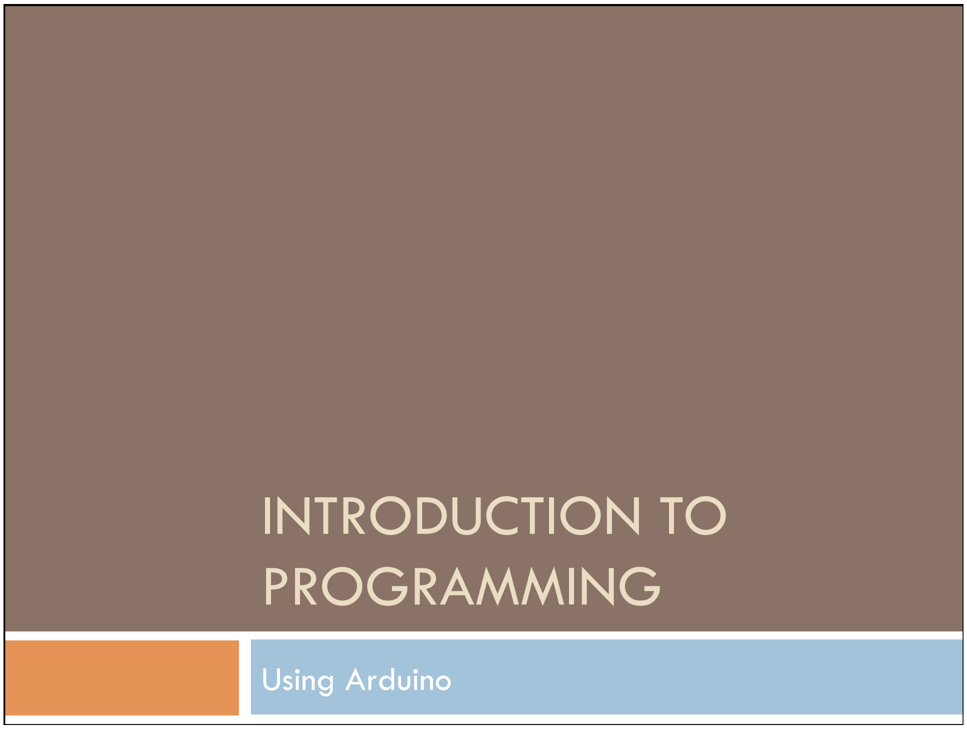
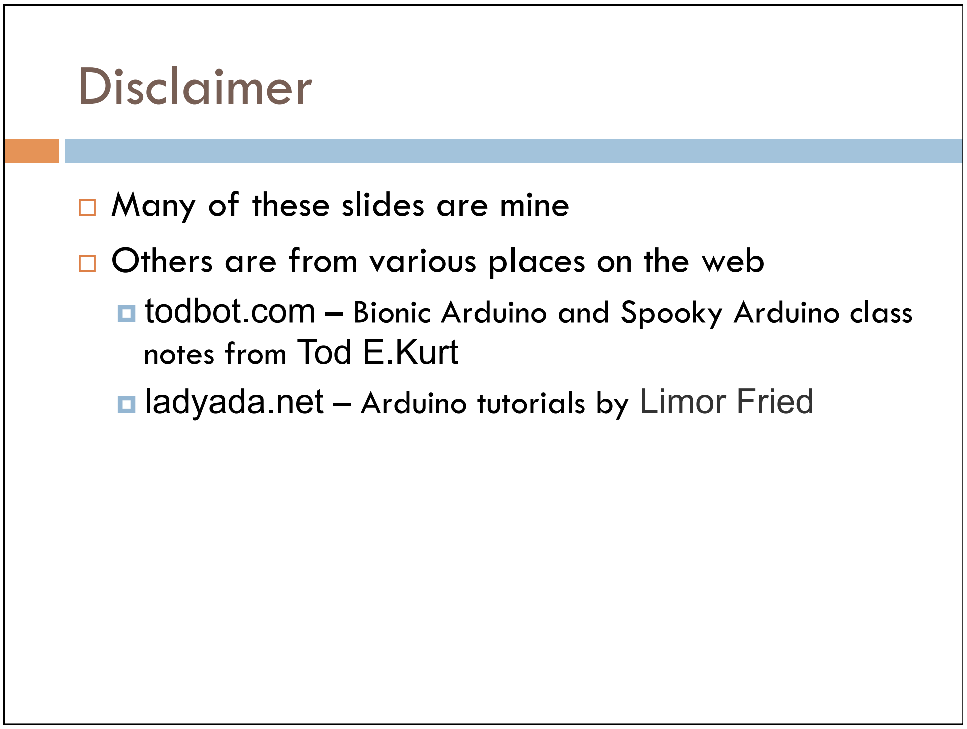


INTRODUCTION TO PROGRAMMING

Using Arduino



Disclaimer

- Many of these slides are mine
- Others are from various places on the web
  - todbot.com – Bionic Arduino and Spooky Arduino class notes from Tod E.Kurt
  - ladyada.net – Arduino tutorials by Limor Fried

What is a program?

- Essentially just a list of actions to take
  - Each line of the program is step to take
  - The program just walks through the steps one at at time
    - Maybe looping too
- It's like a recipe!

Meatloaf…

Meatloaf Recipe
Ingredients:

1 package Lipton Onion Soup Mix
2 pounds lean ground beef
1 large egg
2/3 cup milk
3 Tablespoons catsup
3 Tablespoons brown sugar
1 Tablespoon yellow mustard

Meatloaf…

Directions:

1. Preheat the oven to 350 degrees F.

2. Mix the onion soup mix, ground beef, egg and milk together.

3. Form the combination into a loaf shape in a 13 X 9 X 2 loaf pan.

4. Combine the rest of the ingredients and spoon onto the top of the meatloaf.

5. Bake uncovered, for about an hour.

6. When done, take the meatloaf out of the pan and place on a serving plate. Let stand for 10 minutes before slicing.

Shampoo

1. Lather
2. Rinse
3. Repeat

□ When do you stop?

Shampoo

1. Lather
2. Rinse
3. If this is the first lather, then Repeat
   else stop and towel off

Shampoo

1. Repeat twice {
2. Lather
3. Rinse
4. }

Shampoo

```
For (count=1; count<3; count=count+1)
{
    Lather
    Rinse
}
```

Shampoo

```
For (count=1; count<3; count=count+1)
{
    Lather
    Rinse
}
```

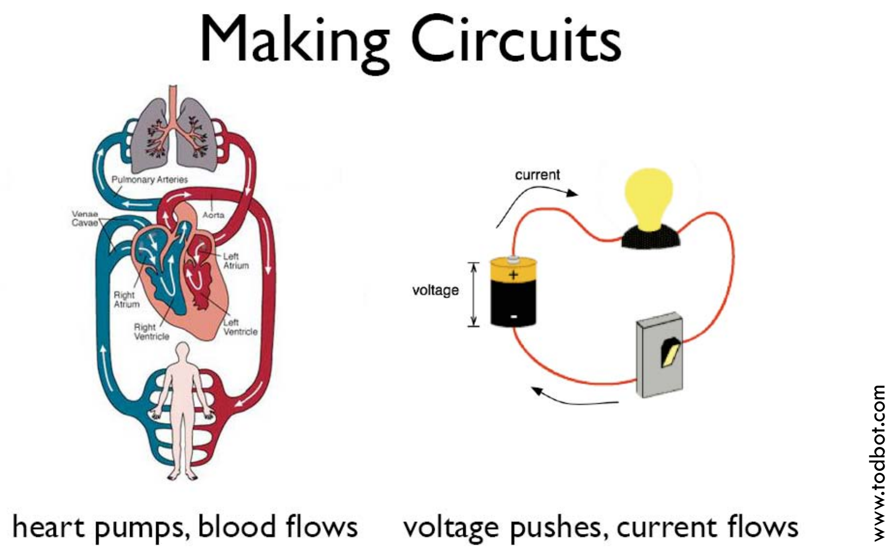
count=1
lather
rinse
count=2
lather
rinse
count=3
continue to next instruction…

Make a light flash

1. Turn light on
2. Wait for 1 second
3. Turn light off
4. Wait for one second
5. repeat

□ We'll come back to this… Let's talk about lights

Electricity



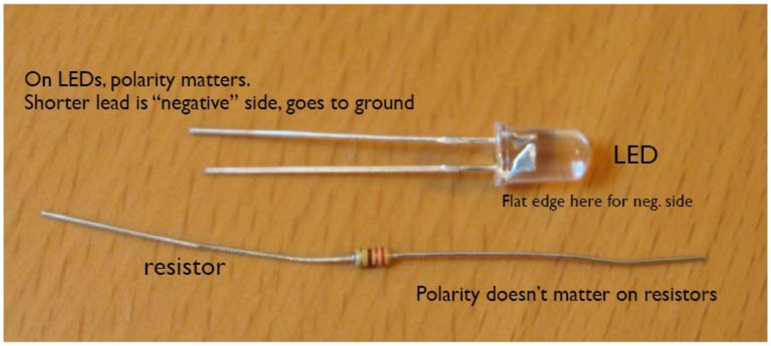
Making Circuits

heart pumps, blood flows

voltage pushes, current flows

www.todbot.com

LEDs and Resistors



On LEDs, polarity matters.
Shorter lead is "negative" side, goes to ground

LED

Flat edge here for neg. side

resistor

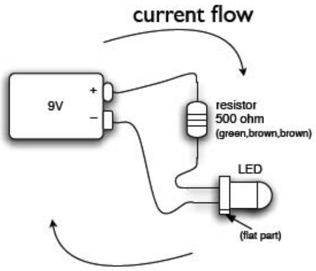
Polarity doesn't matter on resistors

long lead
Anode +

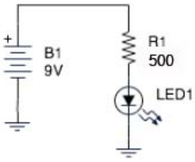
short lead
Cathode -

Current flows from Anode to Cathode
Lights up when current flows

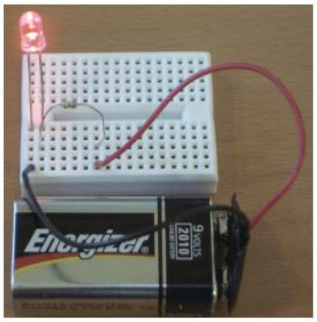
Wiring it up



wiring diagram



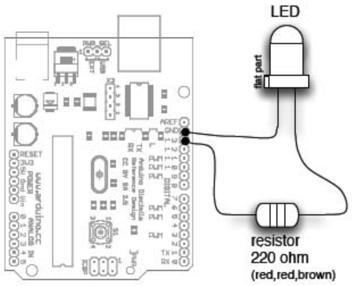
schematic



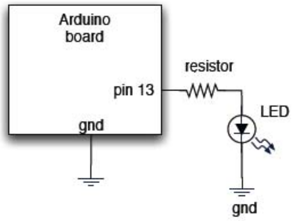
wiring it up

Electricity flows in a loop. Can stop flow by breaking the loop

Wiring it Up

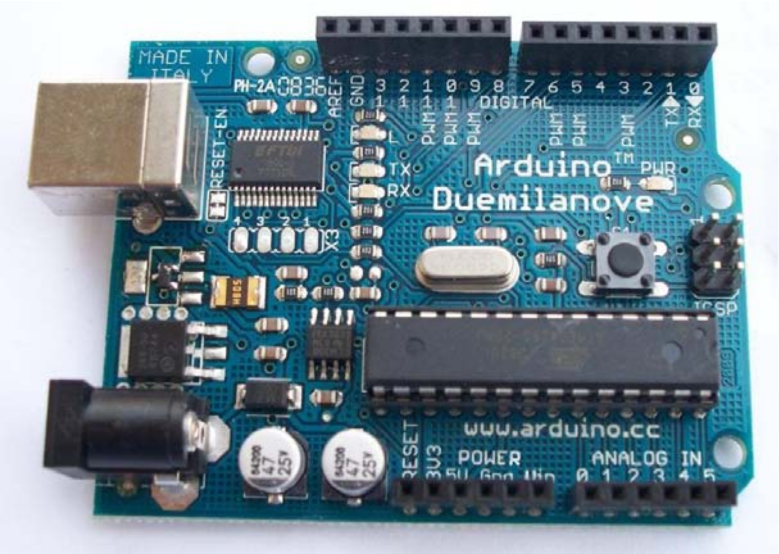


wiring diagram



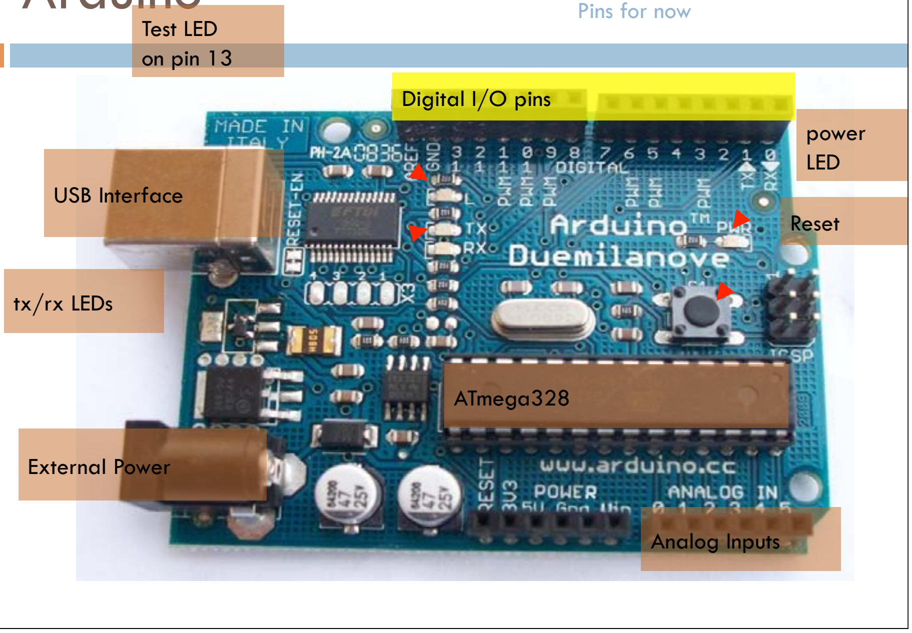
schematic

Arduino

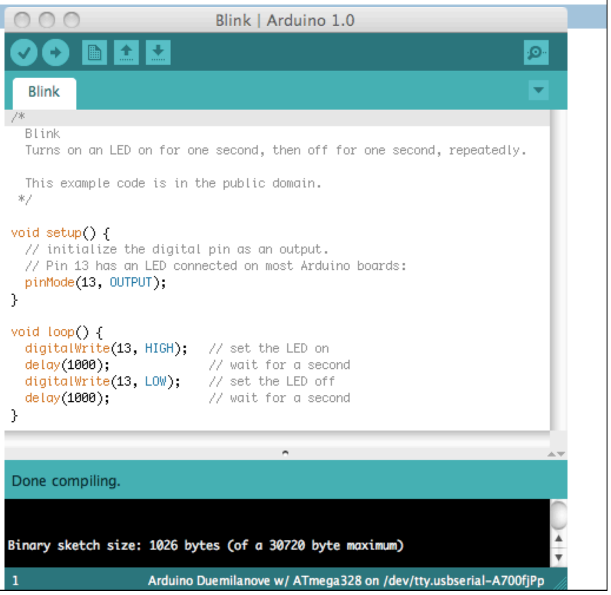


Arduino

Focus on these Digital Pins for now



Arduino Programming



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

  This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // set the LED off
  delay(1000);              // wait for a second
}
```

Done compiling.

