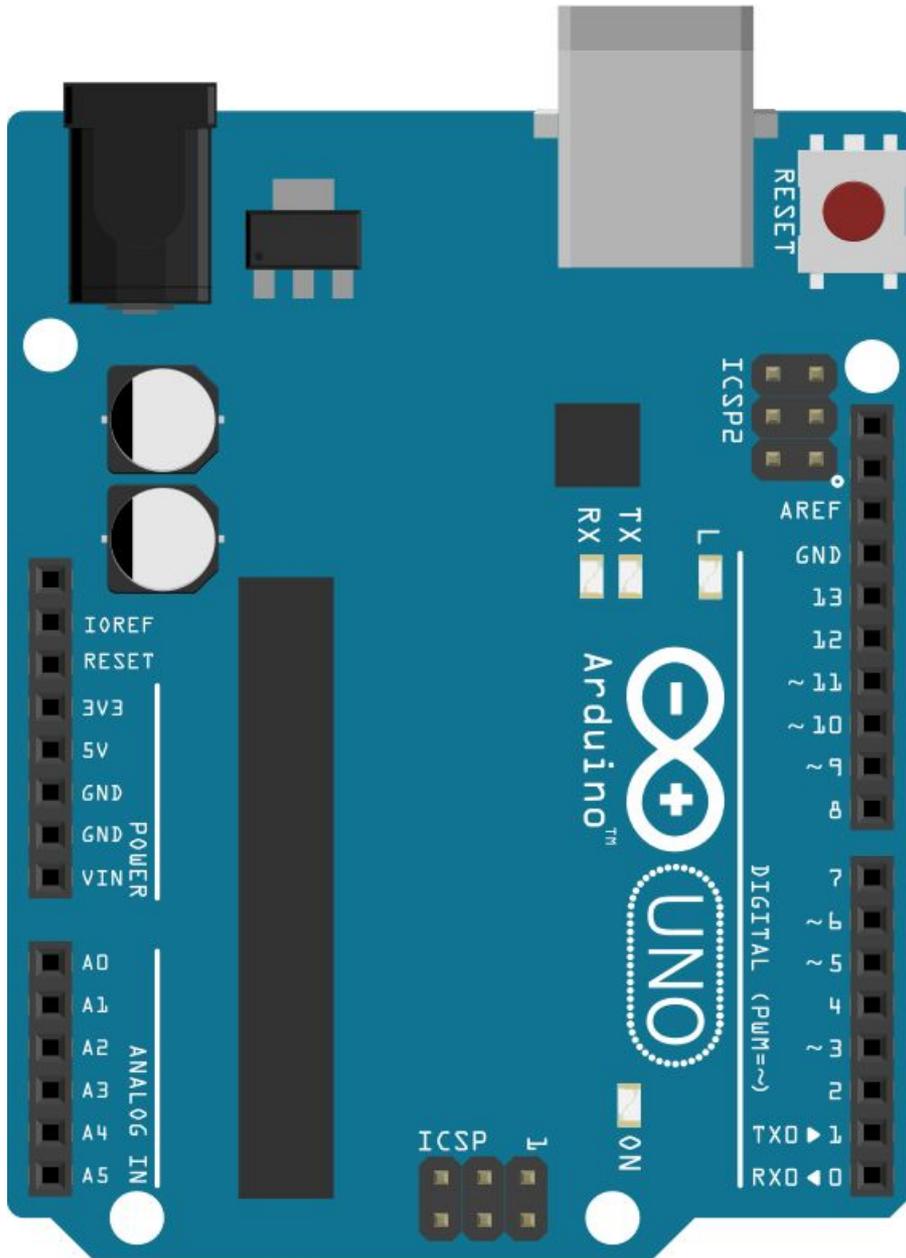
ARDUINO

DIGITAL &

ANALOG PINS

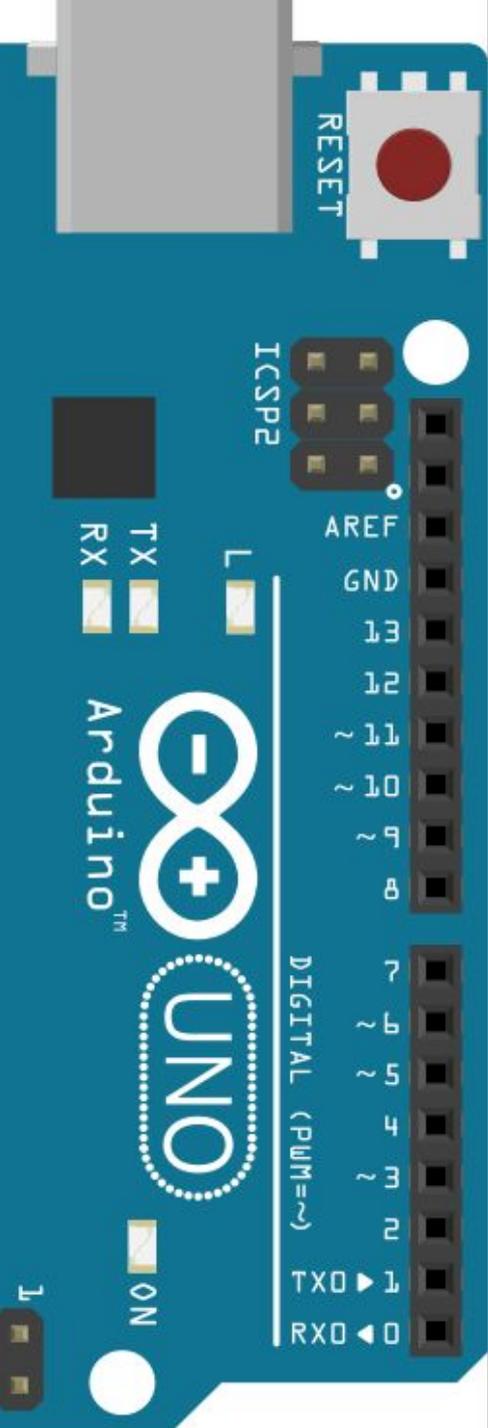
(ANALOG PINS)

PIN A0 TO A5



(DIGITAL PINS)

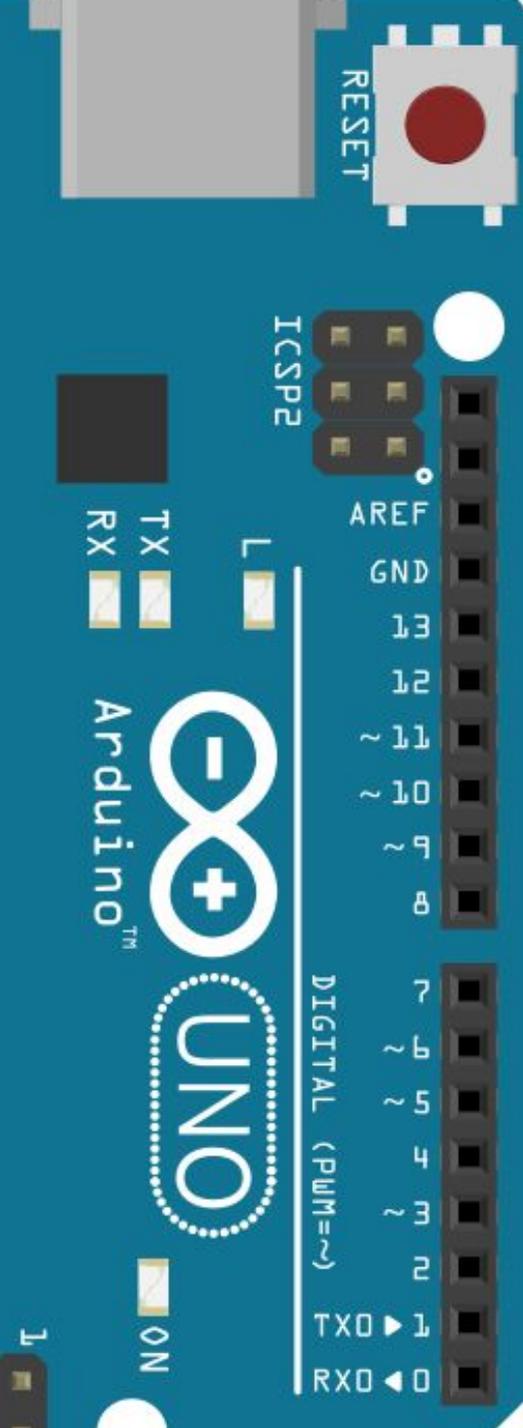
PIN 0 TO 13



(DIGITAL PINS)

PIN 2 TO 13

PINS 0 & 1 ARE USE FOR
COMMUNICATING WITH
COMPUTER, IT IS BEST TO
START WITH PIN 2



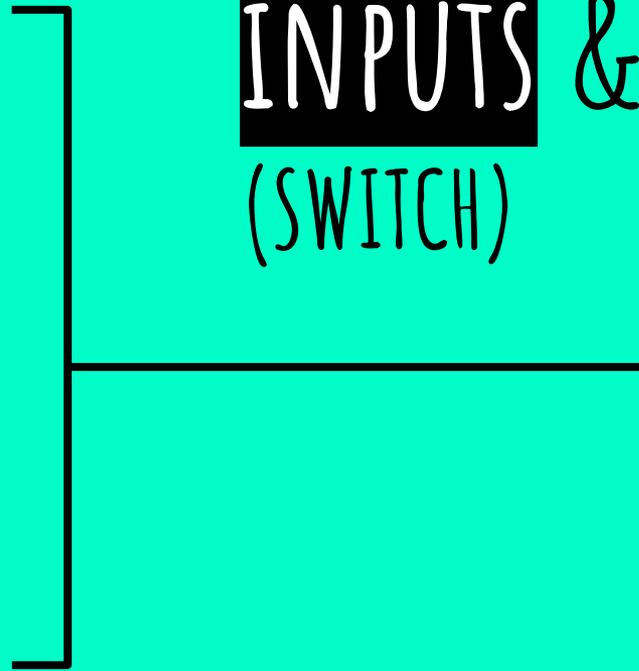
ARDUINO DIGITAL PINS

CAN ACT AS BOTH

INPUTS & **OUTPUTS**

(SWITCH)

(LEDS)



WE USED

```
digitalRead()
```

TO CHECK THE STATE OF

THE BUTTON

ARDUINO

DIGITAL PINS

CAN READ ONLY

2 STATES

WE USED

```
digitalRead()
```

TO CHECK THE STATE OF

OUR PUSH-BUTTON

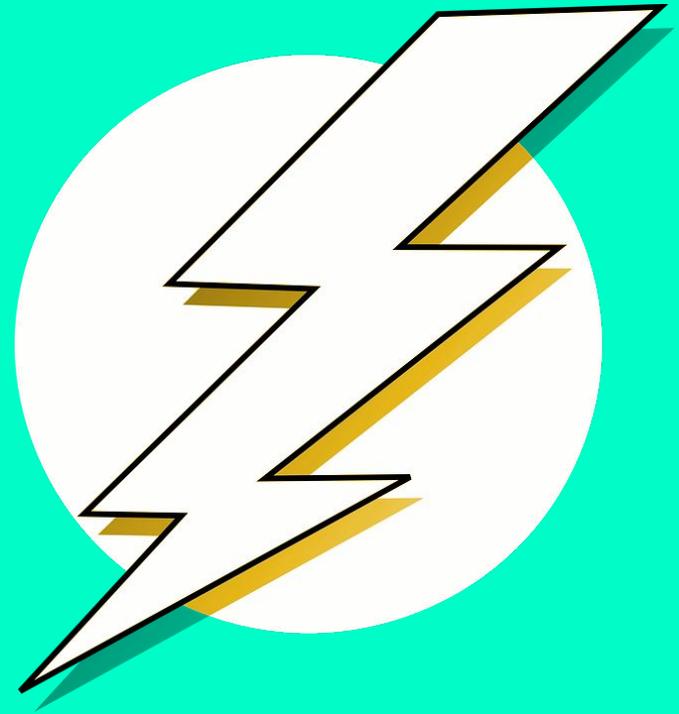
SWITCH

TO CHECK THE PIN STATE

```
digitalRead(pin)
```

HIGH & LOW

HIGH



THERE'S VOLTAGE ON THIS PIN

5V IS SENT TO THIS PIN

LOW



THERE IS NO VOLTAGE ON THIS PIN

WE USED

```
digitalWrite()
```

TO TURN ON OR OFF

THE LEDs.

TO TURN THE PIN

ON

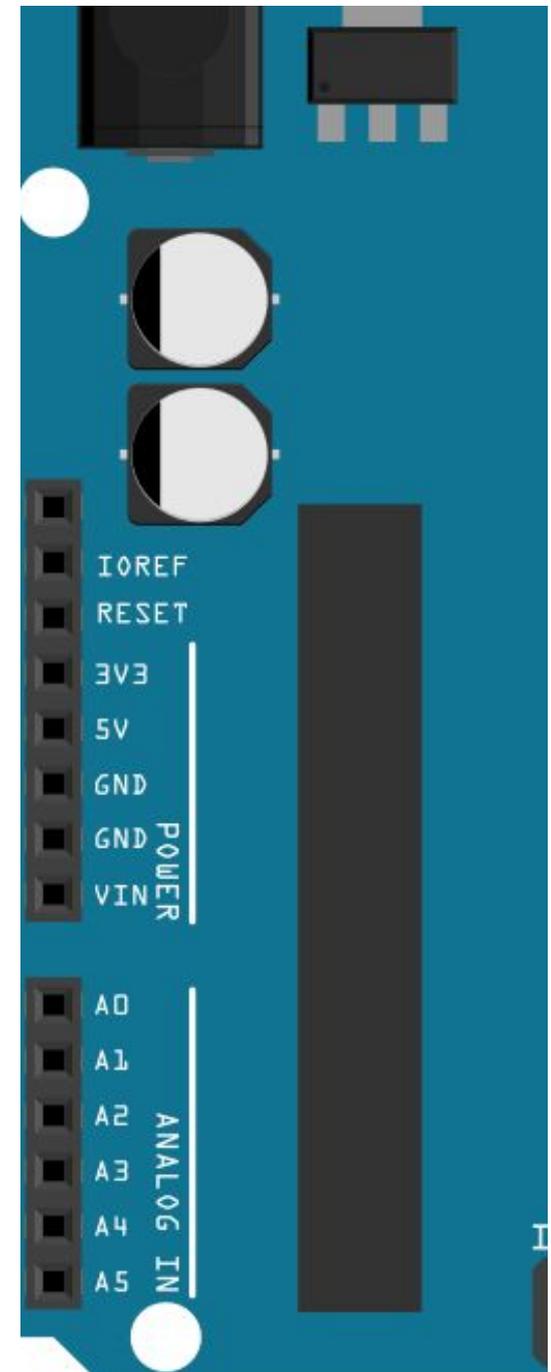
```
digitalWrite(pin, HIGH)
```

TO TURN THE PIN **OFF**

```
digitalWrite(pin, LOW)
```

WHAT CAN
WE DO WITH
ANALOG IN
(A0 TO A5)
PINS ON
ARDUINO ?

