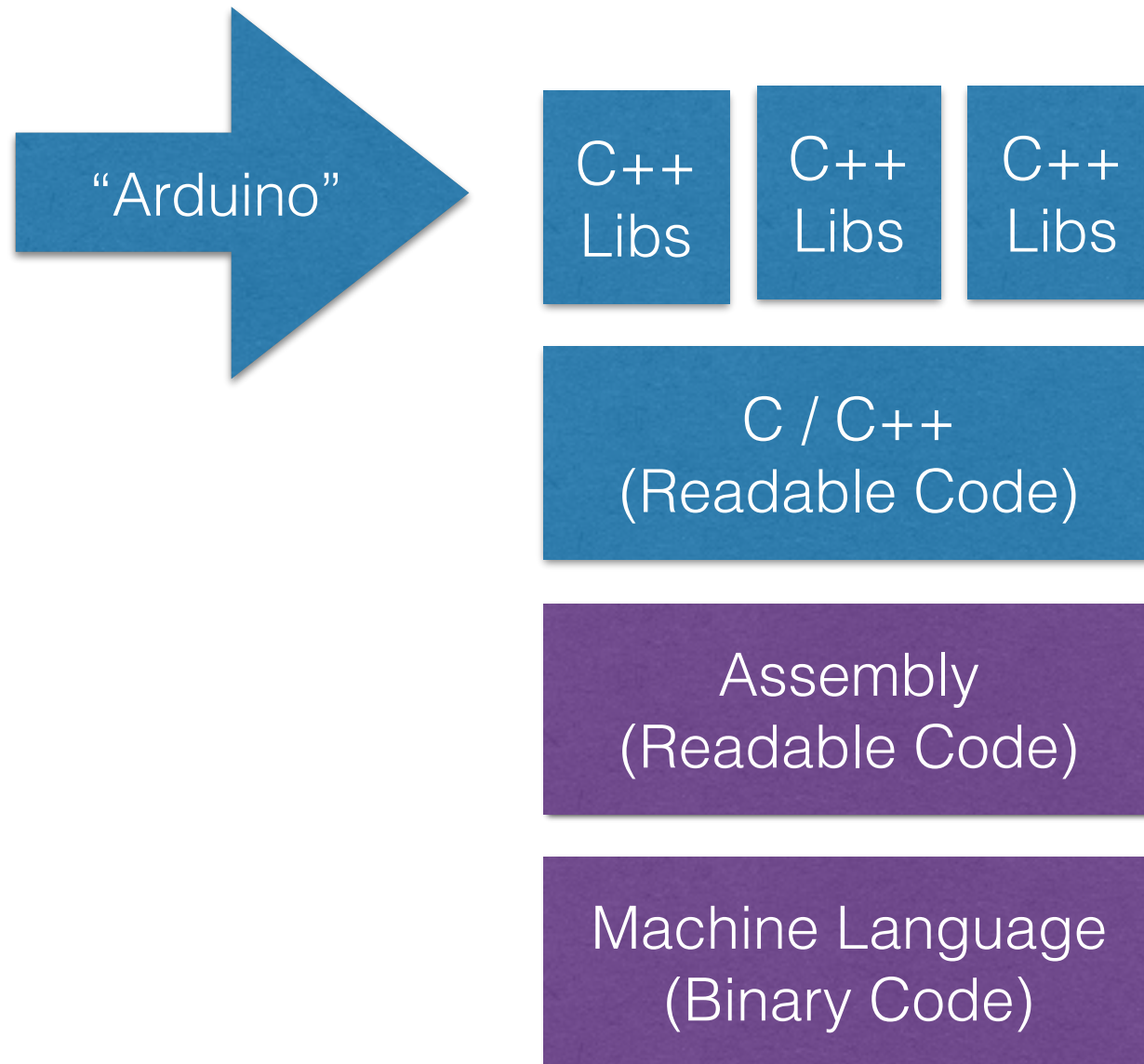


Arduino Programming

(C) 2014 James Lewis
james@baldengineer.com

Arduino Langauge



Arduino Language



C++
Libs

C++
Libs

C++
Libs

C / C++
(Readable Code)

Assembly
(Readable Code)

Machine Language
(Binary Code)

```
// the loop routine runs over  
void loop() {  
  digitalWrite(led, HIGH);  
  delay(1000);  
  digitalWrite(led, LOW);  
  delay(1000);  
}
```

Assembly Code Example

```
in r16, SREG      ; store SREG value  
cli              ; disable interrupts during timed seq  
sbi EECR, EEMPE   ; start EEPROM write  
sbi EECR, EEPE  
out SREG, r16     ; restore SREG value (I-bit)
```

```
0010 00000000C9461000C947E000C947E000C947E0095  
:100010000C947E000C947E000C947E000C947E0068  
:100020000C947E000C947E000C947E000C947E0058
```

Hello World (Blink)

```
/*  
  Blink  
  Turns on an LED on for one second, then off for one second, repeatedly.  
  
  This example code is in the public domain.  
  */  
  
// Pin 13 has an LED connected on most Arduino boards.  
// give it a name:  
int led = 13;  
  
// the setup routine runs once when you press reset:  
void setup() {  
  // initialize the digital pin as an output.  
  pinMode(led, OUTPUT);  
}  
|  
// the loop routine runs over and over again forever:  
void loop() {  
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000);             // wait for a second  
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW  
  delay(1000);             // wait for a second  
}
```

Hello World (Blink)

```
/*  
  Blink  
  Turns on an LED on for one second, then off for one second, repeatedly.  
  
  This example code is in the public domain.  
  */  
  
// Pin 13 has an LED connected on most Arduino boards.  
// give it a name:  
int led = 13;  
  
// the setup routine runs once when you press reset:  
void setup() {  
  // initialize the digital pin as an output.  
  pinMode(led, OUTPUT);  
}  
  
// the loop routine runs over and over again forever:  
void loop() {  
  digitalWrite(led, HIGH);    // turn the LED on (HIGH is the voltage level)  
  delay(1000);                // wait for a second  
  digitalWrite(led, LOW);     // turn the LED off by making the voltage LOW  
  delay(1000);                // wait for a second  
}
```



Variable

Hello World (Blink)

```
/*  
  Blink  
  Turns on an LED on for one second, then off for one second, repeatedly.  
  
  This example code is in the public domain.  
  */
```

```
// Pin 13 has an LED connected on most Arduino boards.  
// give it a name:  
int led = 13;
```

```
// the setup routine runs once when you press reset:
```

```
void setup() {  
  // initialize the digital pin as an output.  
  pinMode(led, OUTPUT);  
}
```

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

```
void loop() {  
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000);             // wait for a second  
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW  
  delay(1000);             // wait for a second  
}
```



Comments

Hello World (Blink)

```
/*  
  Blink  
  Turns on an LED on for one second, then off for one second, repeatedly.  
  
  This example code is in the public domain.  
  */
```

```
// Pin 13 has an LED connected on most Arduino boards.  
// give it a name:  
int led = 13;
```

```
// the setup routine runs once when you press reset:
```

```
void setup() {  
  // initialize the digital pin as an output.  
  pinMode(led, OUTPUT);  
}
```

```
}
```

```
|
```

```
//
```

```
void
```

```
di
```

```
de
```

```
di
```

```
de
```

```
}
```



Comments

Good Comment:
// Blue LED for Activity Indicator

Bad Comment:
// Pin 13

Hello World (Blink)

```
/*  
  Blink  
  Turns on an LED on for one second, then off for one second, repeatedly.  
  
  This example code is in the public domain.  
  */
```

```
// Pin 13 has an LED connected on most Arduino boards.  
// give it a name:  
int led = 13;
```

```
// the setup routine runs once when you press reset:
```

```
void setup() {  
  // initialize the digital pin as an output.  
  pinMode(led, OUTPUT);  
}
```

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

```
void loop() {  
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000);             // wait for a second  
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW  
  delay(1000);             // wait for a second  
}
```



Functions

Hello World (Blink)

```
/*  
  Blink  
  Turns on an LED on for one second, then off for one second, repeatedly.  
  
  This example code is in the public domain.  
  */
```

```
// Pin 13 has an LED connected on most Arduino boards.  
// give it a name:  
int led = 13;
```

```
// the setup routine runs once when you press reset:
```

```
void setup() {  
  // initialize the digital pin as an output.  
  pinMode(led, OUTPUT);  
}
```

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

```
void loop() {  
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000);             // wait for a second  
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW  
  delay(1000);             // wait for a second  
}
```



Functions

Hello World (Blink)

```
/*  
  Blink  
  Turns on an LED on for one second, then off for one second, repeatedly.  
  
  This example code is in the public domain.  
*/  
  
// Pin 13 has an LED connected on most Arduino boards.  
// give it a name:  
int led = 13;  
  
// the setup routine runs once when you press reset:  
void setup() {  
  // initialize the digital pin as an output:  
  pinMode(led, OUTPUT);  
}  
  
// the loop routine runs over and over again forever:  
void loop() {  
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000);             // wait for a second  
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW  
  delay(1000);             // wait for a second  
}
```



Instruction

Hello World (Blink)

```
/*  
  Blink  
  Turns on an LED on for one second, then off for one second, repeatedly.  
  
  This example code is in the public domain.  
  */  
  
// Pin 13 has an LED connected on most Arduino boards.  
// give it a name:  
int led = 13;  
  
// the setup routine runs once when you press reset:  
void setup() {  
  // initialize the digital pin as an output:  
  pinMode(led, OUTPUT);  
}  
  
// the loop routine runs over and over again forever:  
void loop() {  
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000);             // wait for a second  
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW  
  delay(1000);             // wait for a second  
}
```



Instruction

Hello World (Blink)

```
/*  
  Blink  
  Turns on an LED on for one second, then off for one second, repeatedly.  
  
  This example code is in the public domain.  
  */  
  
// Pin 13 has an LED connected on most Arduino boards.  
// give it a name:  
int led = 13;  
  
// the setup routine runs once when you press reset:  
void setup() {  
  // initialize the digital pin as an output:  
  pinMode(led, OUTPUT);  
}  
  
// loop routine runs over and over again forever:  
void loop() {  
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000);             // wait for a second  
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW  
  delay(1000);             // wait for a second  
}
```

Function Call

Hello World (Blink)

