Arduino C- Preguntas 1

# P1: ¿Qué tamaño tiene el tipo de dato double? Compare el tipo double con el tipo float, ¿Qué puede concluir?

El tipo de dato doublé tiene un tamaño de 4 bytes en el Uno y otros boards basado en ATMEGA. Esto es lo mismo que en float con ninguna ganancia en precisión.

Datos tipo float son números con un punto decimal. Estos números están frequentemente usados para aproximisar calores anlogos y continuos asi como tiene una resolución mayor que integers. Estos números pueden ser tan largo como 3.4028235E+38 y tan bajo como -3.4028235E+38. Etan guardados como 32 bits(4 bytes). Floats tienen solo 6-7 digitis decimales de precisión.

No como con otras plataformas, donde puedes conseguir mas precisión unsado un doublé( e.g. hasta 15 digitos). Numeros floars no son exactos z podrían resultar en resultados raros cuando se comparen. Por ejemploi 6.0/3.0 no equivale 2.0, por eso se debería chequear si el valor absoluto de la diferencia entre aquellos numeors es menos que un numero menor. Floating point math también es mas lento que integer math en la ejecuccion de calculaciones. Entonces debería ser evitado, por ejemplo cuando deberi afuncionar en un mayor velocidad. Algunos programadores entonces conviertan calculaciones de floating points a matemáticas de inegeras para mejorar la velocidad.

Example Code:

int x

int y

float z

x = 1

y = x / 2

// y now contains 0, ints can't hold fractions

z = (float)x / 2.0

// z now contains .5 (you have to use 2.0, not 2)

# P2: Explore algunas de las [bibliotecas](https://www.arduino.cc/en/Reference/Libraries) estándar de arduino. Explore tres de ellas y explique para qué sirven y de un ejemplo cada una.

El ambiente arduino puede ser extendido por el uso de bibliotecas asi como la moyoria parte de plataformas de programación. Estas bibliotecas están proveniendo funcionalidad extra para uso en sketches. Varias biblitecas vienen instalado con el IDE, pero también se puede descargar y crear si mismo.

SD:

La biblioteca SD permite leer de y escribir en tarjetas SD, e.g. el Arduino Ethernet Shield. Aquella esta construido en sdfatlib y supports FAT16 y FAT 32 file sistemas de tarjetas SD estandarizados y tarjetas SDHC.Usa cortos 8.3 nombres para files. Paths pueden ser seperados por forward slashes “/”. Esta biblitoca supports la apertura de miltimples archivos. La comunicación entre el microcontrolador y la tarjeta SD usa SPI(Serial Peripheral Interface), que ocurre en los pins digitales 11,12 y 13 para casi todos los arduinos. Adicionalmente otro pin tiene que ser usado para seleccionar la tarjeta SD, que podría ser el hardware SS pin –pin 10 u otro pinespicicado en la call para iniciar la SD.

EEPROM:

El microcontrolador del Board Arduino tiene EEPROM(*Electrically Erasable Programmable Read-Only Memory)* una memoria del cual los valores están guardado cuando el board esta apagado, funciona como una pequeña hard drive. Esta biblioteca permita leer y escribir estos bytes. El Arduino uno con el atmega 328 tiene 512 bytes

Ethernet Shield:

Esta biblioteca permito un Arduino Board de conectar al Internet. Podria servir como un servidor, aceptando conexiones incoming o un client aceptando outgoing. La biblioteca supoorts hasta 4 conxiones concurrentes de cualquier combinación. El Arduino comunica con el shield usando el SPI bus, que esta en los pins digitales 11,12 y 13(UNO) Hay que utiliyar el pin 10 como SS.

# P3: ¿Cuál es la [estructura](https://www.arduino.cc/en/Reference/HomePage) de un programa para un Arduino?

Programas de Arduino pueden ser divididos en tres partes: Estructura, Valores (Variables y constants) y funciones:

1. Estructrura:

* Funciones básicas
  + [setup](https://www.arduino.cc/en/Reference/Setup)()
  + [loop](https://www.arduino.cc/en/Reference/Loop)()
* Estructuras de control
  + [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)
* Ademas Syntax
  + (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)
* Operadores aritmeticas
  + [=](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)
* Operadores comparativas
  + [==](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)
* Operadores booleanos
  + [&&](https://www.arduino.cc/en/Reference/Boolean) (and)
  + [||](https://www.arduino.cc/en/Reference/Boolean) (or)
  + [!](https://www.arduino.cc/en/Reference/Boolean) (not)

2. Variables:

* Constantantes
  + [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)
* Tipos de datos
  + [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)

3. Funciones

* 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)()
* 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)

# P5:

Localice el archivo main.cpp en el directorio donde descomprimió el archivo .zip descargado de la página del proyecto Arduino. La función main en un programa en C es el punto de entrada del programa.

P5a. ¿Cuántas veces se ejecuta la función setup?

1 vez

P5b. ¿Cuántas veces se ejecuta la función loop?

tantas veces como definido en loop

# P5c. ¿Cuál es la ruta del archivo main.cpp en el sistema de archivos?

arduino-1.6.5-r2-windows\hardware\arduino\avr\cores\arduino

/\*

main.cpp - Main loop for Arduino sketches

Copyright (c) 2005-2013 Arduino Team. All right reserved.

