

# Arduino Programming Part 1

ME 120

Mechanical and Materials Engineering

Portland State University

# Overview

## Arduino Environment

### Basic code components

- ❖ Two required functions: `startup()` and `loop()`
- ❖ Variables
- ❖ Calling built-in functions

# References

These notes borrow from

- ❖ Arduino web site
  - ▶ <http://arduino.cc/en/Guide/Environment>
  - ▶ <http://arduino.cc/en/Tutorial/HomePage>
- ❖ Adafruit tutorial #1 and 2
  - ▶ <http://www.ladyada.net/learn/arduino/lesson2.html>
- ❖ Leah Buechley's Introduction to Arduino
  - ▶ [http://web.media.mit.edu/~leah/LilyPad/03\\_arduino\\_intro.html](http://web.media.mit.edu/~leah/LilyPad/03_arduino_intro.html)

# Arduino Web Site References

Overview of the development environment

- ❖ <http://www.arduino.cc/en/Guide/Environment>

Language reference

- ❖ <http://arduino.cc/en/Reference/HomePage>

Code tutorials

- ❖ <http://arduino.cc/en/Tutorial/HomePage>

# Basic Process

## Design the circuit:

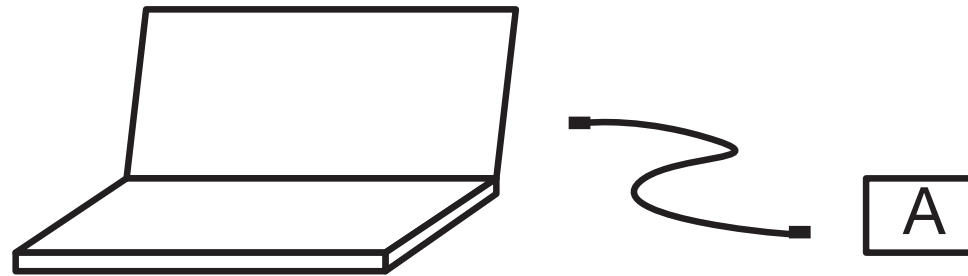
- ❖ What are electrical requirements of the sensors or actuators?
- ❖ Identify analog inputs (sensors)
- ❖ Identify digital inputs & outputs (buttons, LEDs, relays)

## Write the code

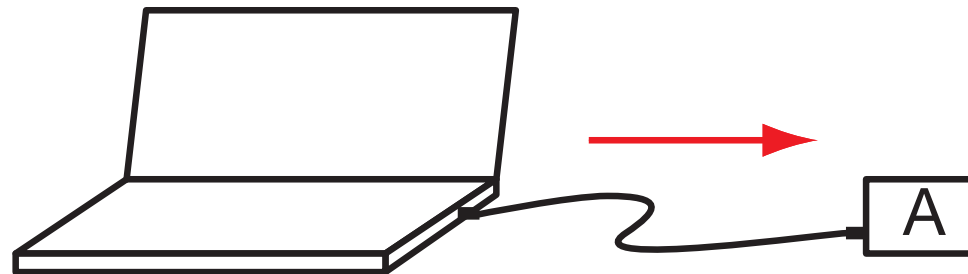
- ❖ Build incrementally
  - ▶ Get the simplest piece to work first
  - ▶ Add complexity and test at each stage
  - ▶ Save and Backup frequently
- ❖ Use variables, not constants
- ❖ Comment liberally

