El lenguaje se divide en :

Estructura, Variable y Funciones.

|  |  |  |
| --- | --- | --- |
| Structure  * [setup](https://www.arduino.cc/en/Reference/Setup)() * [loop](https://www.arduino.cc/en/Reference/Loop)()  Control Structures  * [if](https://www.arduino.cc/en/Reference/If) * [if...else](https://www.arduino.cc/en/Reference/Else) * [for](https://www.arduino.cc/en/Reference/For) * [switch case](https://www.arduino.cc/en/Reference/SwitchCase) * [while](https://www.arduino.cc/en/Reference/While) * [do... while](https://www.arduino.cc/en/Reference/DoWhile) * [break](https://www.arduino.cc/en/Reference/Break) * [continue](https://www.arduino.cc/en/Reference/Continue) * [return](https://www.arduino.cc/en/Reference/Return) * [goto](https://www.arduino.cc/en/Reference/Goto)  Further Syntax  * [;](https://www.arduino.cc/en/Reference/SemiColon) (semicolon) * [{}](https://www.arduino.cc/en/Reference/Braces) (curly braces) * [//](https://www.arduino.cc/en/Reference/Comments) (single line comment) * [/\* \*/](https://www.arduino.cc/en/Reference/Comments) (multi-line comment) * [#define](https://www.arduino.cc/en/Reference/Define) * [#include](https://www.arduino.cc/en/Reference/Include)  Arithmetic Operators  * [=](https://www.arduino.cc/en/Reference/Assignment) (assignment operator) * [+](https://www.arduino.cc/en/Reference/Arithmetic)  (addition) * [-](https://www.arduino.cc/en/Reference/Arithmetic) (subtraction) * [\*](https://www.arduino.cc/en/Reference/Arithmetic) (multiplication) * [/](https://www.arduino.cc/en/Reference/Arithmetic) (division) * [%](https://www.arduino.cc/en/Reference/Modulo) (modulo)  Comparison Operators  * [==](https://www.arduino.cc/en/Reference/If) (equal to) * [!=](https://www.arduino.cc/en/Reference/If) (not equal to) * [<](https://www.arduino.cc/en/Reference/If) (less than) * [>](https://www.arduino.cc/en/Reference/If) (greater than) * [<=](https://www.arduino.cc/en/Reference/If) (less than or equal to) * [>=](https://www.arduino.cc/en/Reference/If) (greater than or equal to)  Boolean Operators  * [&&](https://www.arduino.cc/en/Reference/Boolean) (and) * [||](https://www.arduino.cc/en/Reference/Boolean) (or) * [!](https://www.arduino.cc/en/Reference/Boolean) (not)  Pointer Access Operators  * [\* dereference operator](https://www.arduino.cc/en/Reference/Pointer) * [& reference operator](https://www.arduino.cc/en/Reference/Pointer)  Bitwise Operators  * [&](https://www.arduino.cc/en/Reference/BitwiseAnd) (bitwise and) * [|](https://www.arduino.cc/en/Reference/BitwiseAnd) (bitwise or) * [^](https://www.arduino.cc/en/Reference/BitwiseAnd) (bitwise xor) * [~](https://www.arduino.cc/en/Reference/BitwiseXorNot) (bitwise not) * [<<](https://www.arduino.cc/en/Reference/Bitshift) (bitshift left) * [>>](https://www.arduino.cc/en/Reference/Bitshift) (bitshift right)  Compound Operators  * [++](https://www.arduino.cc/en/Reference/Increment) (increment) * [--](https://www.arduino.cc/en/Reference/Increment) (decrement) * [+=](https://www.arduino.cc/en/Reference/IncrementCompound) (compound addition) * [-=](https://www.arduino.cc/en/Reference/IncrementCompound) (compound subtraction) * [\*=](https://www.arduino.cc/en/Reference/IncrementCompound) (compound multiplication) * [/=](https://www.arduino.cc/en/Reference/IncrementCompound) (compound