```
Binary sketch size: 1026 bytes (of a 30720 byte maximum)
```

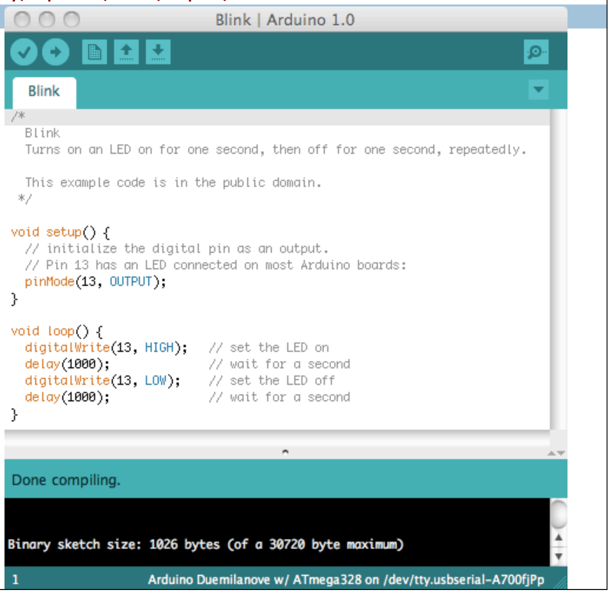
1 Arduino Duemilanove w/ ATmega328 on /dev/tty.usbserial-A700fjPp

Arduino Programming

Verify, Upload, New, Open, Save

Programming area

Notification area



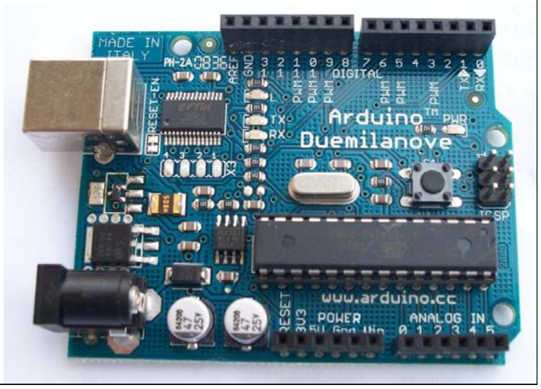
```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // set the LED off
  delay(1000);              // wait for a second
}
```

```
Done compiling.

Binary sketch size: 1026 bytes (of a 30720 byte maximum)
```

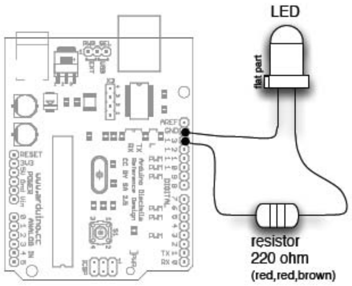
```
1                Arduino Duemilanove w/ ATmega328 on /dev/tty.usbserial-A700fjPp
```

Digital Pins

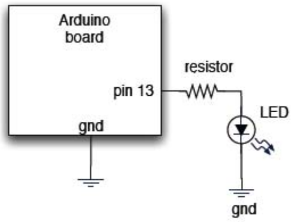
- Each of the digital pins can be set to one of two values
  - High and Low (+5v and 0v)
  - digitalWrite(<pin-number>, <value>);
  - digitalWrite(13, HIGH);
  - digitalWrite(13, LOW);



Wiring it Up



wiring diagram

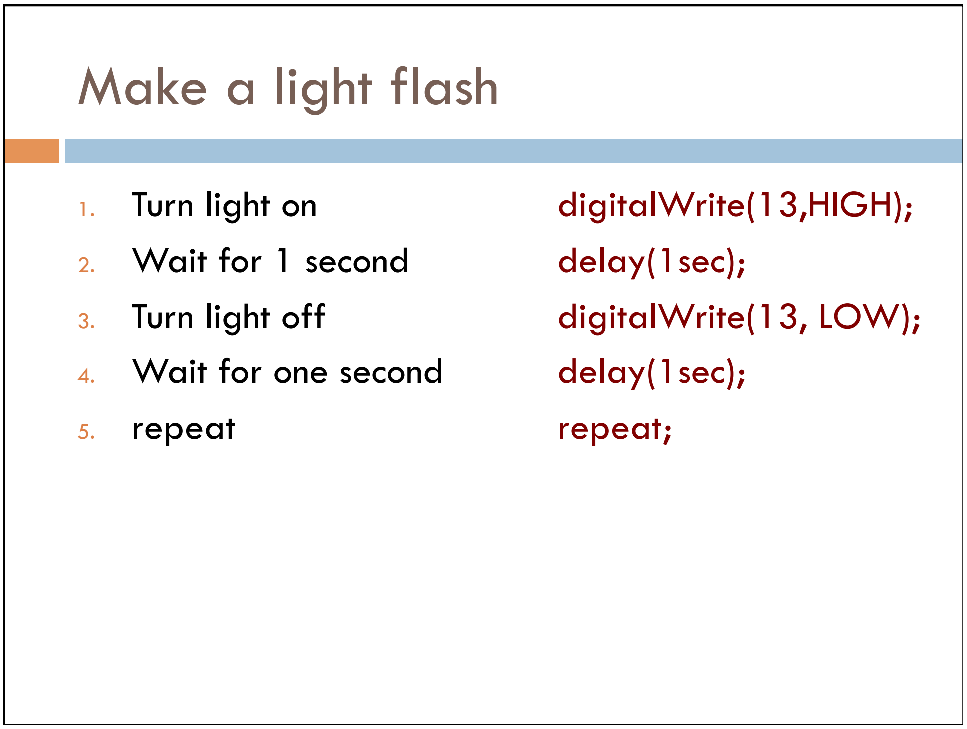


schematic

Arduino Duemilanove board has this circuit built-in
To turn on LED use `digitalWrite(13,HIGH)`

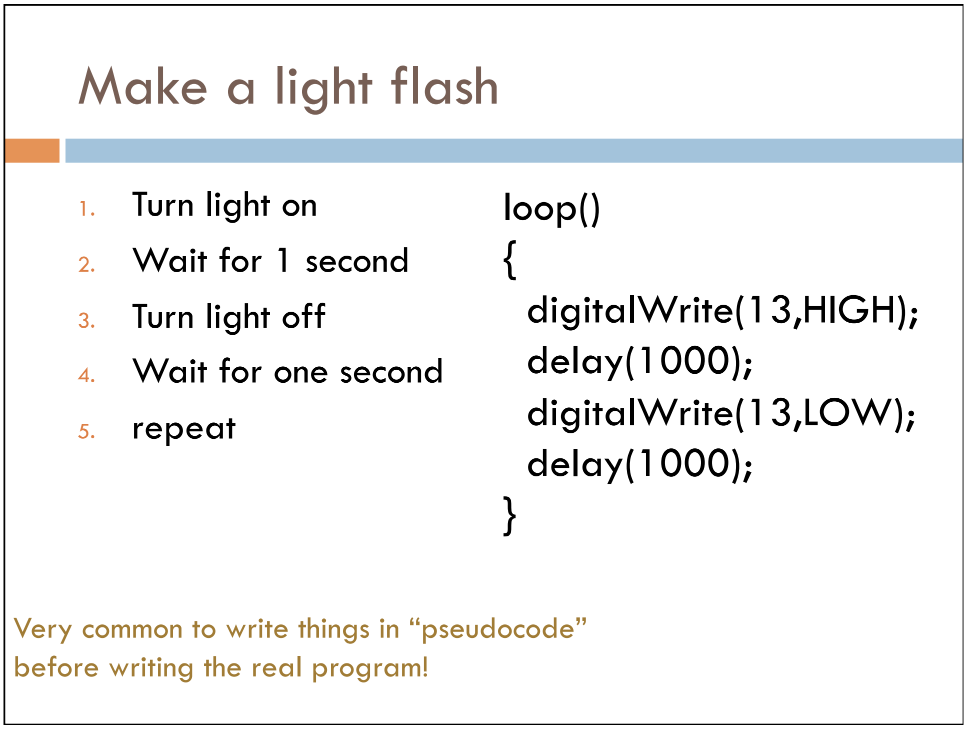
Make a light flash

1. Turn light on
2. Wait for 1 second
3. Turn light off
4. Wait for one second
5. repeat



Make a light flash

1.	Turn light on	digitalWrite(13,HIGH);
2.	Wait for 1 second	delay(1sec);
3.	Turn light off	digitalWrite(13, LOW);
4.	Wait for one second	delay(1sec);
5.	repeat	repeat;

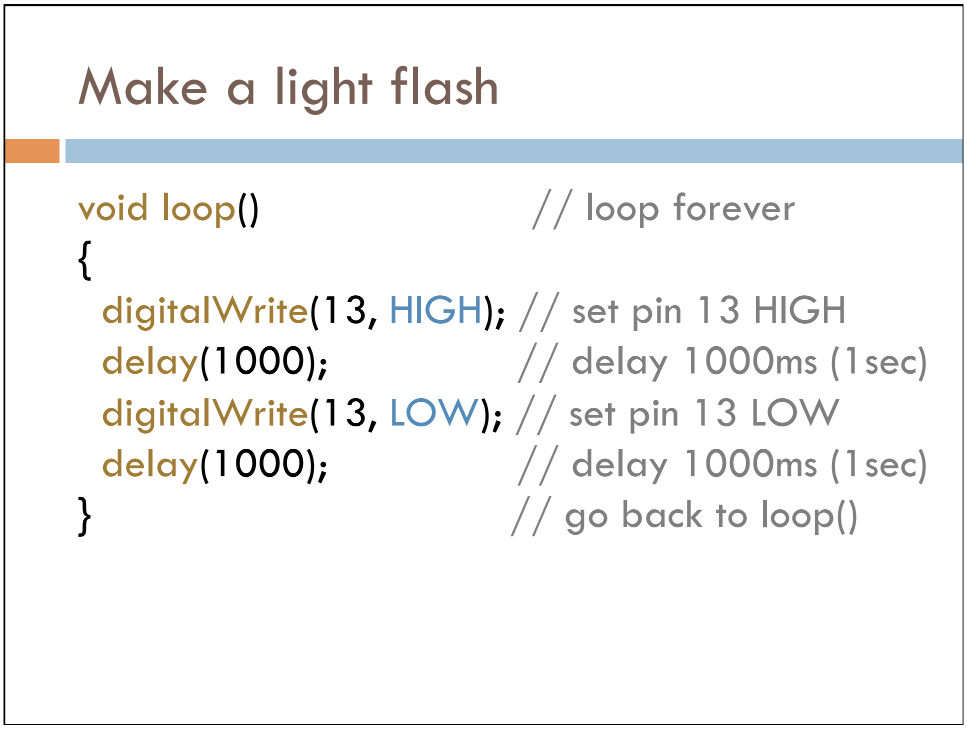


Make a light flash

1. Turn light on
2. Wait for 1 second
3. Turn light off
4. Wait for one second
5. repeat

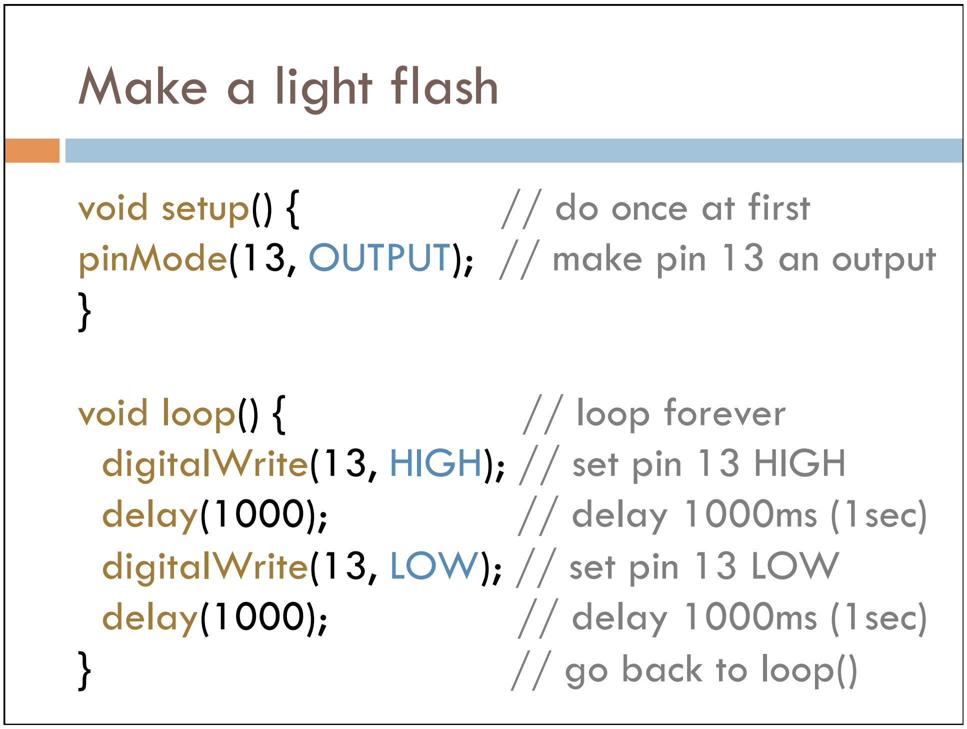
```
loop()
{
  digitalWrite(13,HIGH);
  delay(1000);
  digitalWrite(13,LOW);
  delay(1000);
}
```

Very common to write things in "pseudocode" before writing the real program!



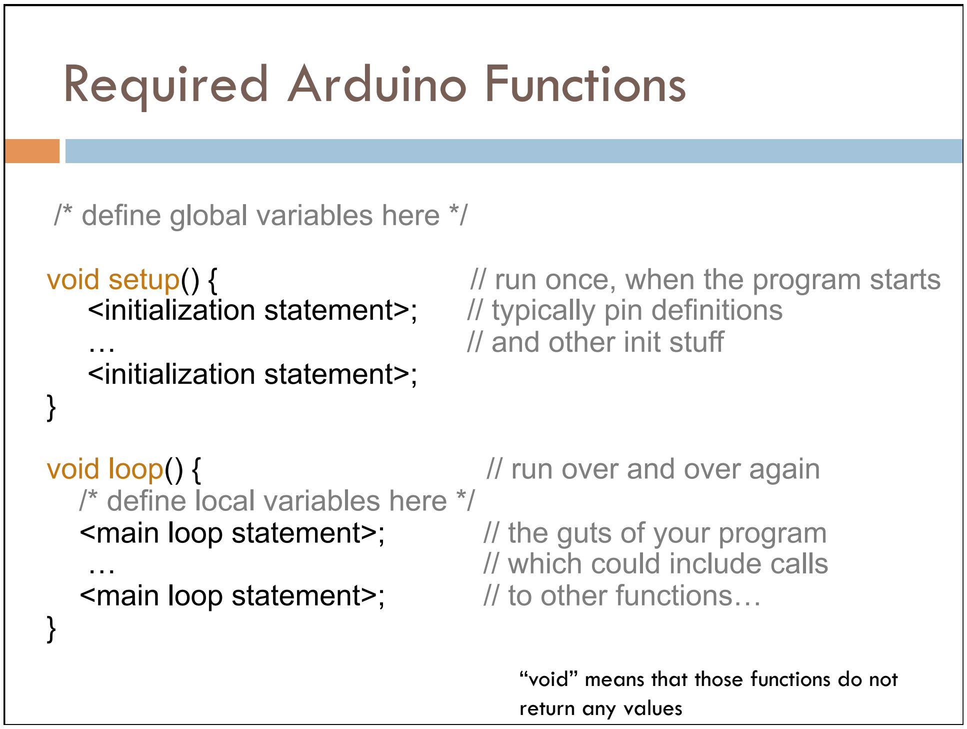
Make a light flash