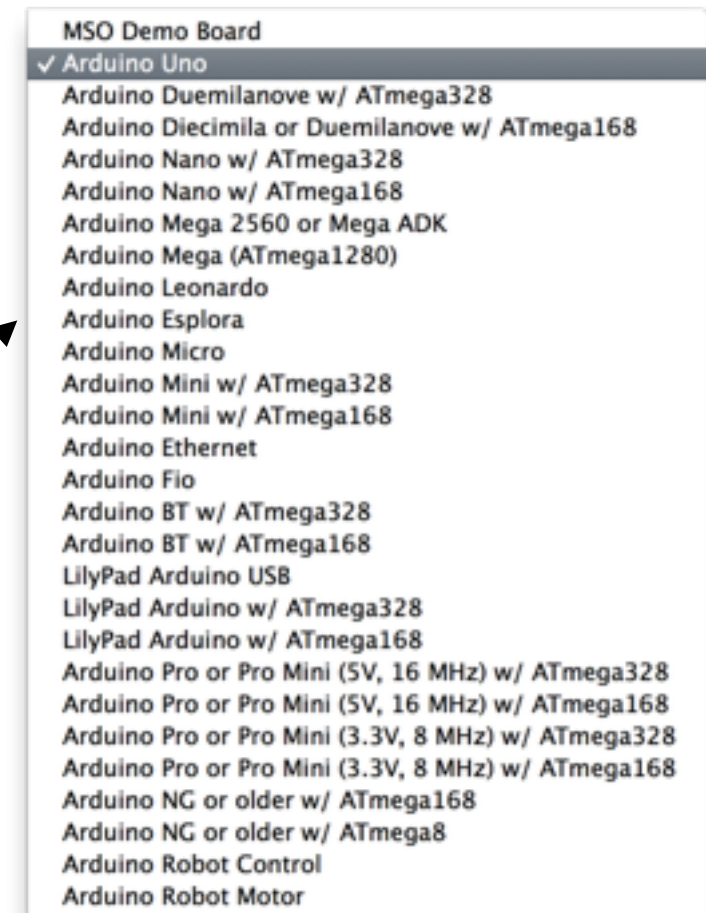
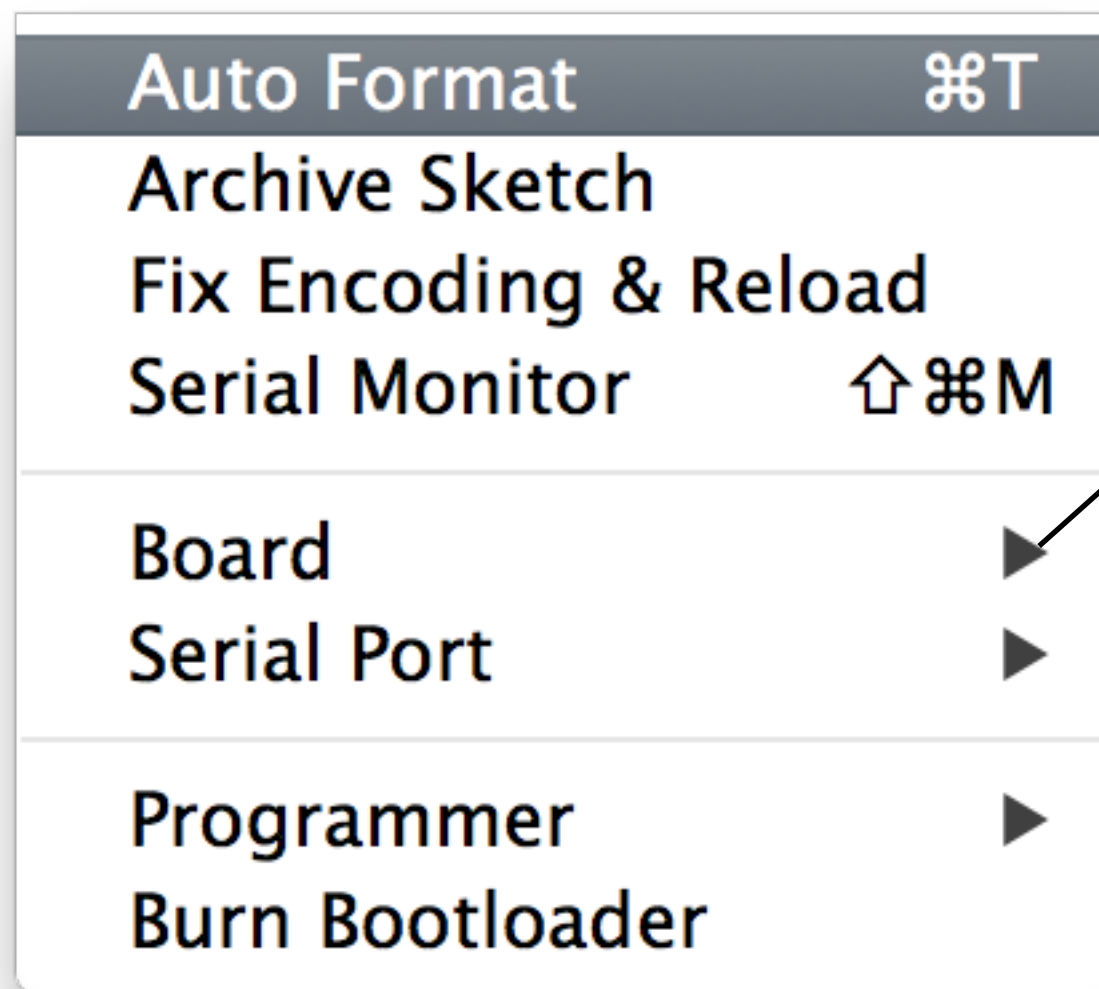
```
/*  
  Blink  
  Turns on an LED on for one second, then off for one second, repeatedly.  
  
  This example code is in the public domain.  
  */  
  
// Pin 13 has an LED connected on most Arduino boards.  
// give it a name:  
int led = 13;  
  
// the setup routine runs once when you press reset:  
void setup() {  
  // initialize the digital pin as an output:  
  pinMode(led, OUTPUT);  
}  
  
// the loop routine runs over and over again forever:  
void loop() {  
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000);             // wait for a second  
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW  
  delay(1000);             // wait for a second  
}
```

Arguments

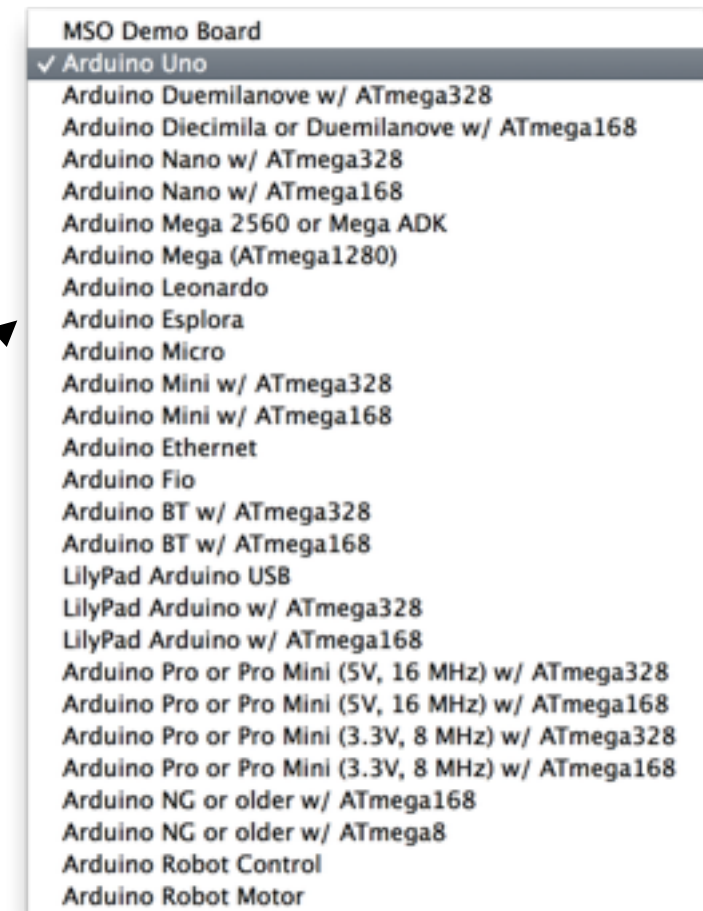
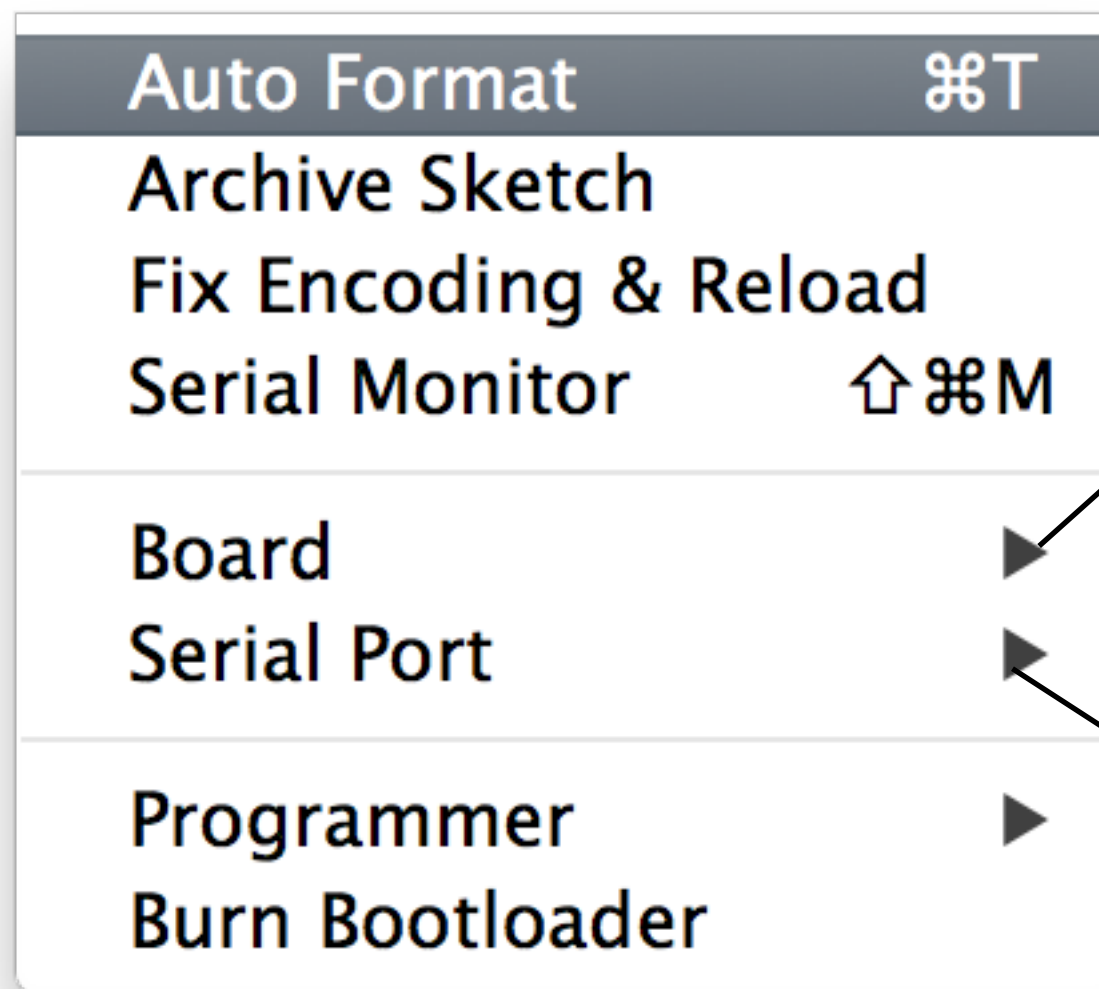
Hello World (Blink)

```
/*  
  Blink  
  Turns on an LED on for one second, then off for one second, repeatedly.  
  
  This example code is in the public domain.  
*/  
  
// Pin 13 has an LED connected on most Arduino boards.  
// give it a name:  
int led = 13;  
  
// the setup routine runs once when you press reset:  
void setup() {  
  // initialize the digital pin as an output.  
  pinMode(led, OUTPUT);  
}  
|  
// the loop routine runs over and over again forever:  
void loop() {  
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000);             // wait for a second  
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW  
  delay(1000);             // wait for a second  
}
```

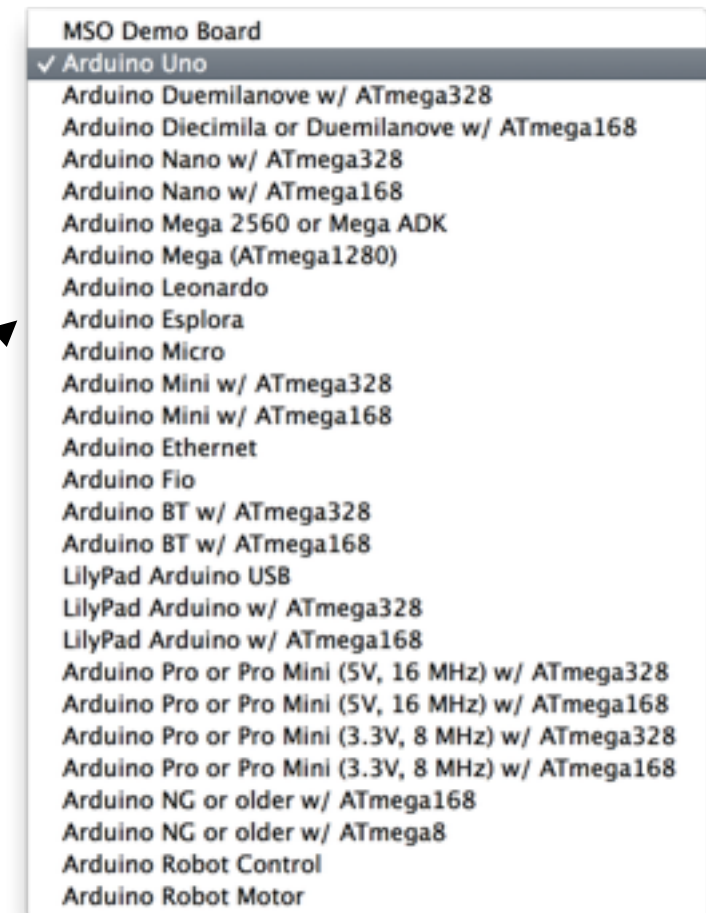
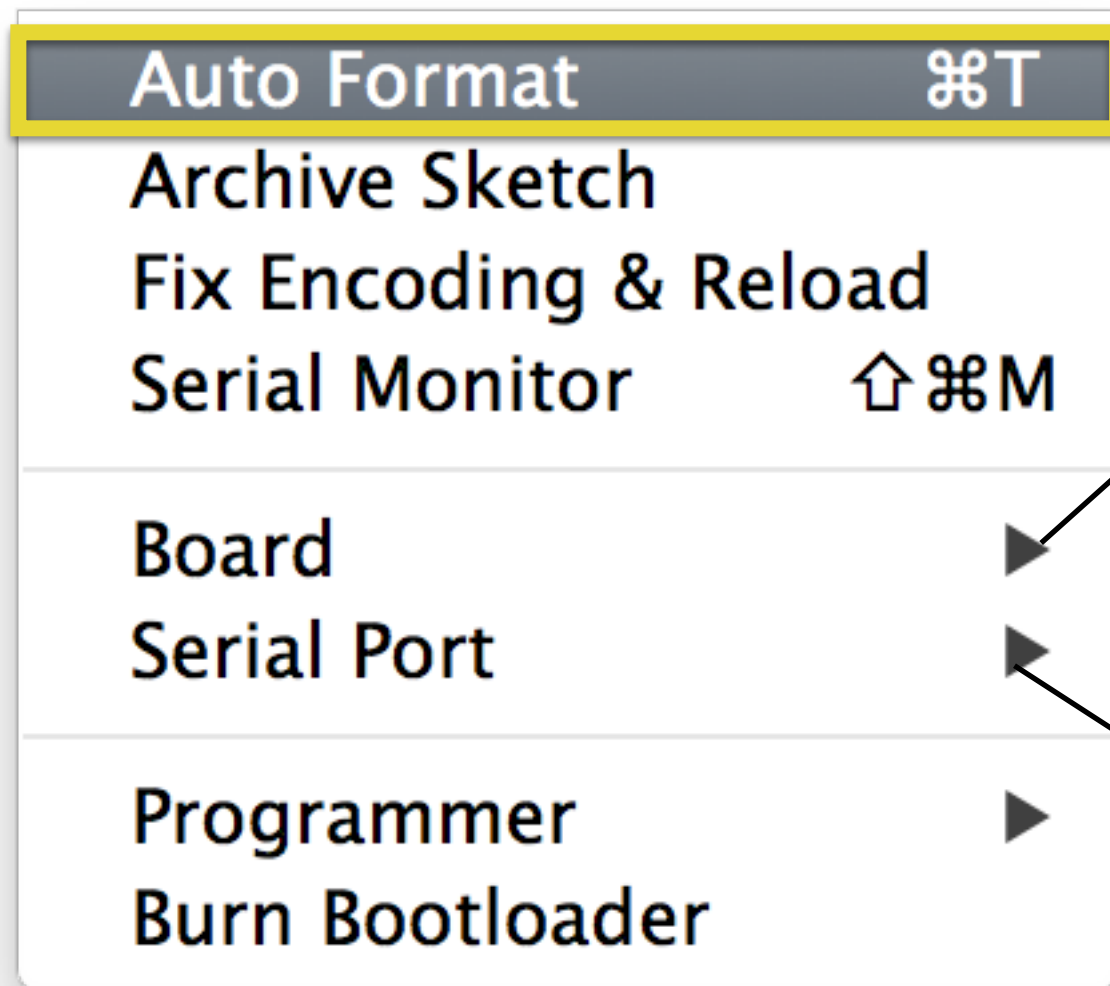

