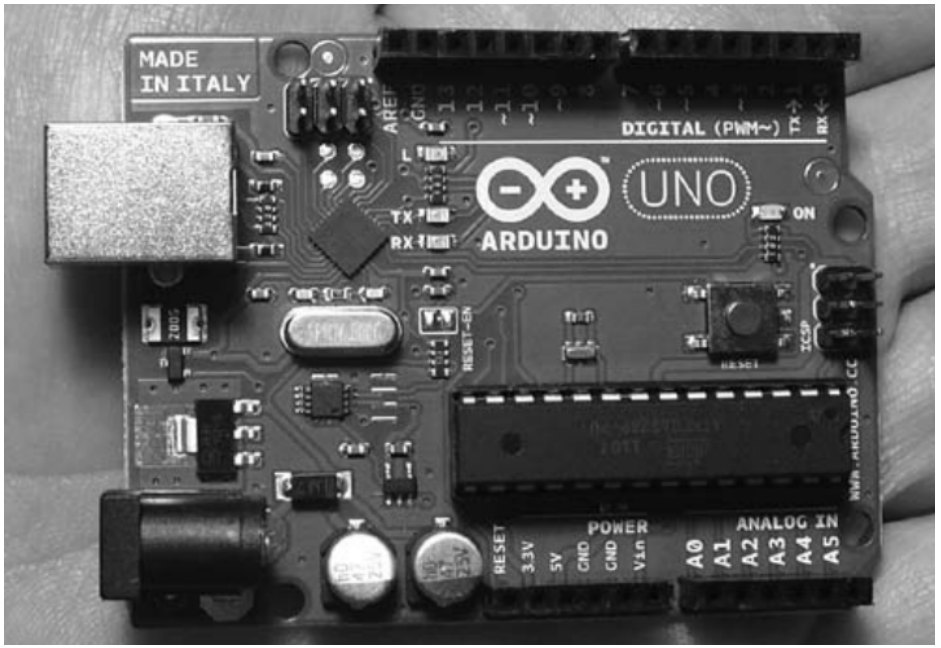
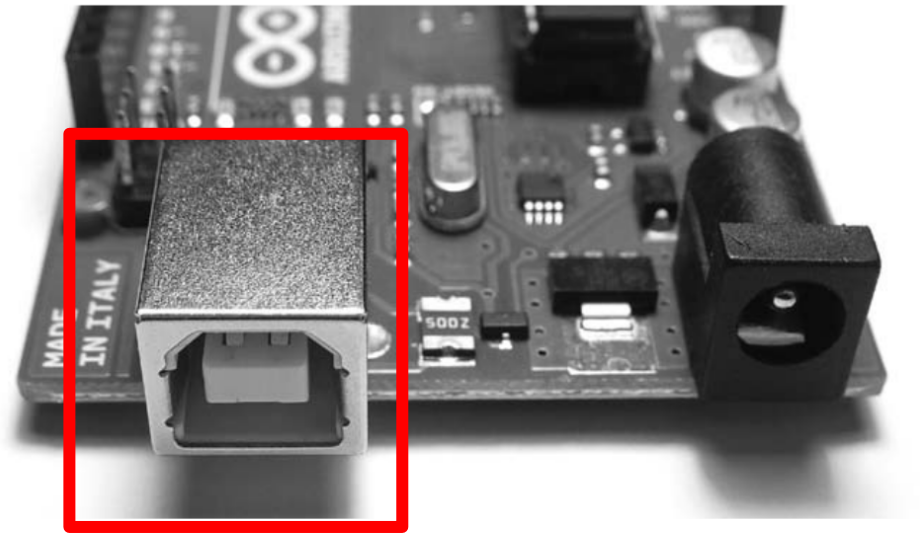
What is Arduino ?



The Arduino is a tiny computer system that can be programmed with your instructions to interact with various forms of input and output

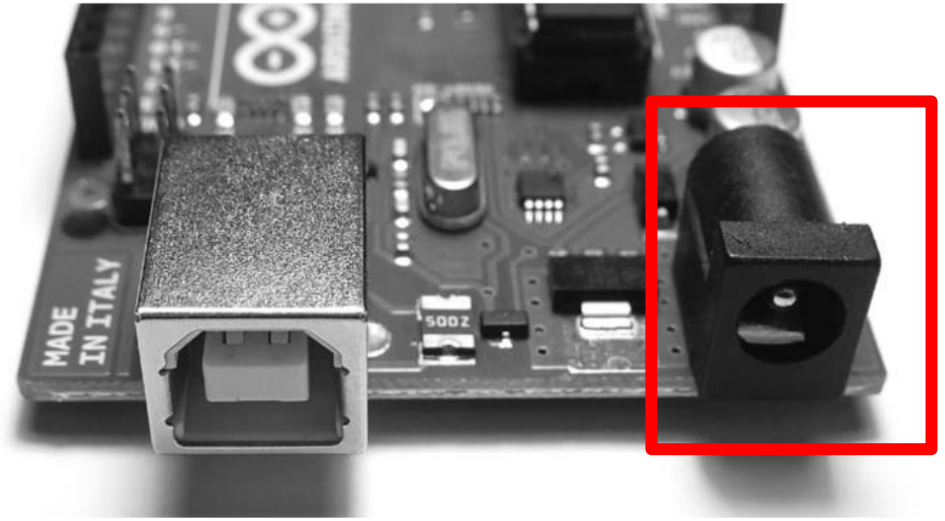
Quick Tour of Arduino



USB Connector

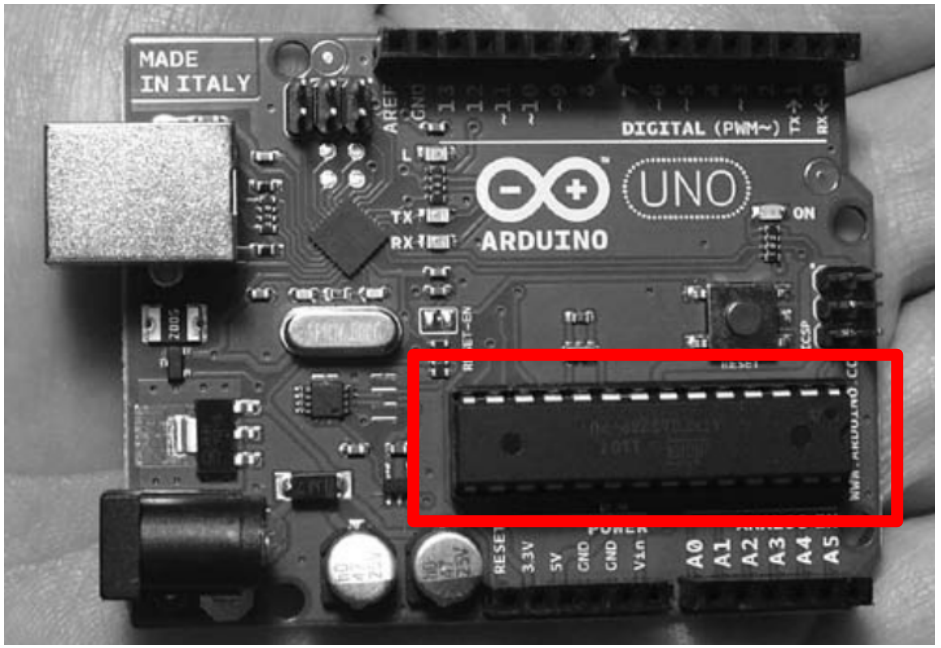
- Connects Arduino to Computer
- To Supply Power
- Upload Instructions to Arduino
- Send and Receive Data