```
void loop()                    // loop forever
{
  digitalWrite(13, HIGH); // set pin 13 HIGH
  delay(1000);            // delay 1000ms (1sec)
  digitalWrite(13, LOW); // set pin 13 LOW
  delay(1000);            // delay 1000ms (1sec)
}                         // go back to loop()
```



Make a light flash

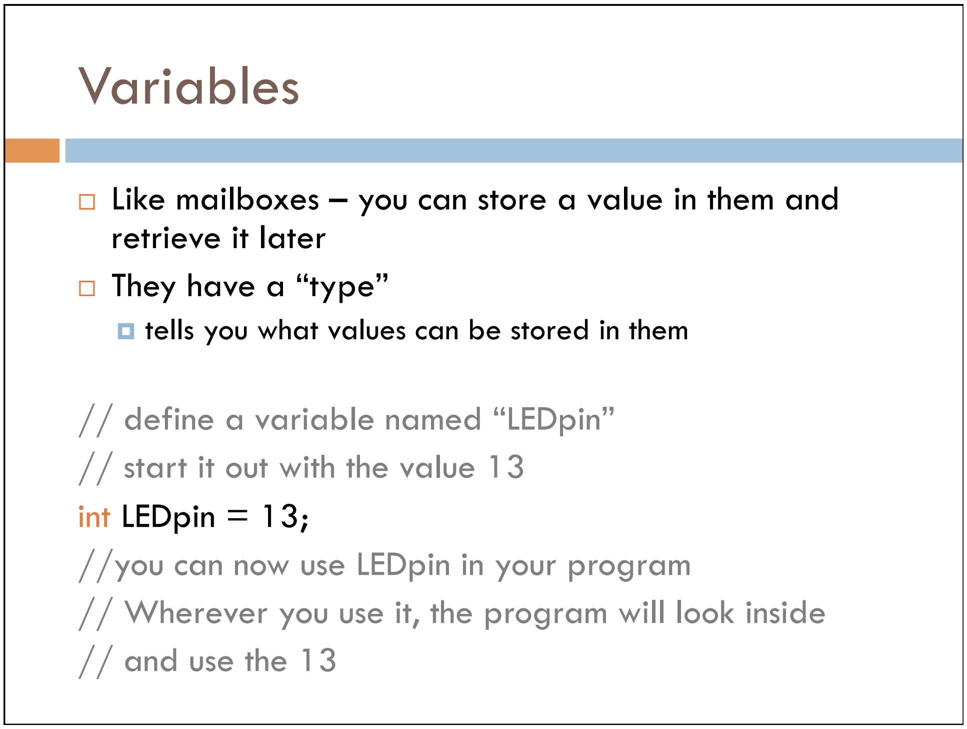
```
void setup() {              // do once at first
pinMode(13, OUTPUT);  // make pin 13 an output
}

void loop() {                  // loop forever
  digitalWrite(13, HIGH); // set pin 13 HIGH
  delay(1000);            // delay 1000ms (1sec)
  digitalWrite(13, LOW); // set pin 13 LOW
  delay(1000);            // delay 1000ms (1sec)
}                         // go back to loop()
```



Required Arduino Functions

```
/* define global variables here */

void setup() {                    // run once, when the program starts
    <initialization statement>;   // typically pin definitions
    …                             // and other init stuff
    <initialization statement>;
}

void loop() {                      // run over and over again
   /* define local variables here */
   <main loop statement>;          // the guts of your program
    …                              // which could include calls
   <main loop statement>;          // to other functions…
}
```

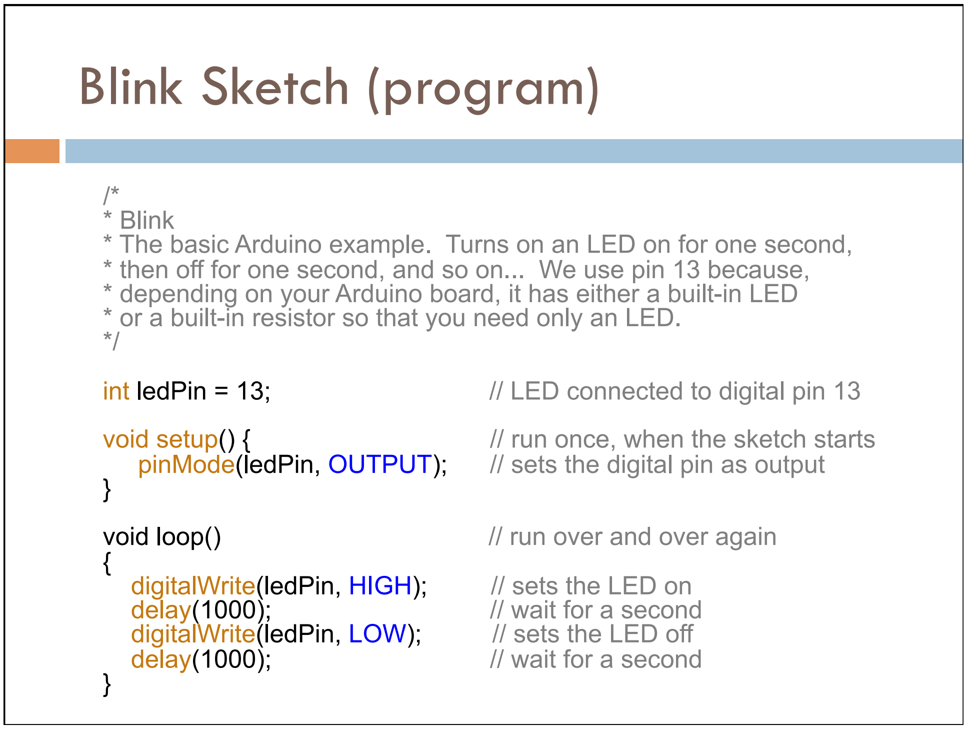
"void" means that those functions do not return any values



Variables

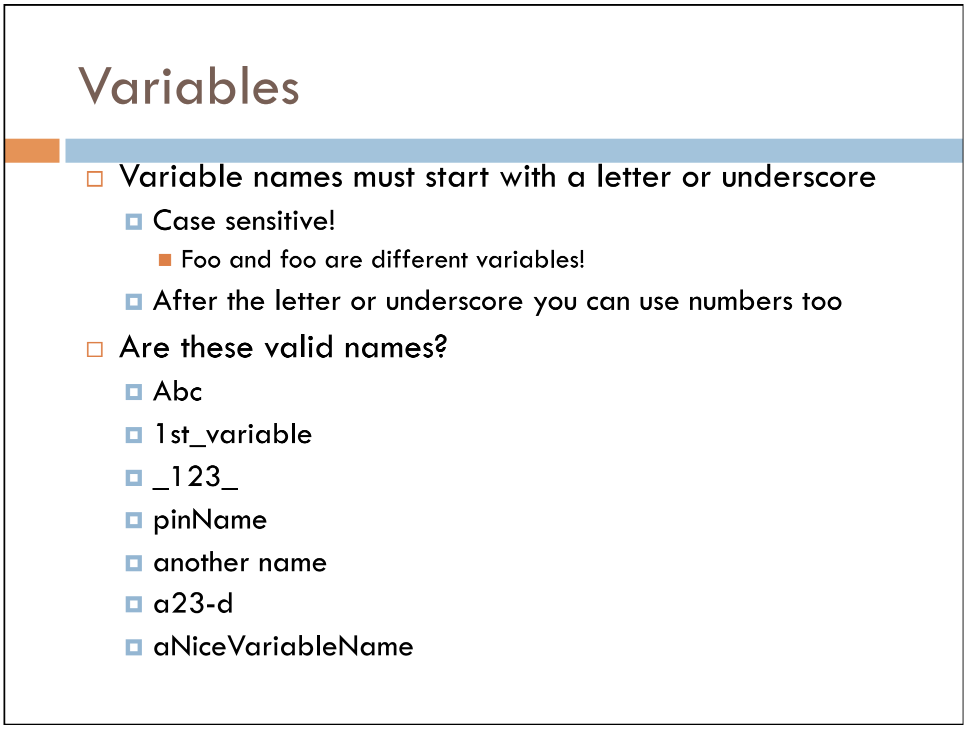
- Like mailboxes – you can store a value in them and retrieve it later
- They have a "type"
  - tells you what values can be stored in them

```
// define a variable named "LEDpin"
// start it out with the value 13
int LEDpin = 13;
//you can now use LEDpin in your program
// Wherever you use it, the program will look inside
// and use the 13
```



Blink Sketch (program)

```
/*
* Blink
* The basic Arduino example.  Turns on an LED on for one second,
* then off for one second, and so on...  We use pin 13 because,
* depending on your Arduino board, it has either a built-in LED
* or a built-in resistor so that you need only an LED.
*/

int ledPin = 13;                    // LED connected to digital pin 13

void setup() {                      // run once, when the sketch starts
    pinMode(ledPin, OUTPUT);        // sets the digital pin as output
}