IDE Tools Menu



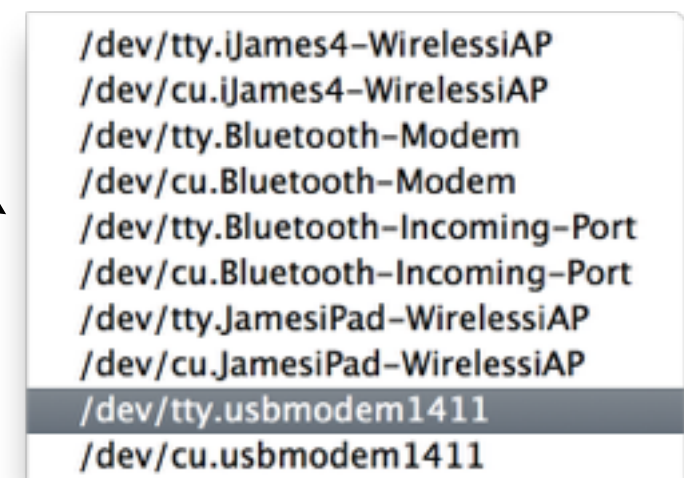
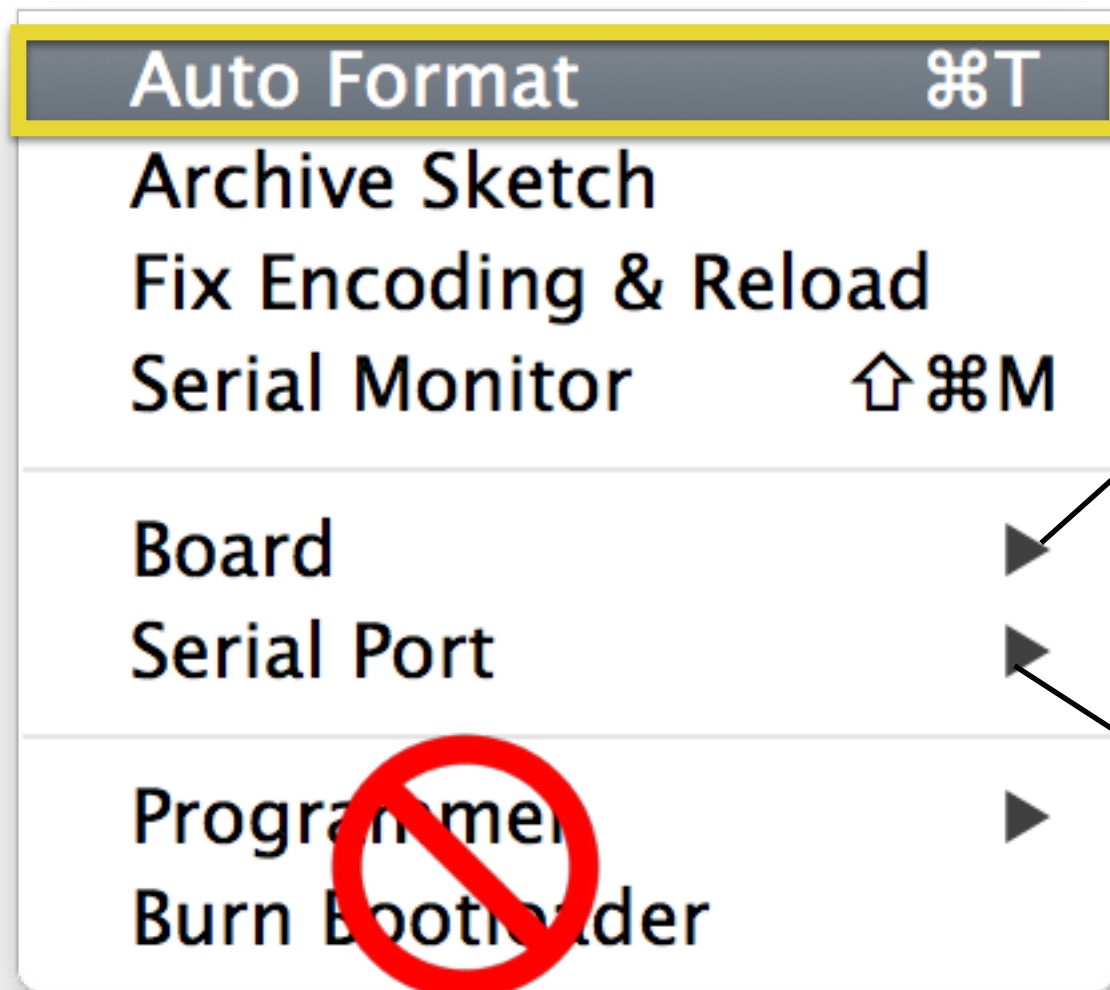
IDE Tools Menu



IDE Tools Menu



IDE Tools Menu



Arduino IDE



The screenshot shows the Arduino IDE window titled "Blink | Arduino 1.0.5". The main editor area displays the Blink example code, which is a simple program to turn an LED on and off repeatedly. The code is as follows:

```
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

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

Below the code editor, the "Compiler Output" window shows the message "Done compiling." followed by the compilation command:

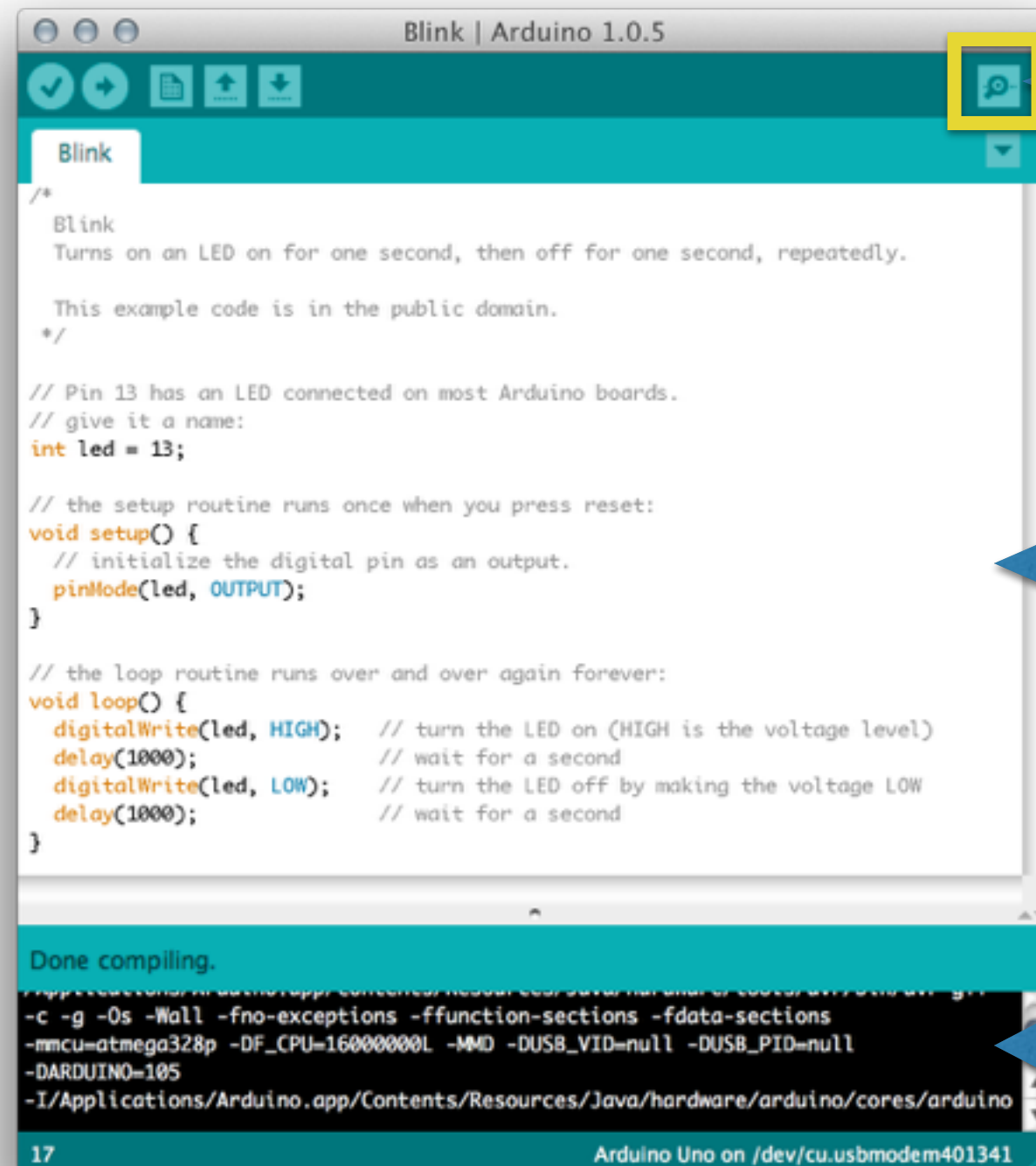
```
-c -g -Os -Wall -fno-exceptions -ffunction-sections -fdata-sections
-mmcu=atmega328p -DF_CPU=16000000L -MMD -DUSB_VID=null -DUSB_PID=null
-DARDUINO=105
-I/Applications/Arduino.app/Contents/Resources/Java/hardware/arduino/cores/arduino
```

The status bar at the bottom indicates "17" and "Arduino Uno on /dev/cu.usbmodem401341".

Type Stuff Here

Compiler Output

Arduino IDE

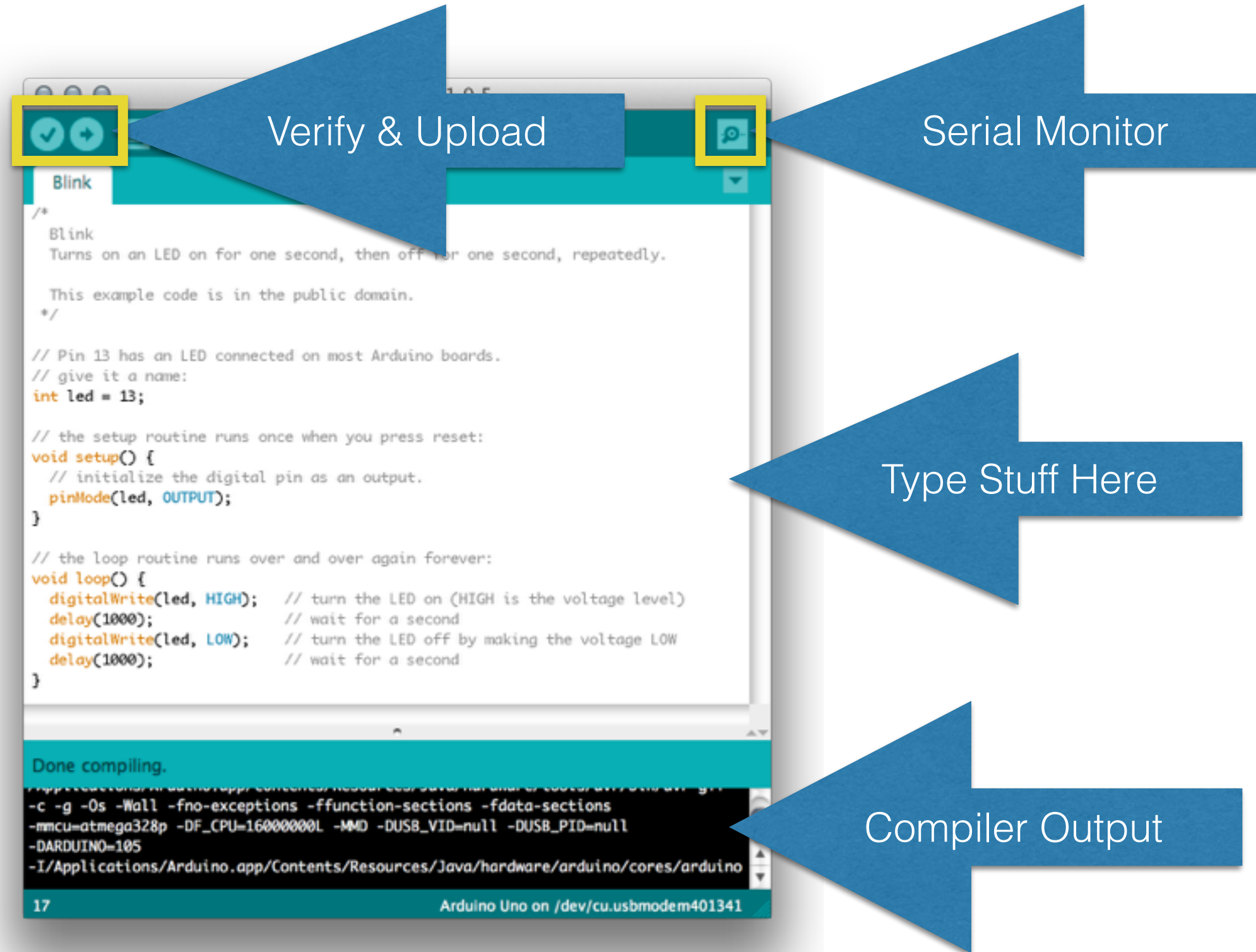


Serial Monitor

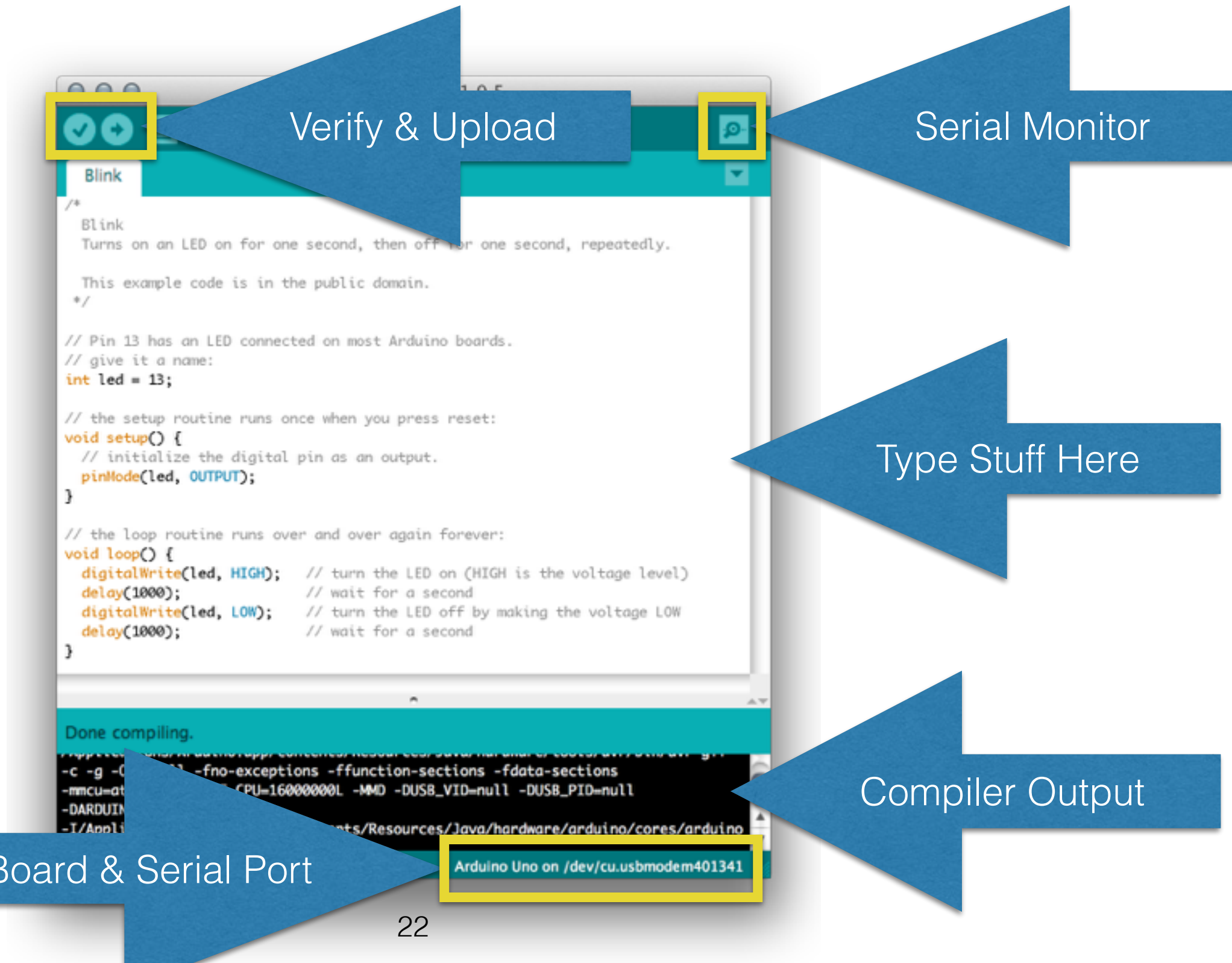
Type Stuff Here

Compiler Output

Arduino IDE



Arduino IDE



Blink Exercise

- Load the Blink Example and program it to your board
- Change the values of delay() to see how it affects the behavior



The screenshot shows the Arduino IDE interface. The top window displays the 'Blink' example code, which is a simple program to turn an LED on and off. The code is as follows:

```
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

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

The bottom window shows the serial monitor output, which includes the message 'Done compiling.' and the command line used to compile the code:

```
-c -g -Os -Wall -fno-exceptions -ffunction-sections -fdata-sections
-mcu=atmega328p -DF_CPU=16000000L -MMD -DUSB_VID=null -DUSB_PID=null
-DARDUINO=105
-I/Applications/Arduino.app/Contents/Resources/Java/hardware/arduino/cores/arduino
```

The status bar at the bottom indicates '17' and 'Arduino Uno on /dev/cu.usbmodem401341'.

Check the correct board and serial port are selected in the tools menu!



```
sketch_mar29b §  
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  Serial.println("Hello World!");  
  delay(2000);  
}
```

Clipboard does not contain a string

7 Arduino Uno on /dev/cu.usbmodem401341

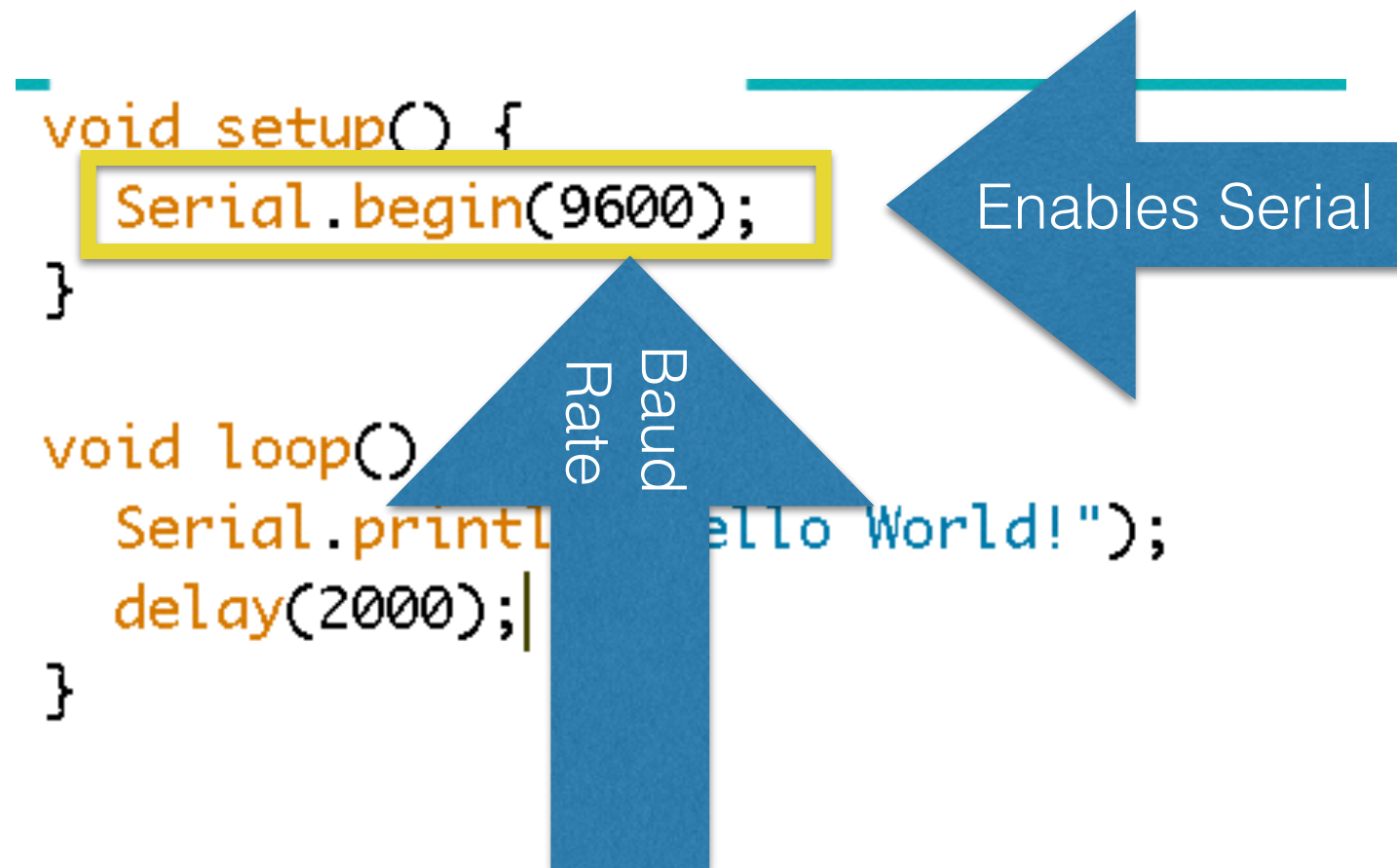
Hello World

Serial Example

Serial objects

```
void setup() {  
    Serial.begin(9600);  
}  
  
void loop() {  
    Serial.println("Hello World!");  
    delay(2000);  
}
```

Serial objects



Serial objects

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

```
void loop() {  
  Serial.println("Hello World!");  
  delay(2000);  
}
```

Print
and Println

Variables
Strings
Control Characters

Serial objects

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

```
void loop() {  
  Serial.println("Hello World!");  
  delay(2000);  
}
```

Print
and Println

Variables
Strings
Control Characters

NOTE: Strings and Variables Can't be used on the same line

Hello World (Serial)

- Load up the serial code to the right
- Exercise:
 - Change the 2000 in delay into a variable.
 - Print value of variable on same line as “Hello World”