(ANALOG PINS)
PIN A0 TO A5



WE USED `analogRead()` TO
CHECK THE VALUES OF
PHOTOCELL

```
const int analogPin = A0;
```

```
void setup() {  
    Serial.begin(9600);  
}
```

```
void loop() {  
    int analogValue = analogRead(analogPin);  
    Serial.println(analogValue);  
    delay(1);  
}
```

WE USE `Serial.println()`

TO CHECK READINGS FROM

YOUR PHOTOCELL IN THE

SERIAL MONITOR

WE USE `Serial.println()`

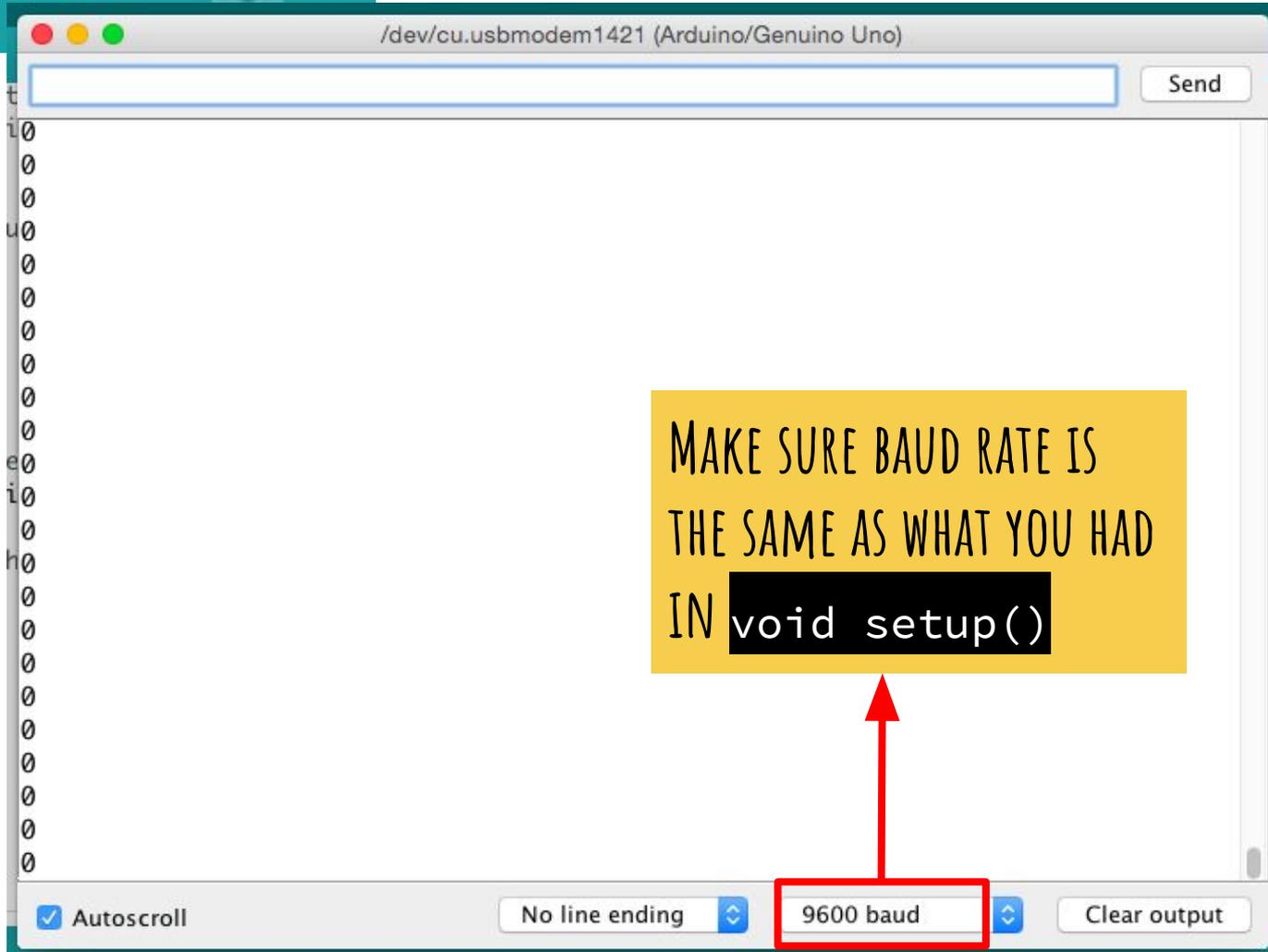
TO DISPLAY READINGS FROM

YOUR PHOTOCELL

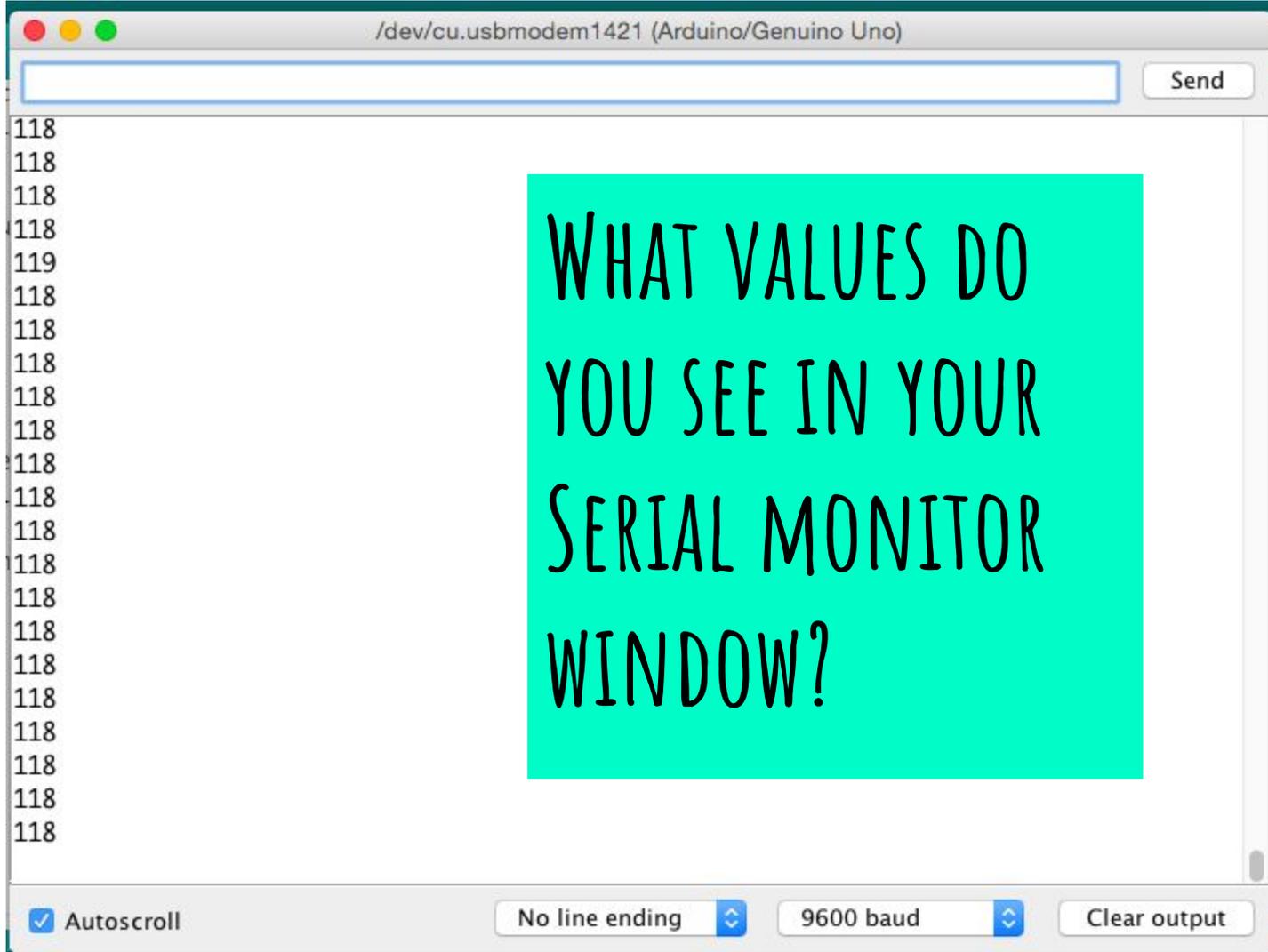
Serial Monitor



Click to open
Serial Monitor Window



MAKE SURE BAUD RATE IS
THE SAME AS WHAT YOU HAD
IN `void setup()`



118
118
118
118
119
118
118
118
118
118
118
118
118
118
118
118
118
118
118
118

WHAT VALUES DO
YOU SEE IN YOUR
SERIAL MONITOR
WINDOW?

Autoscroll

No line ending

9600 baud

Clear output

3. WE USE **IF** ... **ELSE**

STATEMENTS FOR

CHECKING BRIGHTNESS

```
if analogValue is less than 400  
  turn LED on
```

```
if analogValue is more than 400  
  turn LED off
```

```
if analogValue is less than 400  
    turn LED on
```

```
else  
    turn LED off
```

```
const int analogPin = A0;
const int ledPin = 13;
const int threshold = 400;

void setup() {
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  int analogValue = analogRead(analogPin);

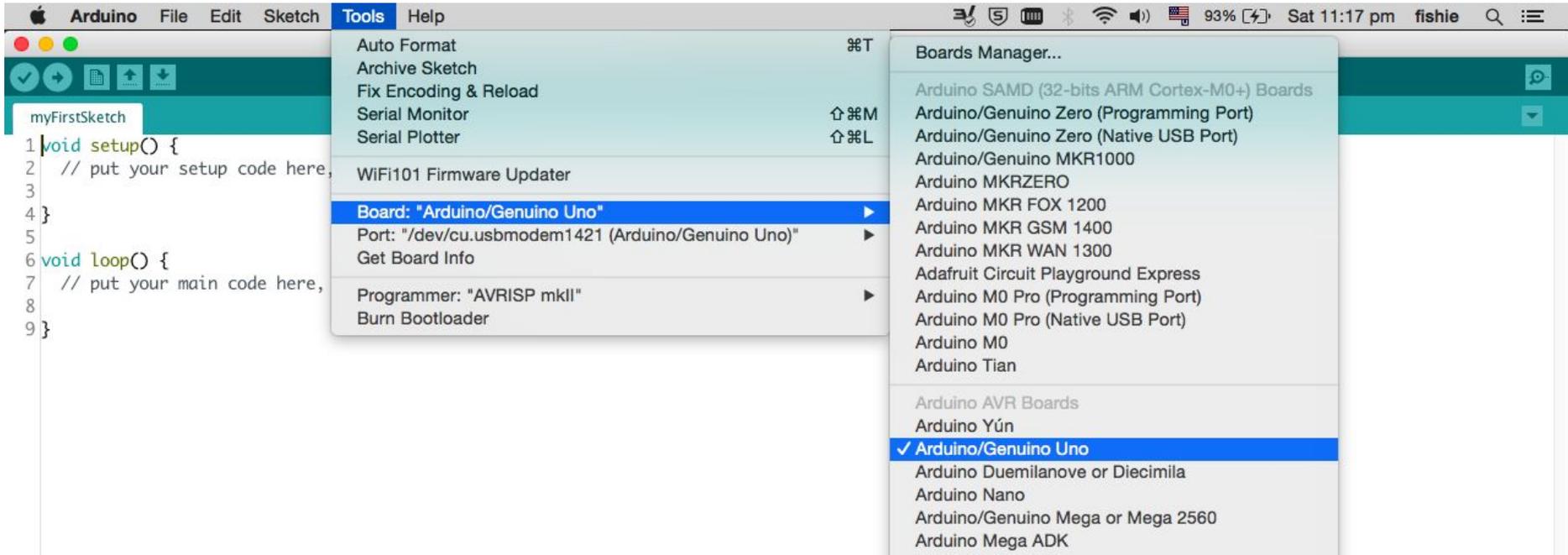
  if (analogValue < threshold) {
    digitalWrite(ledPin, HIGH);
  } else {
    digitalWrite(ledPin, LOW);
  }

  Serial.println(analogValue);
  delay(1);
}
```

ARDUINO HARDWARE & IDE

1. Launch Arduino
2. Plug the “square part” of the USB cable to Arduino
3. Plug the “flatter part” of the USB cable to your computer
4. Arduino on board “ON” LED will light up if it is powered