void loop()                         // run over and over again
{
   digitalWrite(ledPin, HIGH);      // sets the LED on
   delay(1000);                     // wait for a second
   digitalWrite(ledPin, LOW);       // sets the LED off
   delay(1000);                     // wait for a second
}
```

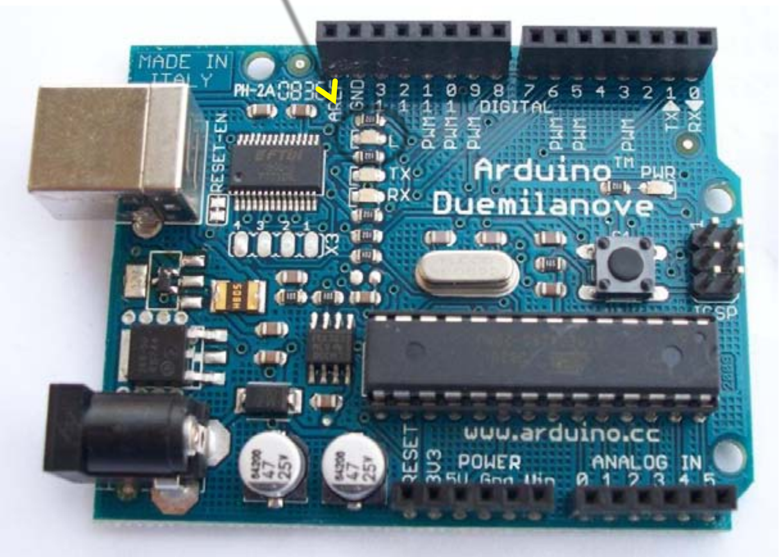


Variables

- Variable names must start with a letter or underscore
  - Case sensitive!
    - Foo and foo are different variables!
  - After the letter or underscore you can use numbers too
- Are these valid names?
  - Abc
  - 1st_variable
  - _123_
  - pinName
  - another name
  - a23-d
  - aNiceVariableName

Arduino

LED



Blink Modifications

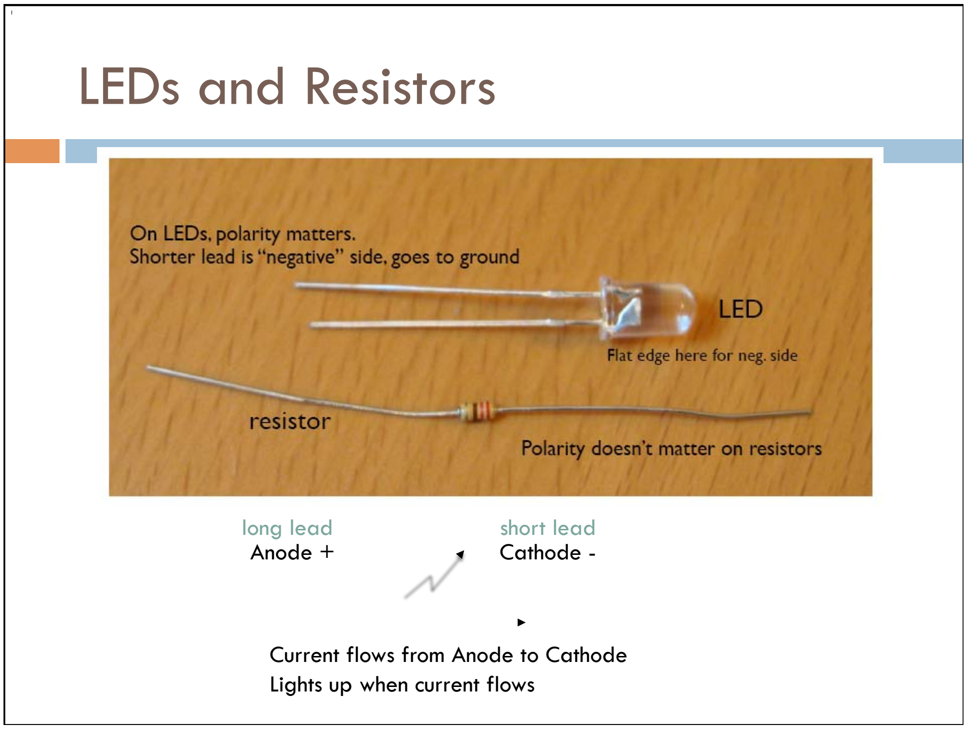
- Change so that blink is on for 500msec and off for 100msec
  - What happens?
- Change so that blink is on for 50msec and off for 50msec
  - What happens?
- Change so that blink is on for 10ms and off for 10ms
  - What happens?

We just made an LED blink…Big Deal?

- Most actuators are switched on and off with a digital output
  - The digitalWrite(pin,value); function is the software command that lets you control almost anything
- LEDs are easy!
  - Motors, servos, etc. are a little trickier, but not much
  - More on that later…
- Arduino has 14 digital pins (inputs or outputs)
  - can easily add more with external helper chips
  - More on that later…

Blink Modifications

- Change to use an external LED rather than the one on the board
  - Connect to any digital pin
  - LED is on if current flows from Anode to Cathode
  - LED is on if the digital pin is HIGH, off if LOW
  - How much current do you use?
    - not more than 20mA
  - How do you make sure you don't use too much?
    - use a resistor
  - Pay attention to current! Use a current-limiting resistor!



LEDs and Resistors

On LEDs, polarity matters.
Shorter lead is “negative” side, goes to ground

LED

Flat edge here for neg. side

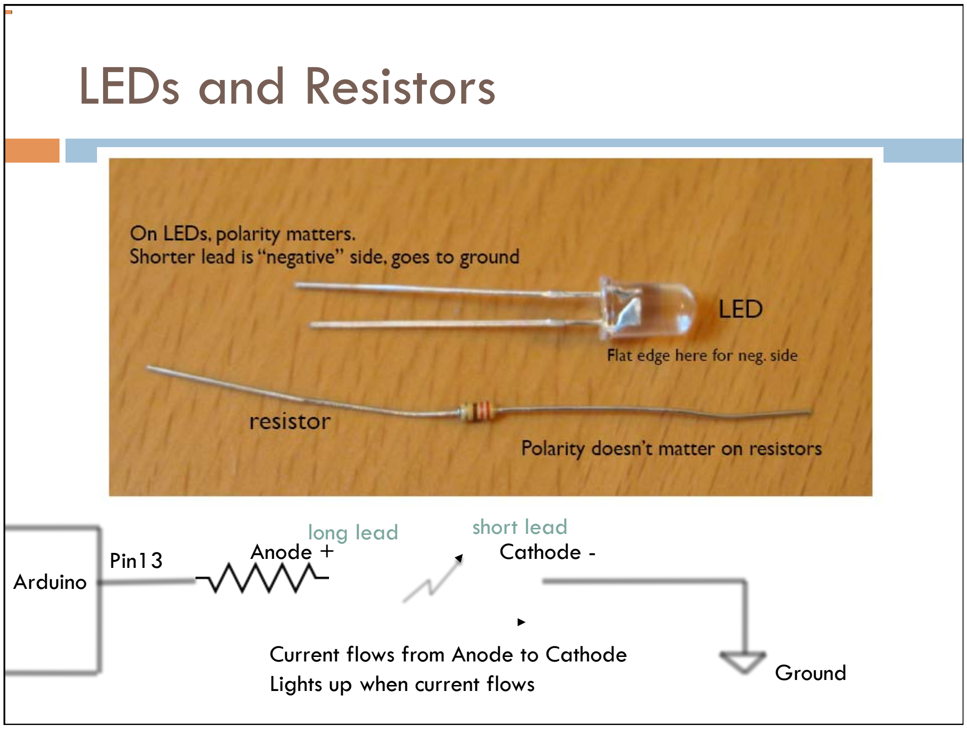
resistor

Polarity doesn’t matter on resistors

long lead
Anode +

short lead
Cathode -

Current flows from Anode to Cathode
Lights up when current flows



LEDs and Resistors

On LEDs, polarity matters.
Shorter lead is “negative” side, goes to ground

LED

Flat edge here for neg. side

resistor

Polarity doesn’t matter on resistors

Arduino

Pin13

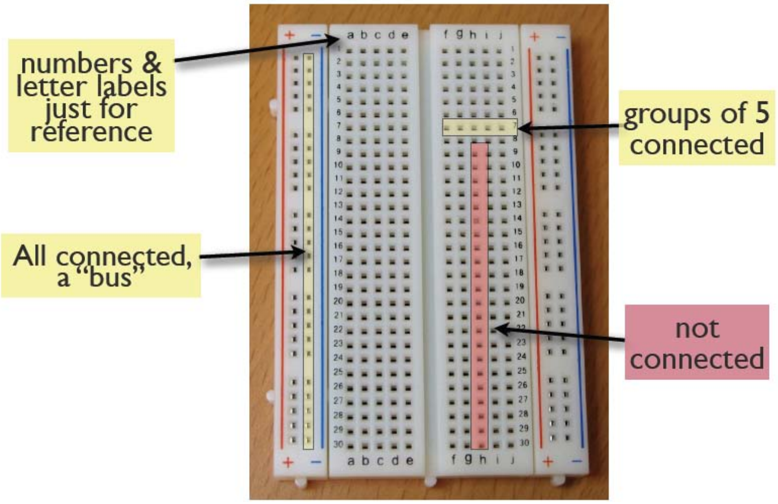
long lead
Anode +

short lead
Cathode -

Current flows from Anode to Cathode
Lights up when current flows

Ground

Proto Boards


numbers & letter labels just for reference

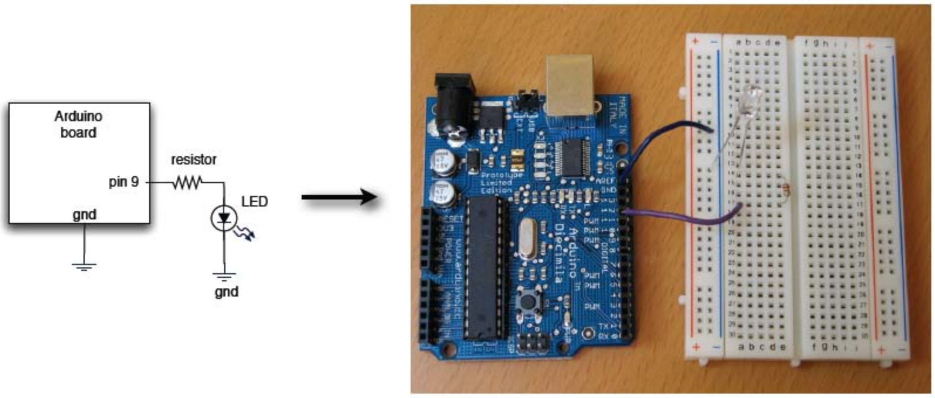
groups of 5 connected

All connected, a “bus”

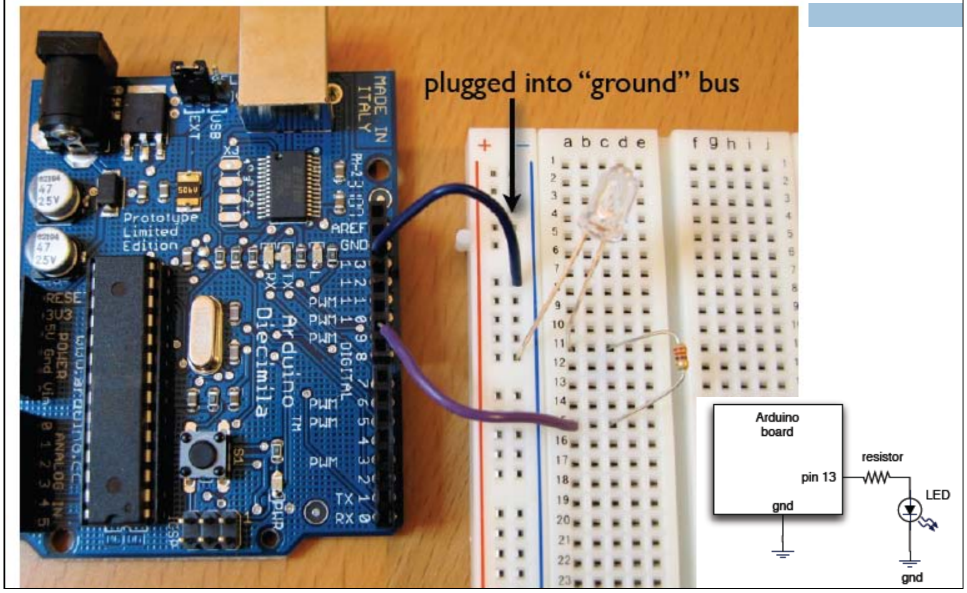
not connected

AKA Solderless Breadboards

Wire it Up

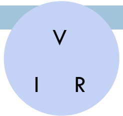


Wire it Up



Current Limiting Resistor

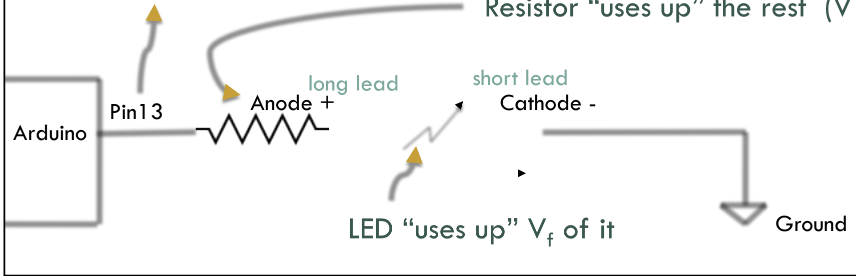
- Ohm's Law
  - V = IR   I = V/R R = V/I



- Every LED has a V_f "Forward Voltage"
  - How much voltage is dropped (used up) passing through the LED

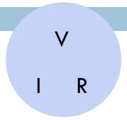
"HIGH" forces output pin to 5v (called V)

Resistor "uses up" the rest ($V - V_f$)



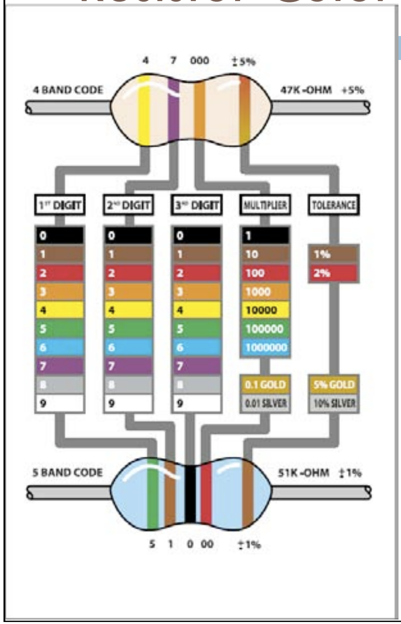
LED "uses up" V_f of it

Current Limiting Resistor



- Ohm's Law
  - V = IR I = V/R R = V/I
- Every LED has a V_f "Forward Voltage"
  - How much voltage is dropped (used up) passing through the LED
- R = $(V - V_f) / I$
  - Example – If V_f is 1.9v (red LED), and V = 5v, and you want 15mA of current (0.015A)
  - R = (5 – 1.9)/0.015 = 3.1/0.015 = 206Ω
  - Exact isn't critical – use next size up, i.e. 220Ω
  - Or be safe and use 330Ω or 470Ω
  - This would result in 9.4mA or 6.6mA which is fine

Resistor Color Codes



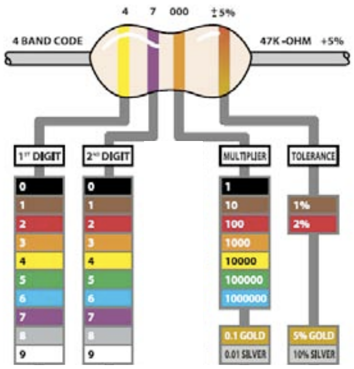
1ST DIGIT	2ND DIGIT	3RD DIGIT	MULTIPLIER	TOLERANCE
0	0	0	1	
1	1	1	10	1%
2	2	2	100	2%
3	3	3	1000	
4	4	4	10000	
5	5	5	100000	
6	6	6	1000000	
7	7	7		
8	8	8	0.1 GOLD	5% GOLD
9	9	9	0.01 SILVER	10% SILVER

What's the color code for a 330Ω resistor?

What's the color code for a 1kΩ resistor?

What's the color code for a 10kΩ resistor

Resistor Color Codes



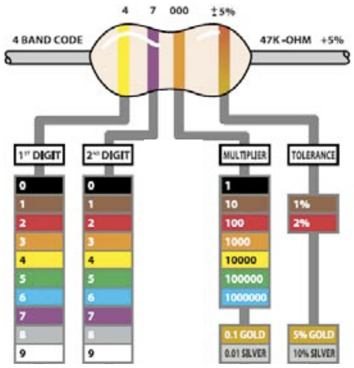
We're using 4-band 5% resistors with a ¼ watt rating

What's the color code for a 330Ω resistor?

What's the color code for a 1kΩ resistor?

What's the color code for a 10kΩ resistor

Resistor Color Codes



We're using 4-band 5% resistors with a ¼ watt rating

What's the color code for a 220Ω resistor?

orange orange brown gold

What's the color code for a 1kΩ resistor?

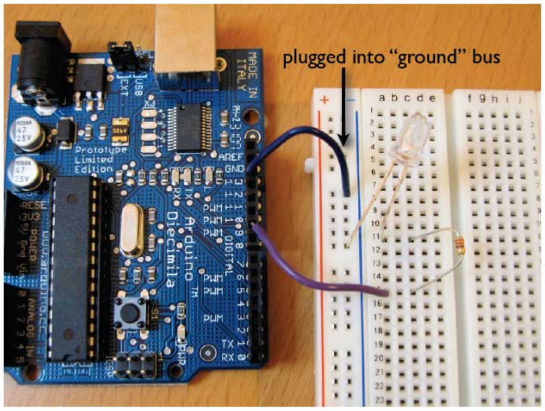
brown black red gold

What's the color code for a 470Ω resistor

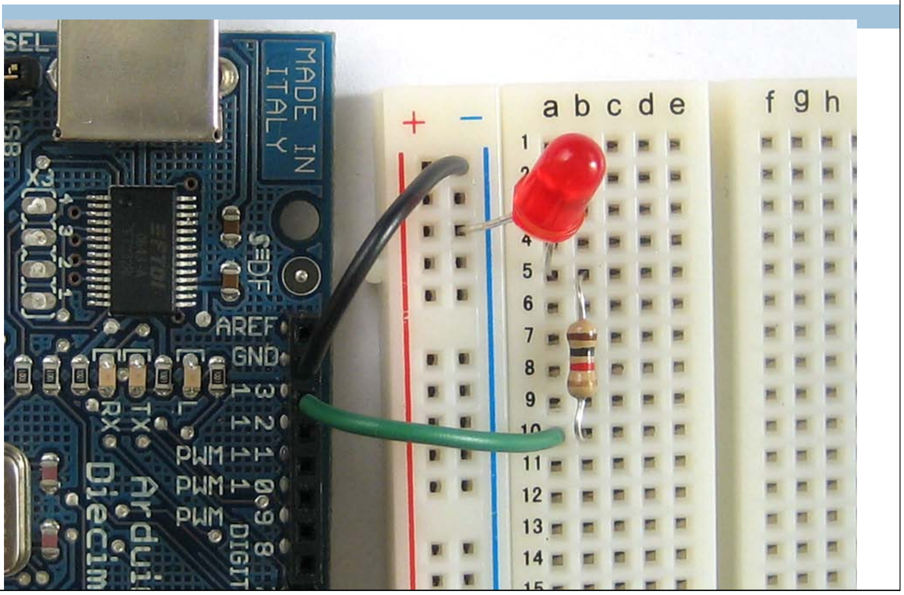
brown black orange gold

Wire it Up

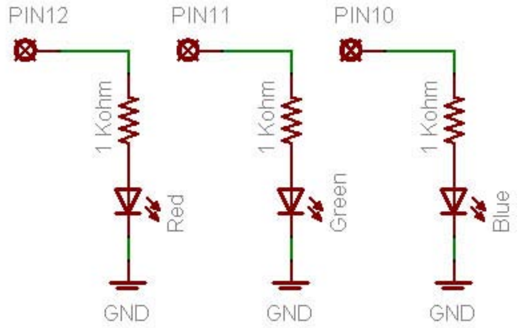
- Wire up an external LED of your choice, and change the Blink program to use that external LED
  - Choose your resistor based on the V_f of the LED you're using
    - Usually 1.8-2.2v
    - Listed on class web site
  - If you don't know Vf pick 330Ω or 470Ω



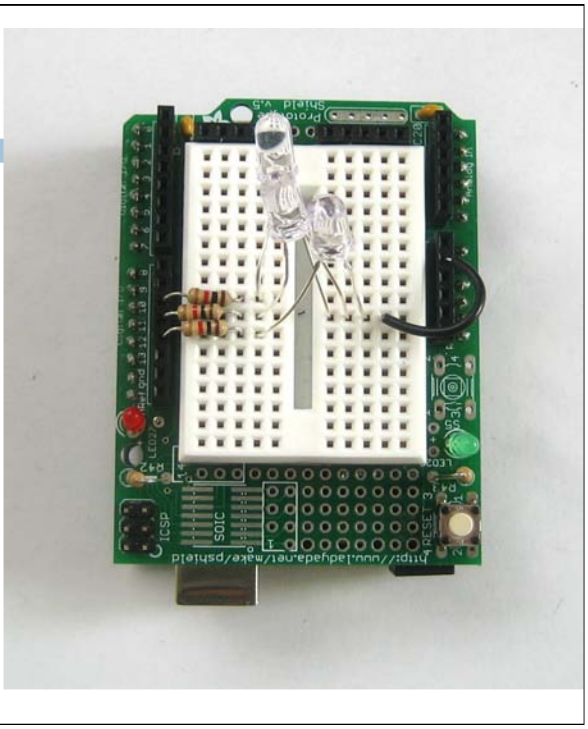
Another view



Multiple LEDs



Multiple LEDs

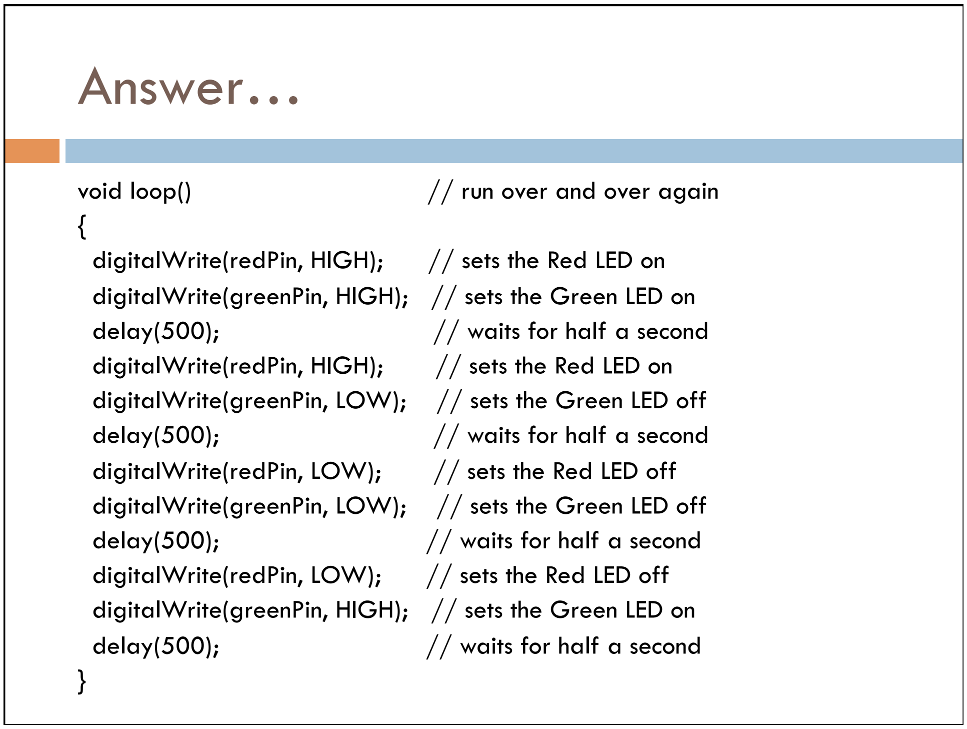


Arduino Code