division) * [%=](https://www.arduino.cc/en/Reference/IncrementCompound) (compound modulo) * [&=](https://www.arduino.cc/en/Reference/BitwiseCompoundAnd) (compound bitwise and) * [|=](https://www.arduino.cc/en/Reference/BitwiseCompoundOr) (compound bitwise or) | VariablesConstants  * [HIGH](https://www.arduino.cc/en/Reference/Constants) | [LOW](https://www.arduino.cc/en/Reference/Constants) * [INPUT](https://www.arduino.cc/en/Reference/Constants) | [OUTPUT](https://www.arduino.cc/en/Reference/Constants) | [INPUT\_PULLUP](https://www.arduino.cc/en/Reference/Constants) * [LED\_BUILTIN](https://www.arduino.cc/en/Reference/Constants) * [true](https://www.arduino.cc/en/Reference/Constants) | [false](https://www.arduino.cc/en/Reference/Constants) * [integer constants](https://www.arduino.cc/en/Reference/IntegerConstants) * [floating point constants](https://www.arduino.cc/en/Reference/Fpconstants)  Data Types  * [void](https://www.arduino.cc/en/Reference/Void) * [boolean](https://www.arduino.cc/en/Reference/BooleanVariables) * [char](https://www.arduino.cc/en/Reference/Char) * [unsigned char](https://www.arduino.cc/en/Reference/UnsignedChar) * [byte](https://www.arduino.cc/en/Reference/Byte) * [int](https://www.arduino.cc/en/Reference/Int) * [unsigned int](https://www.arduino.cc/en/Reference/UnsignedInt) * [word](https://www.arduino.cc/en/Reference/Word) * [long](https://www.arduino.cc/en/Reference/Long) * [unsigned long](https://www.arduino.cc/en/Reference/UnsignedLong) * [short](https://www.arduino.cc/en/Reference/Short) * [float](https://www.arduino.cc/en/Reference/Float) * [double](https://www.arduino.cc/en/Reference/Double) * [string](https://www.arduino.cc/en/Reference/String) - char array * [String](https://www.arduino.cc/en/Reference/StringObject) - object * [array](https://www.arduino.cc/en/Reference/Array)  Conversion  * [char()](https://www.arduino.cc/en/Reference/CharCast) * [byte()](https://www.arduino.cc/en/Reference/ByteCast) * [int()](https://www.arduino.cc/en/Reference/IntCast) * [word()](https://www.arduino.cc/en/Reference/WordCast) * [long()](https://www.arduino.cc/en/Reference/LongCast) * [float()](https://www.arduino.cc/en/Reference/FloatCast)  Variable Scope & Qualifiers  * [variable scope](https://www.arduino.cc/en/Reference/Scope) * [static](https://www.arduino.cc/en/Reference/Static) * [volatile](https://www.arduino.cc/en/Reference/Volatile) * [const](https://www.arduino.cc/en/Reference/Const)  Utilities  * [sizeof](https://www.arduino.cc/en/Reference/Sizeof)() * [PROGMEM](https://www.arduino.cc/en/Reference/PROGMEM) | [Functions](http://arduino.cc/en/Reference/FunctionDeclaration)Digital I/O  * [pinMode](https://www.arduino.cc/en/Reference/PinMode)() * [digitalWrite](https://www.arduino.cc/en/Reference/DigitalWrite)() * [digitalRead](https://www.arduino.cc/en/Reference/DigitalRead)()  Analog I/O  * [analogReference](https://www.arduino.cc/en/Reference/AnalogReference)() * [analogRead](https://www.arduino.cc/en/Reference/AnalogRead)() * [analogWrite](https://www.arduino.cc/en/Reference/AnalogWrite)() - *PWM*  Due & Zero only  * [analogReadResolution](https://www.arduino.cc/en/Reference/AnalogReadResolution)() * [analogWriteResolution](https://www.arduino.cc/en/Reference/AnalogWriteResolution)()  