SELECT → TOOLS / BOARD: / ARDUINO GENUINO UNO

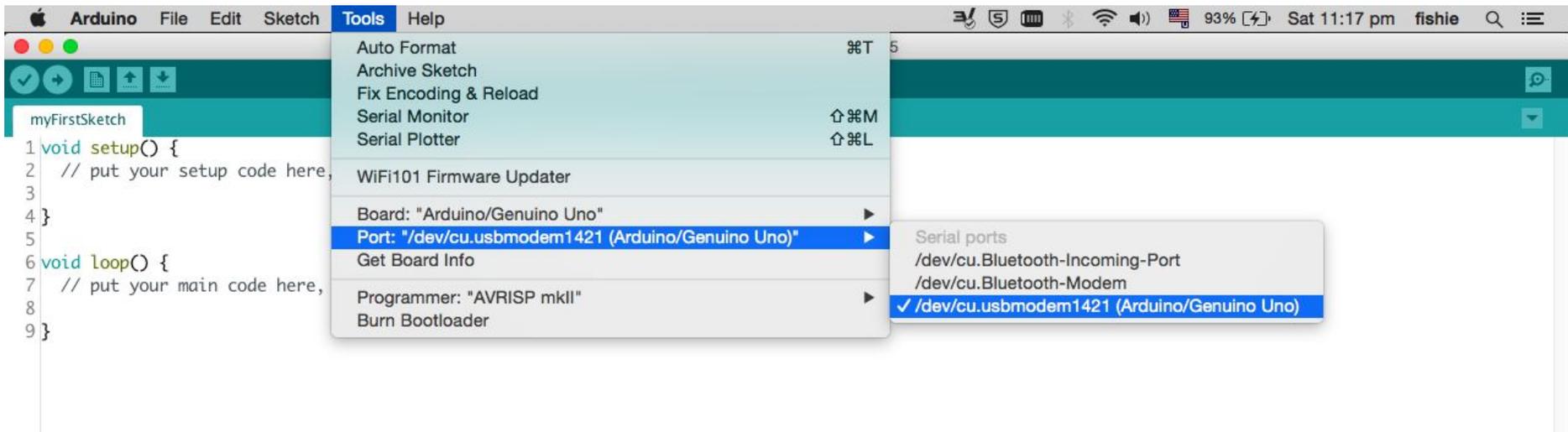


SELECT → TOOLS

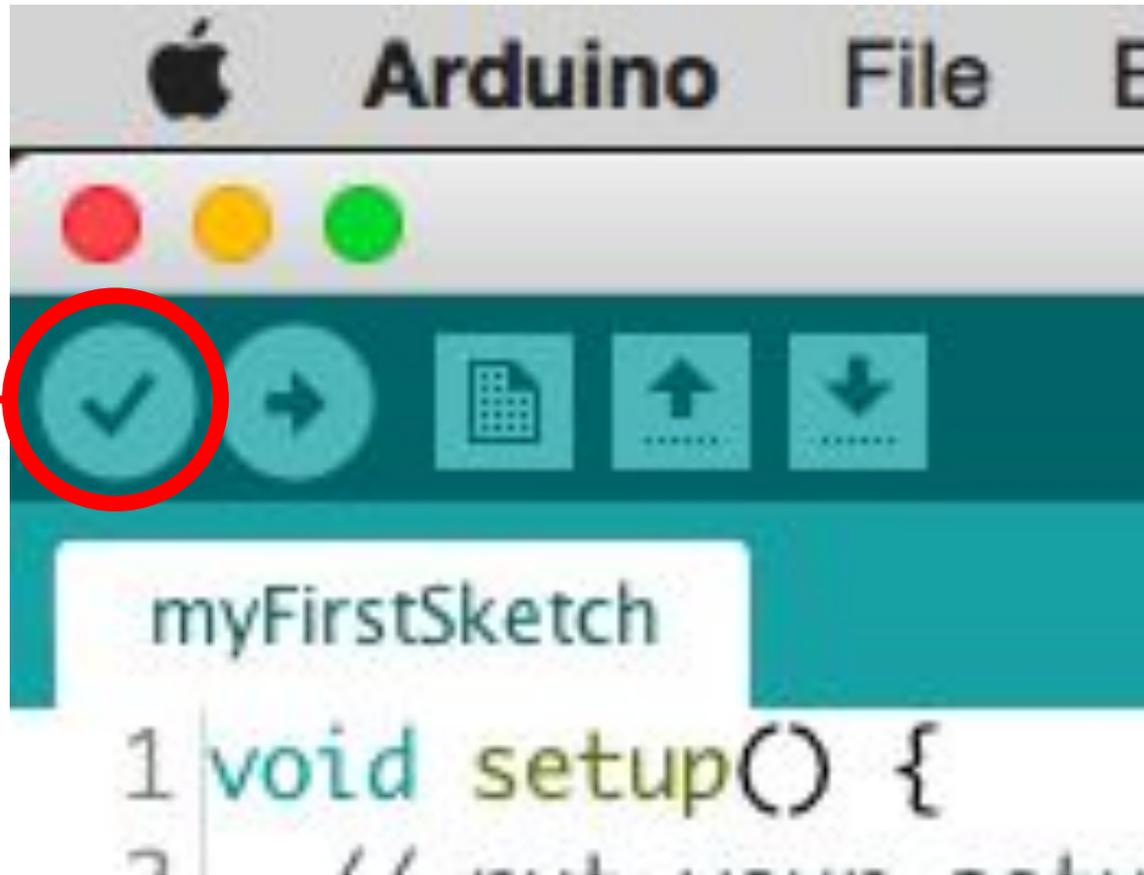
/ PORT:

/ DEV/CU.USBMODEMXXXX (MAC) OR

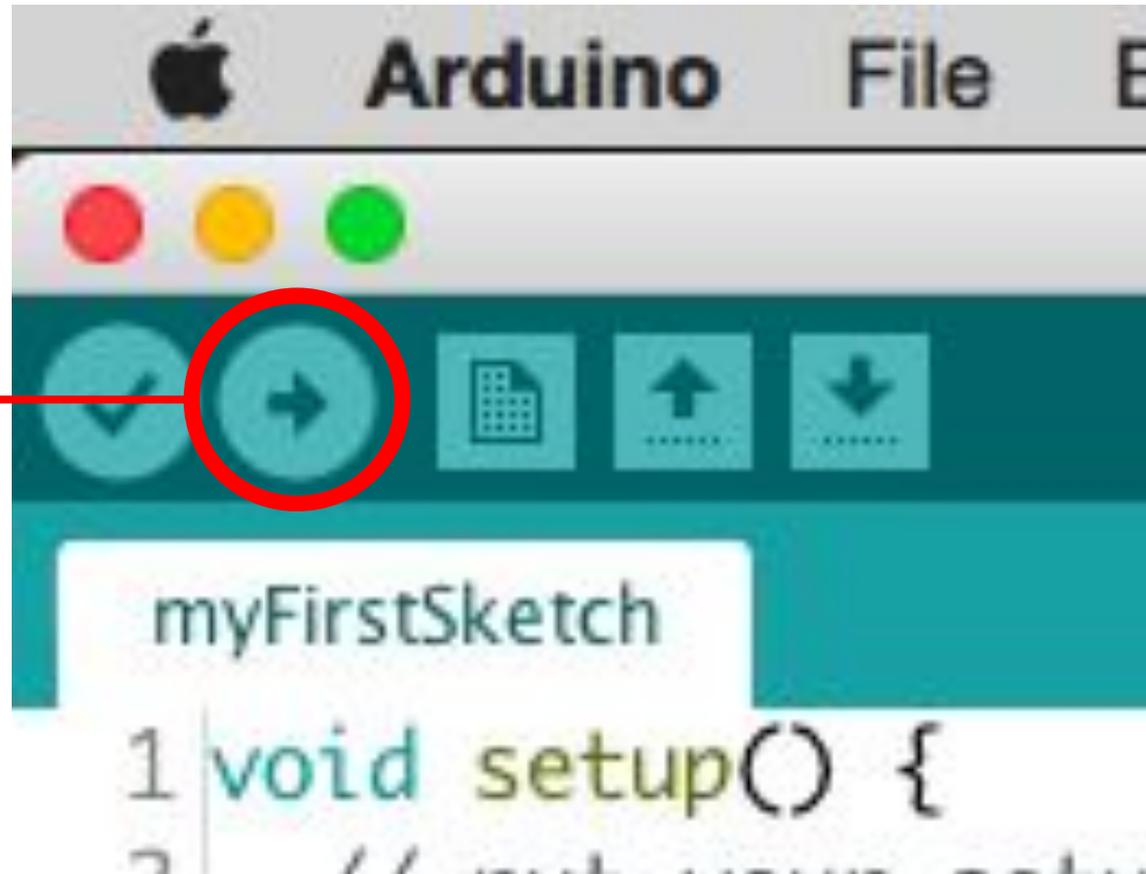
/ COMXXX (WINDOWS)

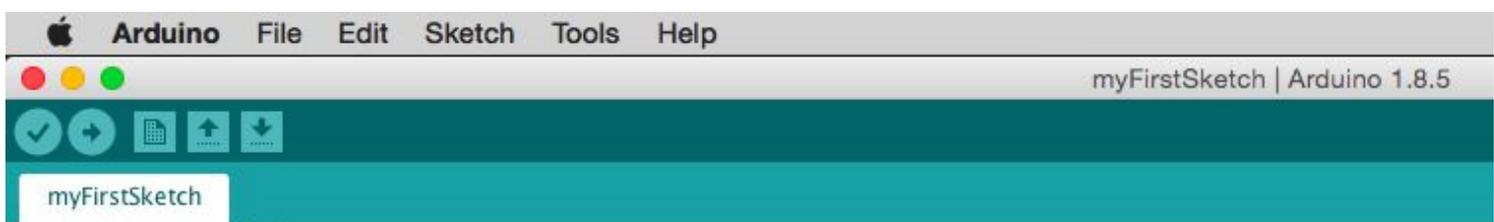


Verify
Code



Upload
Code





myFirstSketch

```
1 void setup() {  
2   // put your setup code here, to run once:  
3  
4 }  
5  
6 void loop() {  
7   // put your main code here, to run repeatedly:  
8  
9 }
```

-----SHORTCUT KEYS-----
CTRL + T = Auto Format code
CTRL + R = Verify code
CTRL + U = Upload code



CHALLENGE

CAN YOU BLINK

THE ONBOARD

LED ?

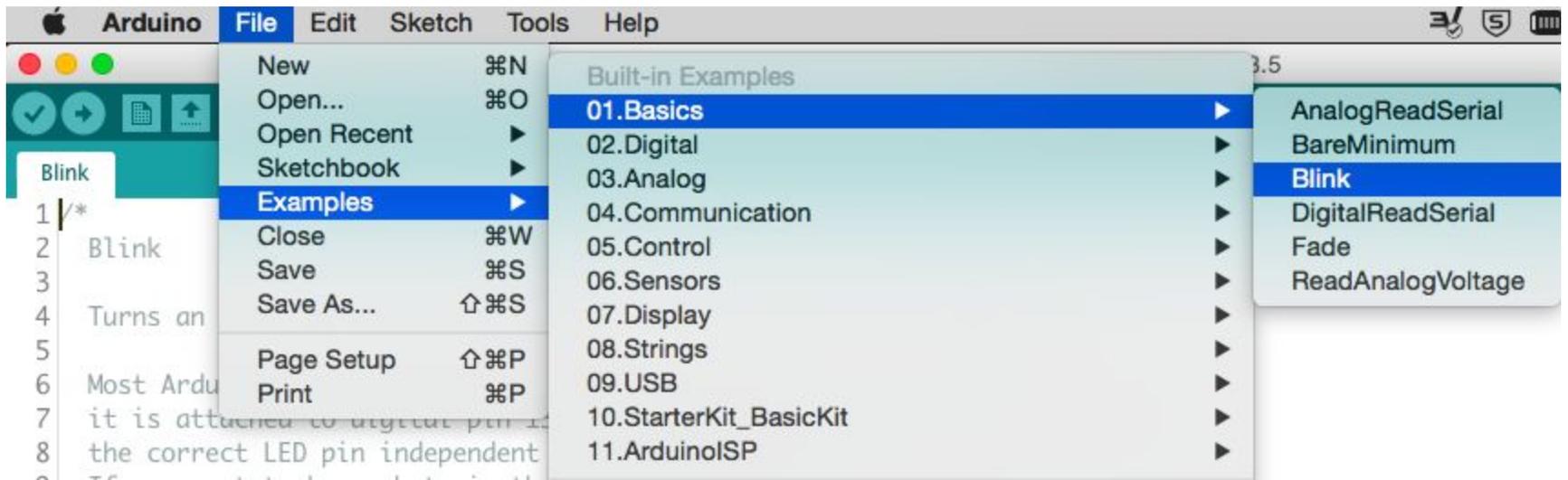
RECAP

DIGITALWRITE()

- We use `digitalWrite()` to turn the LED ON or OFF.
- Basically we are telling Arduino to set the voltage to HIGH (5 VOLTS) (ON) or to LOW (0 VOLT) (OFF)

GO TO ARDUINO

FILE / EXAMPLES / 01. BASICS / BLINK



*

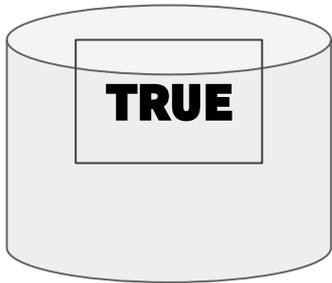
RECAP VARIABLES

*

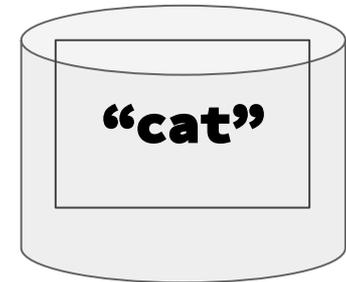
Variables are like buckets

INTRODUCING (VARIABLES)

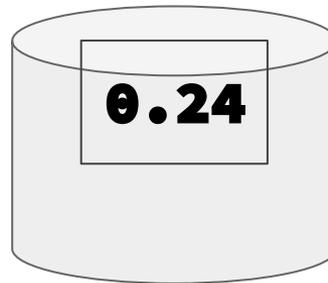
- Imagine variable is like a bucket.
- We can put content into this bucket.
- Each bucket can only contain one item at each time.
- We can change the (type of) content of the bucket.