```
int redPin = 12;                    // Red LED connected to digital pin 12
int greenPin = 11;                  // Green LED connected to digital pin 11

void setup()  {                     // run once, when the program starts
  pinMode(redPin, OUTPUT);          // sets the digital pin as output
  pinMode(greenPin, OUTPUT);        // sets the digital pin as output
}

void loop()   {                     // run over and over again
  digitalWrite(redPin, HIGH);       // sets the Red LED on
  digitalWrite(greenPin, HIGH);     // sets the Green LED on
  delay(500);                       // waits for half a second
  digitalWrite(redPin, LOW);        // sets the Red LED off
  digitalWrite(greenPin, LOW);      // sets the Green LED off
  delay(500);                       // waits for half a second
}
```

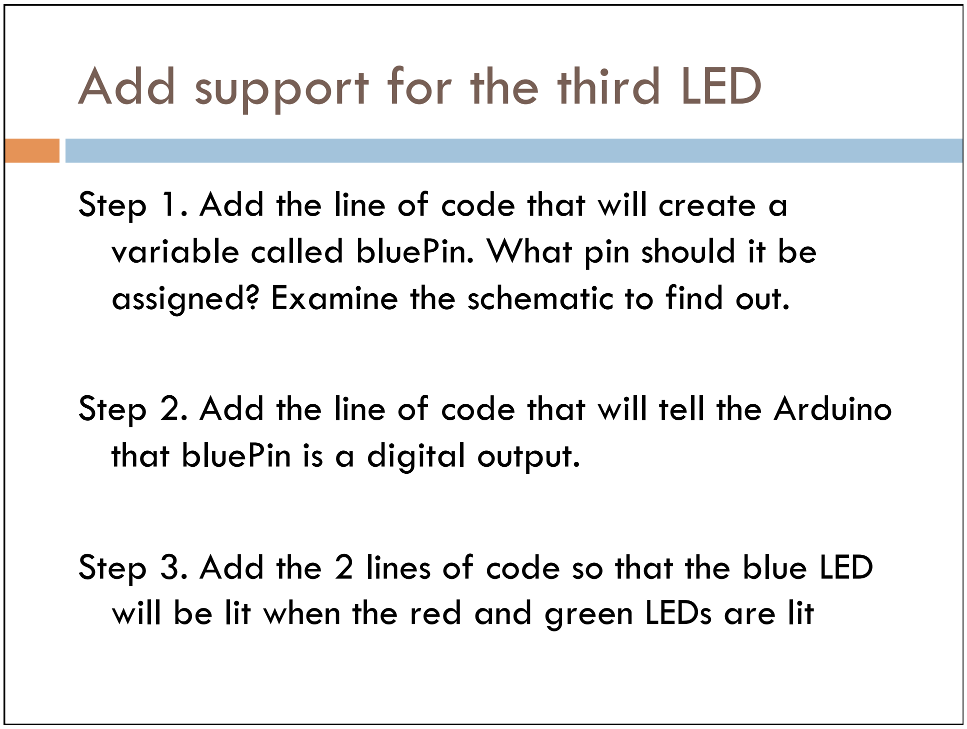
Change the code

- Change the loop() procedure code so that both LEDs are on for 500 ms, then only the red LED is on for 500 ms, then both LEDs are off, and finally only the green LED is on for 500 ms
  - Start with pseudocode…



Answer…

```
void loop()                          // run over and over again
{
  digitalWrite(redPin, HIGH);       // sets the Red LED on
  digitalWrite(greenPin, HIGH);     // sets the Green LED on
  delay(500);                       // waits for half a second
  digitalWrite(redPin, HIGH);       // sets the Red LED on
  digitalWrite(greenPin, LOW);      // sets the Green LED off
  delay(500);                       // waits for half a second
  digitalWrite(redPin, LOW);        // sets the Red LED off
  digitalWrite(greenPin, LOW);      // sets the Green LED off
  delay(500);                       // waits for half a second
  digitalWrite(redPin, LOW);        // sets the Red LED off
  digitalWrite(greenPin, HIGH);     // sets the Green LED on
  delay(500);                       // waits for half a second
}
```

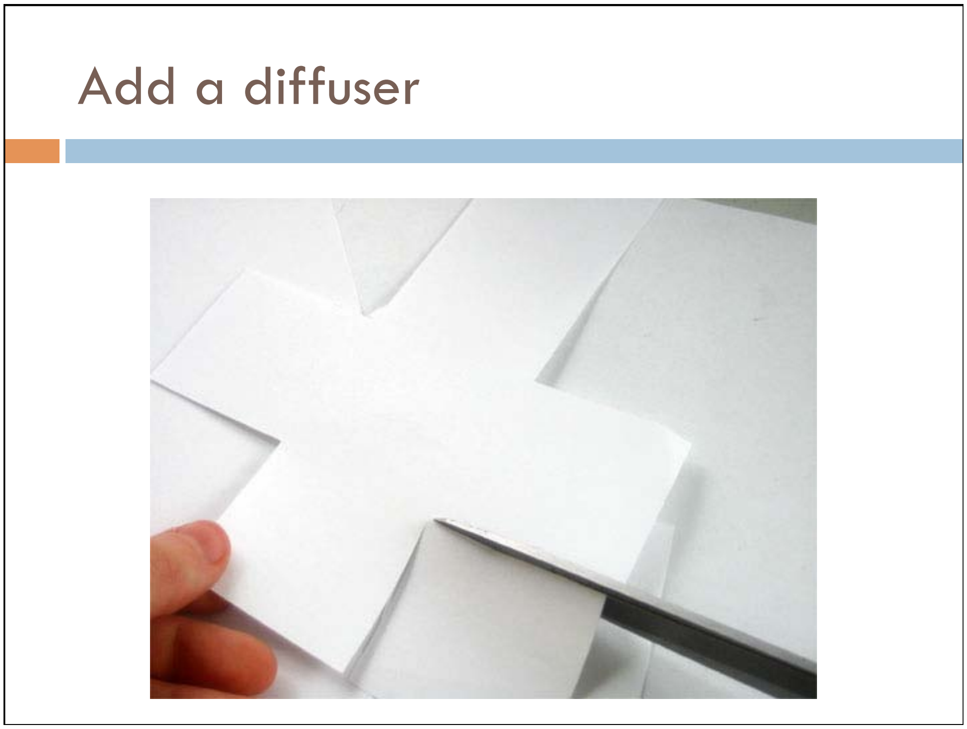


Add support for the third LED

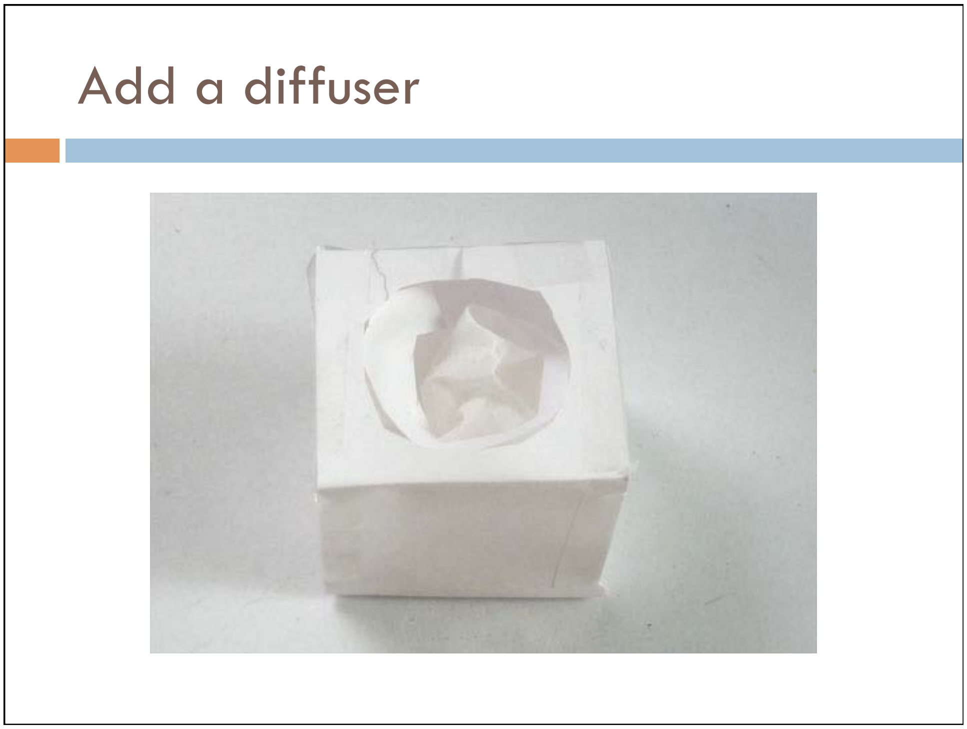
Step 1. Add the line of code that will create a variable called bluePin. What pin should it be assigned? Examine the schematic to find out.

Step 2. Add the line of code that will tell the Arduino that bluePin is a digital output.

Step 3. Add the 2 lines of code so that the blue LED will be lit when the red and green LEDs are lit



Add a diffuser



Add a diffuser

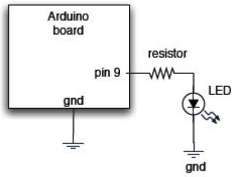
video…. ladyada.net

□ http://www.ladyada.net/learn/arduino/lesson3.html

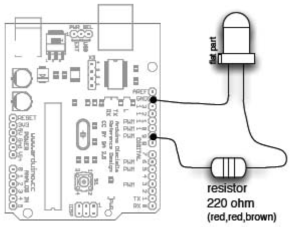
Moving on…

Varying LED Brightness

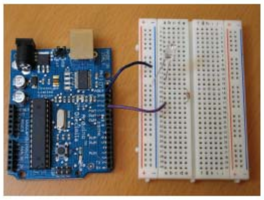
Same circuit as Blink circuit but pin 9 instead of pin 13



schematic



wiring diagram



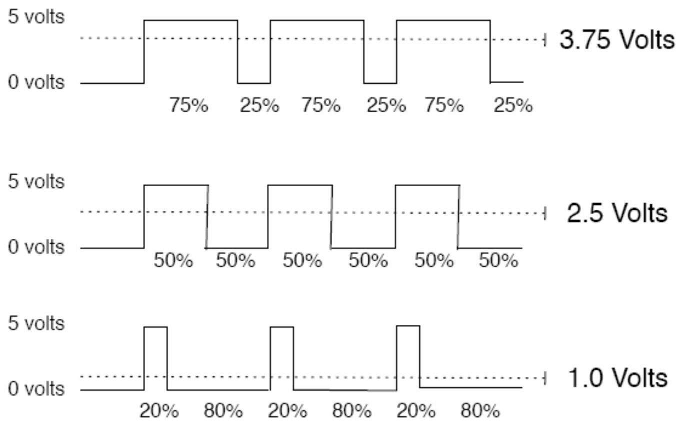
wired up

The PWM pins work with the "`analogWrite(value)`" command where "value" ranges between 0 and 255.
To turn LED to half-bright, use `analogWrite(9,128)`

PWM

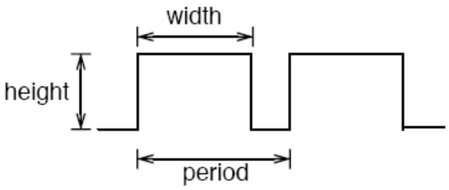
Output voltage is averaged from on *vs.* off time

output_voltage = (on_time / off_time) * max_voltage



PWM

- Used everywhere
  - Lamp dimmers, motor speed control, power supplies, noise making
- Three characteristics of PWM signals
  - Pulse width range (min/max)
  - Pulse period (= 1/pulses per second)
  - Voltage levels (0-5V, for instance)

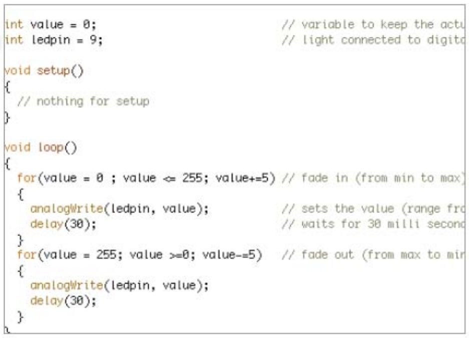


Pulse Width Modulation

- analogWrite(pin, value);
  - value can be 0 to 255
  - Must be one of the "PWM pins" : pins 3, 5, 6, 9, 10, 11

Load "File/Sketchbook/Examples/Analog/Fading"

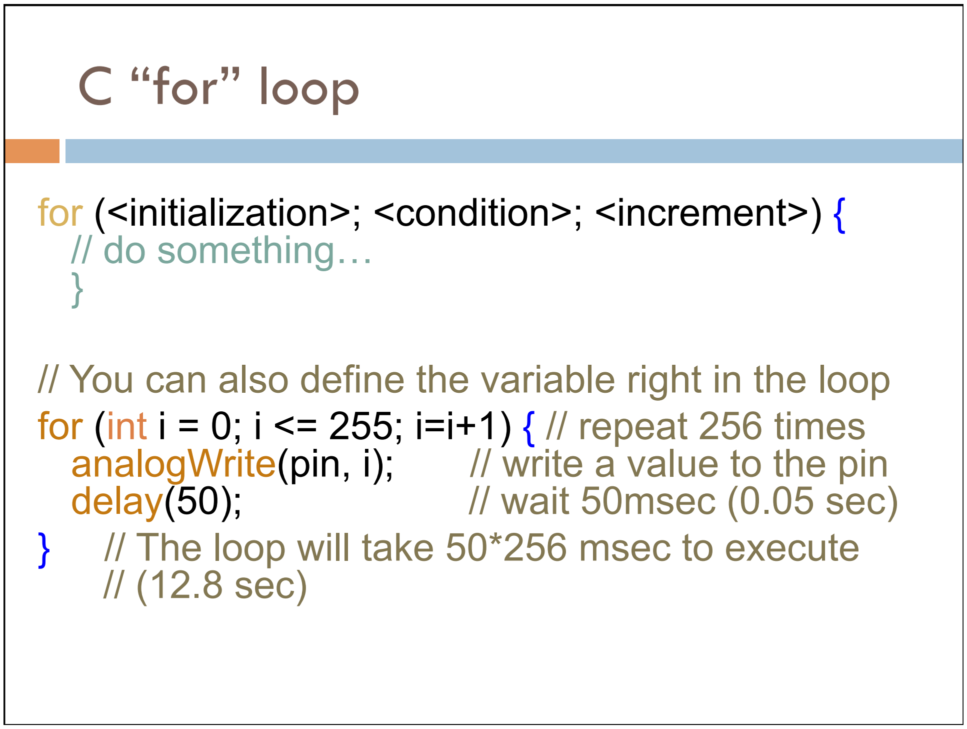
note →