# Writing and Downloading Code

Write sketch on PC

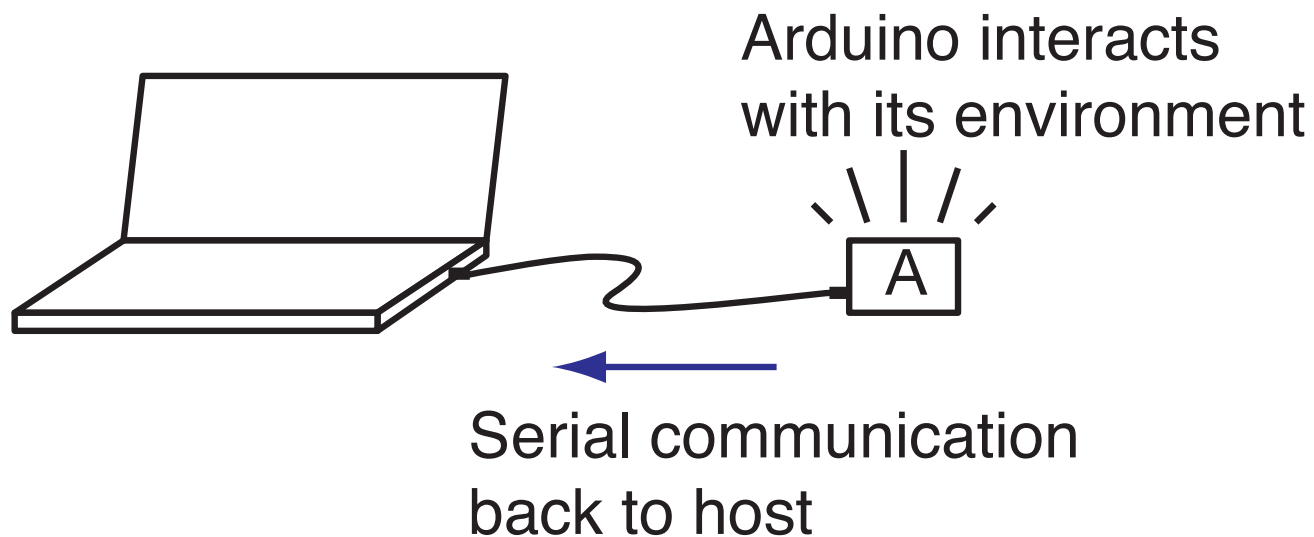


Download sketch to Arduino



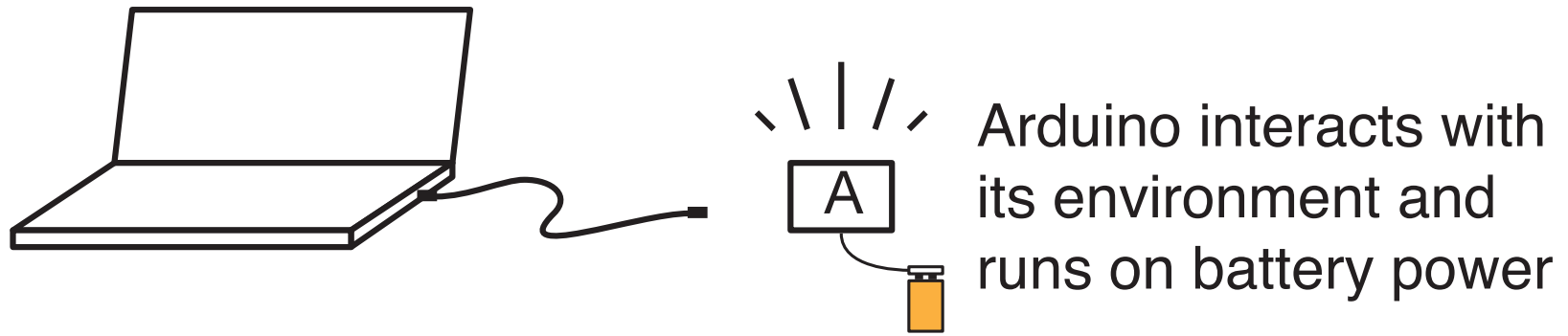
# Running Code While Tethered

Run sketch on Arduino  