`boolean ans = TRUE;`



`string pet = "cat";`



`float value = 0.24;`

INTRODUCING LOOPS

CHALLENGE

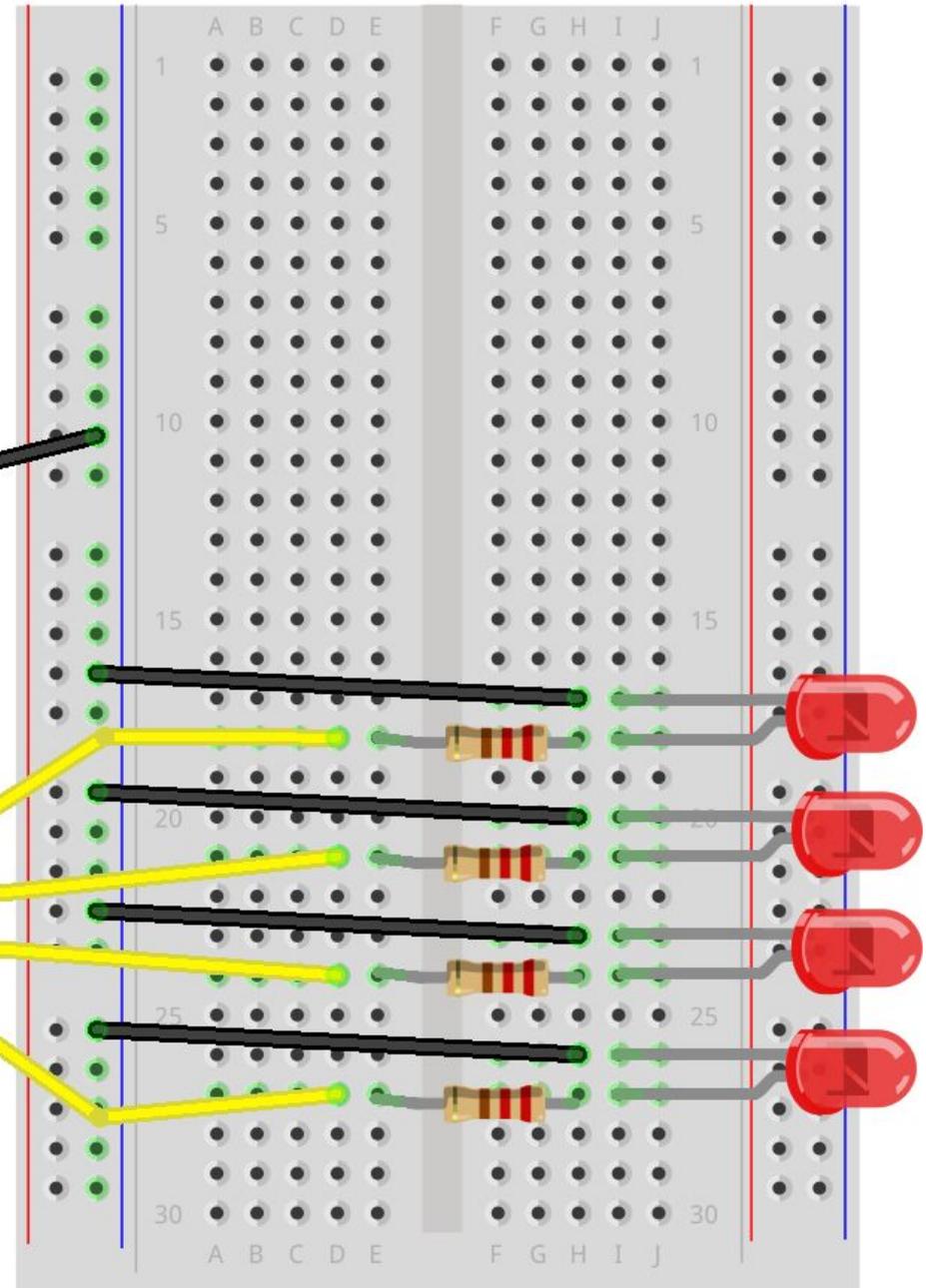
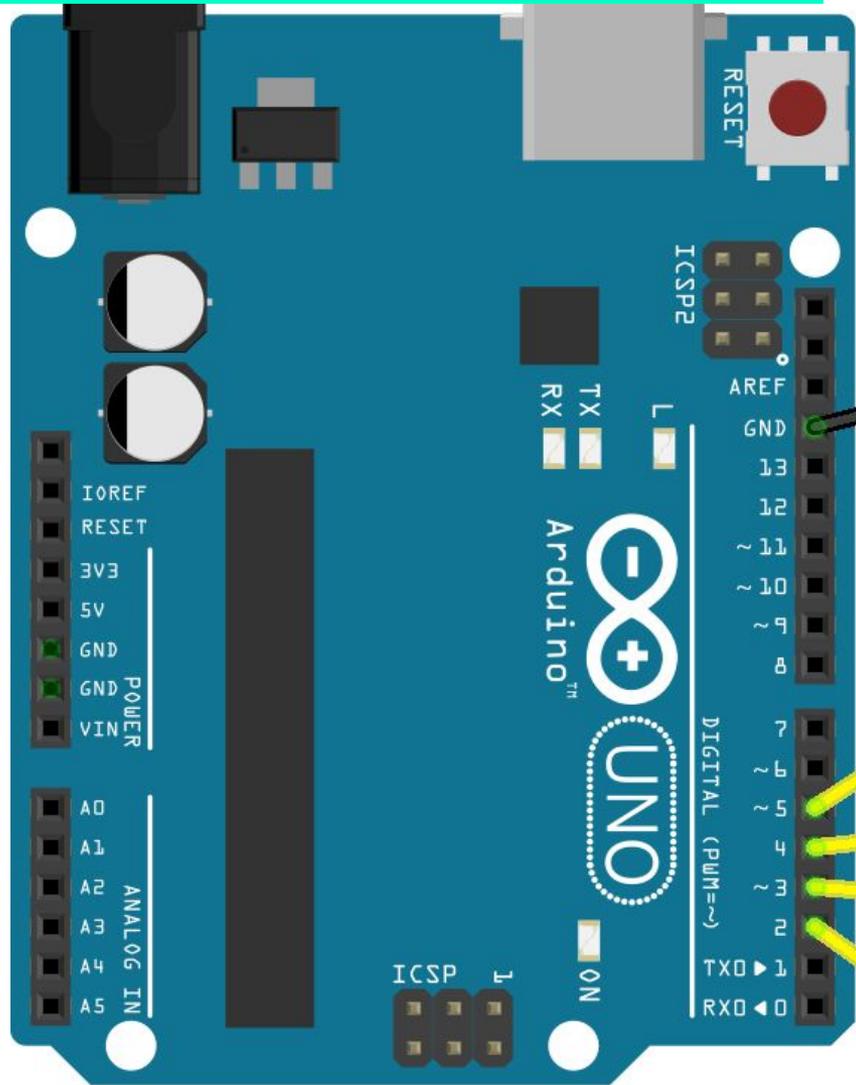
CAN YOU TURN
MULTIPLE LEDS ON &
OFF ONE AFTER
ANOTHER ?

INTRODUCING (FOR LOOPS)

It will repeat the code within the curly bracket as long as the condition is true.

HOW WOULD YOUR
CIRCUIT LOOK LIKE ?

220-ohms resistor for LED
(Red, Red, Brown, Gold)



INTRODUCING (FOR LOOPS)

```
void setup() {  
  pinMode(2, OUTPUT);  
  pinMode(3, OUTPUT);  
  pinMode(4, OUTPUT);  
  pinMode(5, OUTPUT);  
}
```

IMPORTANT TO DECLARE THE MODE OF YOUR PIN
THEY COULD BE EITHER (INPUT) OR (OUTPUT) PINS.

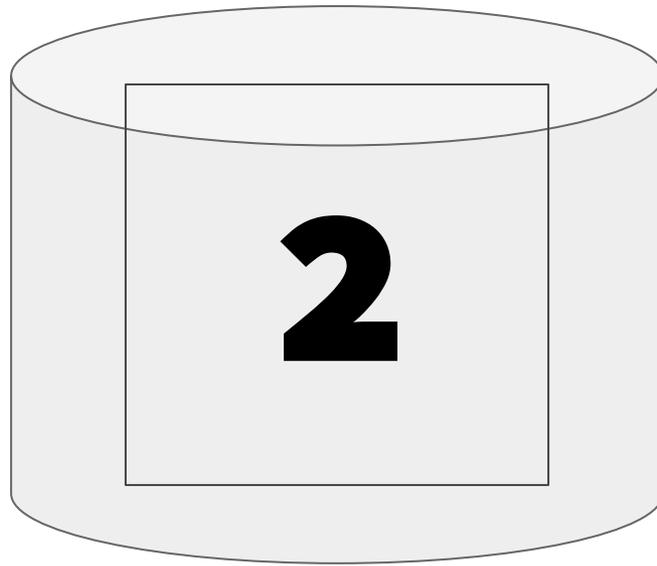
```
void loop() {  
  for ( int pin = 2; pin < 6 ; pin++ )  
  {  
    digitalWrite(pin, HIGH); // turn the LED on  
    delay(30); // wait for 30ms  
    digitalWrite(pin, LOW); // turn the LED off  
    delay(30); // wait for 30ms  
  }  
}
```

INTRODUCING (FOR LOOPS)

```
void loop()  
{  
  
  for (int pin = 2; //start counting from 2  
       pin < 6;    //as long as pin value is less than 6  
       pin ++ )   //add 1 to our last pin value  
  {  
    // TURN ON & OFF LED HERE  
  }  
  
}
```

INTRODUCING (FOR LOOPS)

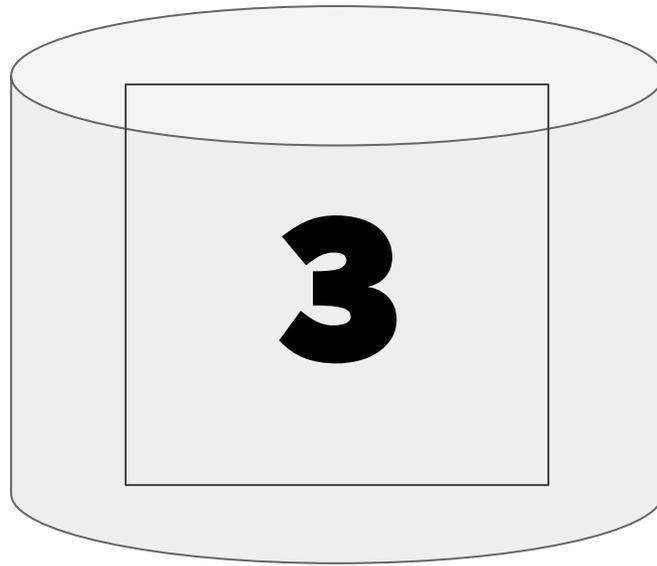
```
for (int pin = 2; pin < 6; pin ++ )
```



```
int pin = 2;
```

INTRODUCING (FOR LOOPS)

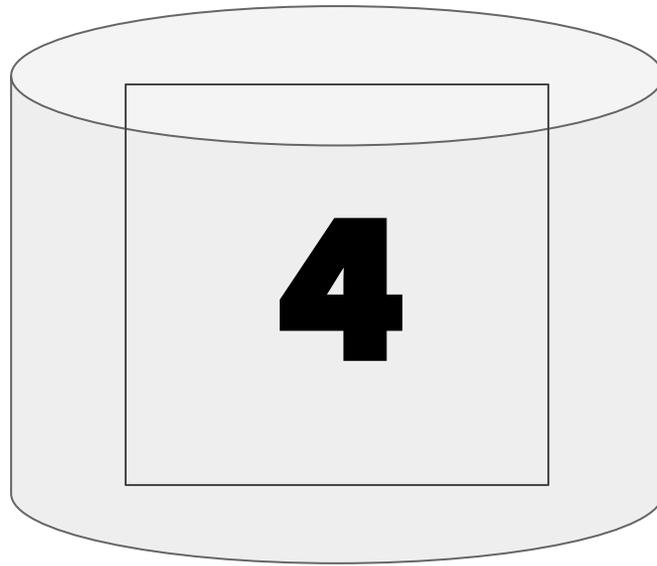
```
for (int pin = 2; pin < 6; pin ++ )
```



```
int pin = 3;
```

INTRODUCING (FOR LOOPS)

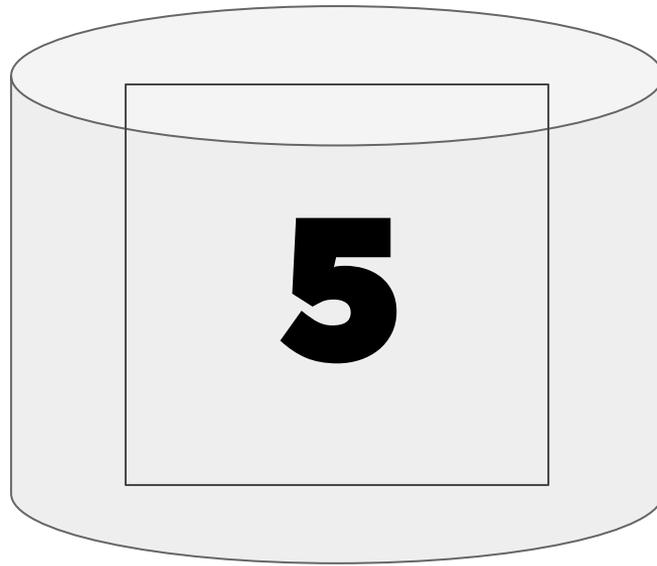
```
for (int pin = 2; pin < 6; pin ++ )
```



```
int pin = 4;
```

INTRODUCING (FOR LOOPS)

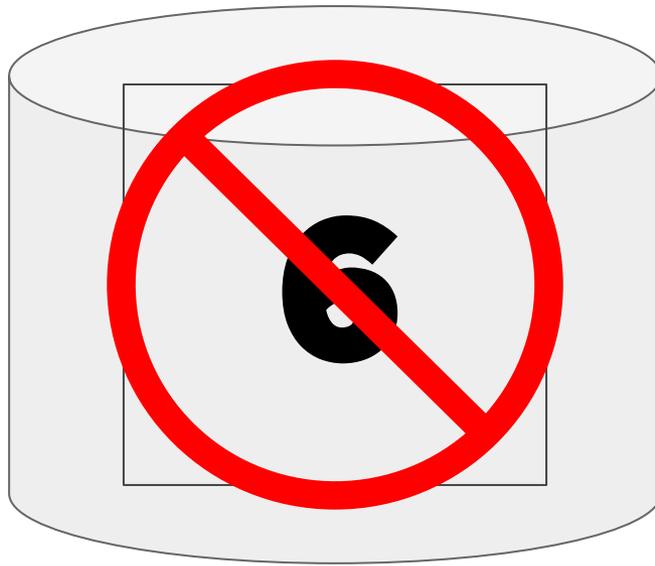
```
for (int pin = 2; pin < 6; pin ++ )
```



```
int pin = 5;
```

INTRODUCING (FOR LOOPS)

```
for (int pin = 2; pin < 6; pin ++ )
```



```
int pin = 6;
```

LET'S MAKE

SOME LEDS FX

CHALLENGE

CAN YOU FADE

ONE LED ON

GRADUALLY ?