Quick Tour of Arduino



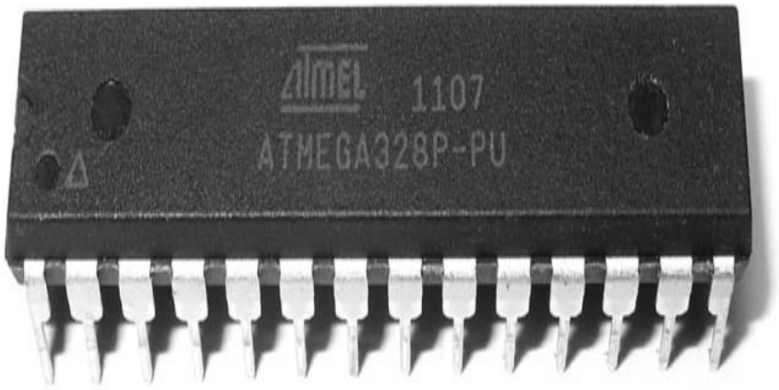
- Power the Arduino with DC 9-12V plug

Quick Tour Arduino

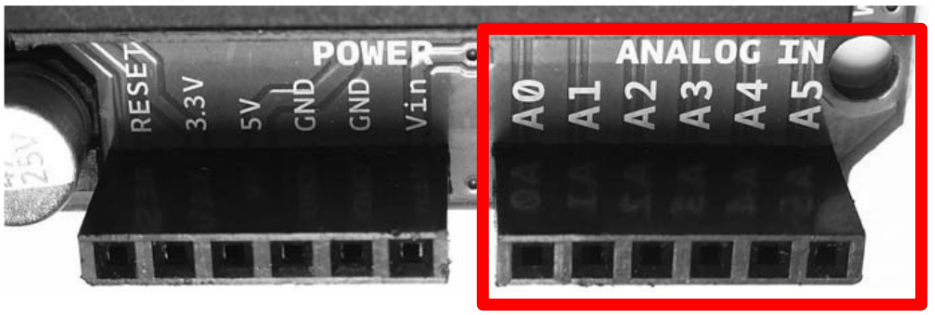


MicroController

- Brain of the Arduino
- Executes Instructions
- Contains Memory



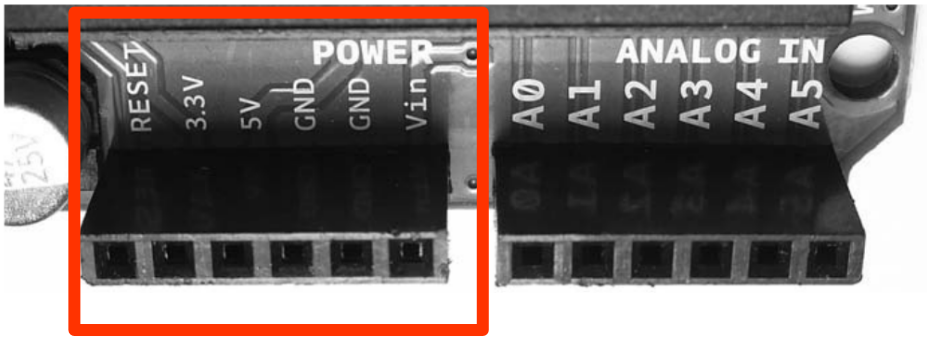
Quick Tour Arduino



Analog Inputs

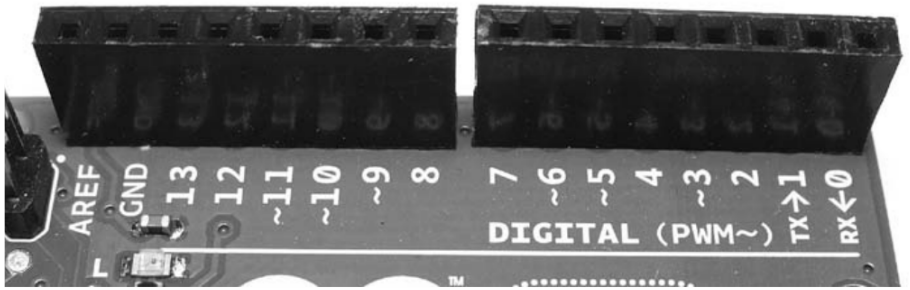
- Six Analog In Pins
- Measure Electrical Signals vary in voltage

Quick Tour of Arduino



- Power Connections
- External RESET

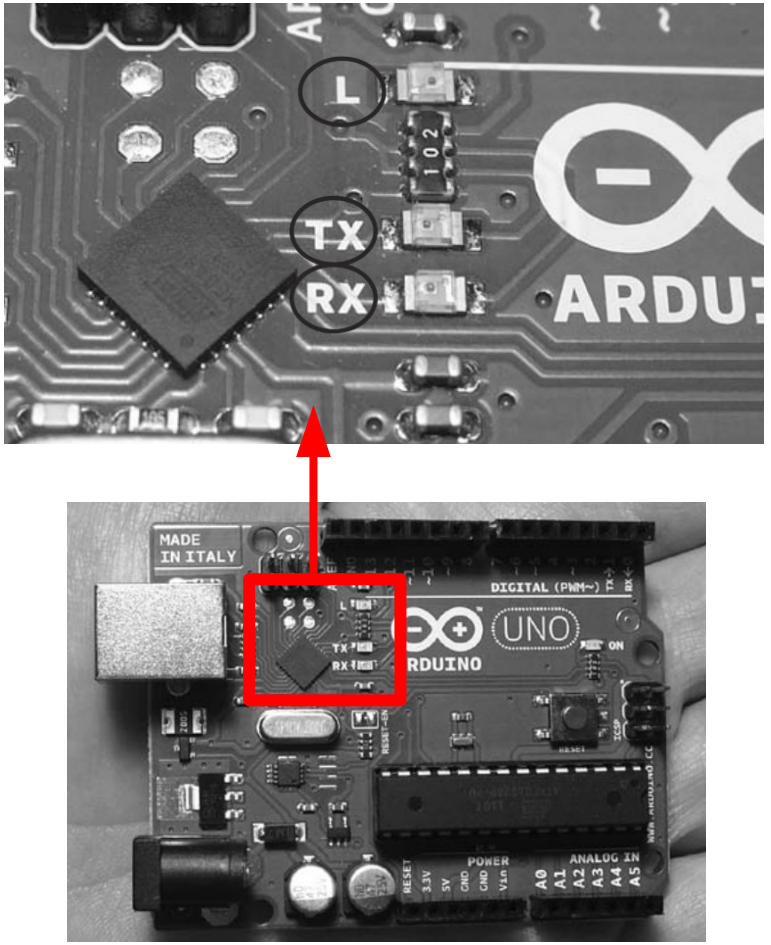
Quick Tour of Arduino



Digital Input and Output Pins

- Detect the prescence of signal or Can Generate on Command
- ‘ ~ ‘ Labelled pins can generate varying electrical signal

