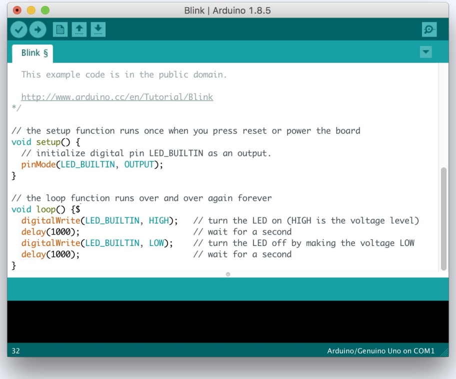
Basics of Arduino Programming



```
  This example code is in the public domain.

  http://www.arduino.cc/en/Tutorial/Blink
*/

// 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
}
```

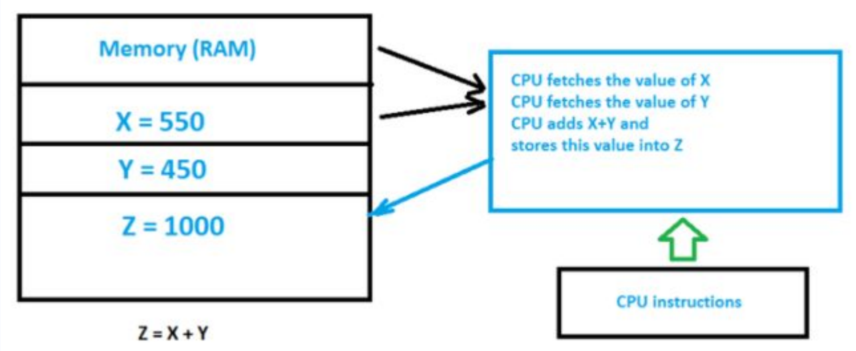
Quick Re-cap!!

- What are microcontrollers?
  - A device used to control an electronic system
  - Brain of any electronic gadget including our satellite
  - We will use Arduino nano board which has the ATmega328P microcontroller
  - We will program it on Arduino IDE

- What is Programming
  - The process of preparing instructions for a device is called programming

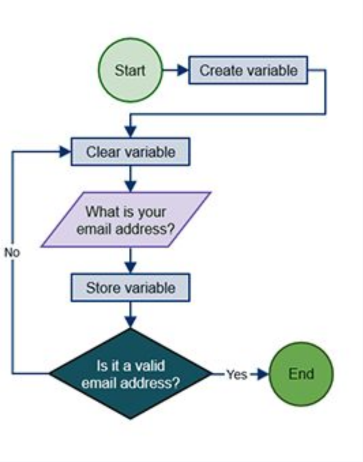
Programming

- What is Programming
  - The process of preparing instructions for a device is called programming. This enables us to perform certain tasks that will return the desired output for valid inputs.



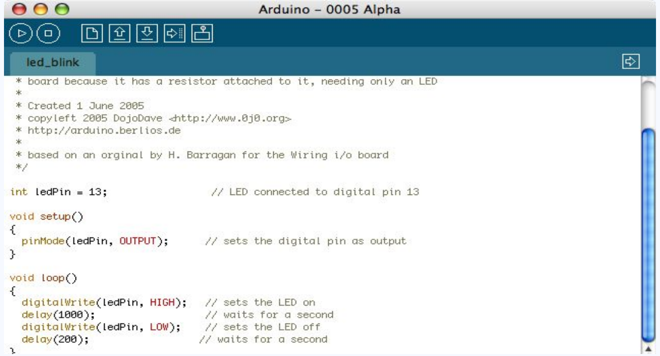
Algorithm/Flowchart

- Set of steps to be followed to accomplish the desired outcome



Source Code

- The actual code that is used to construct the program using the chosen programming language.



```
 * board because it has a resistor attached to it, needing only an LED
 *
 * Created 1 June 2005
 * copyleft 2005 DojoDave <http://www.0j0.org>
 * http://arduino.berlios.de
 *
 * based on an orginal by H. Barragan for the Wiring i/o board
 */

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

void setup()
{
  pinMode(ledPin, OUTPUT);        // sets the digital pin as output
}

void loop()
{
  digitalWrite(ledPin, HIGH);   // sets the LED on
  delay(1000);                  // waits for a second
  digitalWrite(ledPin, LOW);    // sets the LED off
  delay(200);                  // waits for a second
}
```