OOPS THERE'S AN
ISSUE...

1. ARDUINO DIGITAL PINS

CAN ONLY OUTPUT 5V

IT CANNOT VARY THE

OUTPUT VOLTAGE ON ITS

PIN.

2. BUT THERE'S A WAY
AROUND IT!

3. USING PULSE WIDTH MODULATION (PWM)

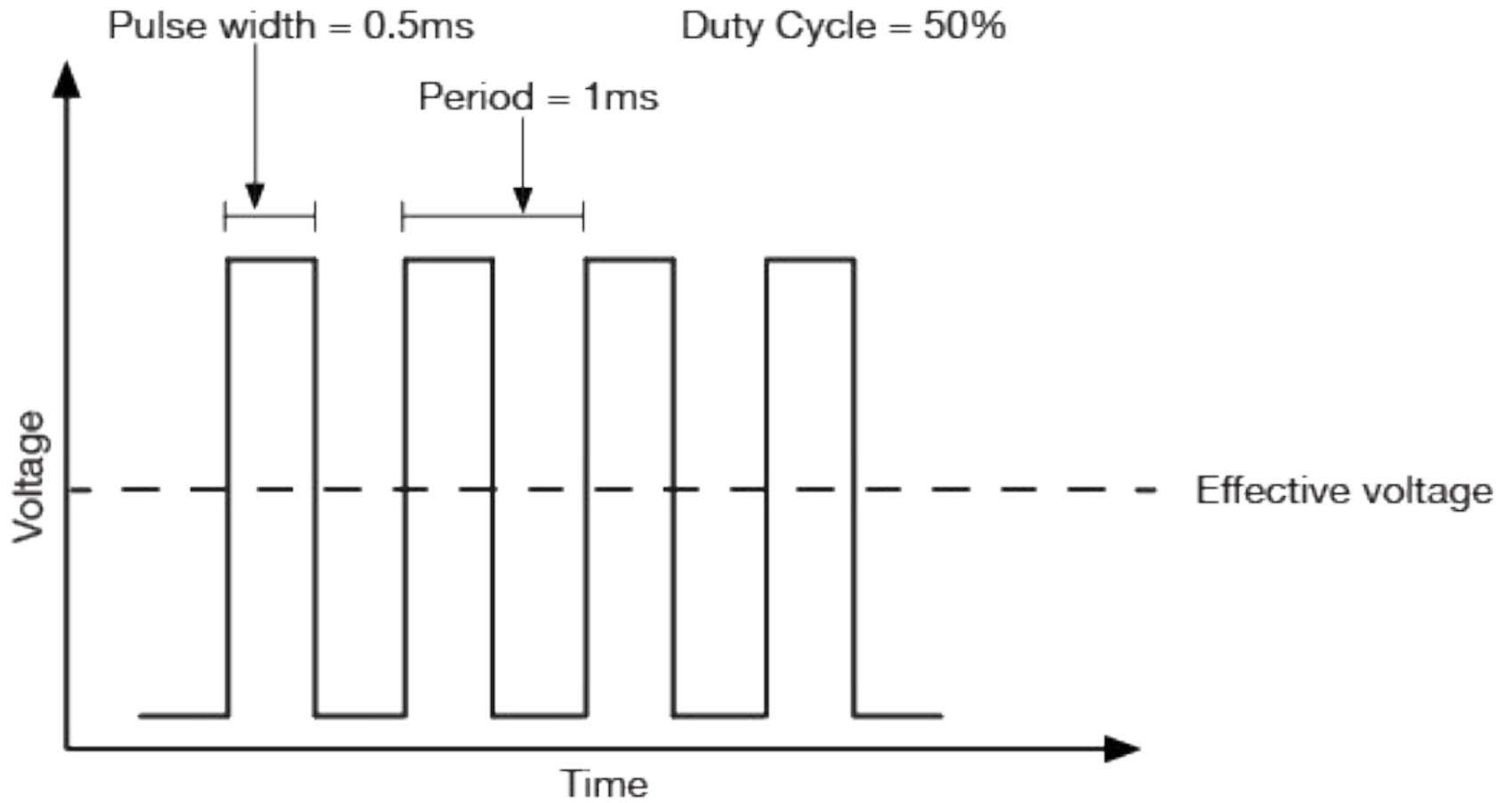
INTRODUCING

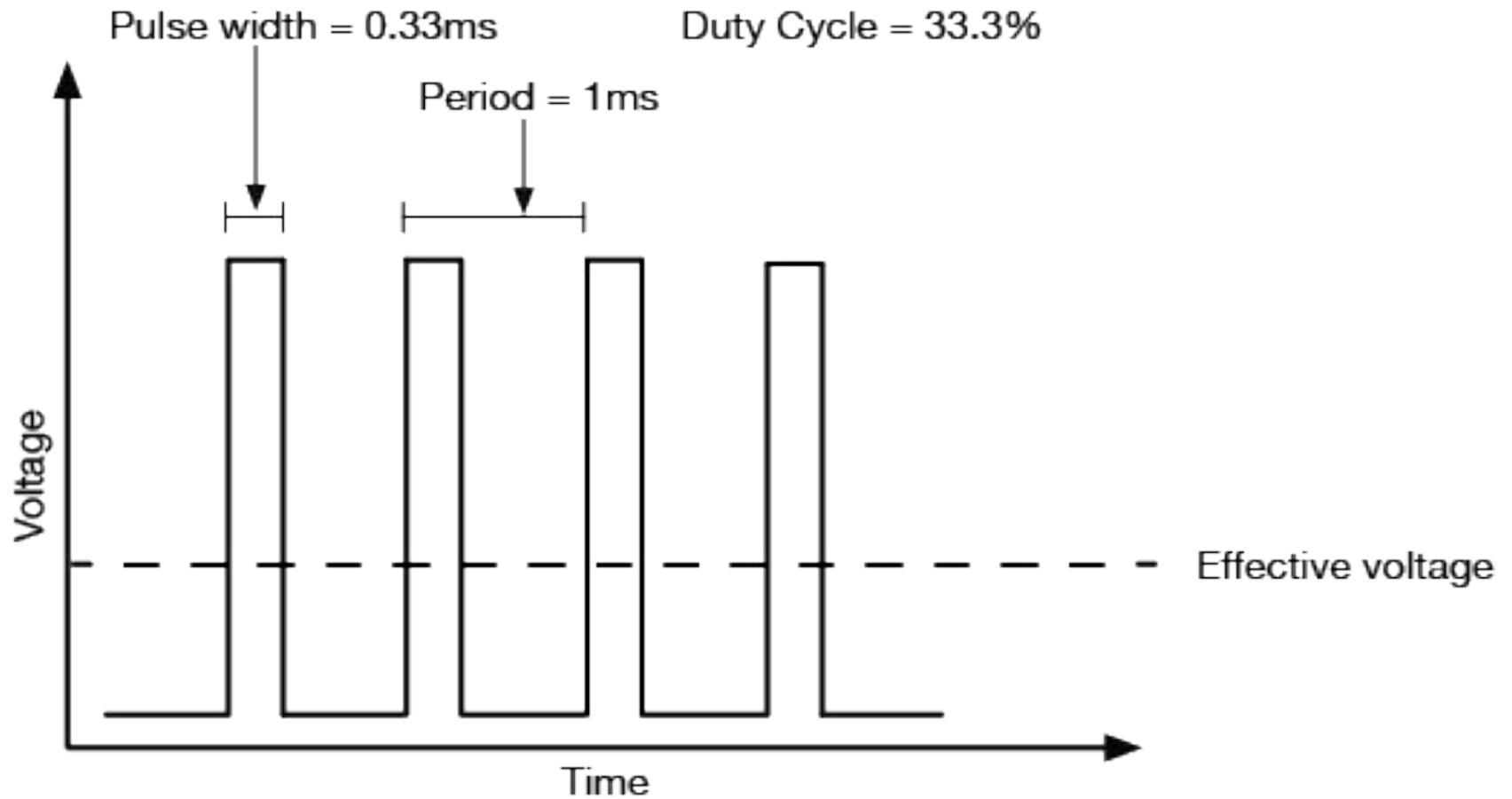
PWM

4. BY RAPIDLY CHANGING
THE PIN HIGH & LOW
FASTER THAN WHAT THE
HUMAN EYE CAN SEE.

5. JUST LIKE HOW ILLUSION
OF MOTION IS CREATED IN
ANIMATION...

6. THE % OF TIME THE PIN
IN HIGH IS CALLED DUTY
CYCLE.





```
analogWrite(pin, duty);
```

PIN REFERS TO THE PIN YOU'RE GOING TO PULSE

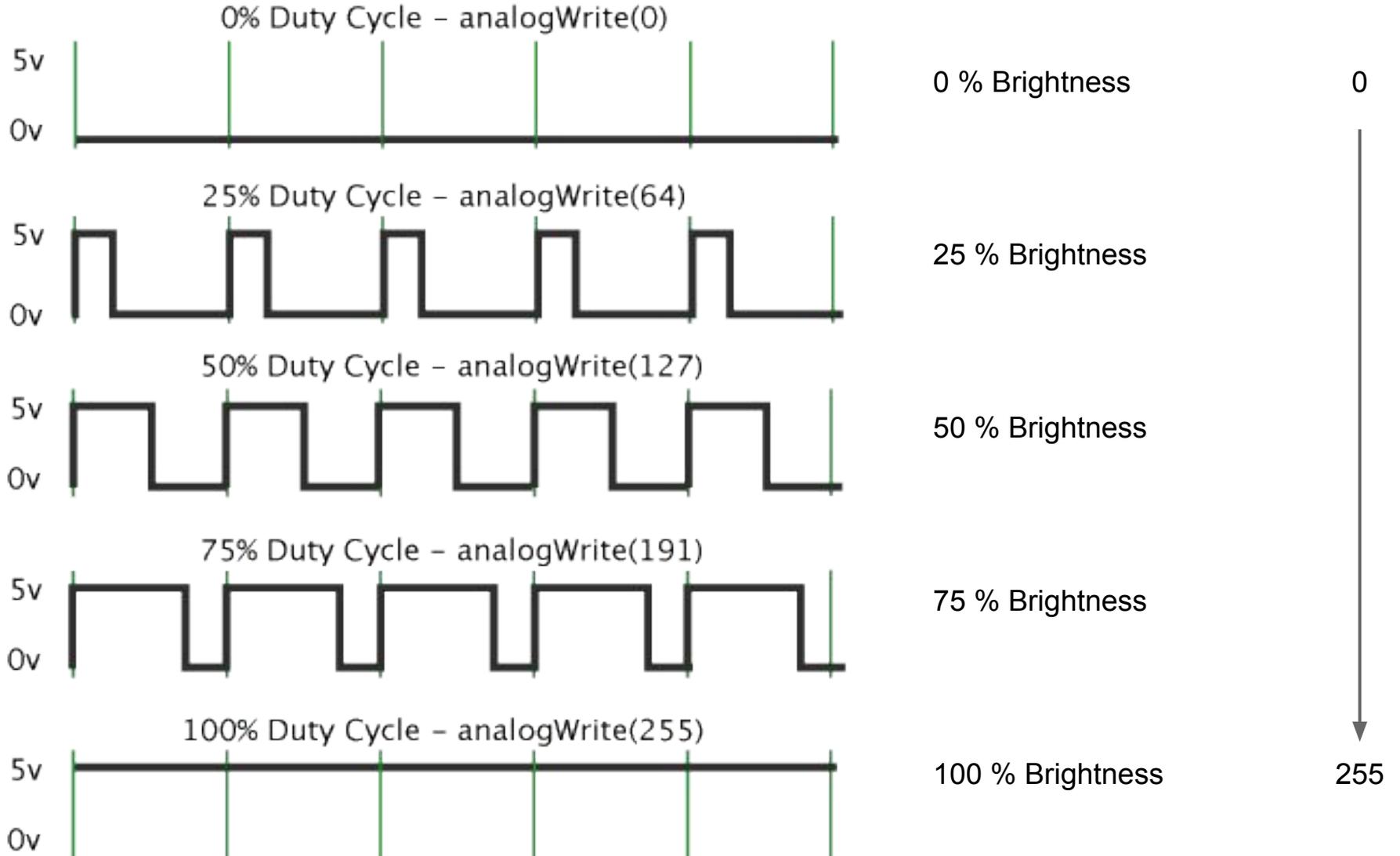
DUTY IS A VALUE FROM 0 – 255.

0 CORRESPONDS TO 0 VOLTS

255 CORRESPONDS TO 5 VOLTS

Every change of one point changes the pseudo-analog output voltage by $5/255$, or 0.0196 volts.