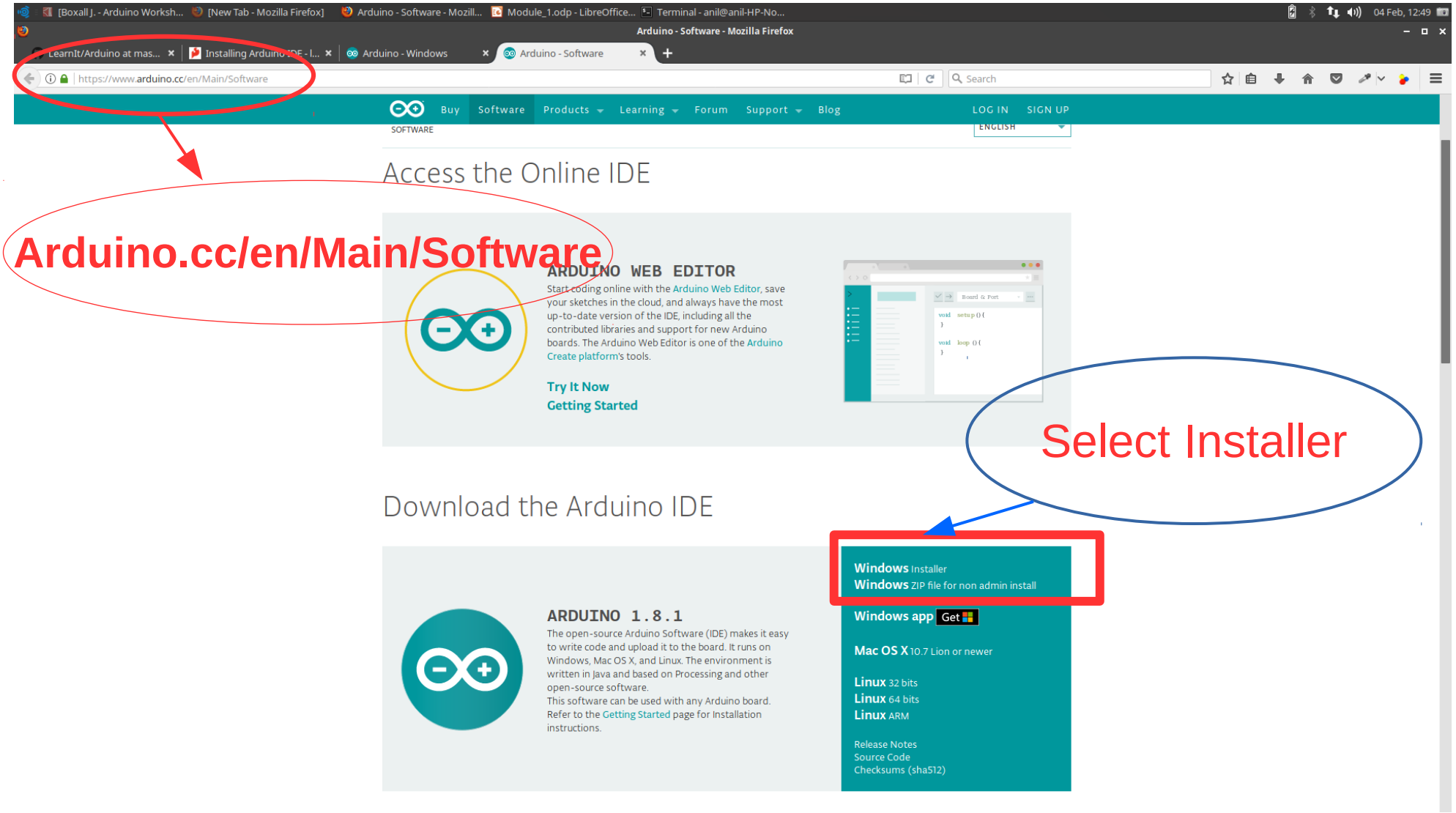
Quick Tour of Arduino



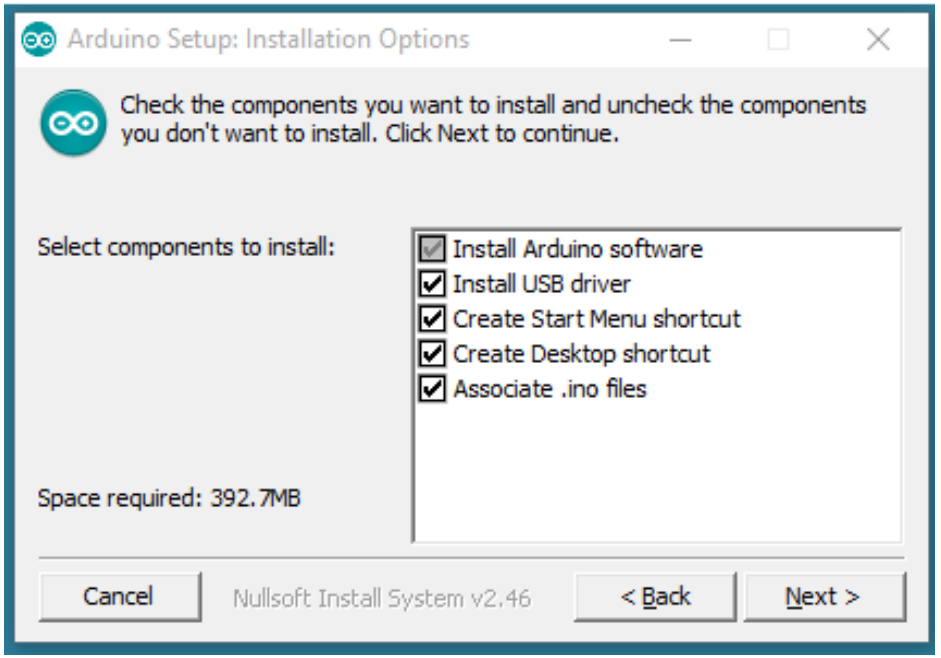
On Board Led's

- TX, RX light up when Data being Transmitted and Received
- L is connected to Digital Pin 13