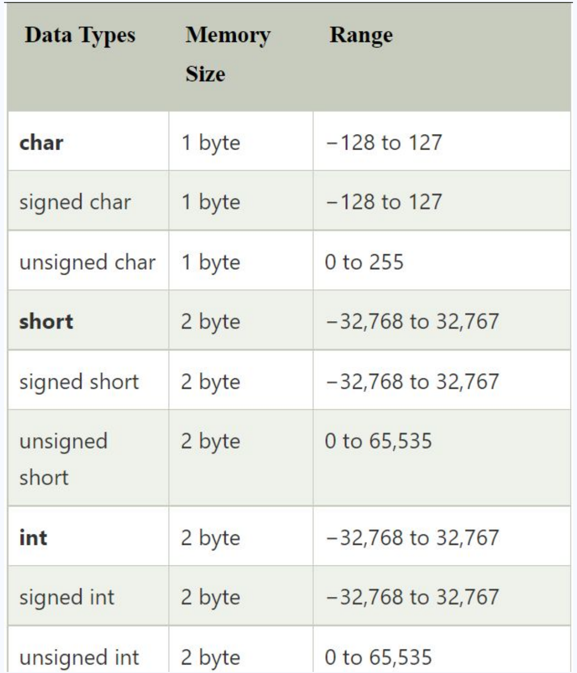
Data Types

- What are Data Types?
  - A data type specifies the type of data that a variable can store such as integer, floating, character, etc.
  - A variable is a name given to a storage idea that our programs can manipulate ex. **int v = 2**

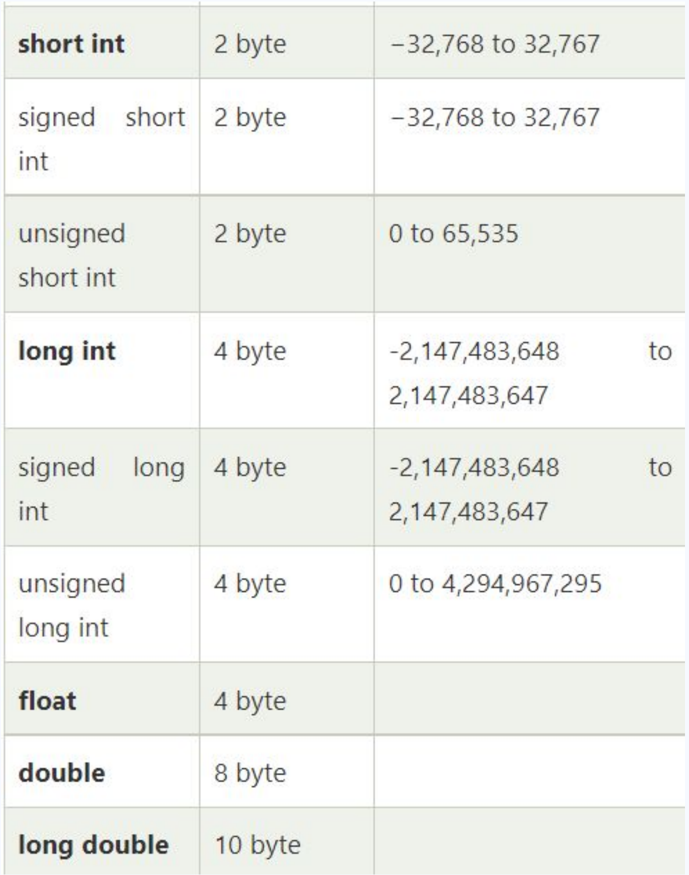
- Basic Data Types:
  - integer - **int**
    - integers
  - character - **char**
    - alphabets, symbols etc.
  - float - **float**
    - also includes fractions/decimals
  - double - **double**
    - same as float with a longer range