\*/

#include <Arduino.h>

// Declared weak in Arduino.h to allow user redefinitions.

int atexit(void (\* /\*func\*/ )()) { return 0; }

// Weak empty variant initialization function.

// May be redefined by variant files.

void initVariant() \_\_attribute\_\_((weak));

void initVariant() { }

int main(void)

{

init();

initVariant();

#if defined(USBCON)

USBDevice.attach();

#endif

setup();

for (;;) {

loop();

if (serialEventRun) serialEventRun();

}

return 0;

}

setup();

for (;;) {

loop();

if (serialEventRun) serialEventRun();

}

return 0;

}

# P6. ¿Qué tamaño en bytes y qué rango de valores tiene los siguientes [tipos de datos](https://www.arduino.cc/en/Reference/HomePage): boolean, char, unsigned char, byte, int, unsigned int, word, long, unsigned long, float, double, string, String, array, void.

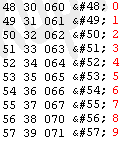
|  |  |  |
| --- | --- | --- |
| Tipo | Longitud (en byte) | Rango |
| *Boolean* | 1 | Limitado a logico *true* y *false* |
| *char* | 1 | Rango: –128 to +127 |
| *unsigned char* | 1 | Rango: 0 to 255 |
| *byte* | 1 | Rango: 0 to 255 |
| *int* | 2 | Rango: –32,768 to 32,767 |
| *unsigned int* | 2 | Rango: 0 to 65,535 |
| *word* | 2 | Rango: 0 to 65,535 |
| *long* | 4 | Rango: –2,147,483,648 to 2,147,483,647 |
| *unsigned long* | 4 | Rango: 0 to 4,294,967,295 |
| *float* | 4 | Rango: –3.4028235E+38 to 3.4028235E+38 |
| *double* | 4: | Rango: –3.4028235E+38 to 3.4028235E+38 |
| *string* | ? | Una tipo data de referencia terminado en null ('\0' ) y construido por un arreglo de characteres |
| *String* | ? | Un objeto del tipo de datos de la referencia |
| *Array* | ? | A sequence of a value type that is referenced by a single variable name |
| *void* | 0 | Una secuencia de un tipo del valor que es referido por un solo nombre de variable |

# P7. ¿Para qué sirve terminar un [string](https://www.arduino.cc/en/Reference/String) en null (código [ASCII 0](http://www.asciitable.com/))?

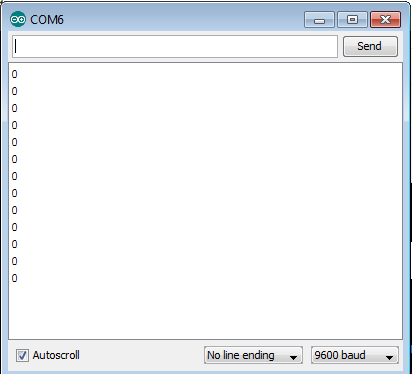
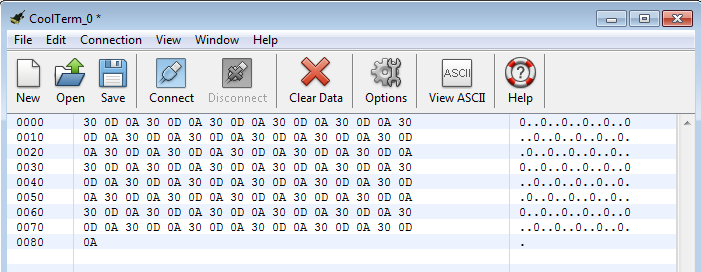
string: strings de texto puedes ser representado en dos maneras, se puede usar el tipo de data string, que es parte del core como en la version 0019, o es posible de crear un string de un array de tipos of char and determinar aquellos cero. Generalmente strings estan terminados en null (ASCII code 0). Esto permita a funciones (como Serial.print()) de decir donde se ubica el fin de un string. Sino se quedaria leyendo bztes subsecuentos de memoria que no son parte del string. Strings siempre estan definidos entre doble comillas, mientras characters adentro de single comillas (‘A’).

# P8. ¿Cuál es el código ASCII para los número del 0 al 9?





P10 Cool Term:



30=0

0D=carriage return

0A=New line feed

P11: ¿Cómo se crean un arreglo? muestre y explique un ejemplo.

Array: un array es una colleccion de variablesque son accesidos con numeros index. En programaccion C pueden ser complicados, pero simples son faciles.

int myInts[6]

int myPins[] = {2, 4, 8, 3, 6}

int mySensVals[6] = {2, 4, -8, 3, 2}

char message[6] = "hello"

Un array puede ser declarado sin inicializar como en myInts. En MzPins un array puede ser declarado sin escojer explicitivamente un tamano. El compile cuenta los elementos y crea un array de tamano apprpiado. Tambien se puede iniciar un arraz como en mzSensVals. Acuerde que en tipo char un elemento mas en la inicialicacion esta requerido: 0.