Software - Download

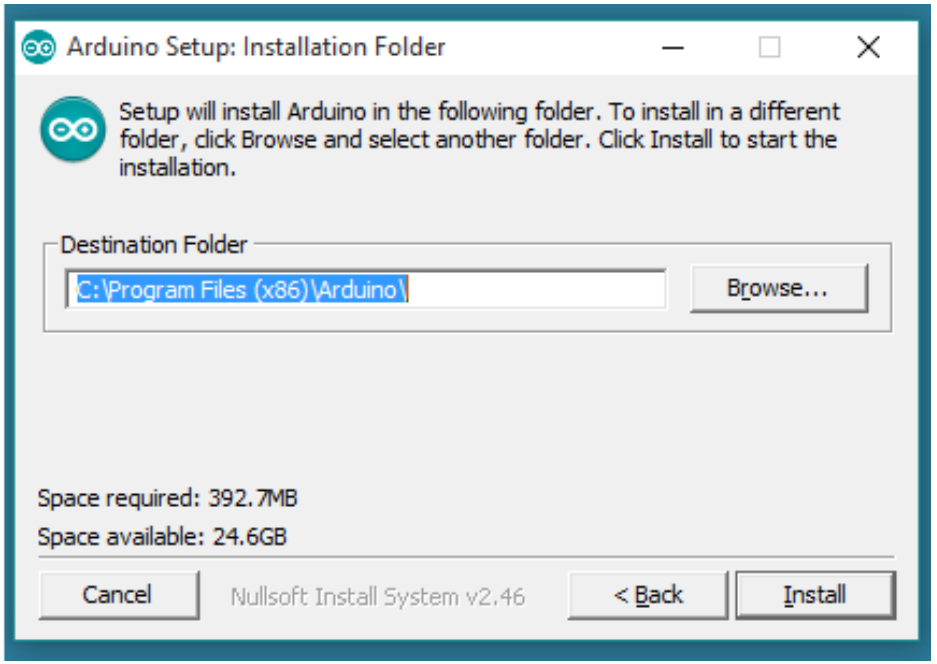


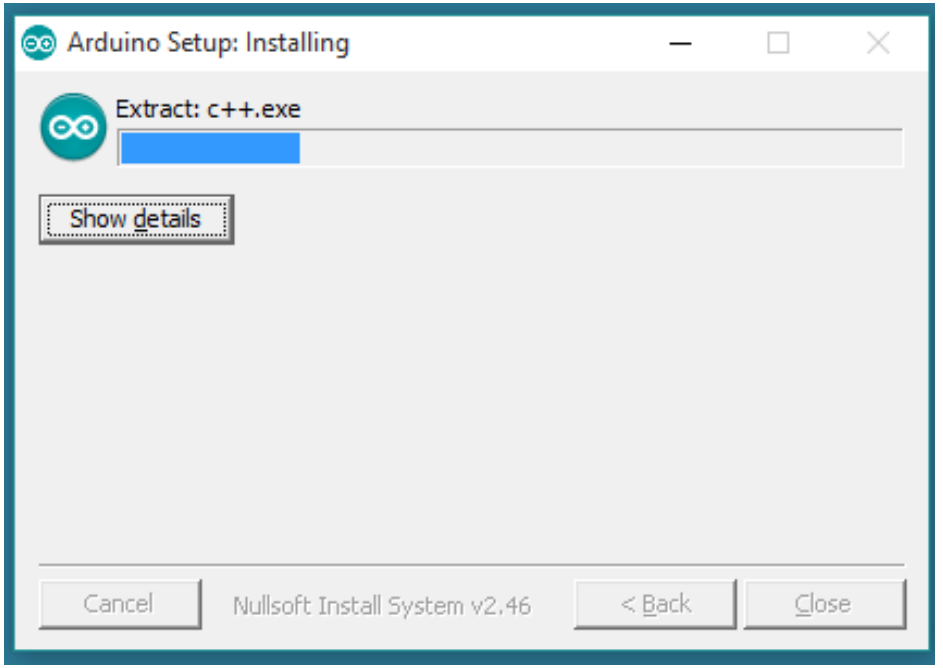
Installation



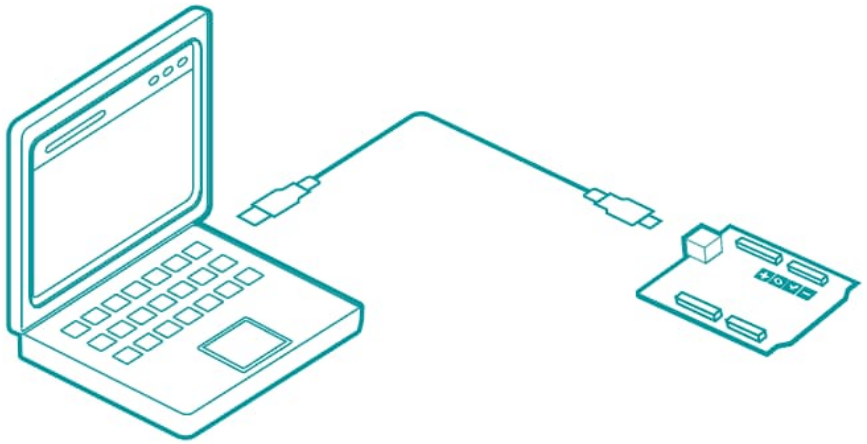
- Double click on the Downloaded Exe File
- Allow OS to install Drivers

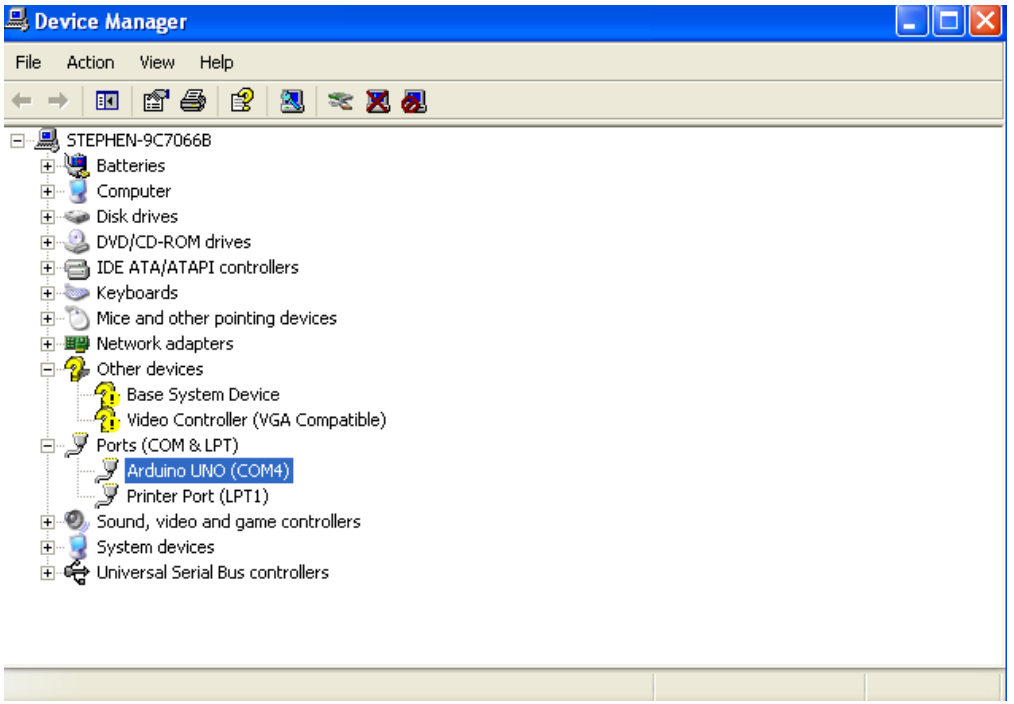
installation





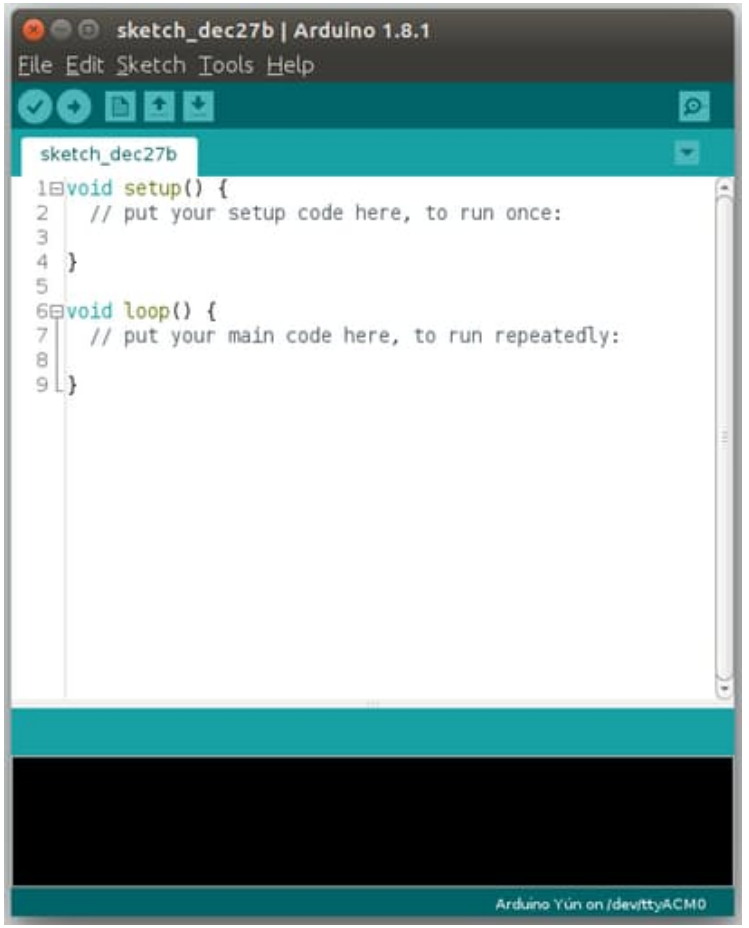
Installation - Driver





- Connect Arduino to Computer using USB Cable
- Open Device Manger
- Look under Ports
- Note Arduino UNO COM port