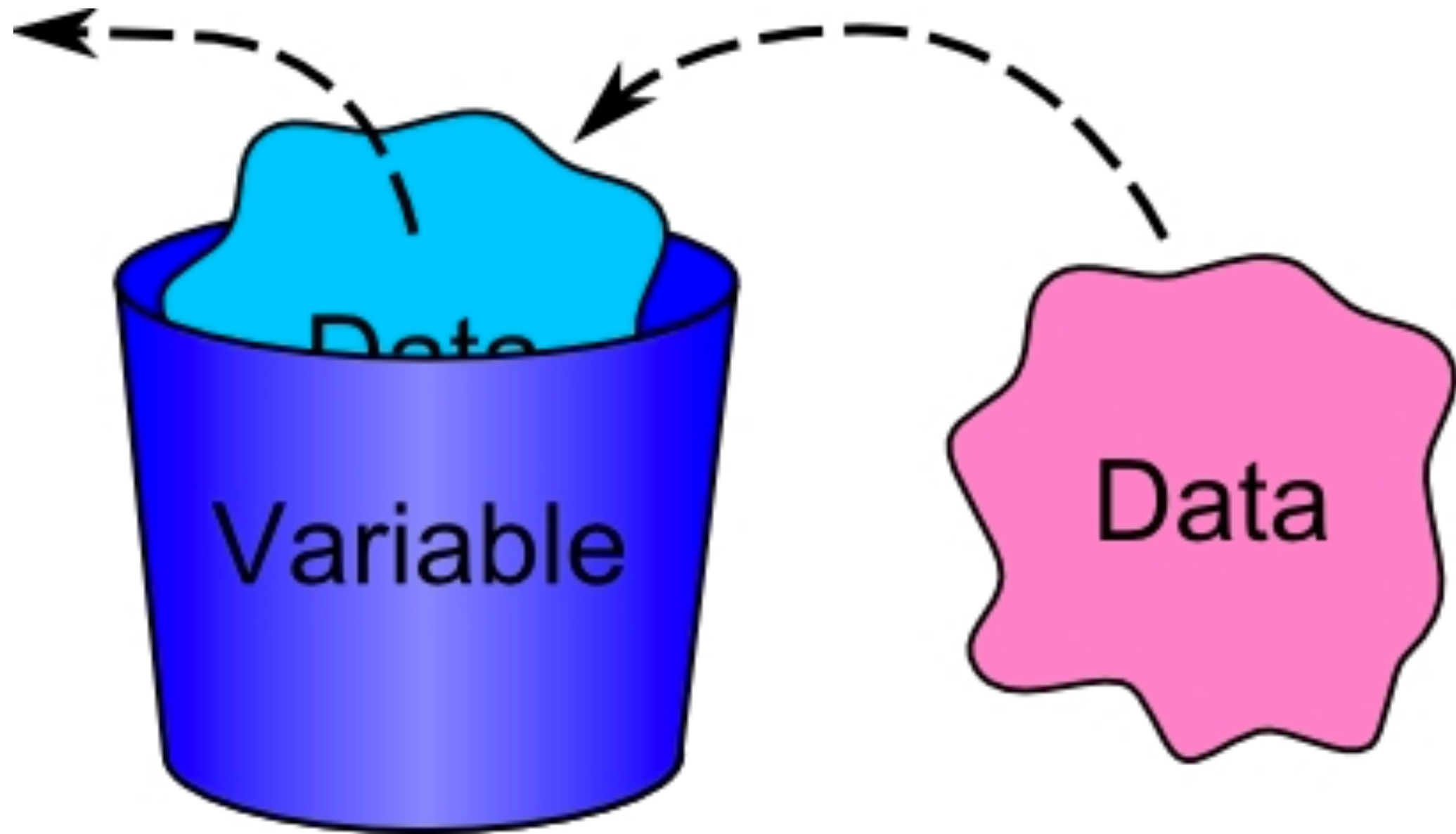


```
sketch_mar29b $
void setup() {
  Serial.begin(9600);
}

void loop() {
  Serial.println("Hello World!");
  delay(2000);
}
```

Clipboard does not contain a string

7 Arduino Uno on



Variables

How Much **Memory** is in your Arduino?

Variable Types

	Bits	Unsigned Range	Signed Range
byte	8	0 to 255	N/A
char	8	0 to 255 'A'..'b'..'X'	N/A
int	16	0 to 65535	-32,767 to 32,766
long	32	0 to 4,294,967,295	-2,147,483,648 to 2,147,483,647
float	32	$\pm 3.4028235E+38$	n/a
double	32	n/a	n/a

Variable Do and Don't

- DO Use Descriptive Names
 - “BlueLED”, “ActivityIndicator”
- DON'T Use Bad Names
 - “Integer”, “Pin13”
- DO Stick to a naming convention
 - Variables are Case Sensitive!
- DON'T use same name for Local and Global Variables

Variable Scope

```
int LEDpin = 13;
int ButtonPin = 2;

void setup() {
    pinMode(LEDpin, OUTPUT);
    pinMode(ButtonPin, INPUT);
}

void loop() {
    int buttonValue = digitalRead(ButtonPin);
    digitalWrite(LEDpin, buttonValue);
}
```

Variable Scope

```
int LEDpin = 13;  
int ButtonPin = 2;
```



Global

```
void setup() {  
    pinMode(LEDpin, OUTPUT);  
    pinMode(ButtonPin, INPUT);  
}
```

```
void loop() {  
    int buttonValue = digitalRead(ButtonPin);  
    digitalWrite(LEDpin, buttonValue);  
}
```

Variable Scope

```
int LEDpin = 13;  
int ButtonPin = 2;
```

Global

```
void setup() {  
    pinMode(LEDpin, OUTPUT);  
    pinMode(ButtonPin, INPUT);  
}
```

```
void loop() {  
    int buttonValue = digitalRead(ButtonPin);  
    digitalWrite(LEDpin, buttonValue);  
}
```

Local to loop()

Variable Don't!

```
int LEDpin = 13;
int ButtonPin = 2;
int buttonValue = 0;

void setup() {
    pinMode(LEDpin, OUTPUT);
    pinMode(ButtonPin, INPUT);
}

void loop() {
    int buttonValue = digitalRead(ButtonPin);
    digitalWrite(LEDpin, buttonValue);
}
```

Variable Don't!

```
int LEDpin = 13;  
int ButtonPin = 2;  
int buttonValue = 0;
```

```
void setup() {  
    pinMode(LEDpin, OUTPUT);  
    pinMode(ButtonPin, INPUT);  
}
```

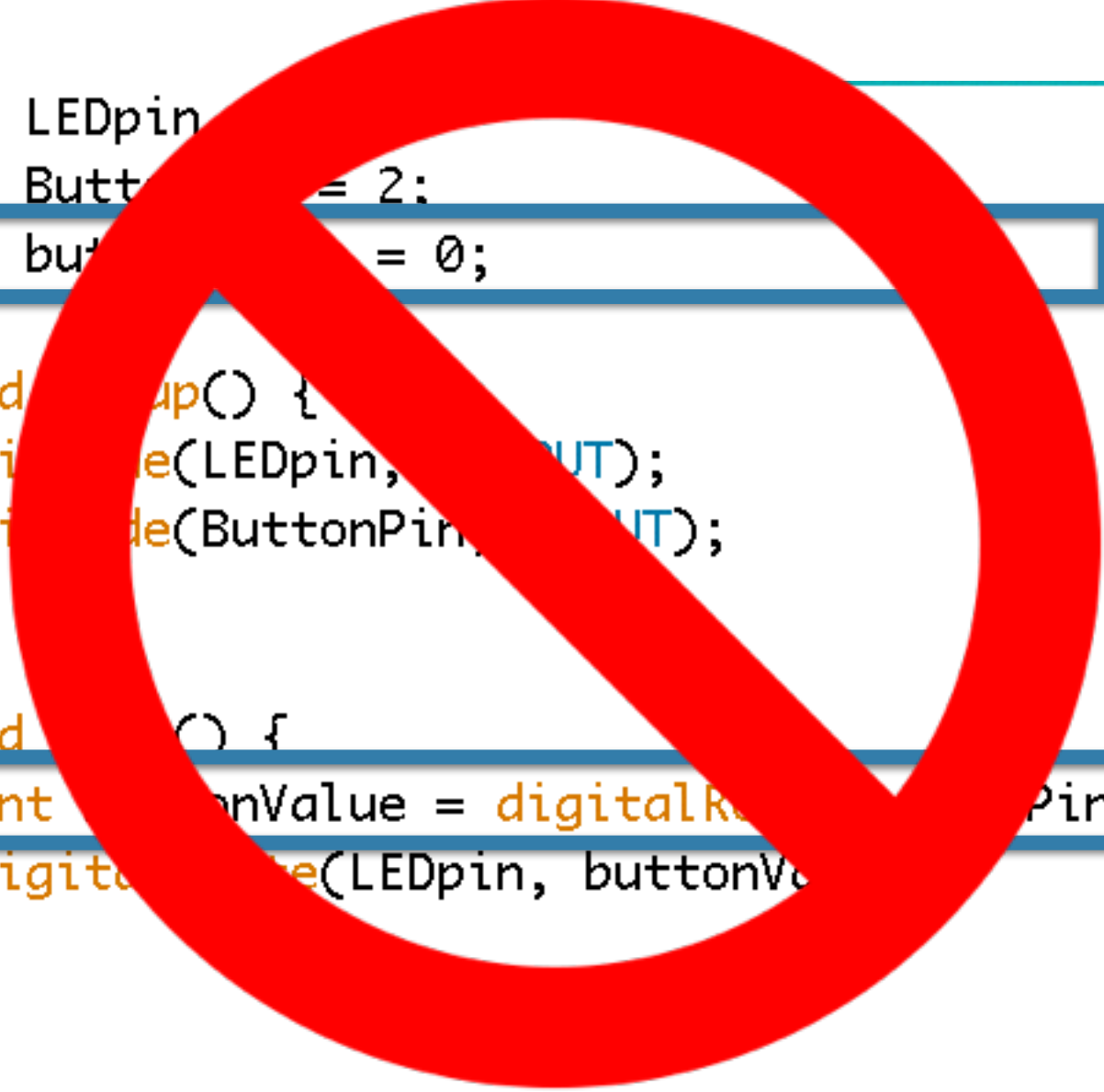
```
void loop() {  
    int buttonValue = digitalRead(ButtonPin);  
    digitalWrite(LEDpin, buttonValue);  
}
```


Variable Don't!

```
int LEDpin = 13;
int ButtonPin = 2;
int buttonValue = 0;

void setup() {
  pinMode(LEDpin, OUTPUT);
  pinMode(ButtonPin, INPUT);
}

void loop() {
  int buttonValue = digitalRead(ButtonPin);
  digitalWrite(LEDpin, buttonValue);
}
```



arrays



```
int analogReadings[6];  
analogReadings[0] = analogRead(A0);  
analogReadings[1] = analogRead(A1);  
analogReadings[2] = analogRead(A2);  
analogReadings[3] = analogRead(A3);  
analogReadings[4] = analogRead(A4);  
analogReadings[5] = analogRead(A5);
```

arrays



```
int analogReadings[6];  
analogReadings[0] = analogRead(A0);  
analogReadings[1] = analogRead(A1);  
analogReadings[2] = analogRead(A2);  
analogReadings[3] = analogRead(A3);  
analogReadings[4] = analogRead(A4);  
analogReadings[5] = analogRead(A5);
```

Size

Elements

arrays

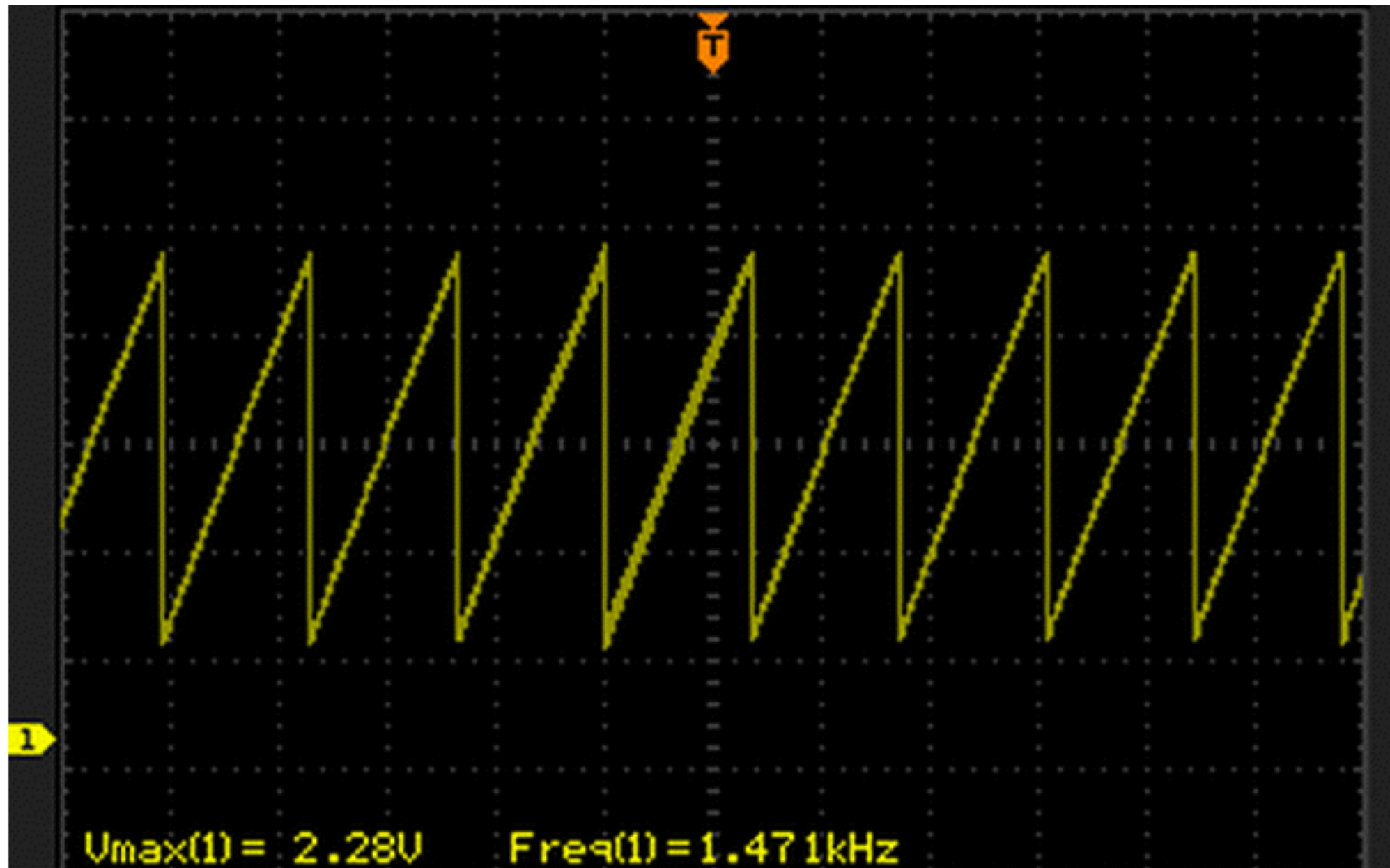


```
int analogReadings[6];  
analogReadings[0] = analogRead(A0);  
analogReadings[1] = analogRead(A1);  
analogReadings[2] = analogRead(A2);  
analogReadings[3] = analogRead(A3);  
analogReadings[4] = analogRead(A4);  
analogReadings[5] = analogRead(A5);
```

Size

Elements

arrays are
0-index based.
So last element
is always
“1 less”
than the size!