Advanced I/O  * [tone](https://www.arduino.cc/en/Reference/Tone)() * [noTone](https://www.arduino.cc/en/Reference/NoTone)() * [shiftOut](https://www.arduino.cc/en/Reference/ShiftOut)() * [shiftIn](https://www.arduino.cc/en/Reference/ShiftIn)() * [pulseIn](https://www.arduino.cc/en/Reference/PulseIn)()  Time  * [millis](https://www.arduino.cc/en/Reference/Millis)() * [micros](https://www.arduino.cc/en/Reference/Micros)() * [delay](https://www.arduino.cc/en/Reference/Delay)() * [delayMicroseconds](https://www.arduino.cc/en/Reference/DelayMicroseconds)()  Math  * [min](https://www.arduino.cc/en/Reference/Min)() * [max](https://www.arduino.cc/en/Reference/Max)() * [abs](https://www.arduino.cc/en/Reference/Abs)() * [constrain](https://www.arduino.cc/en/Reference/Constrain)() * [map](https://www.arduino.cc/en/Reference/Map)() * [pow](https://www.arduino.cc/en/Reference/Pow)() * [sqrt](https://www.arduino.cc/en/Reference/Sqrt)()  Trigonometry  * [sin](https://www.arduino.cc/en/Reference/Sin)() * [cos](https://www.arduino.cc/en/Reference/Cos)() * [tan](https://www.arduino.cc/en/Reference/Tan)()  Characters  * [isAlphaNumeric](https://www.arduino.cc/en/Reference/CharacterAnalysis)() * [isAlpha](https://www.arduino.cc/en/Reference/CharacterAnalysis)() * [isAscii](https://www.arduino.cc/en/Reference/CharacterAnalysis)() * [isWhitespace](https://www.arduino.cc/en/Reference/CharacterAnalysis)() * [isControl](https://www.arduino.cc/en/Reference/CharacterAnalysis)() * [isDigit](https://www.arduino.cc/en/Reference/CharacterAnalysis)() * [isGraph](https://www.arduino.cc/en/Reference/CharacterAnalysis)() * [isLowerCase](https://www.arduino.cc/en/Reference/CharacterAnalysis)() * [isPrintable](https://www.arduino.cc/en/Reference/CharacterAnalysis)() * [isPunct](https://www.arduino.cc/en/Reference/CharacterAnalysis)() * [isSpace](https://www.arduino.cc/en/Reference/CharacterAnalysis)() * [isUpperCase](https://www.arduino.cc/en/Reference/CharacterAnalysis)() * [isHexadecimalDigit](https://www.arduino.cc/en/Reference/CharacterAnalysis)()  Random Numbers  * [randomSeed](https://www.arduino.cc/en/Reference/RandomSeed)() * [random](https://www.arduino.cc/en/Reference/Random)()  Bits and Bytes  * [lowByte](https://www.arduino.cc/en/Reference/LowByte)() * [highByte](https://www.arduino.cc/en/Reference/HighByte)() * [bitRead](https://www.arduino.cc/en/Reference/BitRead)() * [bitWrite](https://www.arduino.cc/en/Reference/BitWrite)() * [bitSet](https://www.arduino.cc/en/Reference/BitSet)() * [bitClear](https://www.arduino.cc/en/Reference/BitClear)() * [bit](https://www.arduino.cc/en/Reference/Bit)()  External Interrupts  * [attachInterrupt](https://www.arduino.cc/en/Reference/AttachInterrupt)() * [detachInterrupt](https://www.arduino.cc/en/Reference/DetachInterrupt)()  Interrupts  * [interrupts](https://www.arduino.cc/en/Reference/Interrupts)() * [noInterrupts](https://www.arduino.cc/en/Reference/NoInterrupts)()  Communication  * [Serial](https://www.arduino.cc/en/Reference/Serial) * [Stream](https://www.arduino.cc/en/Reference/Stream)  USB (32u4 based boards and Due/Zero only)  * [Keyboard](https://www.arduino.cc/en/Reference/MouseKeyboard) * [Mouse](https://www.arduino.cc/en/Reference/MouseKeyboard) |