```
int value = 0;                            // variable to keep the actu
int ledpin = 9;                           // light connected to digita

void setup()
{
  // nothing for setup
}

void loop()
{
  for(value = 0 ; value <= 255; value+=5) // fade in (from min to max)
  {
    analogWrite(ledpin, value);           // sets the value (range fro
    delay(30);                            // waits for 30 milli second
  }
  for(value = 255; value >=0; value-=5)   // fade out (from max to mir
  {
    analogWrite(ledpin, value);
    delay(30);
  }
```

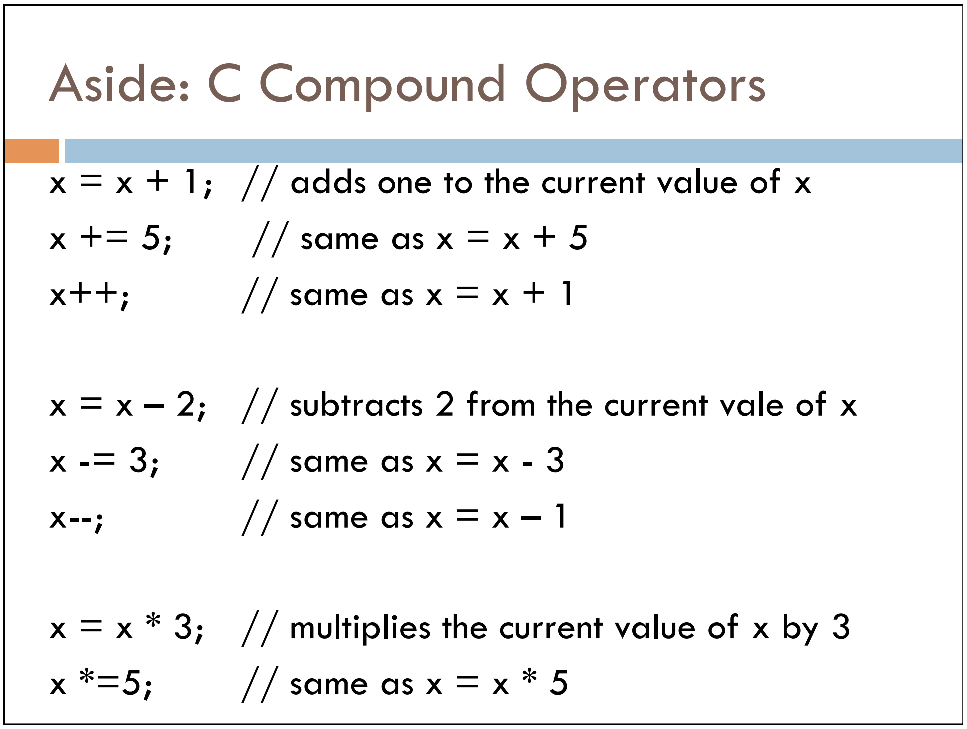
C "for loop"

```
for (<initialization>; <condition>; <increment>) {
   // do something…
   }


int i;   // define an int to use as a loop variable
for (i = 0; i <= 255; i=i+1) { // repeat 256 times
   analogWrite(pin, i);       // write a value to the pin
   delay(50);                 // wait 50msec (0.05 sec)
}     // The loop will take 50*256 msec to execute
      // (12.8 sec)
```



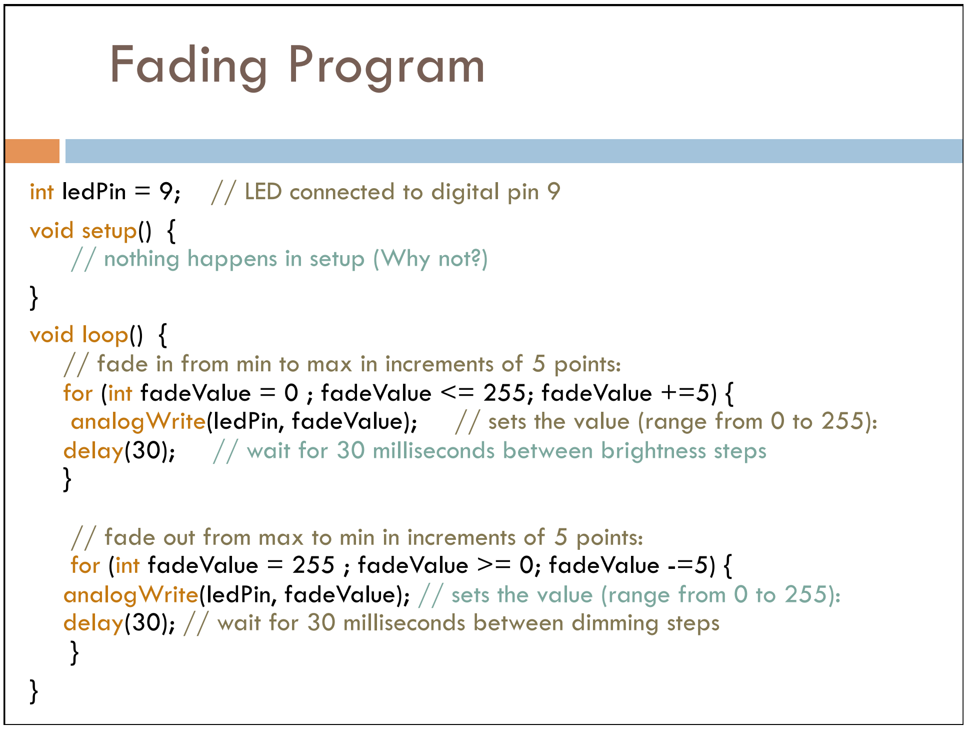
C “for” loop

```
for (<initialization>; <condition>; <increment>) {
   // do something…
   }

// You can also define the variable right in the loop
for (int i = 0; i <= 255; i=i+1) { // repeat 256 times
   analogWrite(pin, i);        // write a value to the pin
   delay(50);                  // wait 50msec (0.05 sec)
}     // The loop will take 50*256 msec to execute
      // (12.8 sec)
```



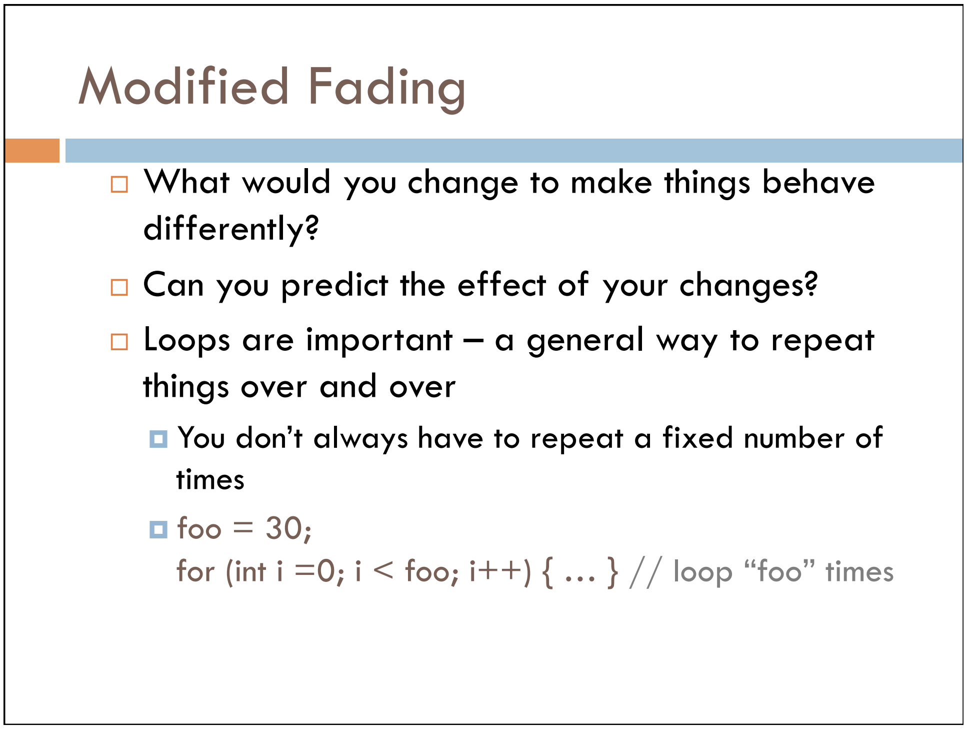
Aside: C Compound Operators

```
x = x + 1;  // adds one to the current value of x
x += 5;      // same as x = x + 5
x++;        // same as x = x + 1

x = x – 2;  // subtracts 2 from the current vale of x
x -= 3;     // same as x = x - 3
x--;        // same as x = x – 1

x = x * 3;  // multiplies the current value of x by 3
x *=5;      // same as x = x * 5
```



Fading Program

```
int ledPin = 9;    // LED connected to digital pin 9
void setup()  {
     // nothing happens in setup (Why not?)
}
void loop()  {
    // fade in from min to max in increments of 5 points:
    for (int fadeValue = 0 ; fadeValue <= 255; fadeValue +=5) {
     analogWrite(ledPin, fadeValue);      // sets the value (range from 0 to 255):
    delay(30);      // wait for 30 milliseconds between brightness steps
    }

     // fade out from max to min in increments of 5 points:
     for (int fadeValue = 255 ; fadeValue >= 0; fadeValue -=5) {
    analogWrite(ledPin, fadeValue); // sets the value (range from 0 to 255):
    delay(30); // wait for 30 milliseconds between dimming steps
     }
}
```



Modified Fading

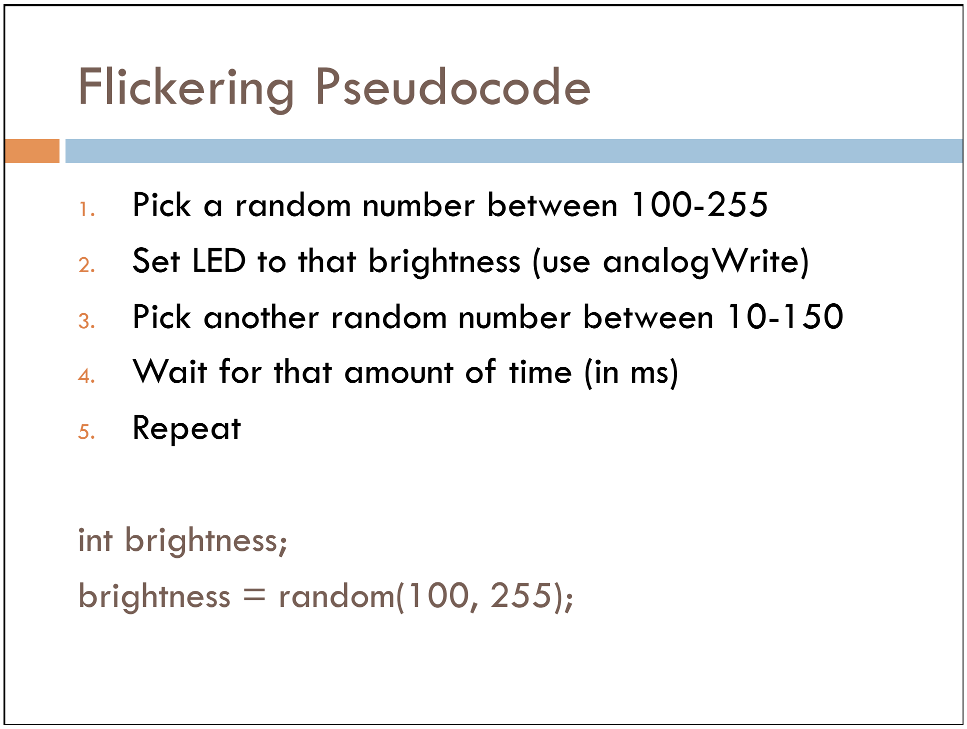
- What would you change to make things behave differently?
- Can you predict the effect of your changes?
- Loops are important – a general way to repeat things over and over
  - You don’t always have to repeat a fixed number of times
  - foo = 30;
    for (int i =0; i < foo; i++) { … } // loop “foo” times

Moving on…

- Write a program to make the LED flicker like a flame
  - Choose a random intensity
  - For a random amount of time
- Use analogWrite(ledPin, val) to change brightness
- Main loop repeats itself forever…
  - Set the value of the brightness to a random value
  - Wait for a random amount of time
  - repeat
- The effect looks like flickering…

Flickering Pseudocode

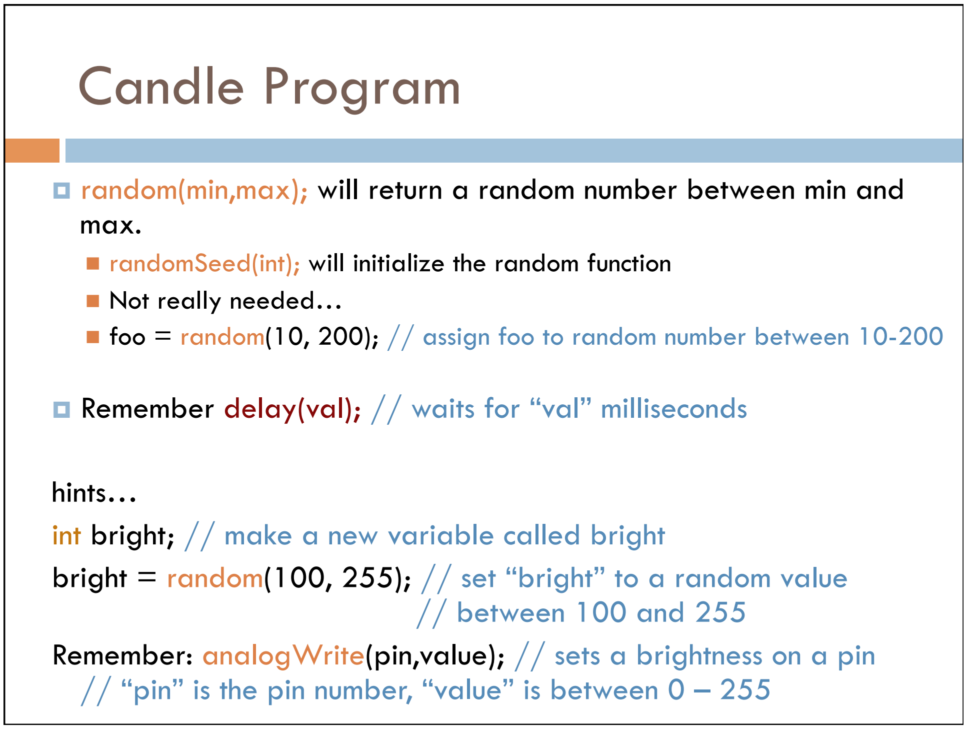
1. Set the LED to a random brightness
2. Wait for a random amount of time
3. repeat



Flickering Pseudocode

1. Pick a random number between 100-255
2. Set LED to that brightness (use analogWrite)
3. Pick another random number between 10-150
4. Wait for that amount of time (in ms)
5. Repeat