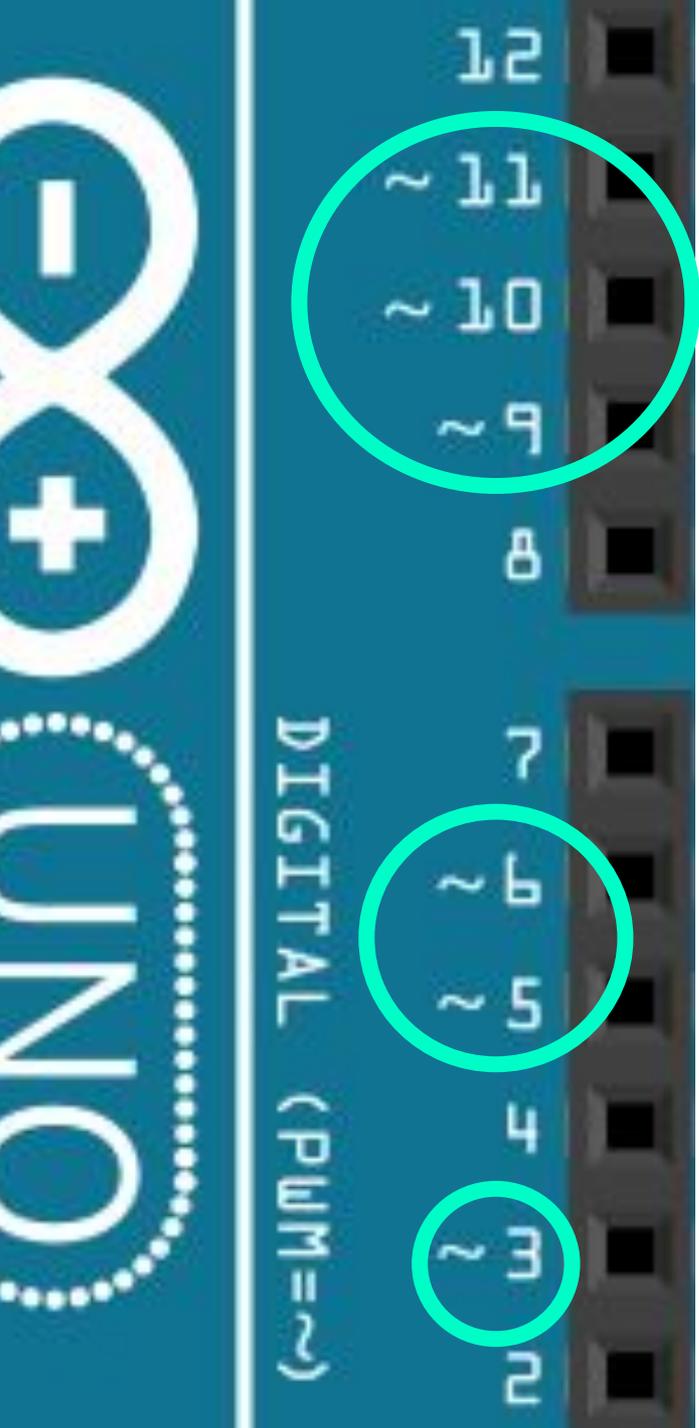
Pulse Width Modulation



7. WHEN THE PIN IS HIGH FOR HALF THE PERIOD AND LOW FOR THE OTHER HALF, THE DUTY CYCLE IS 50%.

IMPORTANT!

A LOWER DUTY CYCLE GIVES
YOU A DIMMER LED THAN A
HIGHER DUTY CYCLE



THERE ARE 6 PINS SET
ASIDE FOR PWM

3, 5, 6, 9, 10, & 11

LOOK FOR ~ NEXT
TO THEIR NUMBERS.

INTRODUCING (PWM)

```
int brightness = 0;
int fadeAmount = 5;
int ledPin = 3;

void setup()
{
  pinMode(ledPin, OUTPUT);
}

void loop() {

  for (brightness = 0; brightness <= 255 ; brightness += fadeAmount)
  {
    analogWrite(ledPin, brightness);
    delay(30);
  }

}
```

CHALLENGE

CAN YOU FADE

ONE LED OFF

GRADUALLY ?

INTRODUCING (PWM)

```
int brightness = 0;
int fadeAmount = 5;
int ledPin = 3;

void setup()
{
  pinMode(ledPin, OUTPUT);
}

void loop() {
  for (brightness = 255; brightness >= 0; brightness -= fadeAmount)
  {
    analogWrite(ledPin, brightness);
    delay(30);
  }
}
```

CHALLENGE

CAN YOU FADE
ONE LED ON &
FADE OFF
GRADUALLY ?

```
int brightness = 0;
int fadeAmount = 5;
int ledPin = 3;

void setup()
{
  pinMode(ledPin, OUTPUT);
}

void loop() {
  for (brightness = 0; brightness <= 255 ; brightness += fadeAmount)
  {
    analogWrite(ledPin, brightness);
    delay(30);
  }

  for (brightness = 255; brightness >= 0; brightness -= fadeAmount)
  {
    analogWrite(ledPin, brightness);
    delay(30);
  }
}
```

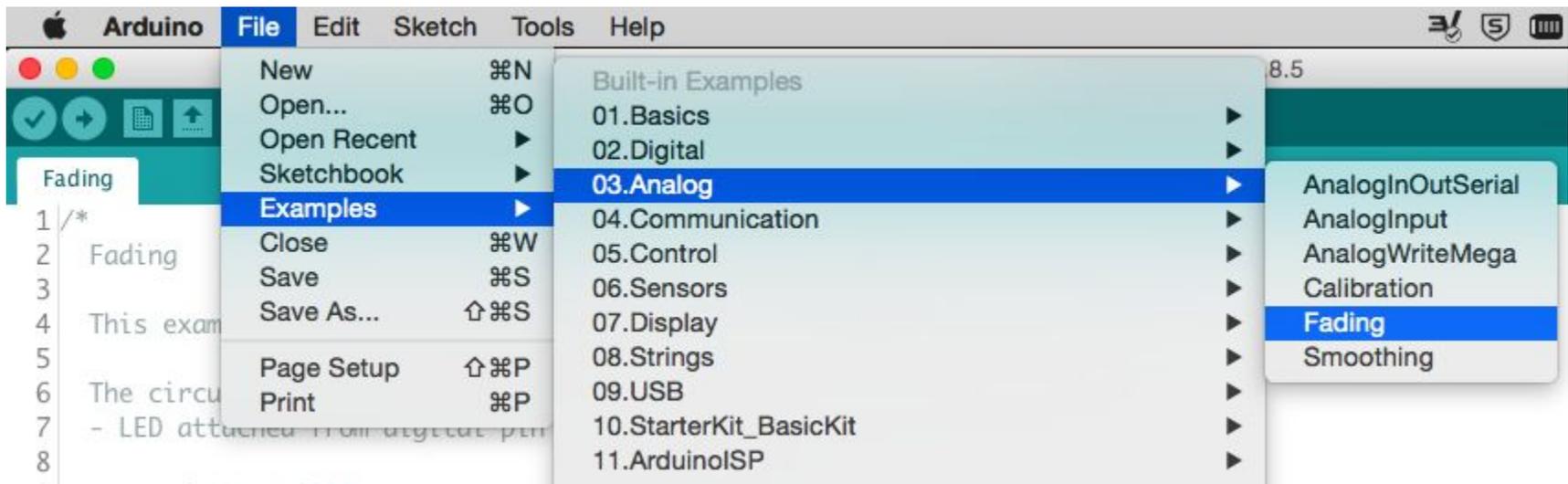
----- shortcut arithmetic -----

++ → add 1 to the last value

+= 5 → add 5 to the last value

-= 5 → minus 5 from the last value

YOU CAN FIND THE EXAMPLE IN ARDUINO
FILE / EXAMPLES / 03. ANALOG / FADING

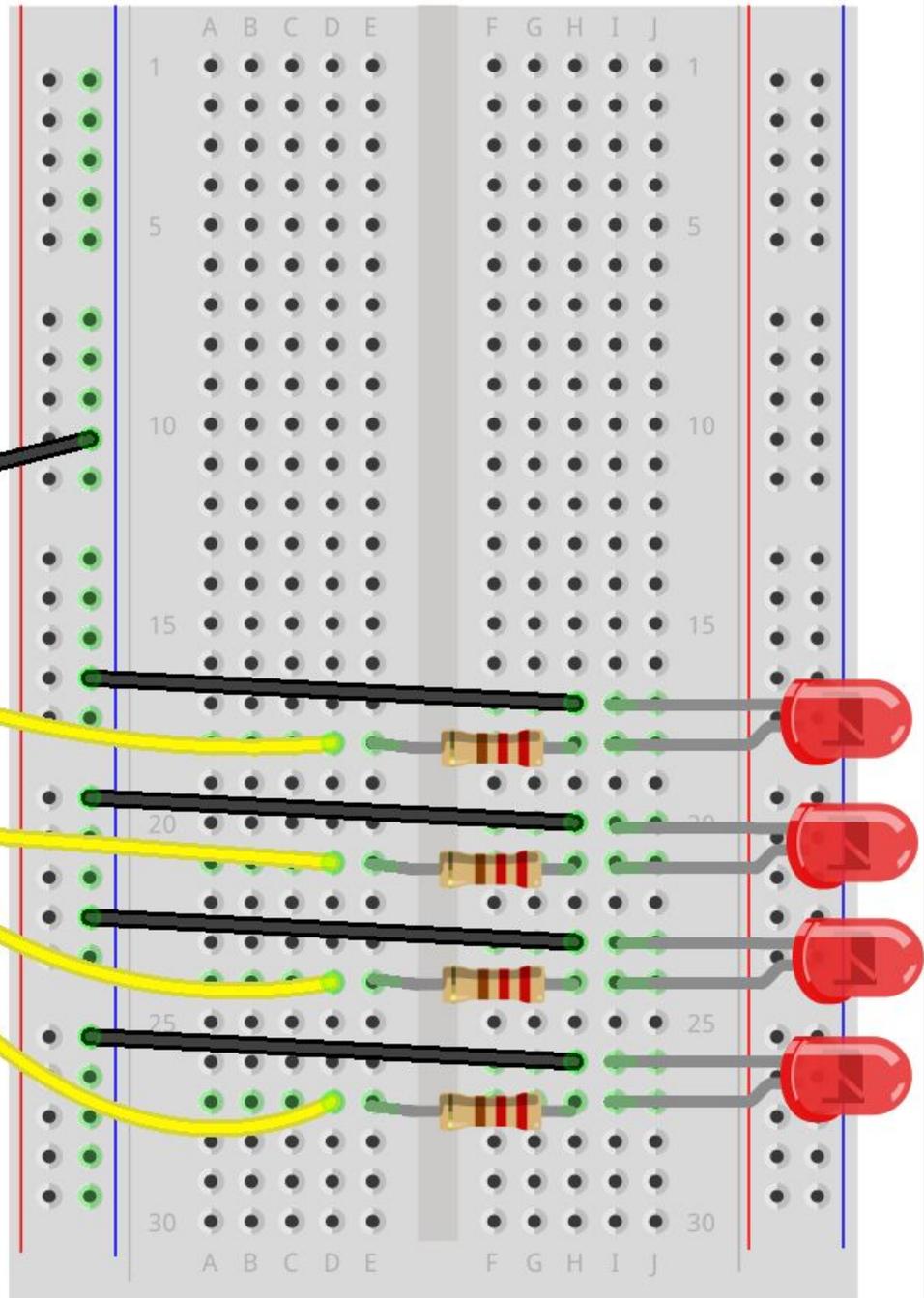
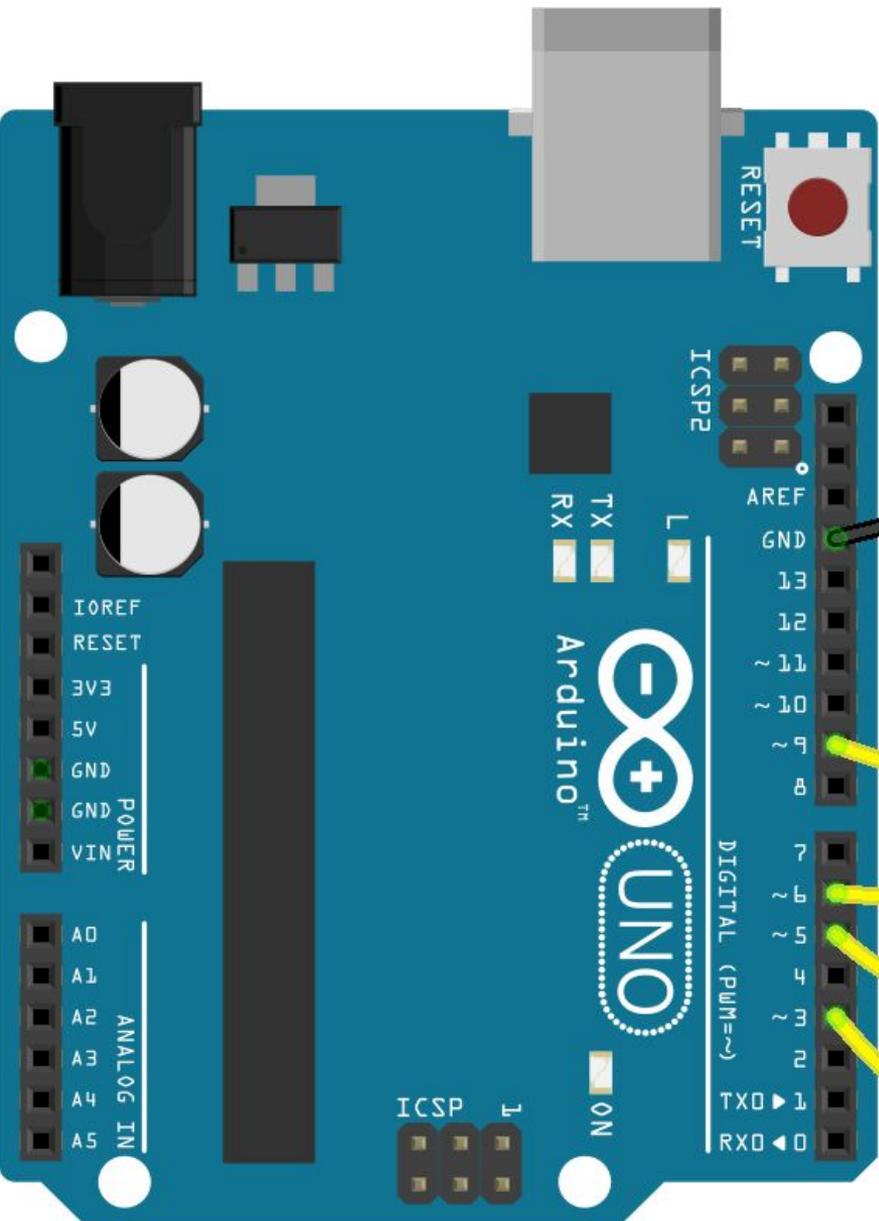