Primitive Data Types

Data Types



Data Types	Memory Size	Range
char	1 byte	−128 to 127
signed char	1 byte	−128 to 127
unsigned char	1 byte	0 to 255
short	2 byte	−32,768 to 32,767
signed short	2 byte	−32,768 to 32,767
unsigned short	2 byte	0 to 65,535
int	2 byte	−32,768 to 32,767
signed int	2 byte	−32,768 to 32,767
unsigned int	2 byte	0 to 65,535

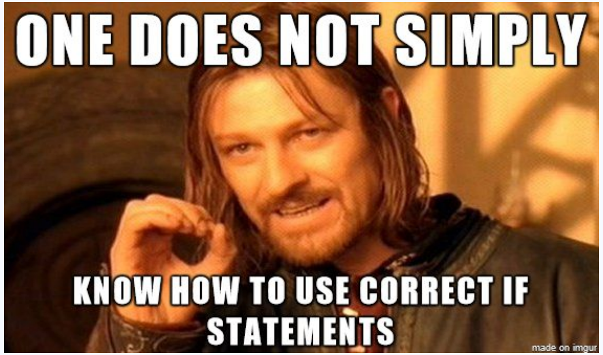


short int	2 byte	−32,768 to 32,767
signed short int	2 byte	−32,768 to 32,767
unsigned short int	2 byte	0 to 65,535
long int	4 byte	-2,147,483,648 to 2,147,483,647
signed long int	4 byte	-2,147,483,648 to 2,147,483,647
unsigned long int	4 byte	0 to 4,294,967,295
float	4 byte	
double	8 byte	
long double	10 byte	