Pin Functions

pinMode()

	Analog (A0..A5)	Digital (0..13)
INPUT	Digital Input, Pull-Up Off	Digital Input, Pull-Up Off
INPUT_PULLUP	Digital Input, Pull-Up On	Digital Input, Pull-Up On
OUTPUT	Digital Output	Digital Output

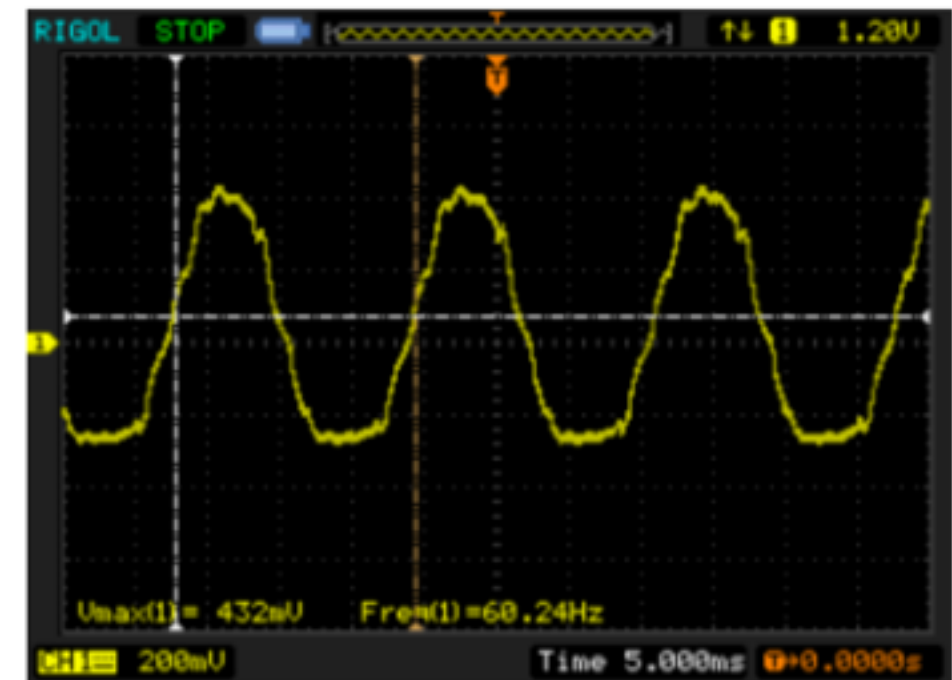
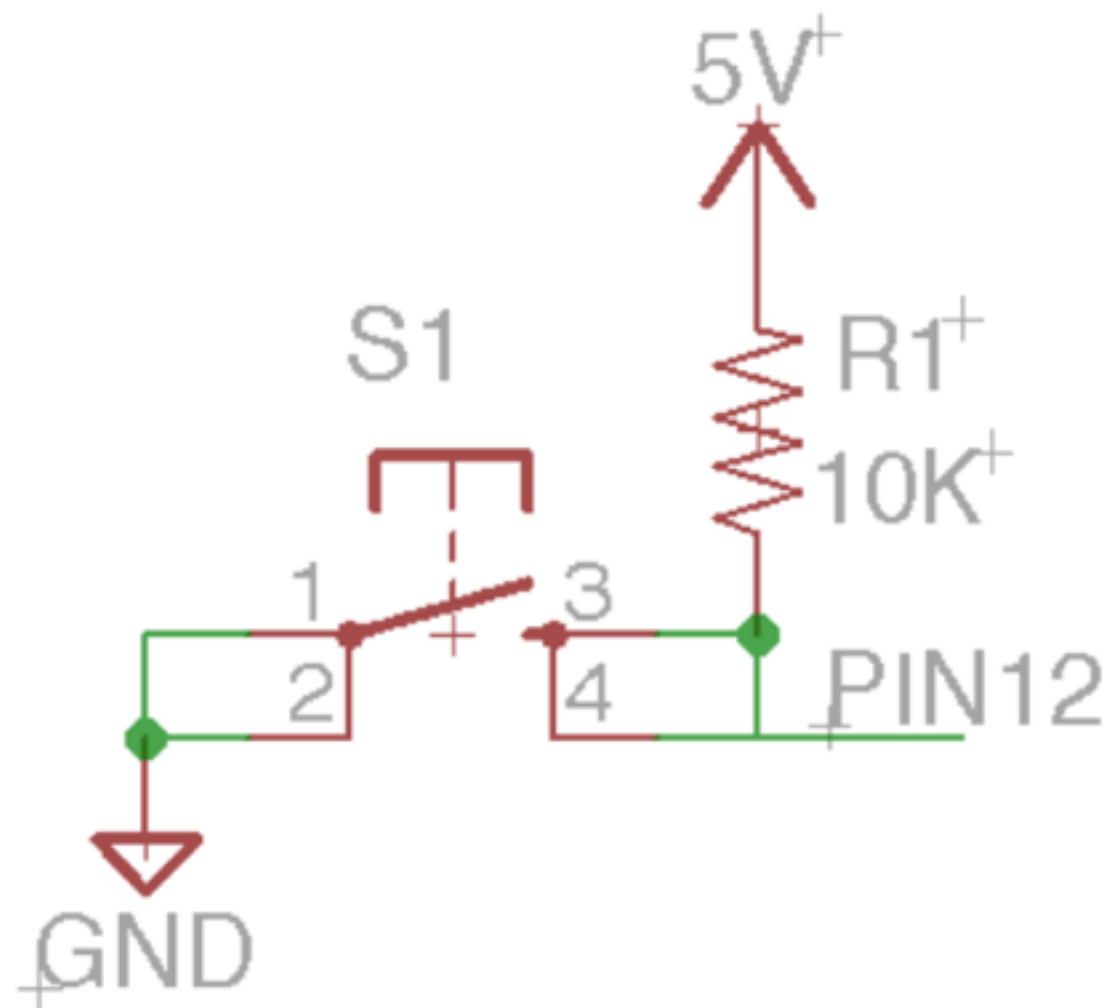
Analog Pins can be used as Digital Pins
pinMode(INPUT, Ax) isn't necessary for analogRead()

digitalRead() & digitalWrite()

```
int ButtonPin = 4;  
int ButtonValue;  
ButtonValue = digitalRead(ButtonPin);
```

```
int LEDPin = 7;  
digitalWrite(LEDPin, HIGH);  
digitalWrite(LEDPin, LOW);
```

Pull-Up Resistor



Picture 2: Oscilloscope Screenshot of Floating Pin



INPUTs almost always need a Pull-Up or Pull-Down
`pinMode(INPUT_PULLUP)` Turns on the Internal Pull-Up Resistor



<http://www.baldengineer.com/tutorials/arduino-pull-ups/>

I/O Exercise

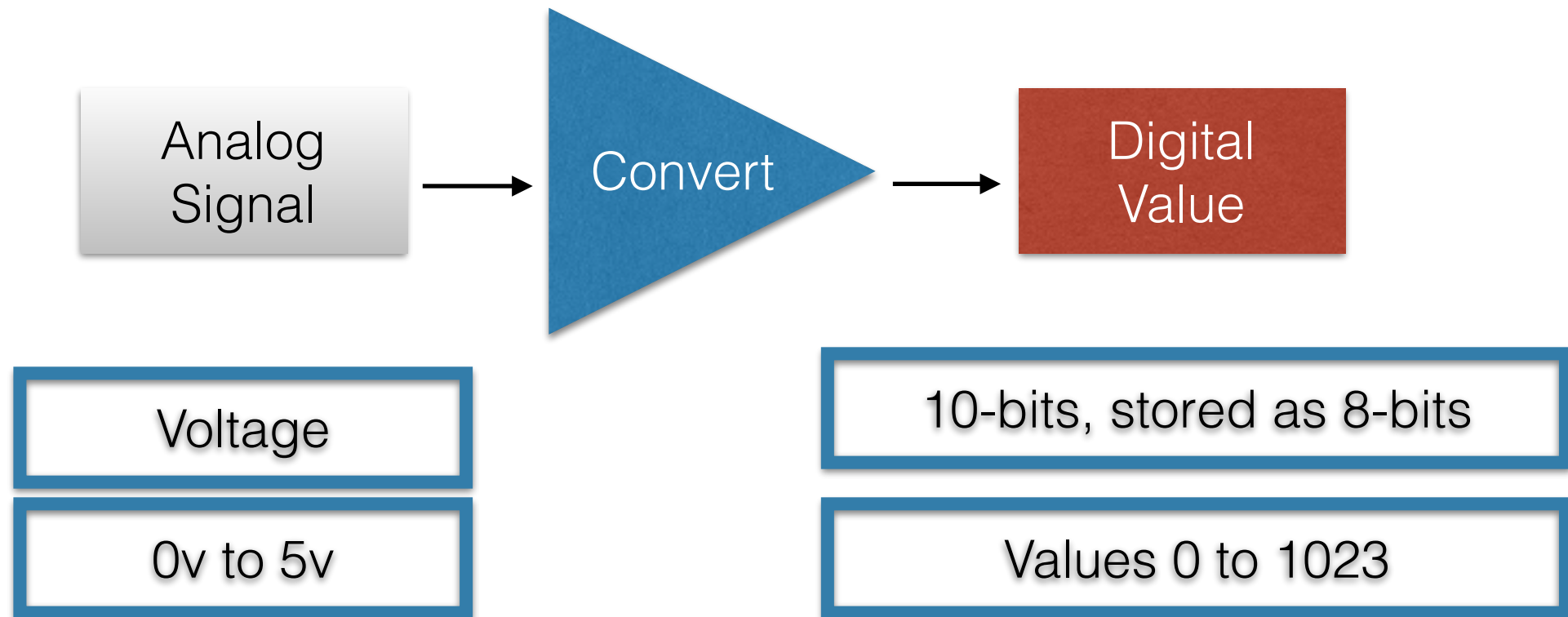
- Objective: Understand Inputs and Outputs
- Exercise: “Fix” the random LED

```
int LEDpin = 13;
int ButtonPin = 2;

void setup() {
  pinMode(LEDpin, OUTPUT);
  pinMode(ButtonPin, INPUT);
}

void loop() {
  int buttonValue = digitalRead(ButtonPin);
  digitalWrite(LEDpin, buttonValue);
}
```

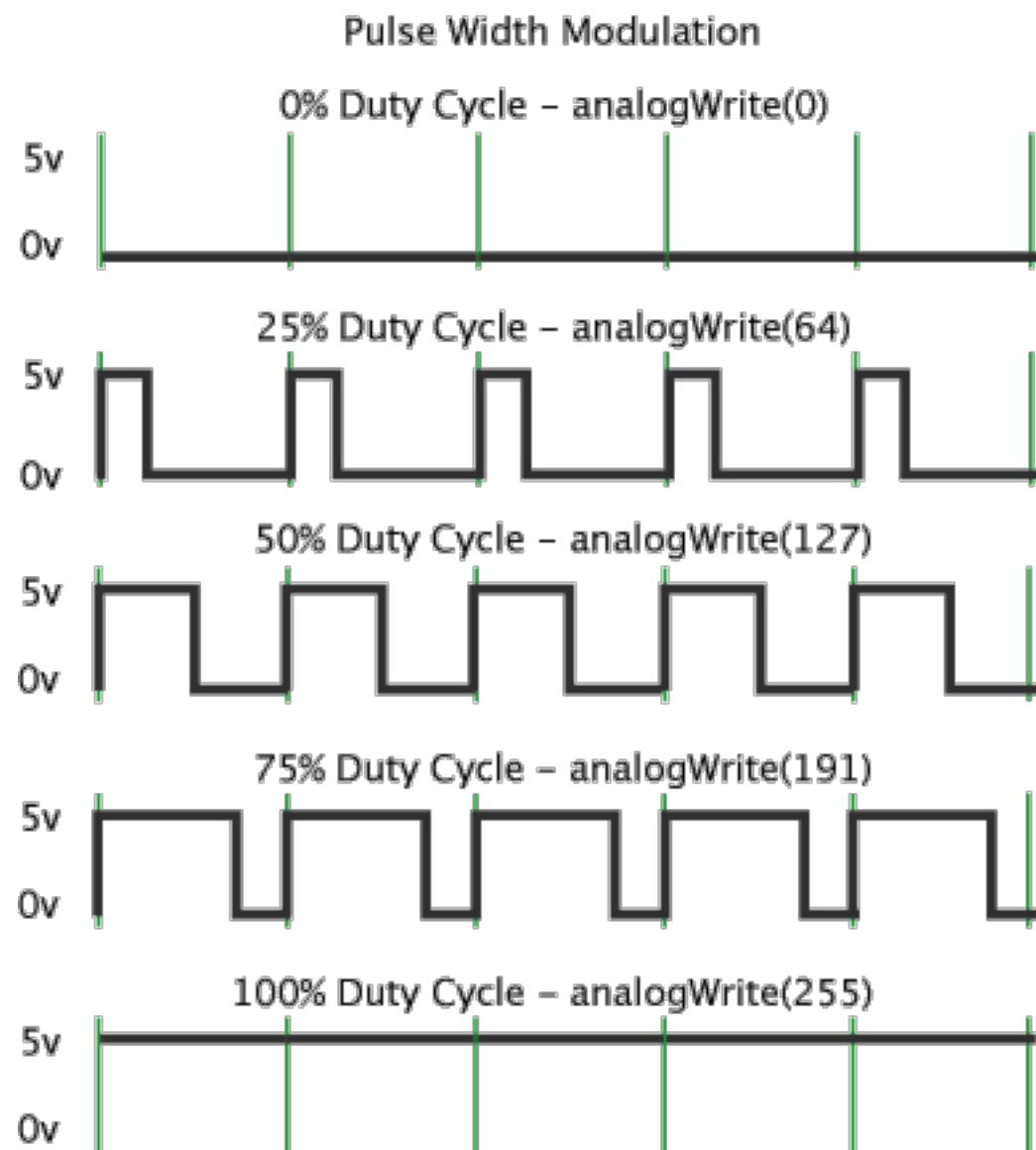
analogRead()