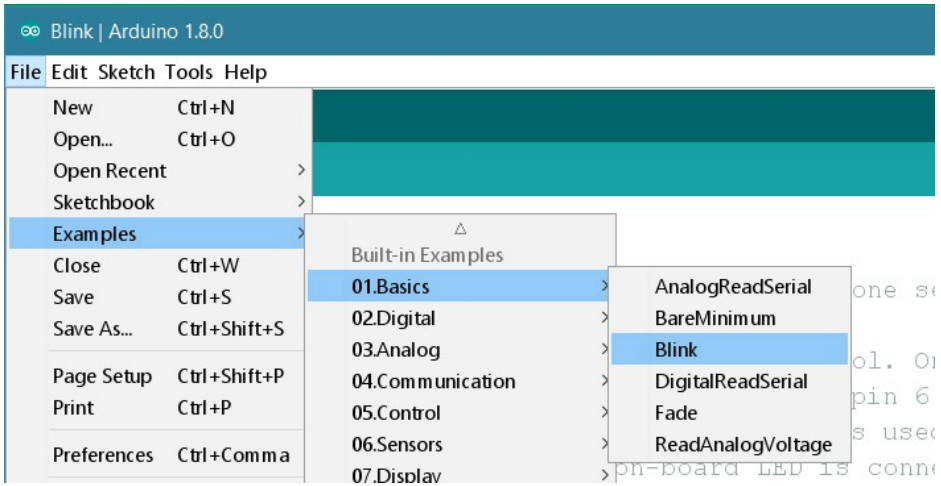
IDE – Test Run

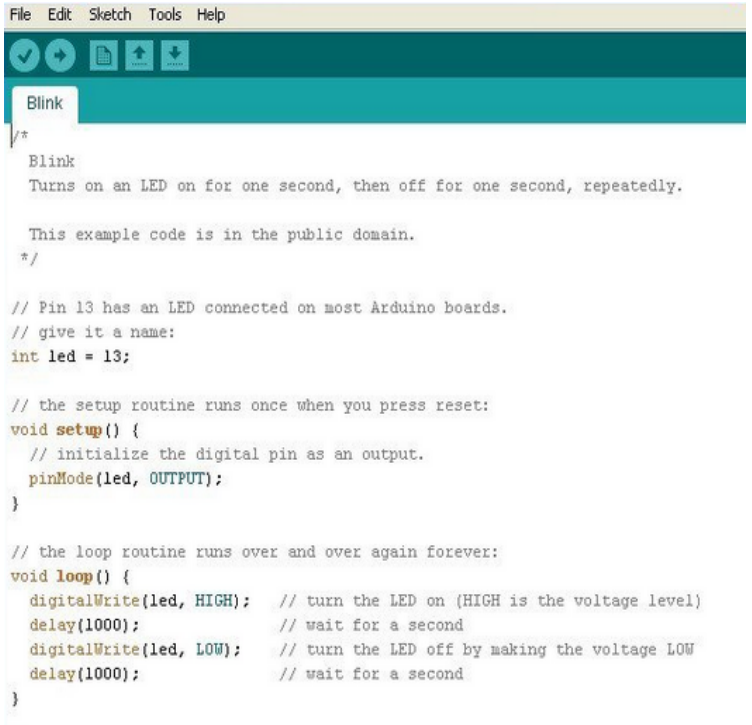


- Double Click on the Arduino Icon in Desktop
- The Tab is known as Sketch
- It Contains the program that tells your board what to do

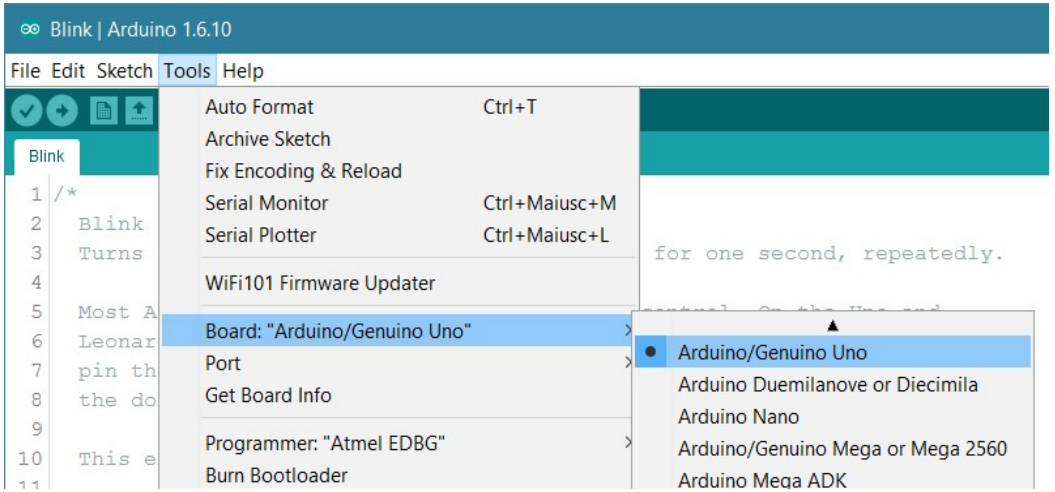
Test Run - Blink It

Go to File->Examples-> Basics-> Blink and click on it

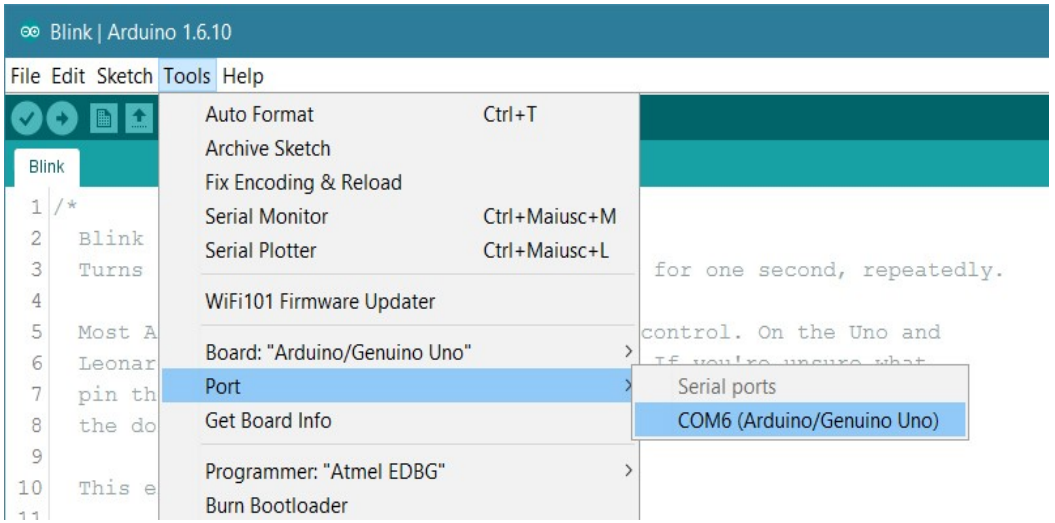




Test Run - Blink It

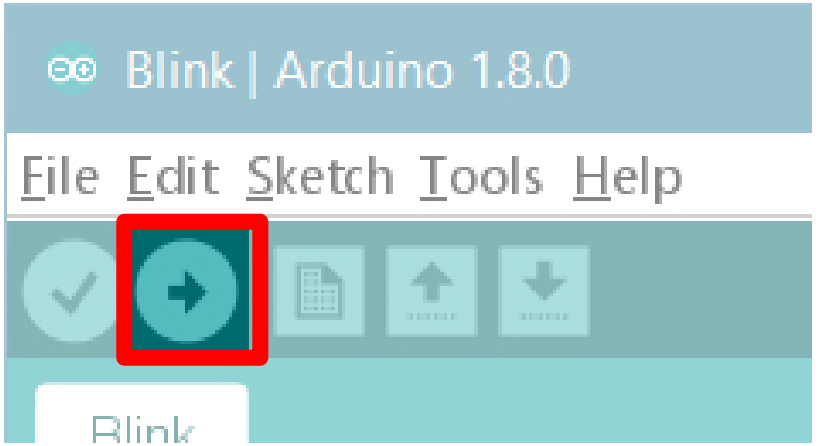


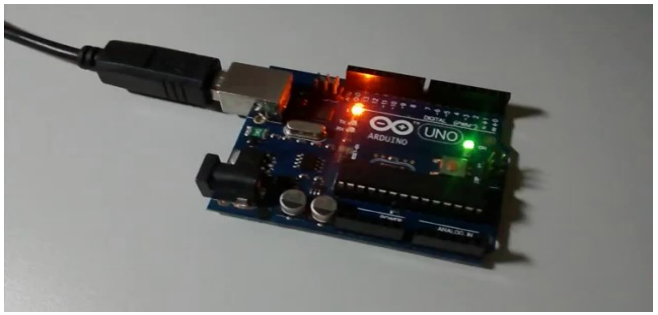
- Select Board type in Tools -> Board
- Select UNO



- Select COM port noted Previously in Tools -> Port

Test Run - Upload





- Click the upload button
- you should see the RX and TX leds on the board flashing
- After few seconds on board led blinks

Introduction to Arduino Programming

Structure

BareMinimum §

```
1 void setup() {
2
3   }
4
5 void loop() {
6
7   }
8
```

- Every Program must contain these two functions
- Setup is preparation

  it is the first function to run and run only once
- Loop is execution

  it is executed continuesly

Fuctions

- Function is block of code that has a name and a block of statements when the function is called
- Setup() and loop() are built-in functions