and send data back to PC



# Running Code Stand-Alone

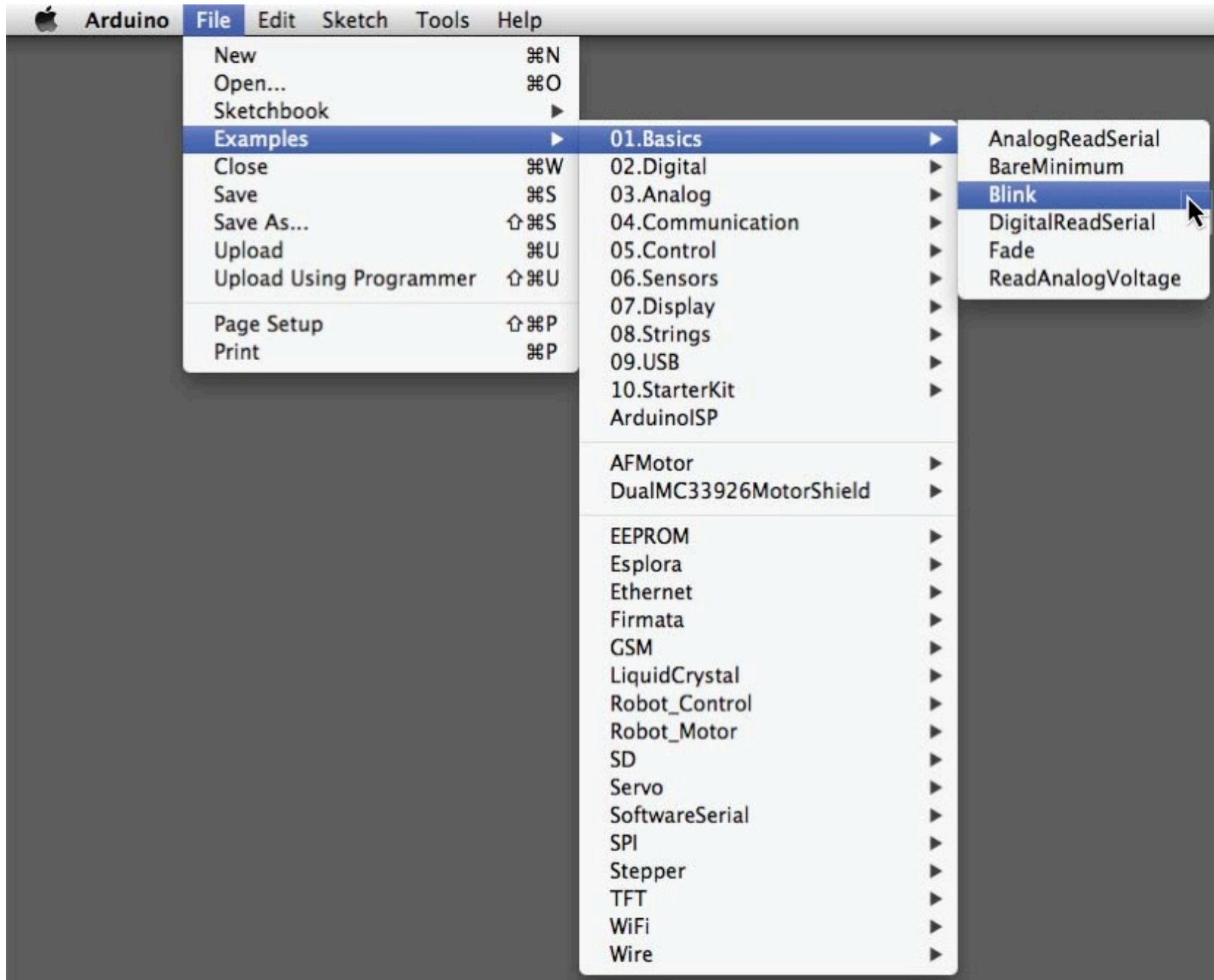
Run Arduino in stand alone mode





Open the example sketch,  
`blink.ino`