P12. ¿Cómo se accede y cómo se escribe un elemento de un arreglo? muestre y explique un ejemplo.

Arrays con indexed con cero(primero elemento en index 0 y ultimo elemnto es index 9 en [10], index 10 sera invalido y contiene informacion random que es **otra memory adress!!!-->escribir aca resuelta en crashes o malfuncion del programa!!**). El compiler C no esta checkeando si un acceso a un array esta adentro de las fronteras legales de un array declarado.

para asignar un valor a un array: MySensVals[0]=10

para recibir un valor de un array: x=mySensVals[4]

Arrays y FOR loops: muchas veces arrays estan manipulados para loops, donde el loop counter es usado como el index para un elemento del array:

int i

for (i = 0

i < 5

i = i + 1) {

Serial.println(myPins[i])

}

P13. Cree un arreglo con su nombre y escriba un programa que envíe cada uno de los caracteres del nombre al puerto serial. Observe el resultado utilizando CoolTerm.

int miNombre[7]={'M','o','r','i','t','z'}

void setup() {

Serial.begin(9600)

}

void loop() {

int i

for (i = 0

i < 6

i = i + 1) {

Serial.println(miNombre[i])

}

}

Nota: int nombre[]={‘M’

...}

→ ={77,...} -->println(int) -->55,55

char nombre[]={‘M’

...}

→ ={77,...} -->println(char) -->77

P14. ¿Cuáles son los operadores de comparación? de ejemplos de cada uno

* [==](https://www.arduino.cc/en/Reference/If) (equal to) 5==5
* [!=](https://www.arduino.cc/en/Reference/If) (not equal to) 5!=6
* [<](https://www.arduino.cc/en/Reference/If) (less than) 5<6
* [>](https://www.arduino.cc/en/Reference/If) (greater than) 5>4
* [<=](https://www.arduino.cc/en/Reference/If) (less than or equal to) 5<=5
* [>=](https://www.arduino.cc/en/Reference/If) (greater than or equal to) 5>=5

P15. Construya programas ejemplo para las siguientes estructuras de control: if, if-else, if-else if- else, for, switch-case, while.

if

if (x > 120){

digitalWrite(LEDpin1, HIGH)

digitalWrite(LEDpin2, HIGH)

}

if-else

if (x > 120){

digitalWrite(LEDpin1, HIGH)

}

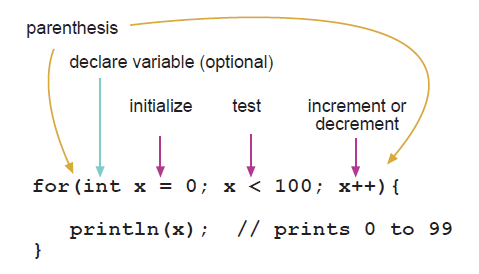
ELSE

{

digitalWrite(LEDpin2, HIGH)

}

if-else if-else

for

for (int x=0

x<120

x++ {

println (x)

}

switch-case

var=x

switch (var) {

case 1:

println (1)

break

case 2:

println (2)

break

default:

println (0)

}

while

while(x>120){

printlln (x)

x++

}

P16. Escriba e implemente el código que se muestra a la derecha. ¿Para qué sirve el modificador [const](https://www.arduino.cc/en/Reference/Const)?

el modificador const es para EVITAR CAMBIOs EN UNA VARIABLE

/\*

Alternate Blink

Turns on one LED on for one second while the other is off, then

reverses the LEDs for 1 second, repeatedly.

Dr. Purdum, 11/13/2014

\*/

// Given each LED pin a name and don't let it be changed by the program:

const int LED1 = 10

const int LED2 = 11

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

void setup() {

// initialize the digital pins as an output.

pinMode(LED1, OUTPUT)

pinMode(LED2, OUTPUT)

}

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

void loop() {

digitalWrite(LED1, HIGH)

// turn LED on (HIGH is the voltage level = 5V = ON)

digitalWrite(LED2, LOW)

// turn LED off by making the voltage LOW (= 0V = OFF)

delay(1000)

// wait for a second

digitalWrite(LED1, LOW)

// turn the LED off by making the voltage LOW

digitalWrite(LED2, HIGH)

// turn LED on (HIGH is the voltage level)

delay(1000)

// wait for a second

}

P17. Compile de nuevo el programa. Compare el mensaje final del proceso de compilación con el mensaje que aparece en la figura de este slide. ¿Qué se puede concluir?

1100 vs 1116

Implemente el código que se muestra en la figura.

P18. ¿Cómo funciona? explique cada línea de código

https://123d.circuits.io/circuits/945513-the-unnamed-circuit/edit#breadboard

Sets the first *num* bytes of the block of memory pointed by *ptr* to the specified *value* (interpreted as an unsigned char).

void \* memset ( void \* ptr, int value, size\_t num )

the sizeof() operator does not give you the number of elements in an array, it gives you the *number of bytes* a thing occupies in memory

sizeof( s ) = 6

sizeof( xs ) = 24