  - we will talk about custom functions later

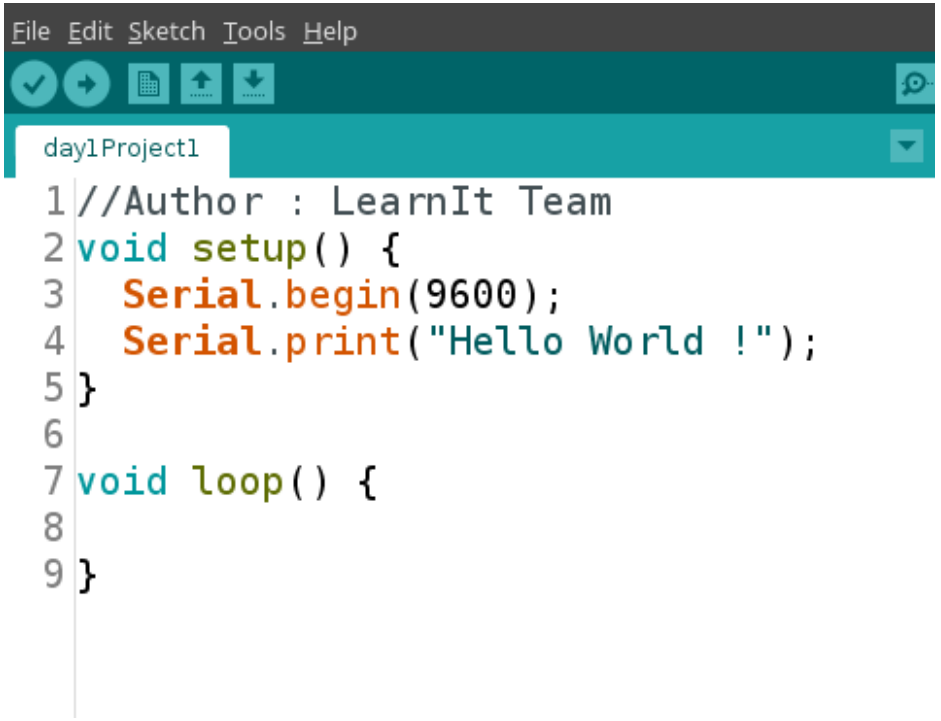
Basic Syntax

- Each Statement Ends with semicolon “;”

  int a = 13;

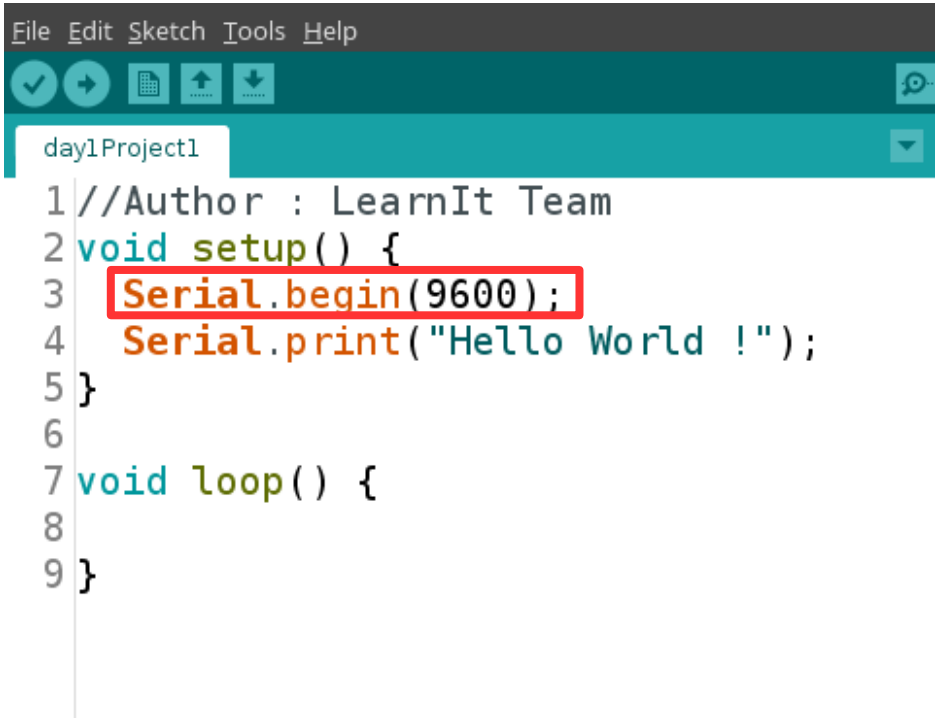
- Curly braces always come in pairs; they are used to define the start and end of functions, loop, conditional statements
- // is used for single line comment
- /* */ used to wrap multiline comment

Hello World



- Usb is used to send and Receive Data Serially
- Serial Port / Com port talks with certain speed, called Baudrate

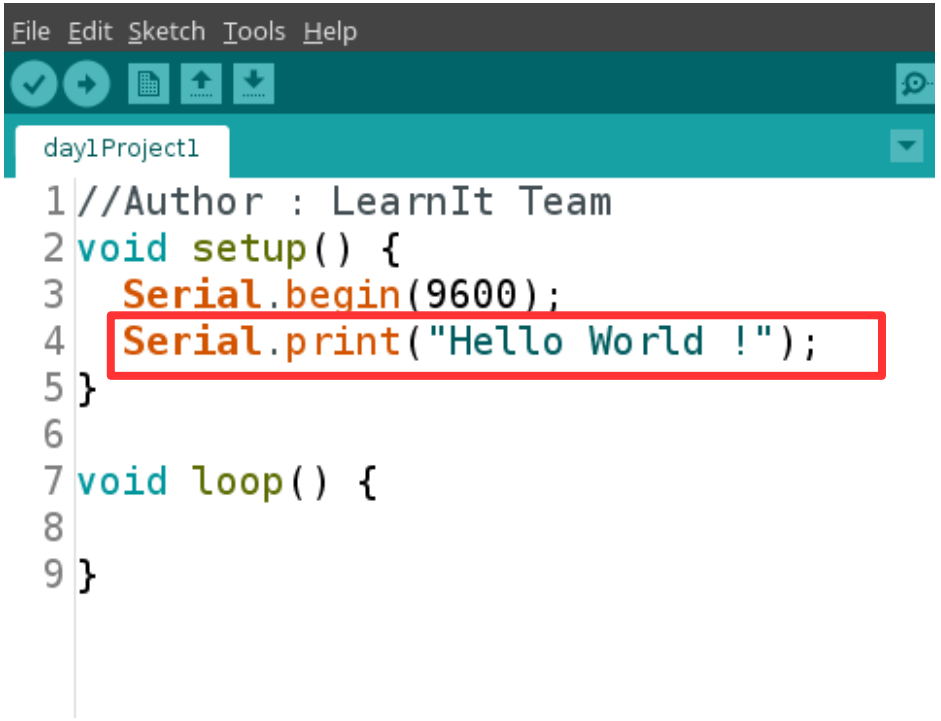
Hello World



Serial.begin(9600) basically tells the Arduio to start talking over serial port at a speed of 9600 baud