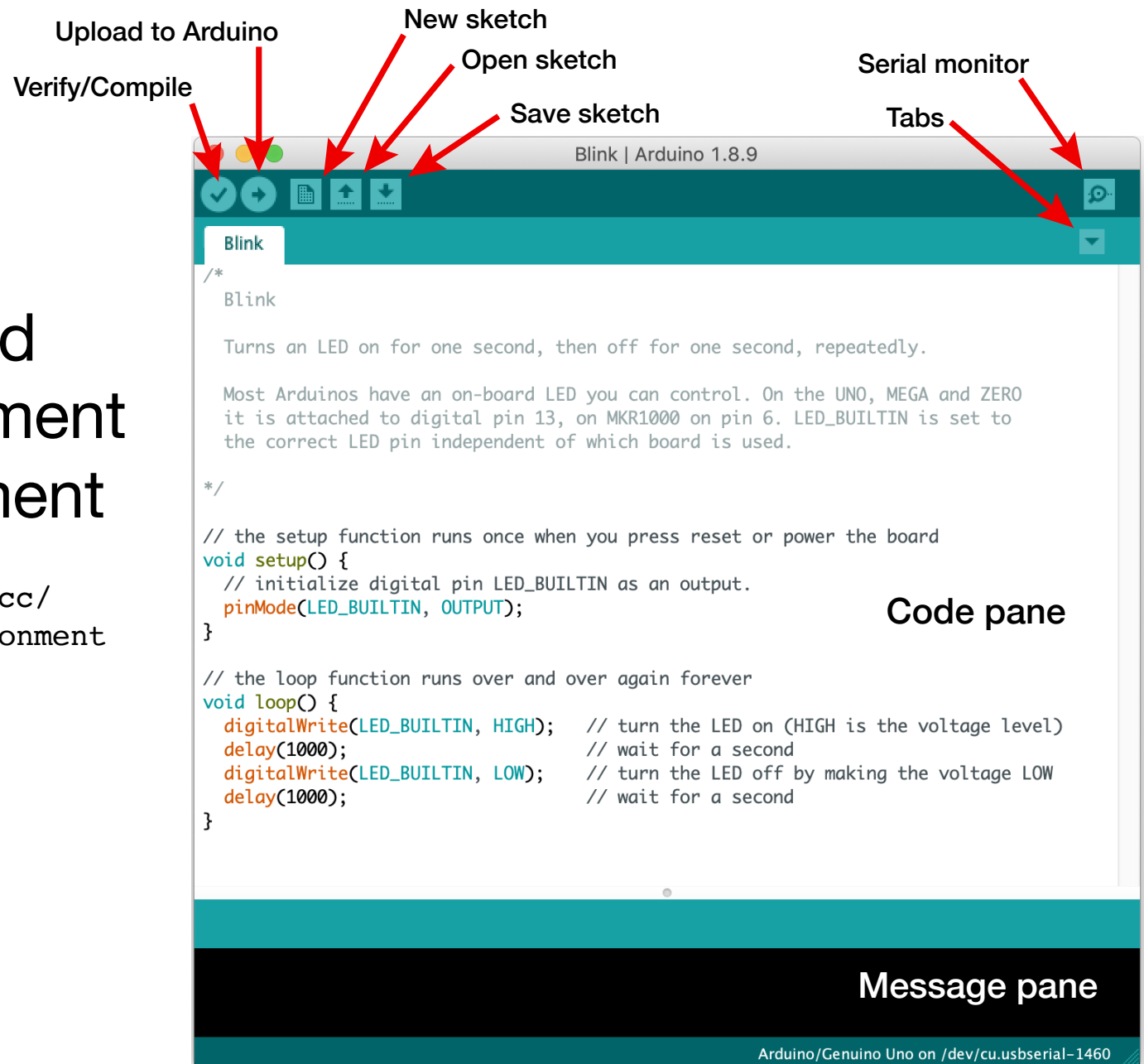
# Load “Blink” from the built-in examples



# Arduino IDE

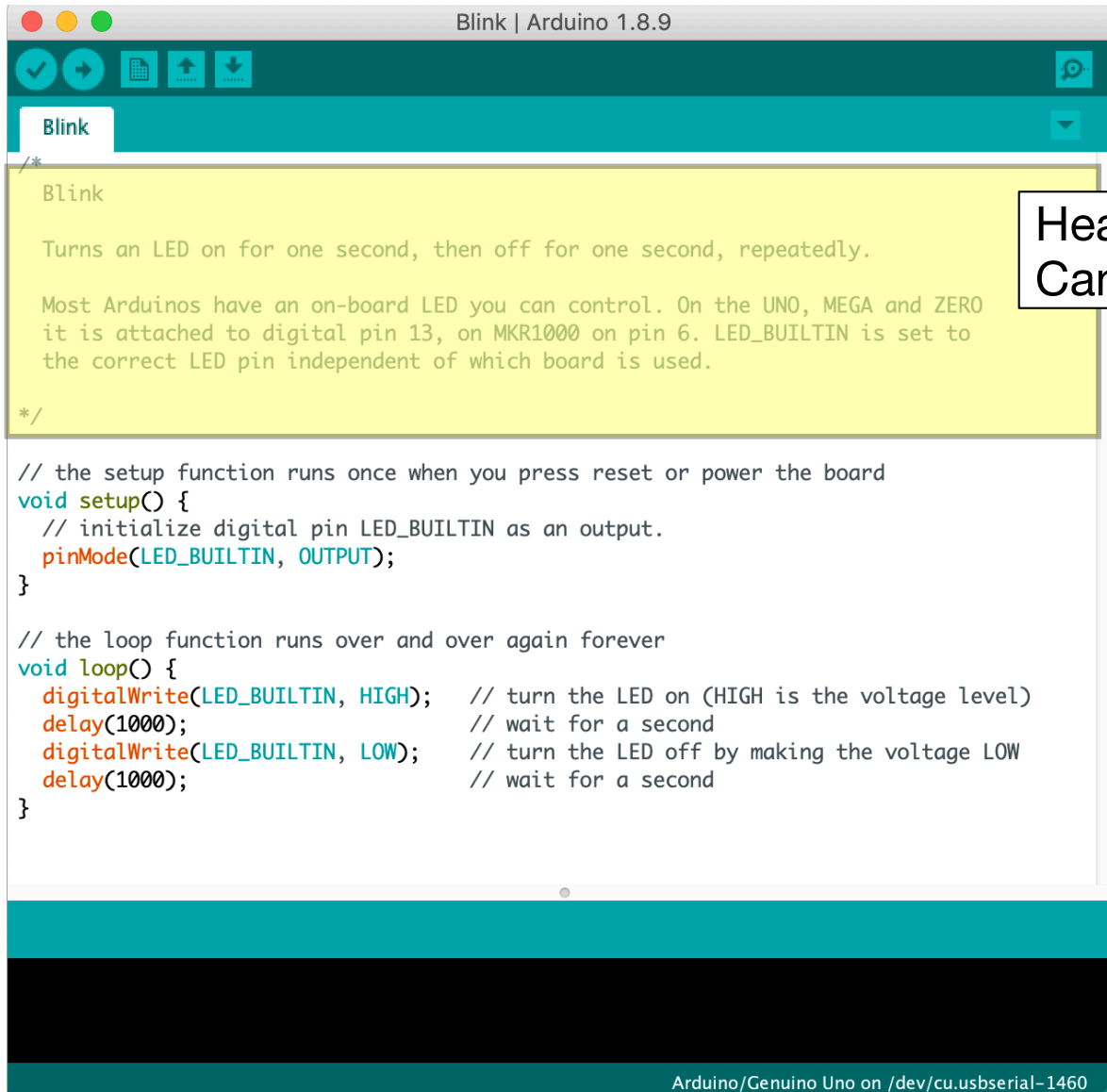
IDE = Integrated  
Development  
Environment