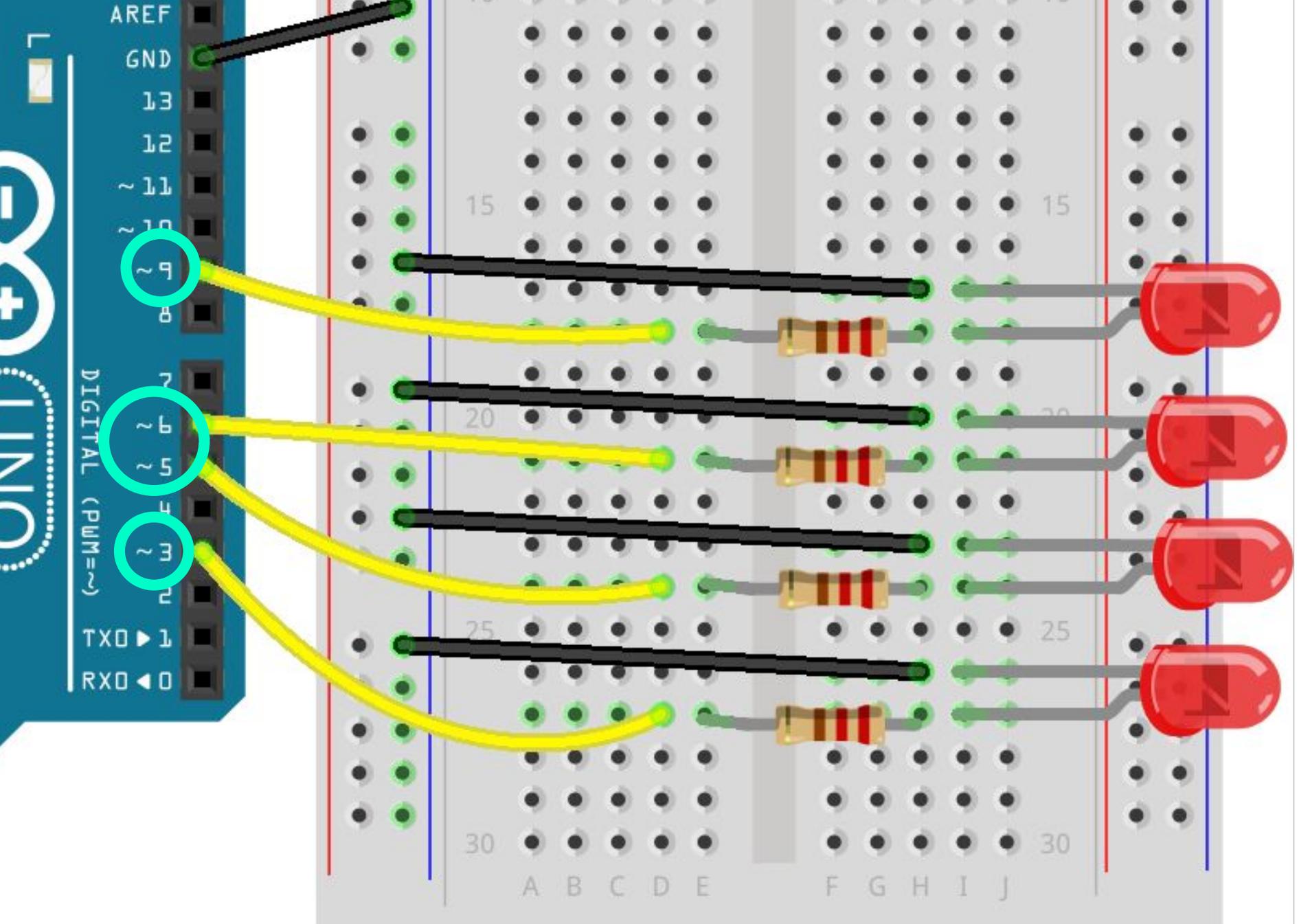
CHALLENGE

CAN YOU FADE ON
& FADE OFF ALL 4
LEDS?

NOTE:

WE NEED TO USE
SPECIFIC ARDUINO
PINS. WHICH ONES?





*

INTRODUCING ARRAYS

*

BEYOND FOR LOOPS

* INTRODUCING ARRAYS *

```
// this is a variable declaration  
// int myPin = 3;
```

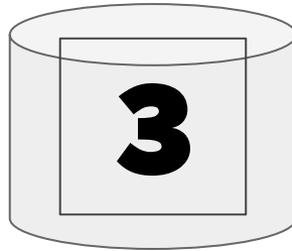
```
// this is a array declaration  
int myPin[] = {3, 5, 6, 9};
```

```
void setup() {  
    Serial.begin(9600);  
  
    Serial.println(myPin[0]);  
    Serial.println(myPin[1]);  
    Serial.println(myPin[2]);  
    Serial.println(myPin[3]);  
}
```

```
void loop () {  
}
```

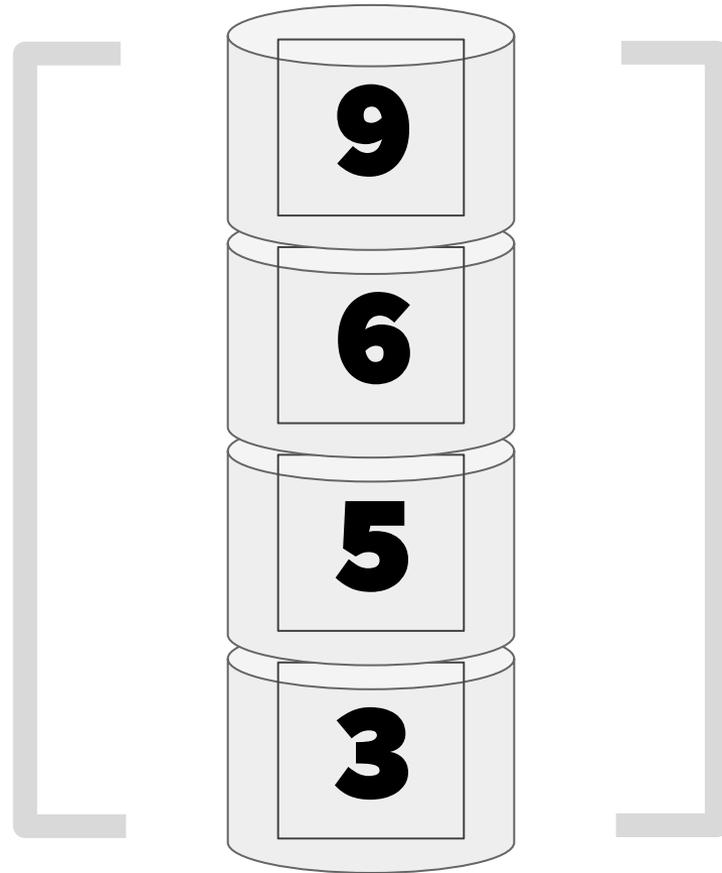
* RECAP VARIABLES *

THIS IS A
VARIABLE
DECLARATION



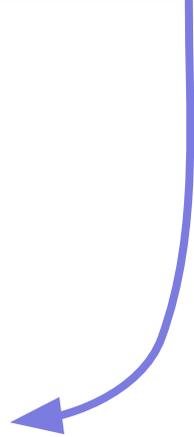
`int myPin = 3;`

* INTRODUCING ARRAYS *

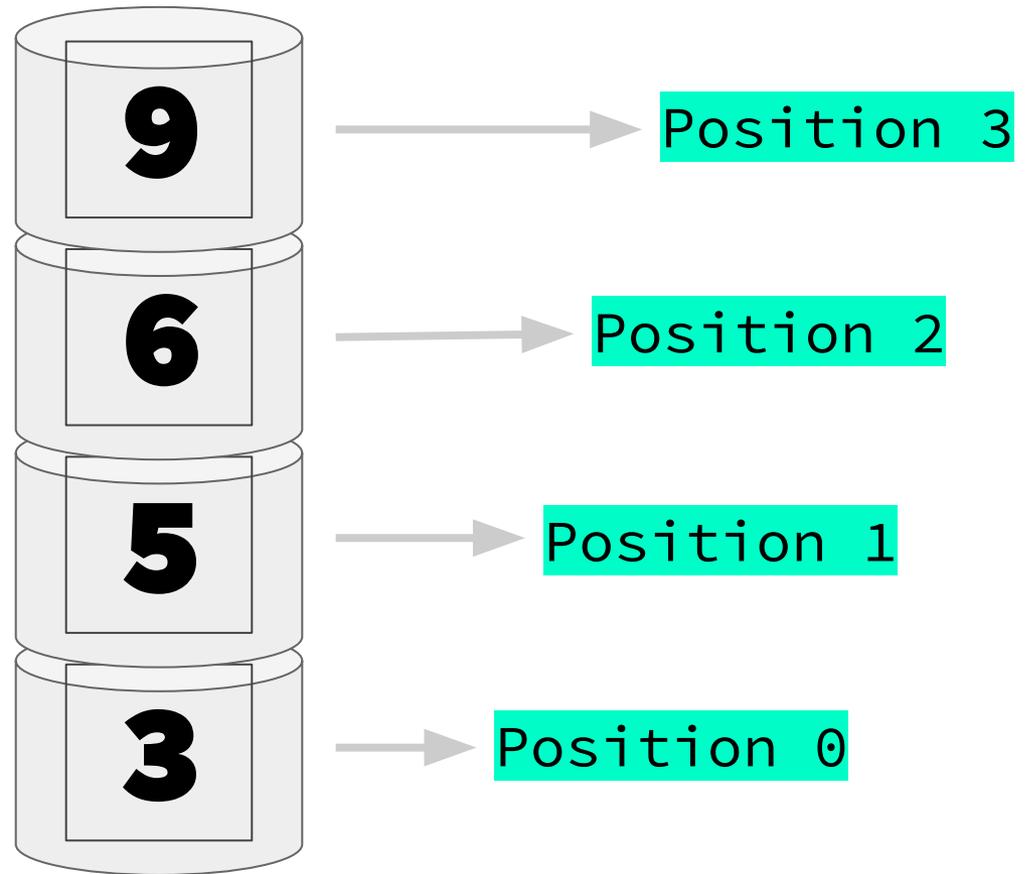


THIS IS AN
ARRAY
DECLARATION

```
int myPin[] = {3, 5, 6, 9};
```



* INTRODUCING ARRAYS *



```
int myPin[] = {3, 5, 6, 9};
```

```
int brightness = 0;
int fadeAmount = 5;
int ledPin[] = {3, 5, 6, 9};

void setup() {
  for (int i = 0; i < 4; i++) {
    pinMode(ledPin[i], OUTPUT);
  }
}

void loop() {

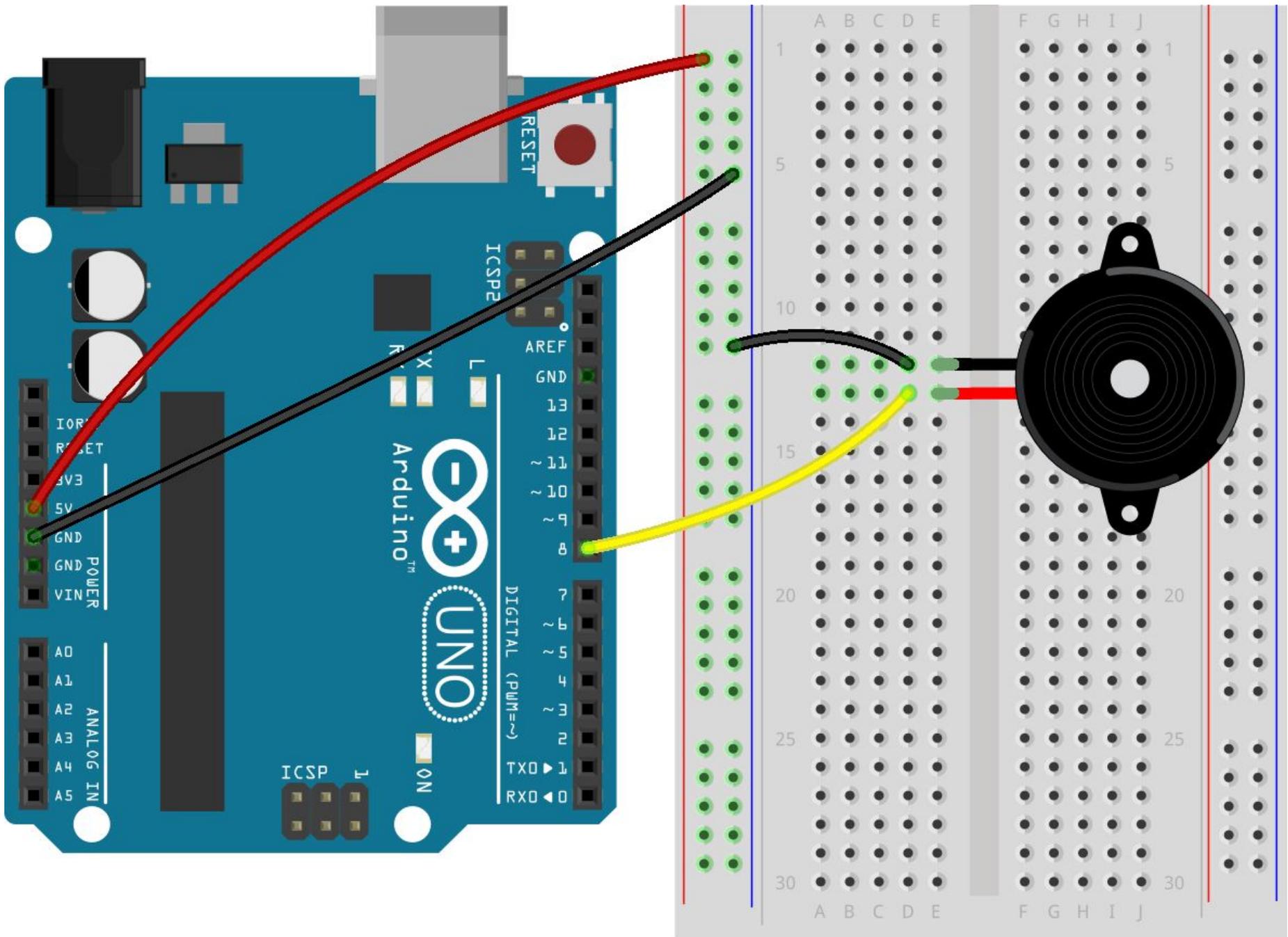
  for (int i = 0; i < 4; i++) {

    for (brightness = 0; brightness <= 255; brightness += fadeAmount)
      {
        analogWrite(ledPin[i], brightness);
        delay(30);
      }

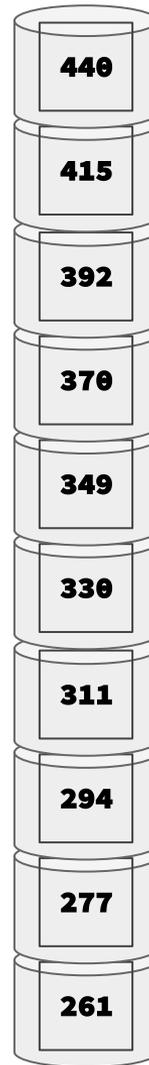
    for (brightness = 255; brightness >= 0; brightness -= fadeAmount)
      {
        analogWrite(ledPin[i], brightness);
        delay(30);
      }
  }
}
```

LET'S MAKE
SOME SOUND!