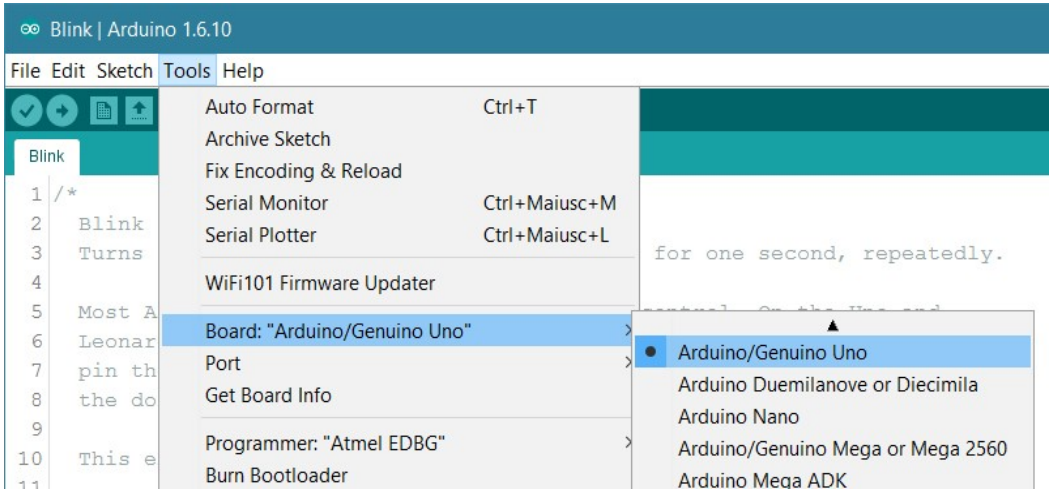
Baudrate – Bits per second

Hello World

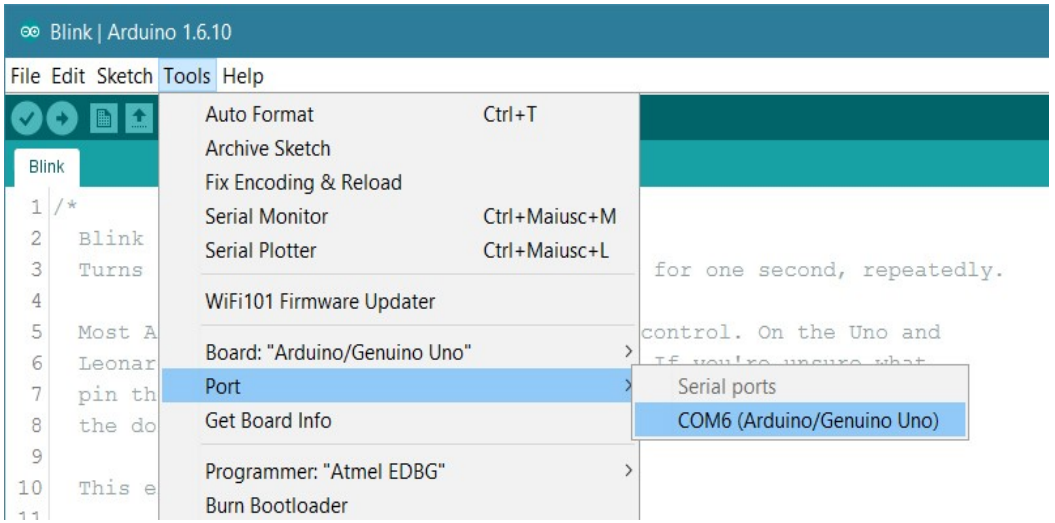


- Serial.print tells the Arduino to send a text line to Serial port
- Notice the Semicolon (;) - every statement must end with semicolon

Select Port and Board

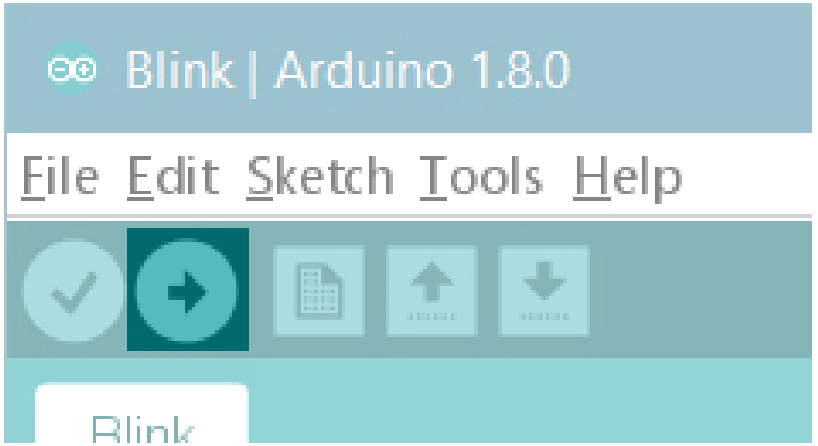


- Select Board type in Tools -> Board
- Select UNO



- Select COM port noted Previously in Tools -> Port

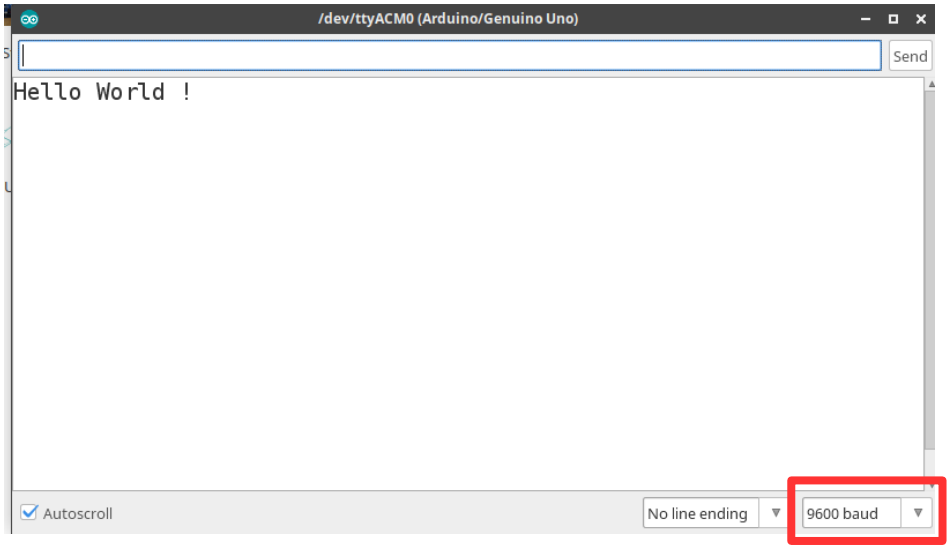
Upload & Open Serial Monitor

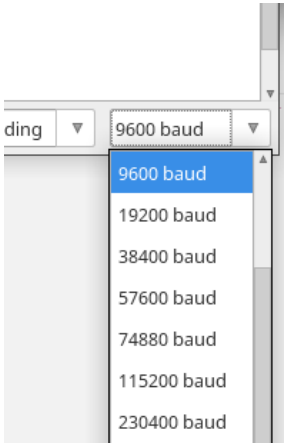


- Upload the Sketch
- Observe the tx & rx blinking on Board

- Click on the magnifying glass to open Serial Monitor

Try with Differnt Baudrates by changing in Serial.begin and Serial Monitor

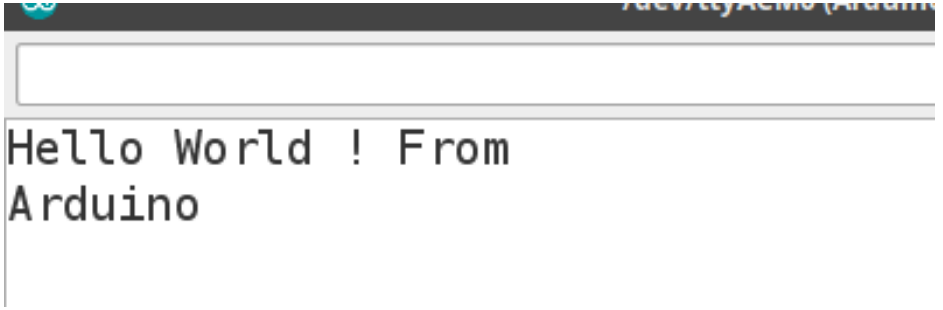




Print vs println

day1Project2

```
1  //Author : LearnIt Team
2  void setup() {
3    Serial.begin(9600);
4    Serial.print("Hello World !");
5    Serial.println(" From");
6    Serial.print("Arduino");
7  }
8
9  void loop() {
10
11 }
```

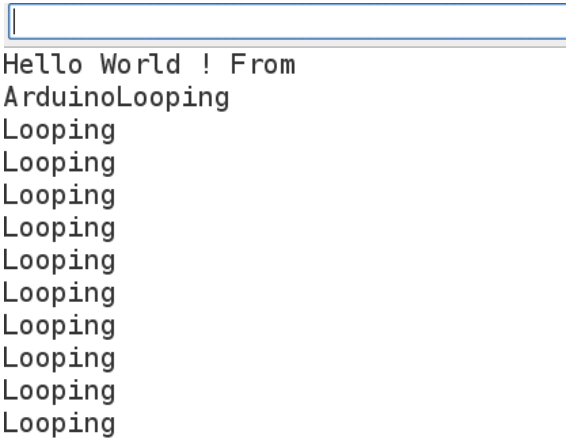


“Arduino “ Starts in new line But “From” is on same ine

Setup vs loop

day1Project3

```
1 //Author : LearnIt Team
2 void setup() {
3   Serial.begin(9600);
4   Serial.print("Hello World !");
5   Serial.println(" From");
6   Serial.print("Arduino");
7 }
8
9 void loop() {
10  Serial.println("Looping");
11 }
```