```
int brightness;
brightness = random(100, 255);
```

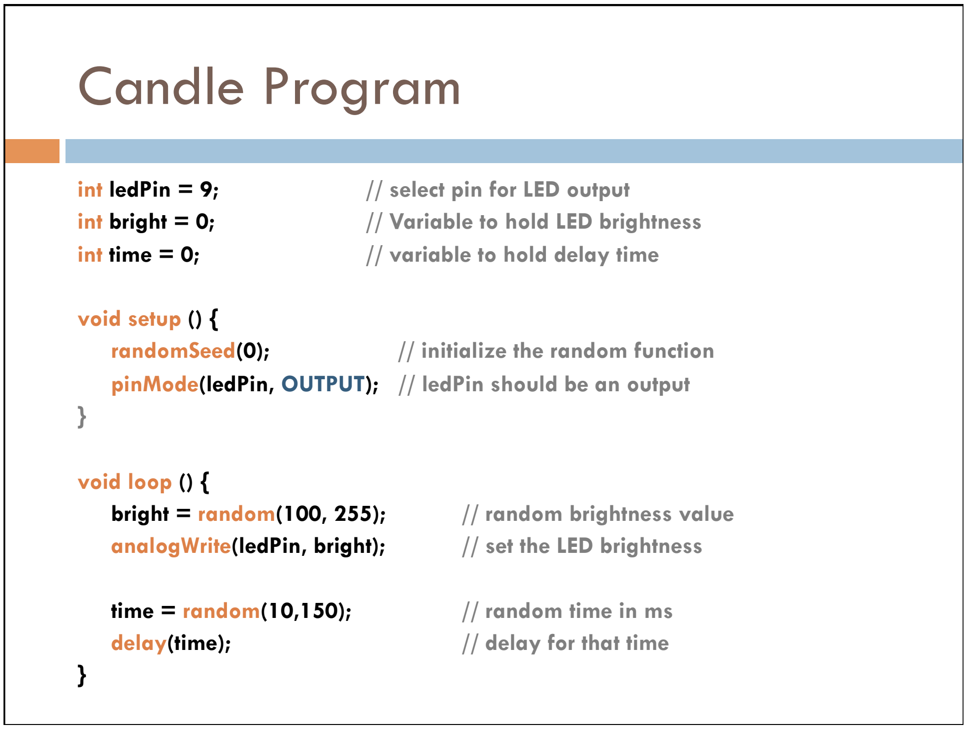


Candle Program

- random(min,max); will return a random number between min and max.
  - randomSeed(int); will initialize the random function
  - Not really needed…
  - foo = random(10, 200); // assign foo to random number between 10-200
- Remember delay(val); // waits for "val" milliseconds

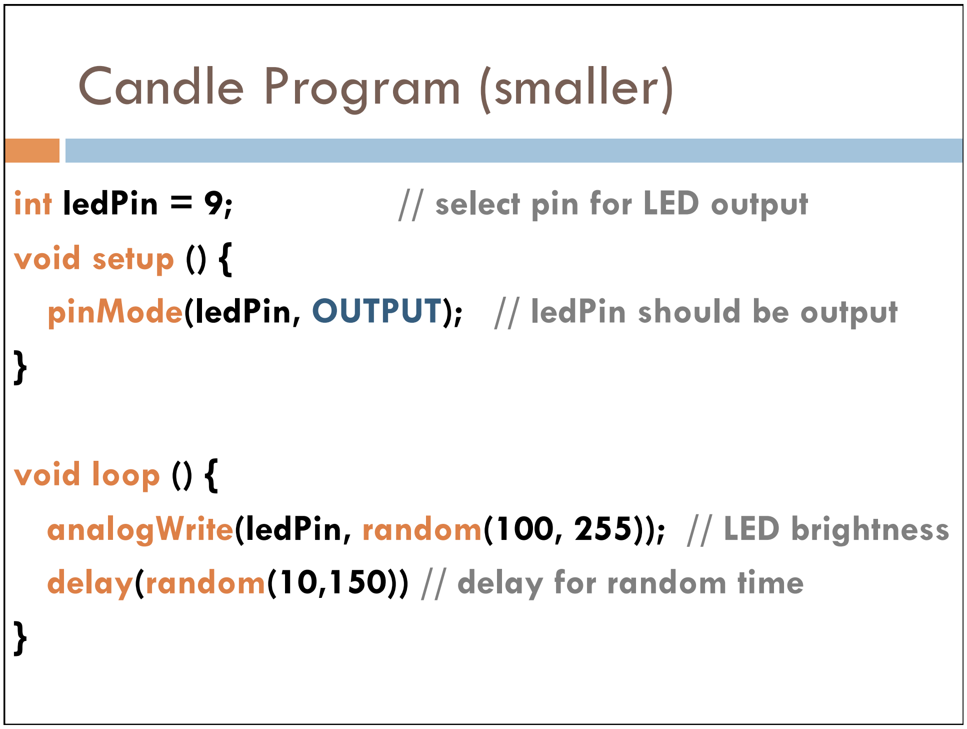
hints…

```
int bright; // make a new variable called bright
bright = random(100, 255); // set "bright" to a random value
                           // between 100 and 255
Remember: analogWrite(pin,value); // sets a brightness on a pin
   // "pin" is the pin number, "value" is between 0 – 255
```



Candle Program

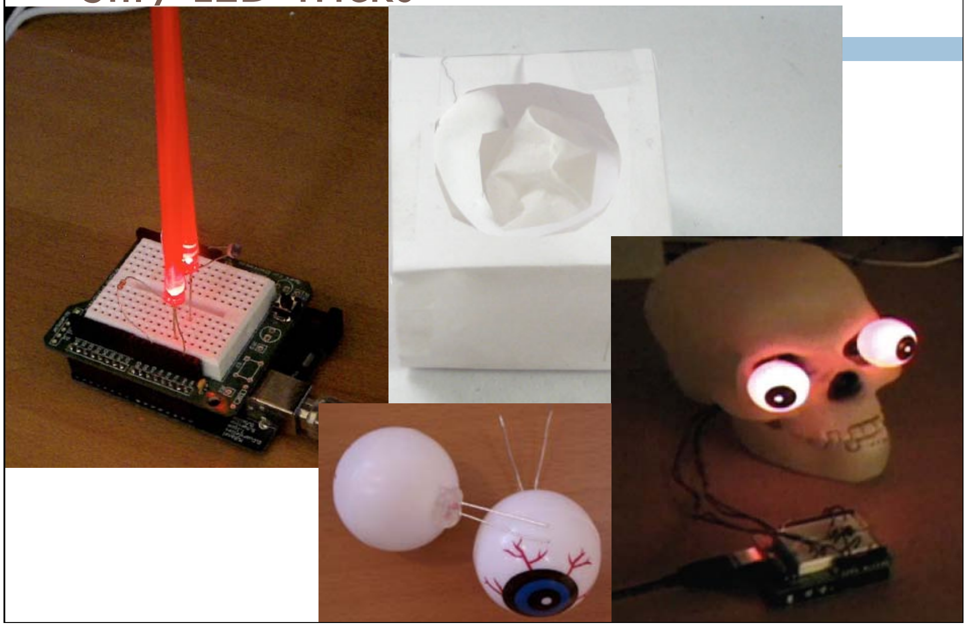
```
int ledPin = 9;                  // select pin for LED output
int bright = 0;                  // Variable to hold LED brightness
int time = 0;                    // variable to hold delay time

void setup () {
   randomSeed(0);                // initialize the random function
   pinMode(ledPin, OUTPUT);   // ledPin should be an output
}

void loop () {
   bright = random(100, 255);       // random brightness value
   analogWrite(ledPin, bright);     // set the LED brightness

   time = random(10,150);           // random time in ms
   delay(time);                     // delay for that time
}
```



Candle Program (smaller)

```
int ledPin = 9;              // select pin for LED output
void setup () {
   pinMode(ledPin, OUTPUT);   // ledPin should be output
}

void loop () {
   analogWrite(ledPin, random(100, 255));  // LED brightness
   delay(random(10,150)) // delay for random time
}
```

Silly LED Tricks

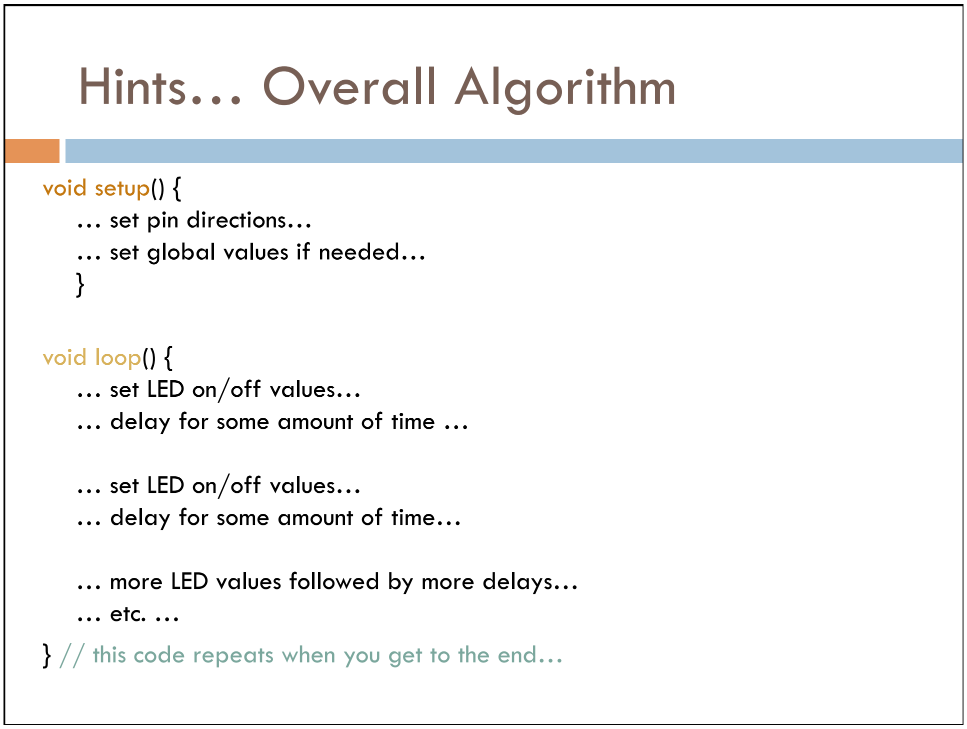


Next Task: 8 LEDs

- connect LEDs (through resistors!) to 8 Arduino pins
  - use pins 2, 3, 4, 5, 6, 7, 9, 8
    - Remember, pwm on pins 3, 5, 6, 9, 10, 11 only…
  - Now you can turn the LEDs on and off with

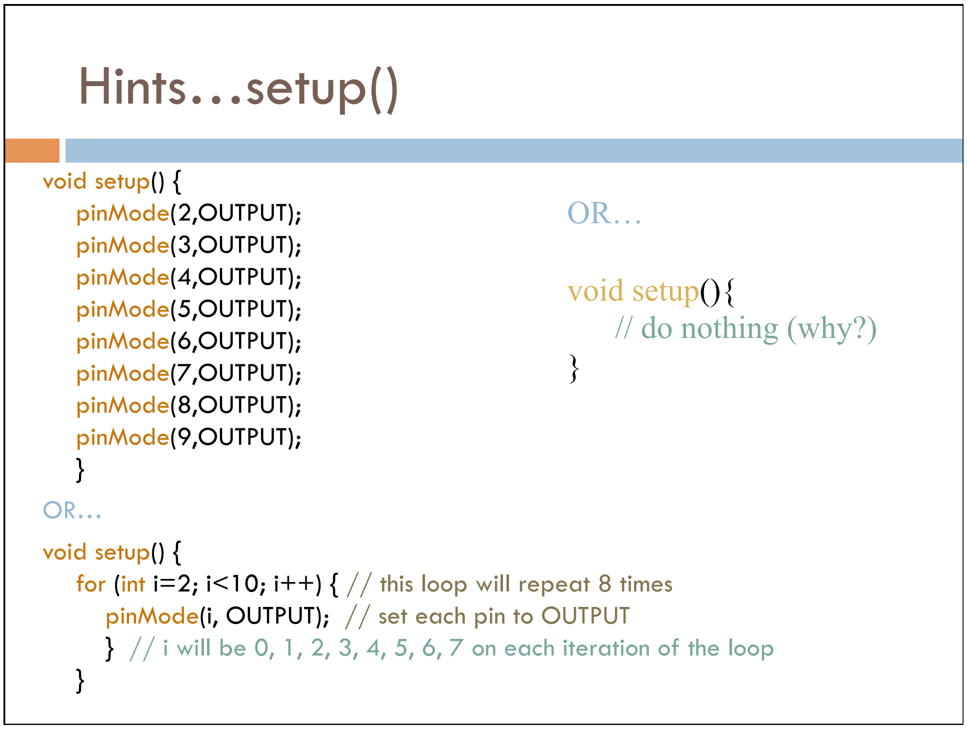
```
digitalWrite(5, HIGH);      // turn LED 5 on
digitalWrite(9, LOW);       // turn LED 9 off
analogWrite(3, 180);        // turn LED 3 partly on
```

  - Use those commands, also delay(), also perhaps loops, and random(min,max) to make the 8 LEDs do something!



Hints… Overall Algorithm

```
void setup() {
    … set pin directions…
    … set global values if needed…
    }

void loop() {
    … set LED on/off values…
    … delay for some amount of time …

    … set LED on/off values…
    … delay for some amount of time…

    … more LED values followed by more delays…
    … etc. …
} // this code repeats when you get to the end…
```



Hints…setup()

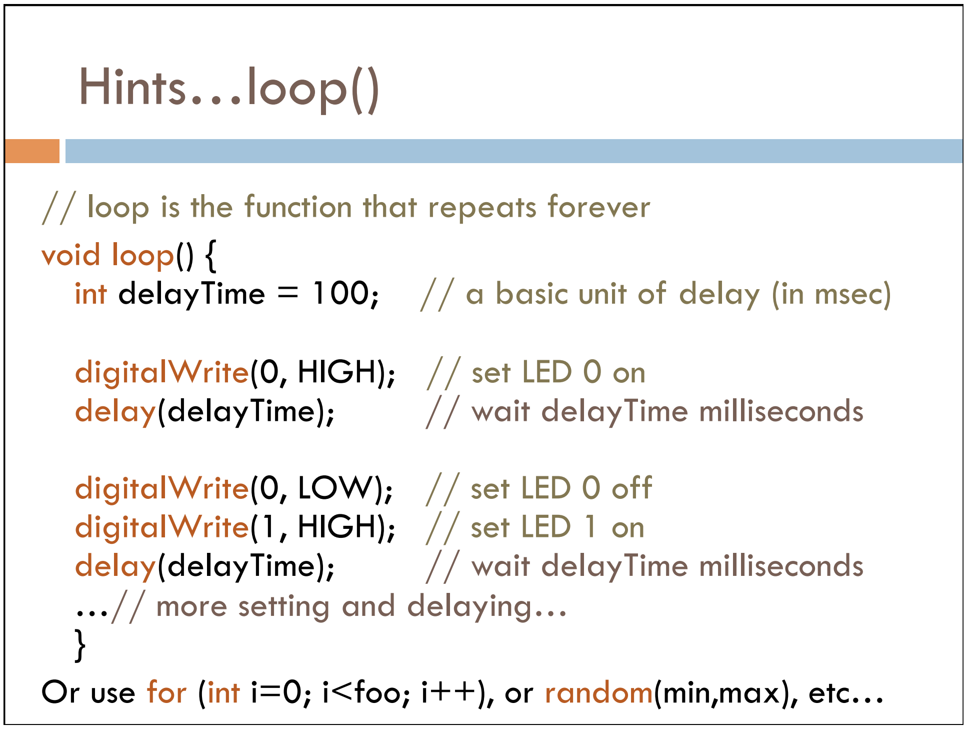
```
void setup() {
    pinMode(2,OUTPUT);
    pinMode(3,OUTPUT);
    pinMode(4,OUTPUT);
    pinMode(5,OUTPUT);
    pinMode(6,OUTPUT);
    pinMode(7,OUTPUT);
    pinMode(8,OUTPUT);
    pinMode(9,OUTPUT);
    }
```

OR…