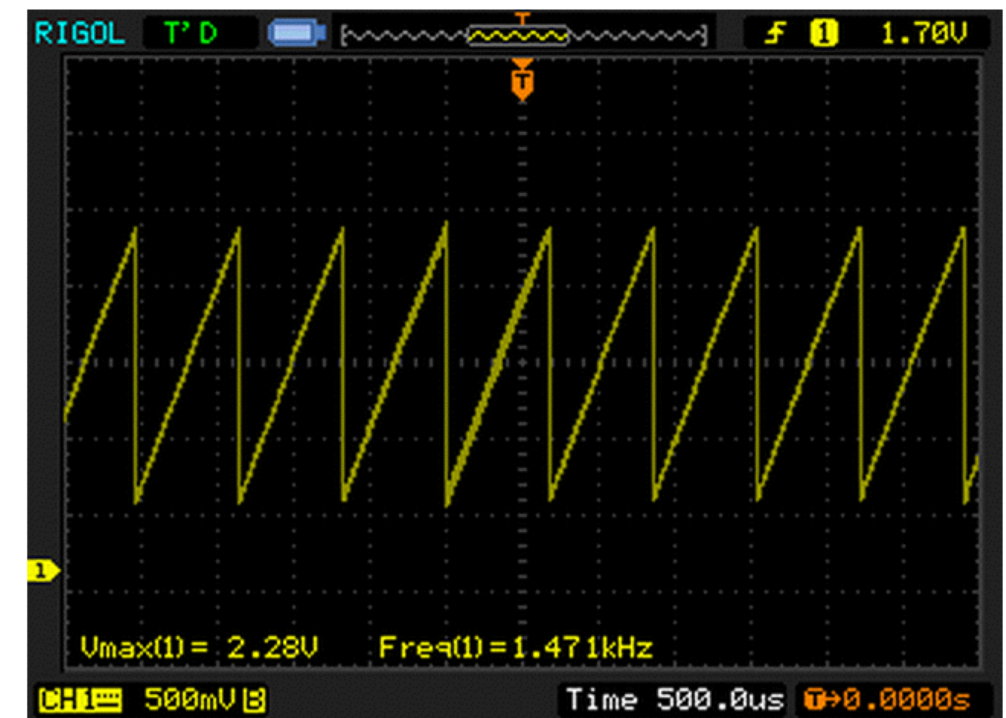
$$\frac{5 \text{ volts}}{1023 \text{ Steps}} = 4.887\text{mV per Step}$$

Calling `analogRead()` on an Analog Pin,
automatically converts to Input

analogWrite()



Pulse Width Modulation (PWM)



Actual Analog

`analogWrite()` isn't Analog (Except on the Due)
Uno Pins: 3, 5, 6, 9, 10, 11

Analog Exercise

- Connect Pin 6 to Analog 0
Run this code
- Then, disconnect from Pin 6
While Running the Code

```
int PWMpin = 6;
void setup() {
    Serial.begin(9600);
    pinMode(PWMpin, OUTPUT);
}

void loop() {
    analogWrite(PWMpin, 127);
    Serial.print("Analog 0: ");
    int analogReading = analogRead(A0);
    Serial.println(analogReading);
}
```


Analog Exercise

- Connect Pin 6 to Analog 0
Run this code
- Then, disconnect from Pin 6
While Running the Code

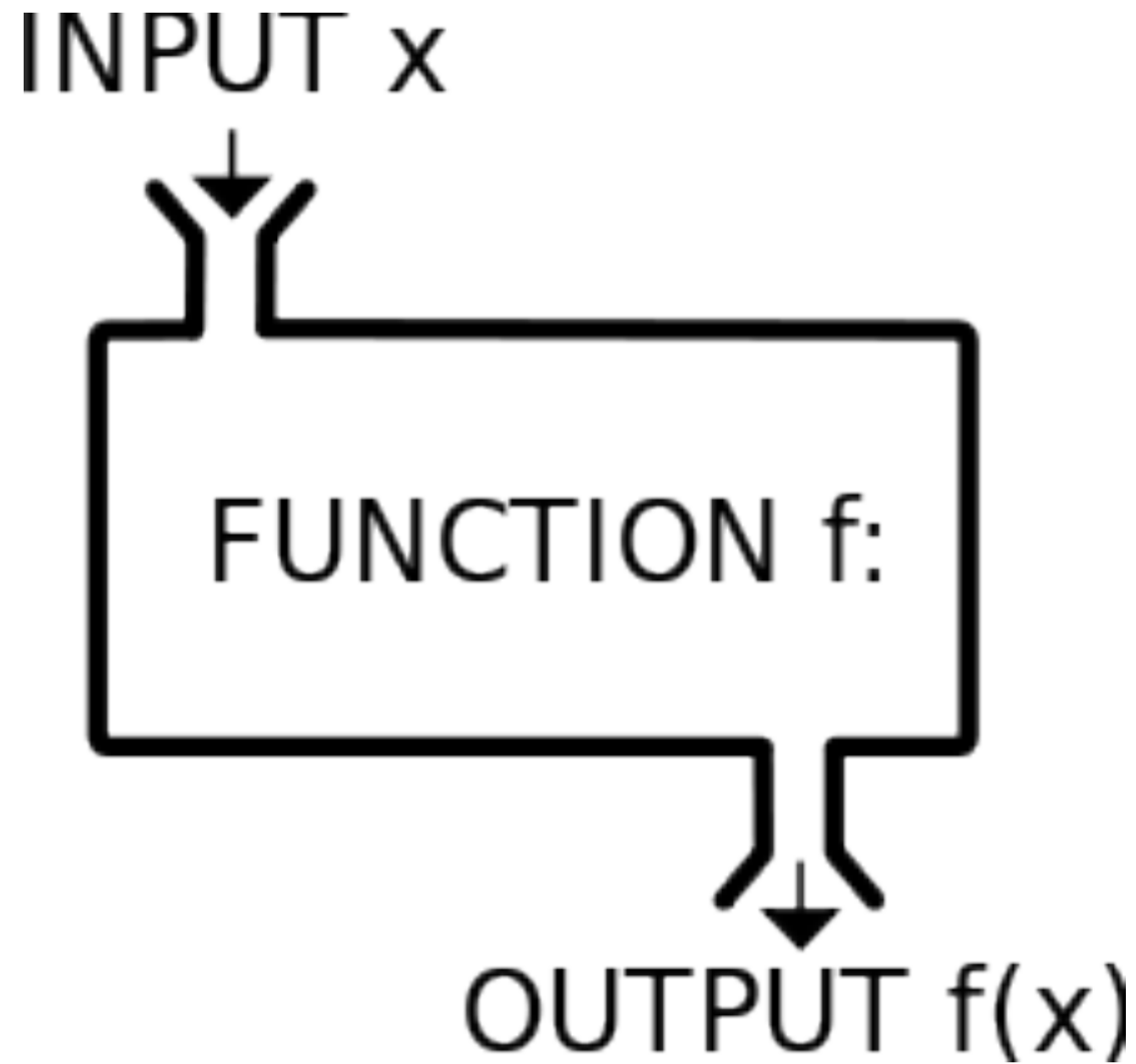
```
int PWMpin = 6;
void setup() {
    Serial.begin(9600);
    pinMode(PWMpin, OUTPUT);
}

void loop() {
    analogWrite(PWMpin, 127);
    Serial.print("Analog 0: ");
    int analogReading = analogRead(A0);
    Serial.println(analogReading);
}
```



Remember to open the Serial Monitor!

A0, A1, A2..A5 are integers!



functions()

Functions

Getting Data Back

```
int getButtonValue(int pinNumber) {  
    int buttonValue = digitalRead(pinNumber);  
    return buttonValue;  
}
```

Functions

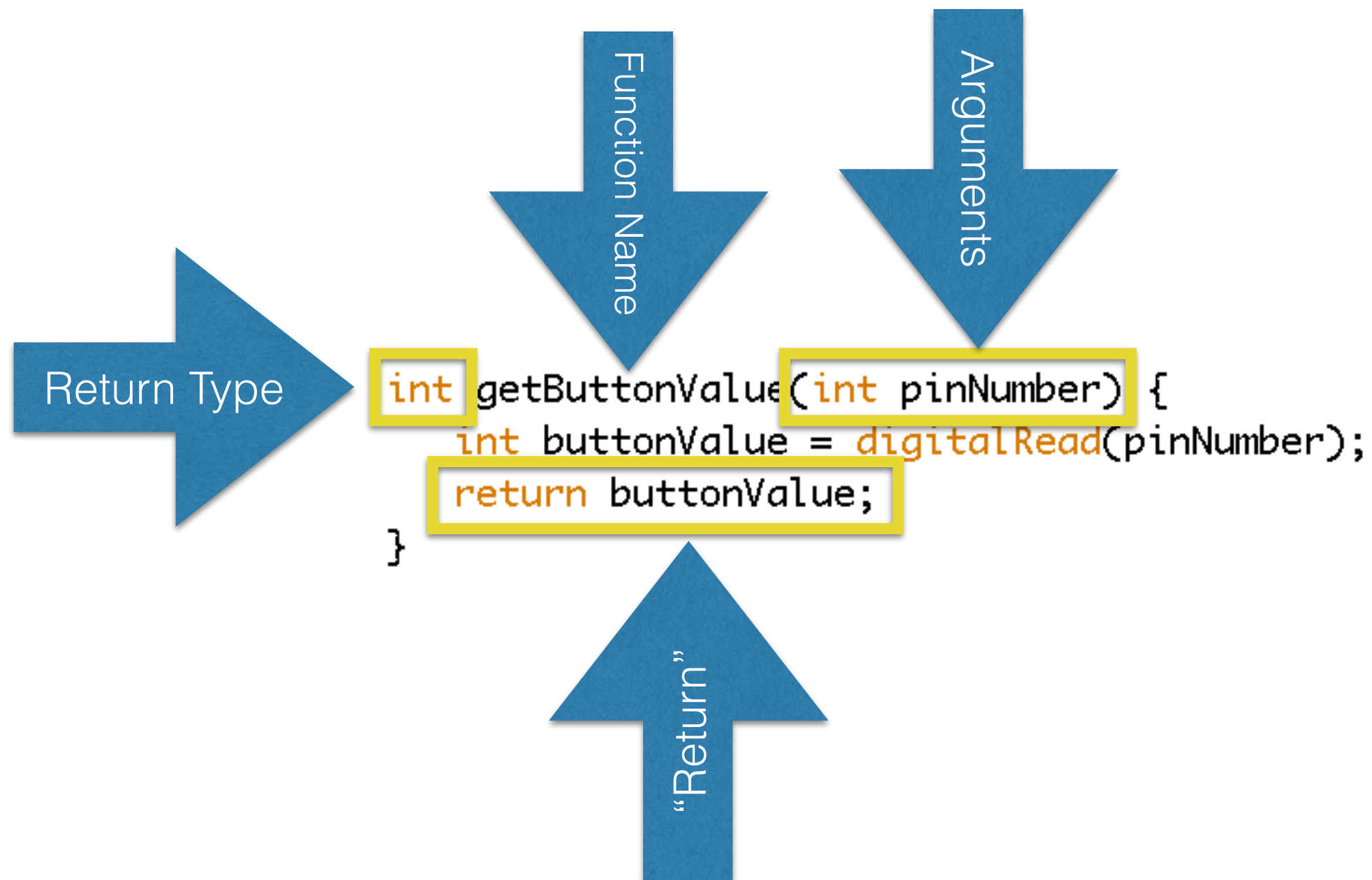
Getting Data Back

```
int getButtonValue(int pinNumber) {  
    int buttonValue = digitalRead(pinNumber);  
    return buttonValue;  
}
```