* [**setup**](https://www.arduino.cc/en/Reference/Setup)**() (estructura)**Solo corre una vez, inicia el programa, se declaran valores.

## **void (tipos de data) (variable)**

* Declarar funciones. It indicates that the function is expected to
* return no information to the function from which it was called.

# **loop()(estructura)**

* allowing your program to change and respond

## **-I/O ANALOGICAS**

## analogWrite()

analogWrite()  
  
Description  
  
Writes an analog value (PWM wave) to a pin. Can be used to light a LED at varying brightnesses or drive a motor at various speeds. After a call to analogWrite(), the pin will generate a steady square wave of the specified duty cycle until the next call to analogWrite() (or a call to digitalRead() or digitalWrite() on the same pin). The frequency of the PWM signal on most pins is approximately 490 Hz. On the Uno and similar boards, pins 5 and 6 have a frequency of approximately 980 Hz. Pins 3 and 11 on the Leonardo also run at 980 Hz.  
  
On most Arduino boards (those with the ATmega168 or ATmega328), this function works on pins 3, 5, 6, 9, 10, and 11. On the Arduino Mega, it works on pins 2 - 13 and 44 - 46. Older Arduino boards with an ATmega8 only support analogWrite() on pins 9, 10, and 11.9, 10, and 11.

analogRead()  
  
Description  
  
Reads the value from the specified analog pin. The Arduino board contains a 6 channel (8 channels on the Mini and Nano, 16 on the Mega), 10-bit analog to digital converter. This means that it will map input voltages between 0 and 5 volts into integer values between 0 and 1023. This yields a resolution between readings of: 5 volts / 1024 units or, .0049 volts (4.9 mV) per unit. The input range and resolution can be changed using analogReference().  
  
It takes about 100 microseconds (0.0001 s) to read an analog input, so the maximum reading rate is about 10,000 times a second.  
  
Syntax  
  
analogRead(pin)

## **---------------------------------------------------**

## **-I/O DIGITALES-**

## **digitalWrite() (función)**

* Write a [HIGH](https://www.arduino.cc/en/Reference/Constants) or a [LOW](https://www.arduino.cc/en/Reference/Constants) value to a digital pin.
* seteado como OUTPUT : 5V (or 3.3V on 3.3V boards) for HIGH, 0V (ground) for LOW.
* como INPUT, digitalWrite() will enable (HIGH) or disable (LOW) the internal pullup on the input pin.
* sintaxis : digitalWrite(pin, value) donde value : [HIGH](https://www.arduino.cc/en/Reference/Constants) or [LOW](https://www.arduino.cc/en/Reference/Constants)

## **digitalRead() (funcion)**

* Reads the value from a specified digital pin, either [HIGH](https://www.arduino.cc/en/Reference/Constants) or [LOW](https://www.arduino.cc/en/Reference/Constants). sintaxis : digitalRead(pin)

## **pinMode() (funcion)**

* Configures the specified pin to behave either as an input or an output.
* sintaxis : pinMode(pin, mode) donde mode : [INPUT](https://www.arduino.cc/en/Reference/Constants), [OUTPUT](https://www.arduino.cc/en/Reference/Constants), or [INPUT\_PULLUP](https://www.arduino.cc/en/Reference/Constants)
* ------------------------------------------------------------------------

## **int (tipo de dato)(variable) Integers are your primary data-type for number storage.**

## int var = val; var - your int variable name

## val - the value you assign to that variable

**if (conditional) and ==, !=, <, > (comparison operators)**

x == y (x is equal to y)  
 x != y (x is not equal to y)  
 x < y (x is less than y)   
 x > y (x is greater than y)   
 x <= y (x is less than or equal to y)   
 x >= y (x is greater than or equal to y)

if (someVariable > 50)  
{  
 // do something here  
}

## **if / else**

## **boolean (tipo de dato)(variable)** A boolean holds one of two values, [true](https://www.arduino.cc/en/Reference/Constants) or [false](https://www.arduino.cc/en/Reference/Constants).

**&& (logical and)**

True only if both operands are true, e.g.

if (digitalRead(2) == HIGH && digitalRead(3) == HIGH) { // read two switches   
 // ...  
}  *is true only if both inputs are high.*

### **|| (logical or)**

True if either operand is true, e.g.

if (x > 0 || y > 0) {  
 // ...  
}

*is true if either x or y is greater than 0.*

### **! (not)**

True if the operand is false, e.g.

if (!x) {   
 // ...  
}

is true if x is false (i.e. if x equals 0).

**Examples**

if (a >= 10 && a <= 20){} // true if a is between 10 and 20

## **string (tipo de dato)(variable)**

## **Arrays(tipo de dato)(variable)**

An array is a collection of variables that are accessed with an index number

creando :

int myInts[6];  
 int myPins[] = {2, 4, 8, 3, 6};  
 int mySensVals[6] = {2, 4, -8, 3, 2};  
 char message[6] = "hello";

## **Variables**

A variable is a way of naming and storing a value for later use by the program, such as data from a sensor or an intermediate value used in a calculation.

## **float()**

#### Description

Converts a value to the [float](https://www.arduino.cc/en/Reference/Float) data type  
Syntax  
float(x)  
Parameters  
x: a value of any type

## 