```
void setup() {
    for (int i=2; i<10; i++) { // this loop will repeat 8 times
        pinMode(i, OUTPUT);  // set each pin to OUTPUT
        }  // i will be 0, 1, 2, 3, 4, 5, 6, 7 on each iteration of the loop
    }
```

OR…

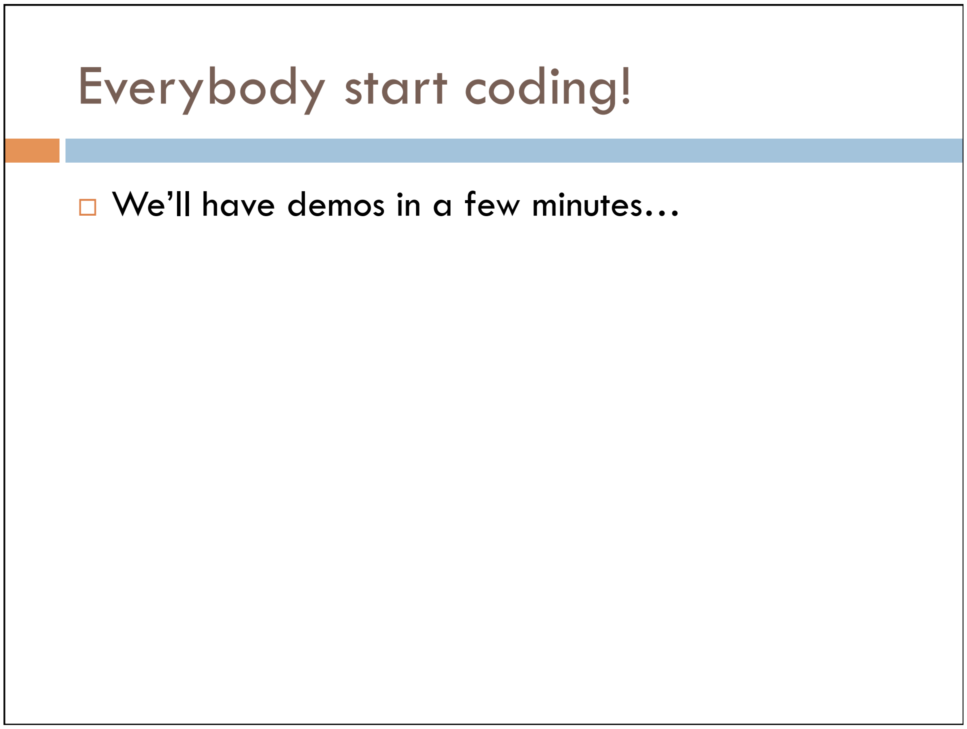
```
void setup(){
    // do nothing (why?)
}
```



Hints…loop()

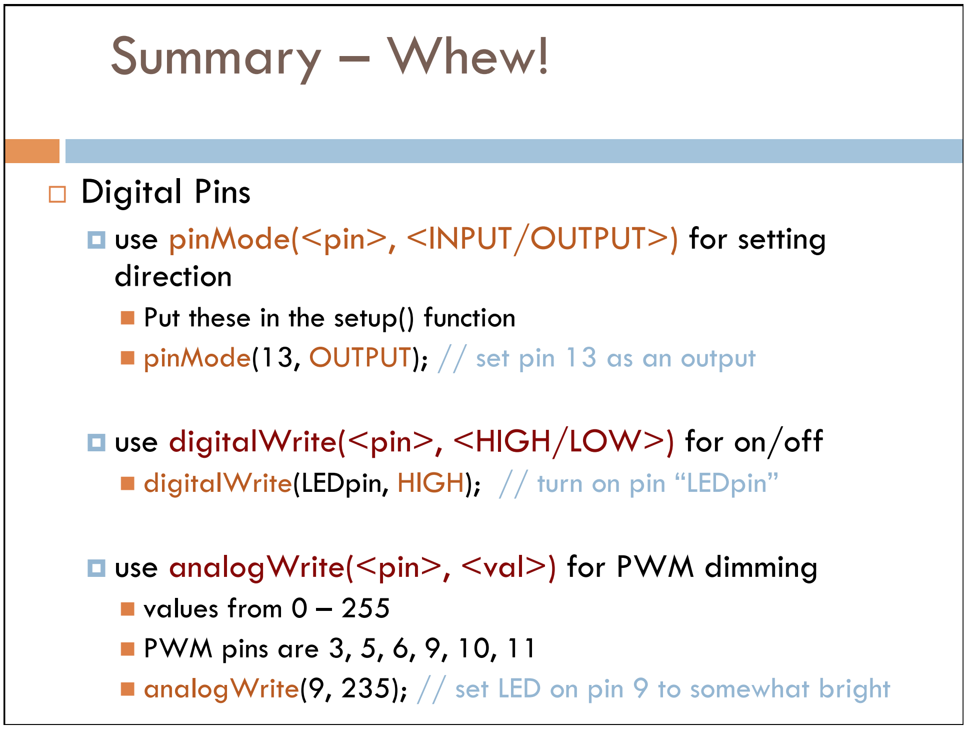
```
// loop is the function that repeats forever
void loop() {
   int delayTime = 100;    // a basic unit of delay (in msec)

   digitalWrite(0, HIGH);   // set LED 0 on
   delay(delayTime);        // wait delayTime milliseconds

   digitalWrite(0, LOW);    // set LED 0 off
   digitalWrite(1, HIGH);   // set LED 1 on
   delay(delayTime);        // wait delayTime milliseconds
   …// more setting and delaying…
   }
```

Or use for (int i=0; i<foo; i++), or random(min,max), etc…



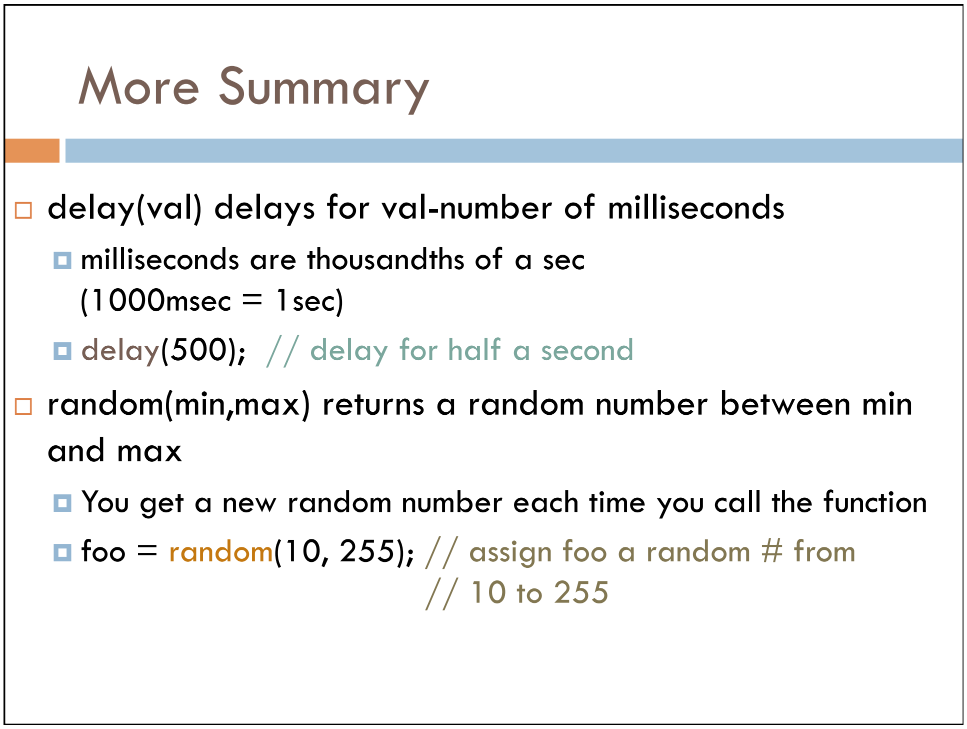
Everybody start coding!

□ We'll have demos in a few minutes…



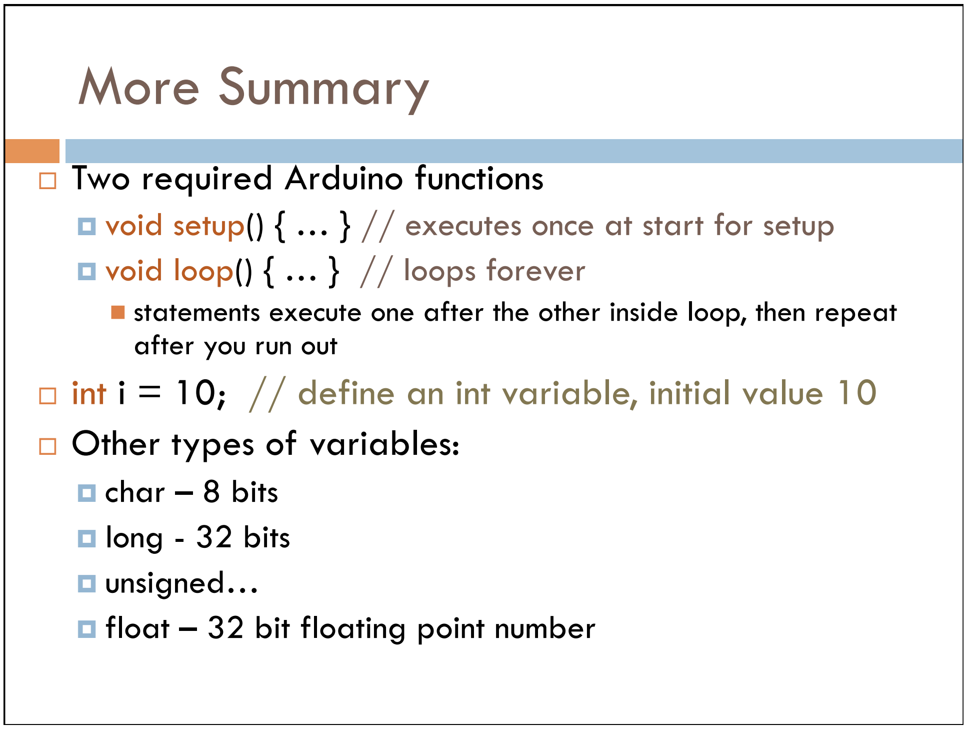
Summary – Whew!

- Digital Pins
  - use pinMode(<pin>, <INPUT/OUTPUT>) for setting direction
    - Put these in the setup() function
    - pinMode(13, OUTPUT); // set pin 13 as an output
  - use digitalWrite(<pin>, <HIGH/LOW>) for on/off
    - digitalWrite(LEDpin, HIGH); // turn on pin "LEDpin"
  - use analogWrite(<pin>, <val>) for PWM dimming
    - values from 0 – 255
    - PWM pins are 3, 5, 6, 9, 10, 11
    - analogWrite(9, 235); // set LED on pin 9 to somewhat bright



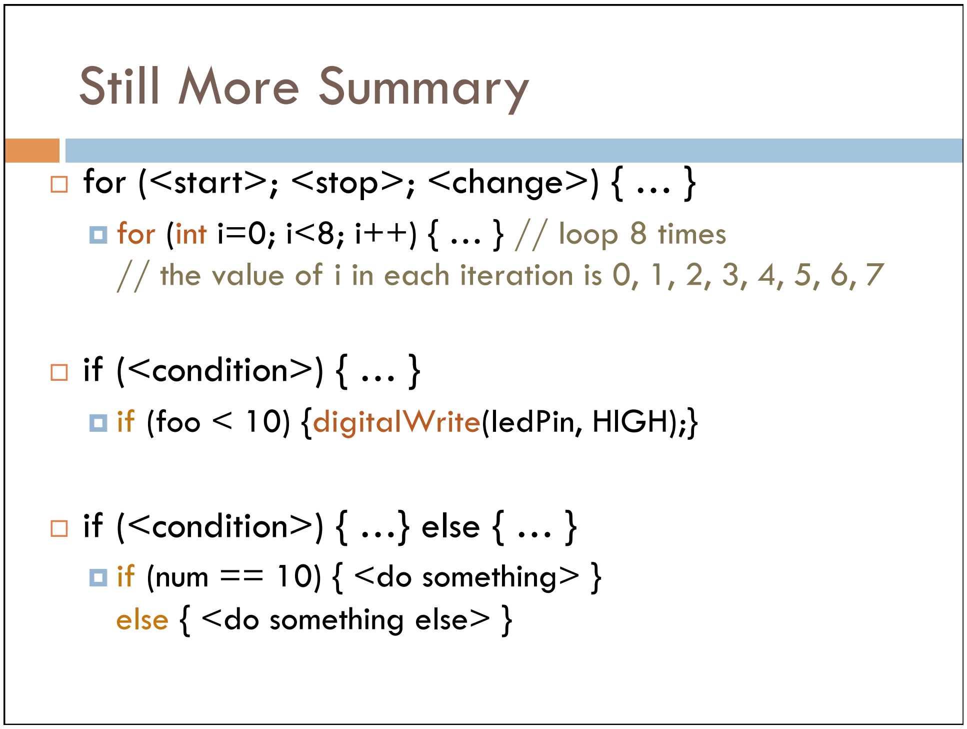
More Summary

- delay(val) delays for val-number of milliseconds
  - milliseconds are thousandths of a sec (1000msec = 1sec)
  - delay(500); // delay for half a second
- random(min,max) returns a random number between min and max
  - You get a new random number each time you call the function
  - foo = random(10, 255); // assign foo a random # from
    // 10 to 255



More Summary

- Two required Arduino functions
  - void setup() { … } // executes once at start for setup
  - void loop() { … } // loops forever
    - statements execute one after the other inside loop, then repeat after you run out
- int i = 10; // define an int variable, initial value 10
- Other types of variables:
  - char – 8 bits
  - long - 32 bits
  - unsigned…
  - float – 32 bit floating point number

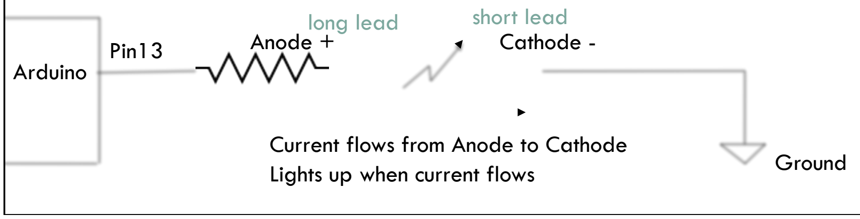


Still More Summary

- for (<start>; <stop>; <change>) { … }
  - for (int i=0; i<8; i++) { … } // loop 8 times
    // the value of i in each iteration is 0, 1, 2, 3, 4, 5, 6, 7
- if (<condition>) { … }
  - if (foo < 10) {digitalWrite(ledPin, HIGH);}
- if (<condition>) { …} else { … }
  - if (num == 10) { <do something> }
    else { <do something else> }

Last Summary (for now)

- □ LEDs – turn on when current flows from anode to cathode
  - Always use a current-limiting resistor!
  - Remember your resistor color codes
  - 220-470 ohm are good, general-purpose values for LEDs
  - Drive from Arduino on digital pins
  - Use PWM pins if you want to use analogWrite for dimming



Current flows from Anode to Cathode
Lights up when current flows

Resources

- □ http://arduino.cc/en/Tutorial/HomePage
- □ http://www.ladyada.net/learn/arduino/index.html
- □ http://todbot.com/blog/bionicarduino/
- □ http://todbot.com/blog/spookyarduino/
- □ http://sheepdogguides.com/arduino/aht0led.htm