Statement in loop is repeting

Delay

day1Project4

```
1 //Author : LearnIt Team
2 void setup() {
3   Serial.begin(9600);
4   Serial.print("Hello World !");
5   Serial.println(" From");
6   Serial.print("Arduino");
7 }
8
9 void loop() {
10  Serial.println("Looping");
11  delay(1000);
12 }
```

- Delay Pauses the program for the amount of time (in miliseconds) specified as parameter
- it brings all other activities to a halt

Variables

- A variable is a place to store a piece of data
- It has a name, a value, and a type

  int a = 13;
- creates a variable whose name is a, whose value is 13, and whose type is int

day1Project5

```
1 //Author : LearnIt Team
2 int a = 13;
3 void setup() {
4   Serial.begin(9600);
5   Serial.println("value of a is");
6   Serial.print(a);
7 }
8
9 void loop() {
10 }
```

Variable types & Rules

Type	Description
int	Integer max size is 2^8
float	Floating point value
char	Character
String	String

- consist of any letters (a to z and A to Z)
- contain the numbers 0 to 9, but may not start with a number, e.g. 3var is not allowed, but var3 is allowed
- Variables may not have the same names as Arduino language keywords, e.g. you can not have a variable named int
- Variables must have unique names i.e. you can not have two variables with the same name
- Variable names are case sensitive, so Count and count are two different variables
- Variables may not contain any special characters, except the underscore (_), e.g. top_score

Variable Scope

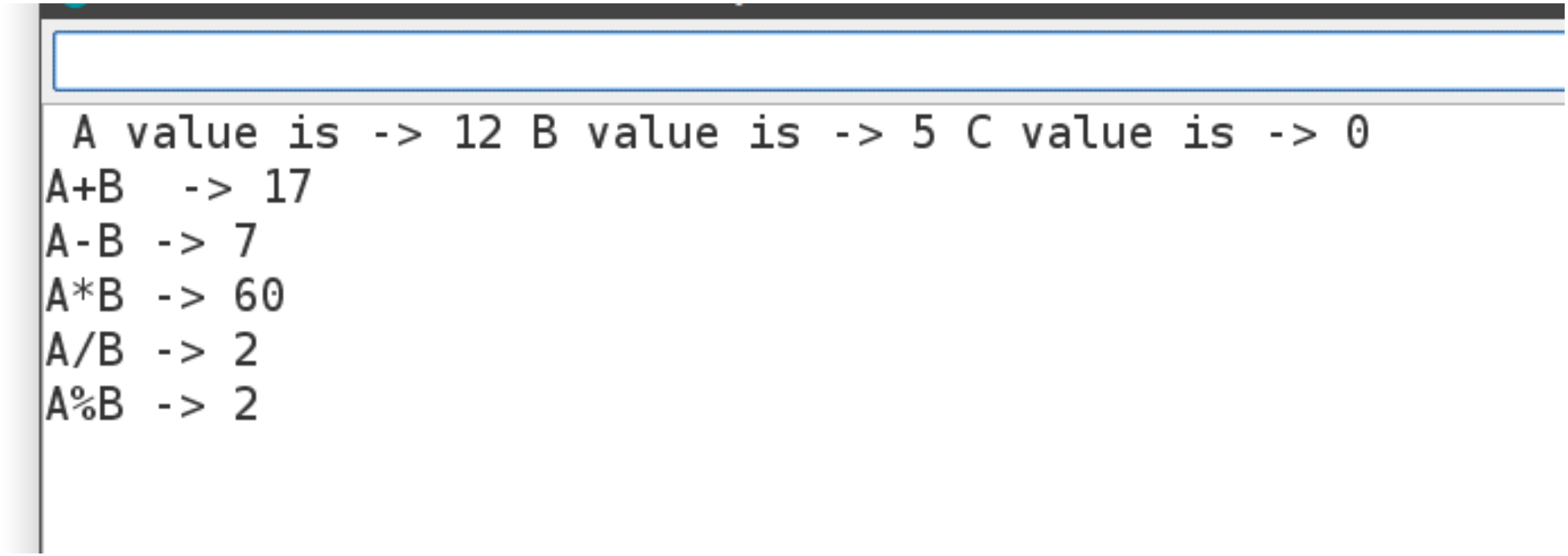
- scope is the lifetime and visibility a variable has in a program
  - Global : A variable with global scope is visible from its point of definition to the end of the file in which it is defined
  - Local : A variable with local scope extends from its point of definition to the end of the function block in which it is define

```
1  int gval;  // any function will see this variable
2
3  void setup()
4  {
5    // ...
6  }
7
8  void loop()
9  {
10   int i;    // "i" is only "visible" inside of "loop"
11   float f;  // "f" is only "visible" inside of "loop"
12   // ...
13 }
```

Arithmetic operators

Symbol	Name	Example	Explanation
=	assignment	Led = 13	Store the value 13 at variable led
+	addition	A = B + C	Add the variable B,C and assign the result to C
-	subtraction	A = B - C	Subtract C from B and assign the result to A
*	multiplication	A = B*C	Multiply B and C and assign the result to A
/	division	A = B/C	Divide B with C and assign the result to A
%	modulo	A = B%C	Calculate the Remainder of division B/C and assign it to the A

```
// Author : LearnIt team
int A = 12;
int B = 5;

void setup() {
  int C;
  Serial.begin(9600);
  Serial.print(" A value is -> ");
  Serial.print(A);
  Serial.print(" B value is -> ");
  Serial.print(B);
  Serial.print(" C value is -> ");
  Serial.println(C);

  int sum = A + B;
  Serial.print("A+B  -> ");
  Serial.println(sum);

  int diff = A - B;
  Serial.print("A-B -> ");
  Serial.println(diff);

  int mul = A * B;
  Serial.print("A*B -> ");
  Serial.println(mul);

  int divi = A / B;
  Serial.print("A/B -> ");
  Serial.println(divi);

  int modulo = A % B;
  Serial.print("A%B -> ");
  Serial.println(modulo);
}

void loop() {

}
```


A value is -> 12 B value is -> 5 C value is -> 0
A+B -> 17
A-B -> 7
A*B -> 60
A/B -> 2
A%B -> 2

Comparison Operators

Result of statement with comparision operator is either TRUE or FALSE

Symbol	Name	Example	Explanation
==	Is equal to	X == Y	X is equal to y
!=	Not equal to	X != Y	X is not equal to Y
<	Less than	X < Y	X less than Y
>	Greater than	X > Y	X greater than Y
<=	Lessthan or equal to	X <= Y	X less than or equal to Y
>=	Greater than or equal to	X >= Y	X greater than or equal to Y

0 < x < 1 is invalid

day1Project7

```
// Author : LearnIt team
int A=12,B=13,C=12;
void setup() {
  Serial.begin(9600);
  Serial.print("A==A    is ");
  Serial.println(A==A);
  Serial.print("A==B    is ");
  Serial.println(A==B);

  Serial.print("A!=A    is ");
  Serial.println(A!=A);
  Serial.print("A!=B    is ");
  Serial.println(A!=B);

  Serial.print("A<B     is ");
  Serial.println(A<B);
  Serial.print("C>B     is ");
  Serial.println(C>B);

  Serial.print("A<=B    is ");
  Serial.println(A<=B);

  Serial.print("B<=B    is ");
  Serial.println(B<=B);

  Serial.print("A=B     is ");
  Serial.println(A=B);
}
void loop() {
}


```

```
A==A     is 1
A==B     is 0
A!=A     is 0
A!=B     is 1
A<B      is 1
C>B      is 0
A<=B     is 1
B<=B     is 1
A=B      is 13
```

Boolean Operators

Symbol	Name	Example	Explanation
&&	Logical AND	X>Y && Y<Z	True only if both conditions are true
||	Logical OR	X>Y || Y< Z	True if any one of condition is true
!	NOT	!X	Invert the boolean value X

Lets start Tinkering