<http://www.arduino.cc/en/Guide/Environment>



# Common Code Structure

# Code Structure: Header



```
/*
  Blink

  Turns an LED on for one second, then off for one second, repeatedly.

  Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
  it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
  the correct LED pin independent of which board is used.

  */

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

Header provides information.  
Can also contain code

# Code Structure: setup function



```
Blink | Arduino 1.8.9
Blink
/*
  Blink

  Turns an LED on for one second, then off for one second, repeatedly.

  Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
  it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
  the correct LED pin independent of which board is used.

  */


// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}

Arduino/Genuino Uno on /dev/cu.usbserial-1460
```

**setup** function is executed only once at the start

# Code Structure: loop function



```
Blink | Arduino 1.8.9
Blink
/*
  Blink

  Turns an LED on for one second, then off for one second, repeatedly.

  Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
  it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
  the correct LED pin independent of which board is used.

  */

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}

Arduino/Genuino Uno on /dev/cu.usbserial-1460
```

**loop function is repeated indefinitely**

# Details of the Blink Code



# Preparing an Output Pin with `pinMode`

```
Blink | Arduino 1.8.9
Blink
/*
  Blink

  Turns an LED on for one second, then off for one second, repeatedly.

  Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
  it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
  the correct LED pin independent of which board is used.

  */

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}

Arduino/Genuino Uno on /dev/cu.usbserial-1460
```

`pinMode(LED_BUILTIN, Output)`  
Prepare pin number “LED\_BUILTIN”  
for outputs of 5 volts.  
“LED\_BUILTIN” is a pre-defined  
variable

# Turn on an output pin with `digitalWrite`

```
Blink | Arduino 1.8.9
Blink
/*
  Blink

  Turns an LED on for one second, then off for one second, repeatedly.

  Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
  it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
  the correct LED pin independent of which board is used.

  */

// the setup function runs once when you press reset or power
void setup() {
  // initialize digital pin LED_BUILTIN as an output:
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}

Arduino/Genuino Uno on /dev/cu.usbserial-1460
```

`digitalWrite(LED_BUILTIN, HIGH)`  
Set the LED\_BUILTIN pin to HIGH,  
which means “on”

`delay(1000);`  
Tell the microcontroller to do  
nothing for 1000 ms = 1 s

# Turn off an output pin with `digitalWrite`

```
Blink | Arduino 1.8.9
Blink
/*
  Blink

  Turns an LED on for one second, then off for one second, repeatedly.

  Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
  it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
  the correct LED pin independent of which board is used.

  */

// the setup function runs once when you press reset or power
void setup() {
  // initialize digital pin LED_BUILTIN as an output
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}

Arduino/Genuino Uno on /dev/cu.usbserial-1460
```

`digitalWrite(LED_BUILTIN, LOW)`  
Set the LED\_BUILTIN pin to LOW,  
which means “off”

`delay(1000);`  
Tell the microcontroller to do  
nothing for 1000 ms = 1 s