PLAYING A TONE



* INTRODUCING ARRAYS *



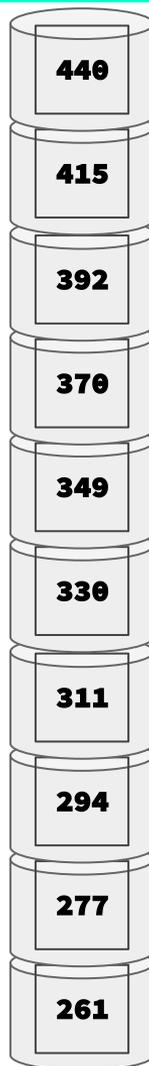
```
int tones[] = {261, 277, 294, 311, 330, 349, 370, 392, 415, 440};
```

* INTRODUCING ARRAYS *

To play a note of a particular pitch, you specify the frequency.

The different frequencies for each note are kept in an array.

An array is like a list. So, a scale can be played by playing each of the notes in the list in turn.



```
int tones[] = {261, 277, 294, 311, 330, 349, 370, 392, 415, 440};
```

```
int speakerPin = 8;

int numTones = 10;
int tones[] = {261, 277, 294, 311, 330, 349, 370, 392, 415, 440};

//           mid C  C#  D   D#  E   F   F#  G   G#  A

void setup() {
  for (int i = 0; i < numTones; i++) {
    tone(speakerPin, tones[i]);
    delay(500);
  }
  noTone(speakerPin);
}

void loop() {
}
```

The 'for' loop will count from 0 to 9 using the variable 'i'.

To get the frequency of the note to play at each step, we use 'tones[i]'.

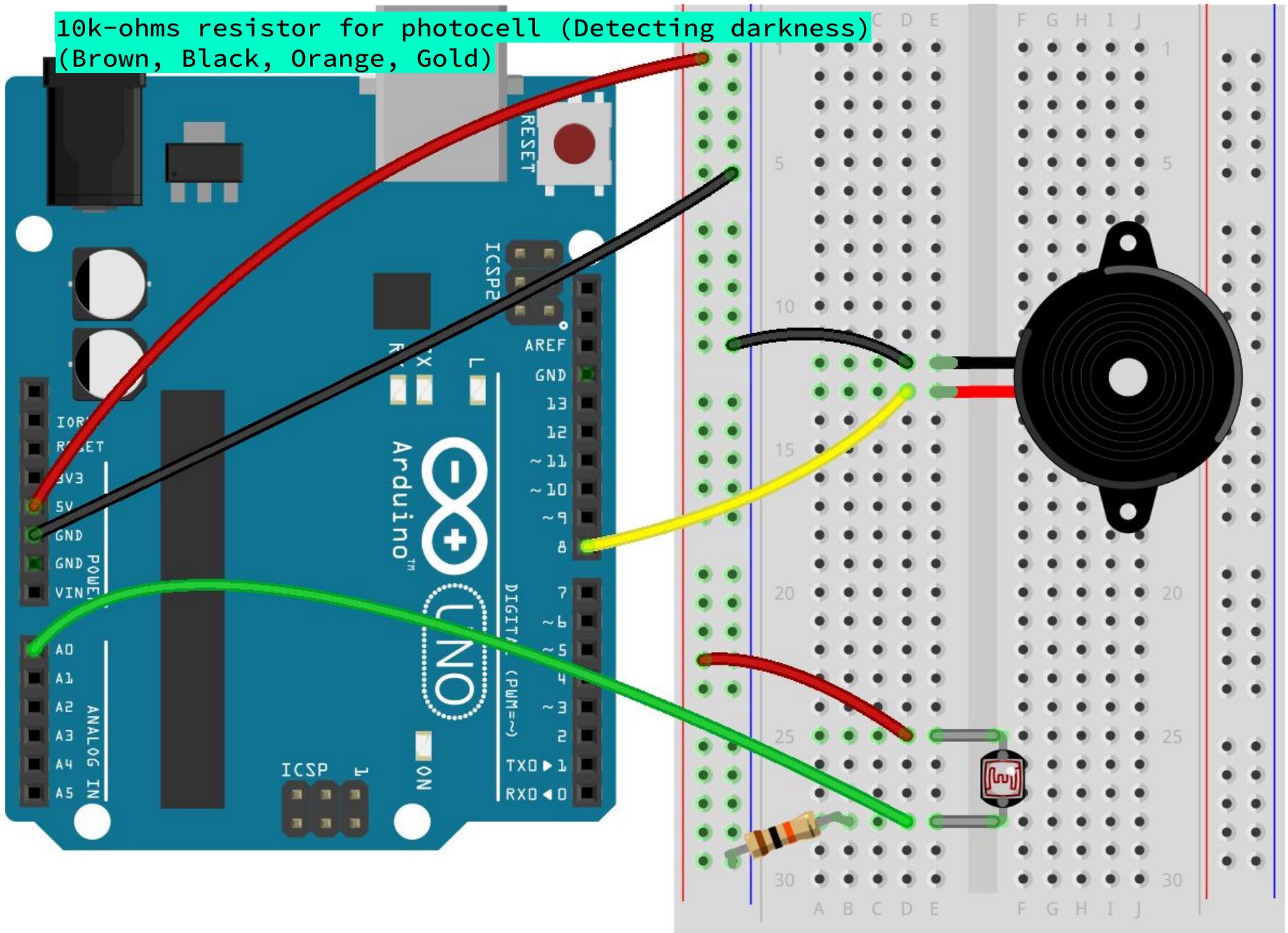
This means, the value in the 'tones' array at position 'i'.

So, for example, 'tones[0]' is 261, 'tones[1]' is 277 etc.

To play the tune again, just press the reset button on Arduino.

LIGHT THEREMIN
(CONTROL SOUND
WITH LIGHT!)

10k-ohms resistor for photocell (Detecting darkness)
(Brown, Black, Orange, Gold)



```
int speakerPin = 8;  
int photocellPin = 0;
```

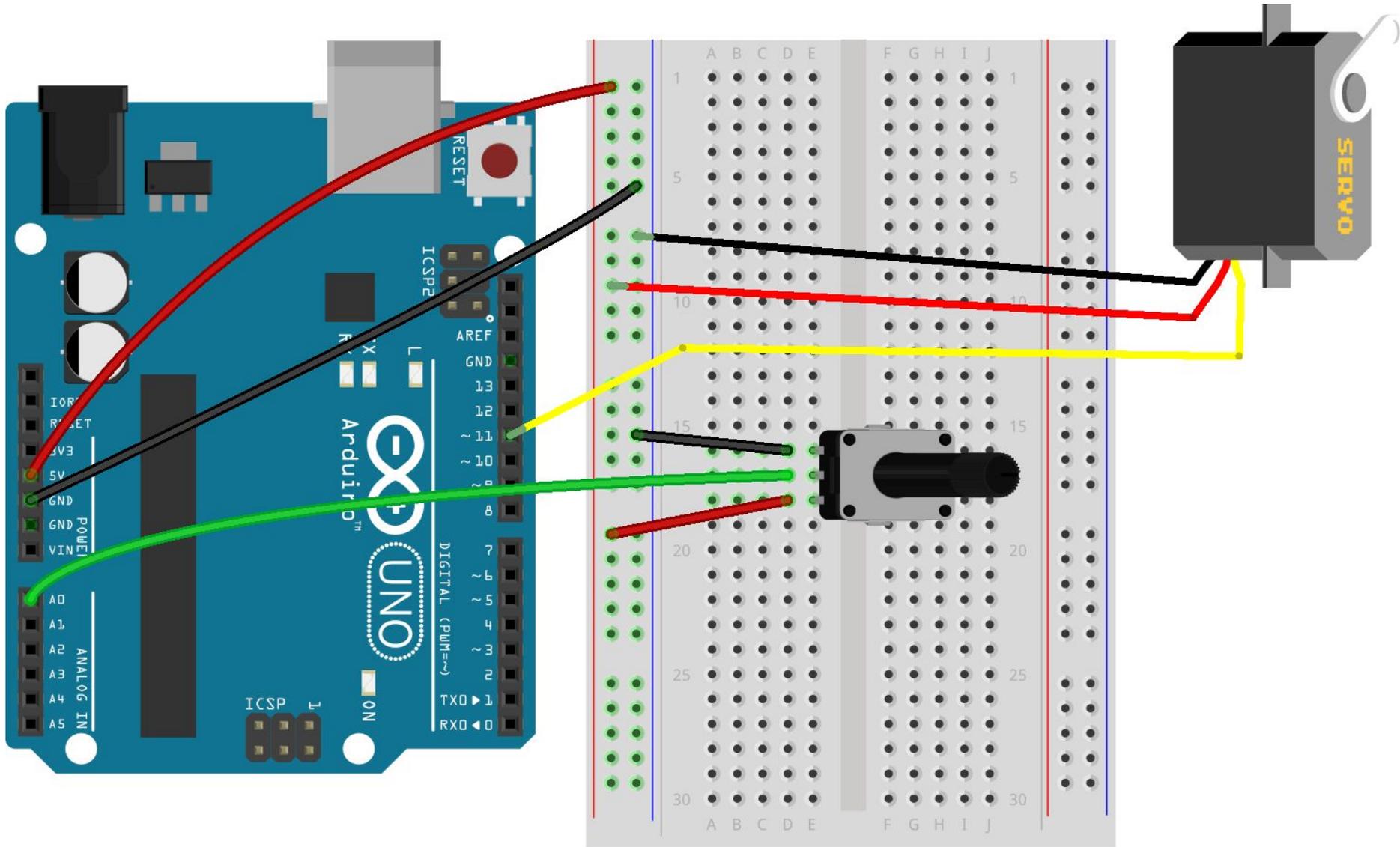
```
void setup()  
{  
}
```

```
void loop()  
{  
  int reading = analogRead(photocellPin);  
  int pitch = map(reading, 0, 1023, 200, 500);  
  tone(speakerPin, pitch);  
}
```

Read from photocell



TURNING A SERVO MOTOR



```
#include <Servo.h>
```

```
int potPin = 0;
```

```
int servoPin = 11;
```

```
Servo servo;
```

```
void setup() {  
    servo.attach(servoPin);  
}
```

Attach servo motor
to Pin 11



```
void loop() {  
    int reading = analogRead(potPin);    // 0 to 1023  
    int angle = reading / 6;             // 0 to 180-ish  
    servo.write(angle);  
}
```

LET'S DO A
QUICK RECAP

RECAP

DIGITALWRITE()

- We use `digitalWrite()` to turn the LED ON or OFF.
- Basically we are telling Arduino to `set` the voltage to **HIGH** (ON) or to **LOW** (OFF)

RECAP

DIGITALREAD()

- We use `digitalRead()` to know if the pushbutton switch is pressed or released.
- It is used to `read` the state of a switch.

RECAP

(1. VARIABLES)

- Imagine variable is like a bucket.
- We can put content into this bucket.
- Each bucket can only contain one item at each time.
- We can change the (type of) content of the bucket.

RECAP

(2.1 FOR LOOPS)

- It will repeat the code within the curly bracket as long as the condition is true.
- We use for loops to repeat specific tasks
- E.g. `for (int a = 0; a < 10; a++) { Serial.print(.); }`

RECAP

(2.2 ARRAYS)

- `int myPin[] = {3, 5, 6, 9};`
- `int tones[] = {261, 277, 294, 311, 330};`

RECAP

(3. PWM)

*IF YOUR OUTPUT REQUIRES A RANGE OF ACTIONS
RATHER THAN ON/OFF STATES USE ANALOGWRITE()

- Only Pins marker tilde (~) are PWM capable pins.
- We use `analogWrite()` to change the brightness of LEDs.
- We can also use `analogWrite()` to move servo motors and create sounds on piezo buzzers.

RECAP

(4. TONE)

- `tone(pin, frequency)`
- `tone(pin, frequency, duration)`

RECAP

(4. SERVO)

- `#include <Servo.h>`
- `Servo myservo; //Create servo object`
- `myservo.attach(9); //Attach servo on pin 9`