Tip: The Arduino IDE, doesn't require "prototyping"

Functions

Getting Data Back



Functions

Returning Nothing



Return Type

```
void flashLED(int pinNumber, int delayTime) {  
    digitalWrite(pinNumber, HIGH);  
    delay(delayTime);  
    digitalWrite(pinNumber, LOW);  
    delay(delayTime);  
}
```

If the function doesn't return anything, declare it as void

Function Exercise

- “Re-Write” the built-in Blink Example to use a Function
- Exercise: add a “argument” to adjust the delay time

```
// Pin 13 has an LED connected on most Arduino boards.  
// give it a name:  
int led = 13;  
  
// the setup routine runs once when you press reset:  
void setup() {  
    // initialize the digital pin as an output.  
    pinMode(led, OUTPUT);  
}  
  
// the function we created  
// by copying and pasting the code from loop()  
void flashLED() {  
    digitalWrite(led, HIGH);  
    delay(1000);  
    digitalWrite(led, LOW);  
    delay(1000);  
}  
  
// the loop routine runs over and over again forever:  
void loop() {  
    flashLED(); // repeats calls to flashLED()  
}
```



Control Structures

if-statements

```
int buttonValue = digitalRead(2);  
if (buttonValue == HIGH) {  
    digitalWrite(13, HIGH);  
} else {  
    digitalWrite(13, LOW);  
}
```

```
int buttonValue = digitalRead(2);  
if ((buttonValue == HIGH) || (buttonValue == LOW)) {  
    digitalWrite(13, HIGH);  
} else {  
    digitalWrite(13, LOW);  
}
```

if-statements

```
int buttonValue = digitalRead(2);  
if (buttonValue == HIGH) {  
    digitalWrite(13, HIGH);  
} else {  
    digitalWrite(13, LOW);  
}
```

if-statements

```
int buttonValue = digitalRead(2);  
if (buttonValue == HIGH) {  
    digitalWrite(13, HIGH);  
} else {  
    digitalWrite(13, LOW);  
}
```

```
int buttonValue = digitalRead(2);  
if ((buttonValue == HIGH) || (buttonValue == LOW)) {  
    digitalWrite(13, HIGH);  
} else {  
    digitalWrite(13, LOW);  
}
```

if-statements

```
int buttonValue = digitalRead(2);  
if (buttonValue == HIGH) {  
    digitalWrite(13, HIGH);  
} else {  
    digitalWrite(13, LOW);  
}
```

```
int buttonValue = digitalRead(2);  
if ((buttonValue == HIGH) || (buttonValue == LOW)) {  
    digitalWrite(13, HIGH);  
} else {  
    digitalWrite(13, LOW);  
}
```

control operators

<code>==</code>	Equal to
<code>></code> <code>>=</code>	Greater than (or equal)
<code><</code> <code><=</code>	Less Than (or equal)
<code>!=</code>	Not Equal to

<code> </code>	OR
<code>&&</code>	AND
<code> </code>	Bitwise OR
<code>&</code>	Bitwise AND

#1 if-statement mistake

```
int buttonValue = digitalRead(2);  
if (buttonValue = HIGH) {  
    digitalWrite(13, HIGH);  
} else {  
    digitalWrite(13, LOW);  
}
```

```
int buttonValue = digitalRead(2);  
if (buttonValue == HIGH) {  
    digitalWrite(13, HIGH);  
} else {  
    digitalWrite(13, LOW);  
}
```

#1 if-statement mistake

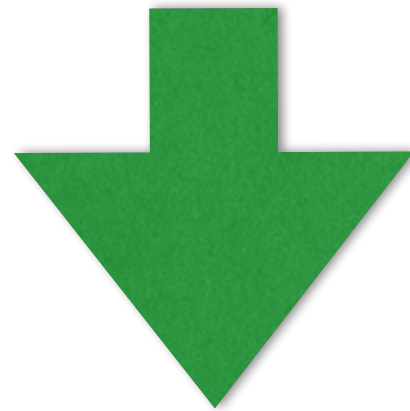
```
int buttonValue = digitalRead(2);  
if (buttonValue = HIGH) {  
    digitalWrite(13, HIGH);  
} else {  
    digitalWrite(13, LOW);  
}
```

```
int buttonValue = digitalRead(2);  
if (buttonValue == HIGH) {  
    digitalWrite(13, HIGH);  
} else {  
    digitalWrite(13, LOW);  
}
```

= != ==

for() loop example

```
int analogReadings[6];  
analogReadings[0] = analogRead(A0);  
analogReadings[1] = analogRead(A1);  
analogReadings[2] = analogRead(A2);  
analogReadings[3] = analogRead(A3);  
analogReadings[4] = analogRead(A4);  
analogReadings[5] = analogRead(A5);
```



```
for(int x; x<6; x++)  
    analogReadings[x] = analogRead(x);  
|
```

for() loop

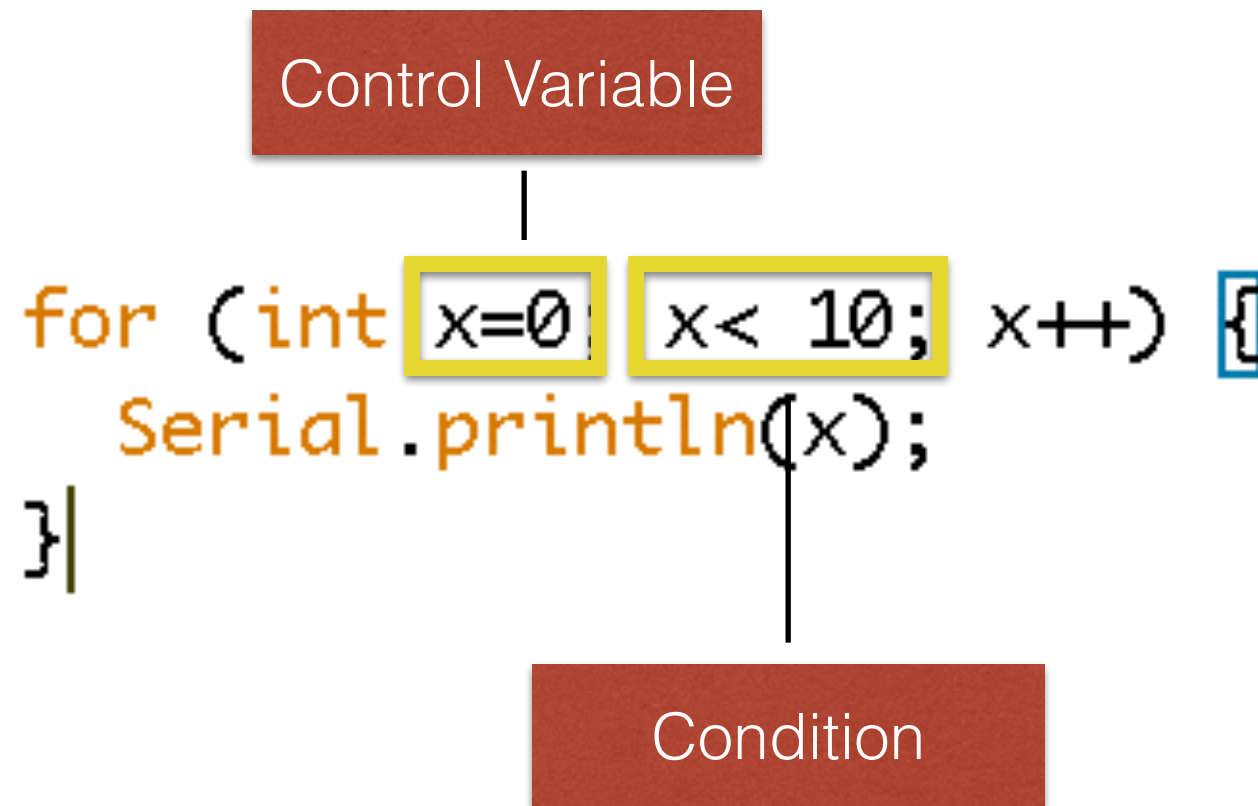
```
for (int x=0; x< 10; x++) {  
    Serial.println(x);  
}
```

for() loop

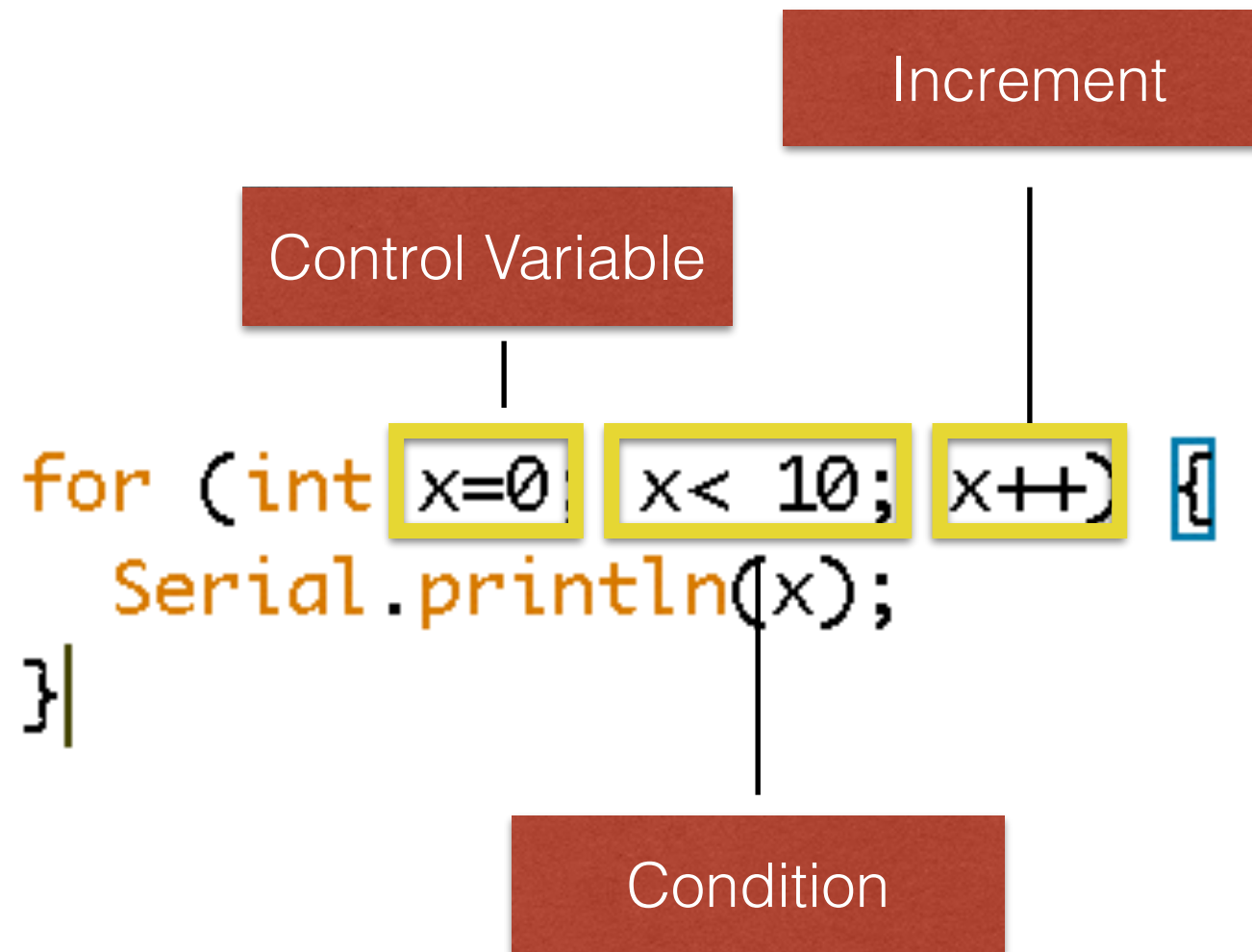
Control Variable

```
for (int x=0 x< 10; x++) {  
    Serial.println(x);  
}
```

for() loop



for() loop



Array and For Exercise

- Use an Array and two for-loops to read analog inputs, then display then
- Notice the difference in brackets between the two loops

```
int analogReadings[6];

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

void loop() {
    for(int x; x<6; x++)
        analogReadings[x] = analogRead(x);

    for(int x; x<6; x++) {
        Serial.print("A");
        Serial.print(x);
        Serial.print(": ");
        Serial.println(analogReadings[x]);
    }
    delay(2000);
    Serial.println();
}
```

while() loop

```
while (digitalRead(2) == HIGH) {  
    Serial.println(x);  
}
```

while() loop

Condition

```
while [digitalRead(2) == HIGH] {  
    Serial.println(x);  
}
```

while() loop

Condition

```
while [digitalRead(2) == HIGH] {  
    Serial.println(x);  
}
```

Loop conditions are same as “if conditions”

while() and Serial

- This program echoes whatever is on the serial buffer back out

```
void setup() {  
    Serial.begin(9600);  
}  
  
void loop() {  
    while (Serial.available() > 0) {  
        char incomingChar = Serial.read();  
        Serial.print(incomingChar);  
    }  
}
```

More information?

Visit

baldengineer.com