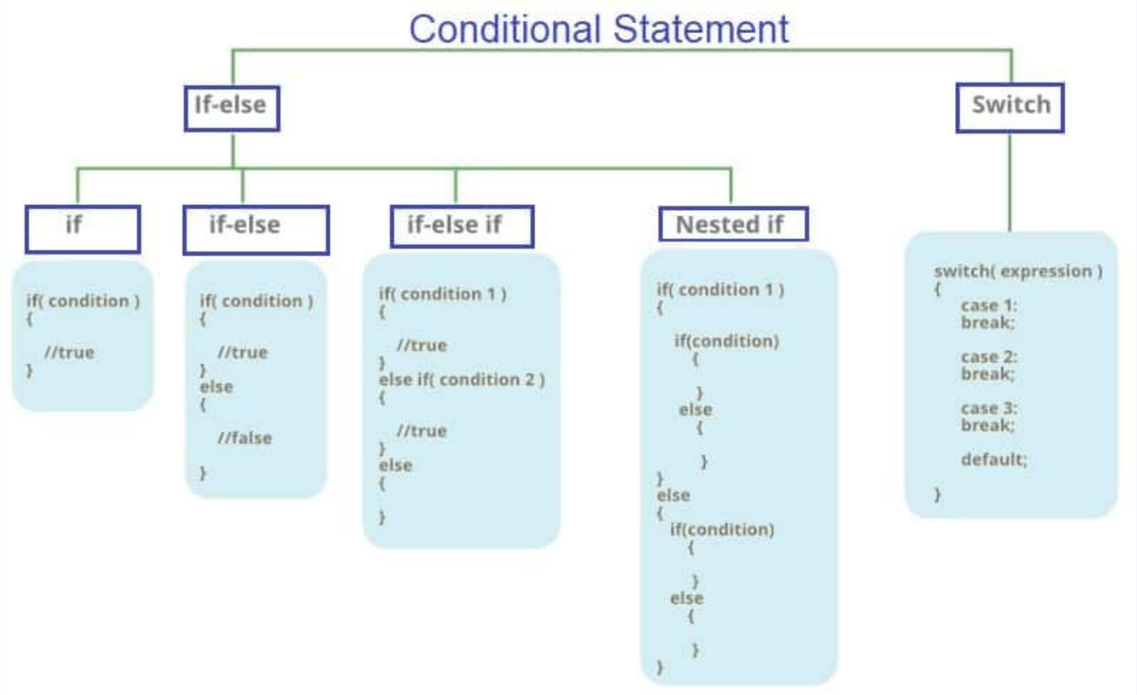
Conditionals

- **If - Else Statement**

  -
    ```
    int a=10;
    int b=15;
    if(a>b)
      Serial.print(“a is greater”);
    else
      Serial.print(“b is greater”);
    ```



Conditionals



```
if( condition )
{
   //true
}
```

```
if( condition )
{
   //true
}
else
{
   //false
}
```

```
if( condition 1 )
{
   //true
}
else if( condition 2 )
{
   //true
}
else
{
}
```

```
if( condition 1 )
{
   if(condition)
    {
    }
   else
    {
     }
}
else
{
  if(condition)
   {
   }
  else
   {
   }
}
```

```
switch( expression )
{
    case 1:
    break;

    case 2:
    break;

    case 3:
    break;

    default;
}
```

Loops

- **For Loop**

  -
    ```
    int i;
    int n =5;
    for(i=0;i<n;i++)
    {
        Serial.println(“hello world”);
    }
    ```
  - **Infinite For Loop**
    ```
    for( ; ;)
    {
         Serial.println(“infinite loop”);
    }
    ```

Loops

- **While Loop**

  -
    ```
    int n=5;
    int i=0;
    while(i<n)
    {
        Serial.println(“hello world”);
        i++;
    }
    ```

  - **Infinite While Loop**
    ```
    while(1)
    {
         Serial.println(“infinite loop”);
    }
    ```

Array

- Array is the variable that stores elements of a similar data type.

- Basic Types Arrays:
    - One Dimensional Array
    - Two Dimensional Array

-
```
int arr[5] = { 1,2,3,4,5};
for(int i=0;i<5;i++)
{
   Serial.println(arr[i]);
}
```

Does your dog bite?
No, but it can hurt you in other ways.
Array indexing starts at 1

Function

- Functions are used to accomplish a task in programming. A function can take parameters and process them to get the desired output.

-
  ```
  void main()
  {
     int a;
     add();
     a = addition();
     Serial.print(a);
  }
  ```

Function

- void main()

```
{
   int a, x=2, y=3;
   addd(x,y);
   add();
   a = addition();
   Serial.print(a);
}

void addd(int x, int y)
{
    int c;
    c=x+y;
    Serial.print(c);
}
```

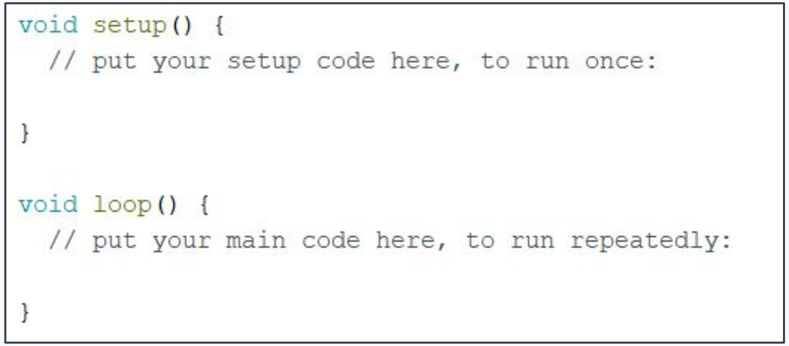
```
void add()
{
   int c,a=5, b=10;
   c=a+b;
   Serial.print(c);
}

int addition()
{
   int c,a=5, b=10;
   c=a+b;
   return(c);
}
```

Structure of a program

- void setup()
  - Gets executed first
  - Runs only one time in the whole program

- void loop()
  - Is executed after the setup function
  - It is an infinite loop and keep on running until stopped.

- void here refers that the function returns nothing when called



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```