

<https://www.arduino.cc/en/Reference/HomePage>

Arduino

Referencia idiomática. Los programas de Arduino se pueden dividir en tres partes principales: estructura, valores (variables y constantes) y funciones.

Structure – Estructura

Structure – Estructura	
Structure	Estructura
setup()	Configuración ()
loop()	bucle ()

Control Structures – Estructuras de Control

Structure – Estructura	
Control Structures	Estructuras de Control
if	Si
if...else	Si otro
for	para
switch case	Caja del interruptor
while	mientras
do... while	Hacer mientras
break	descanso
continue	continuar
return	regreso
goto	ir

Further Syntax – Sintaxis adicional

Structure – Estructura	
Further Syntax	Sintaxis adicional
;(semicolon)	;(punto y coma)
{ }(curly braces)	{ }(Llaves)
// (single line comment)	// (comentario de una sola línea)
/* */ (multi-line comment)	/* */ (Comentario de varias líneas)
#define	#definir
#include	#incluir
;(semicolon)	;(punto y coma)

Arithmetic Operators – Operadores aritméticos

Structure – Estructura	
Arithmetic Operators	Operadores aritméticos
= (assignment operator)	= (operador de asignación)
+ (addition)	+ (adición)
- (subtraction)	- (sustracción)
* (multiplication)	* (multiplicación)
/ (division)	/ (División)
% (modulo)	% (módulo)

Comparison Operators - Operadores de comparación

Structure – Estructura	
Comparison Operators	Operadores de comparación
== (equal to)	== (igual a)
!= (not equal to)	!= (no igual a)
< (less than)	< (menos de)
> (greater than)	> (mayor que)
<= (less than or equal to)	<= (menor o igual que)
>= (greater than or equal to)	>= (mayor o igual que)

Boolean Operators - Operadores booleanos

Structure – Estructura	
Boolean Operators	Operadores booleanos
&& (and)	&& (y)
(or)	(o)
! (not)	! (no)

Pointer Access Operators - Operadores de acceso de puntero

Structure – Estructura	
Pointer Access Operators	Operadores de acceso de puntero
* dereference operator	* operador de desreferencia
& reference operator	& operador de referencia

Bitwise Operators - Operadores de bit a bit

Structure – Estructura	
Bitwise Operators	Operadores de bit a bit
& (bitwise and)	& (bit a bit y)
(bitwise or)	(bit a bit o)
^ (bitwise xor)	^ (bit xor)

~ (bitwise not)	~ (bitwise no)
<< (bitshift left)	<< (Bitshift izquierda)
>> (bitshift right)	>> (bits hacia la derecha)

Compound Operators - Operadores compuestos

Structure – Estructura	
Compound Operators	Operadores compuestos
++ (increment)	++ (incremento)
-- (decrement)	- (decremento)
+= (compound addition)	+ = (adición del compuesto)
-= (compound subtraction)	- = (substracción compuesta)
*= (compound multiplication)	* = (multiplicación compuesta)
/= (compound division)	/ = (división del compuesto)
%= (compound modulo)	% = (compuesto modulo)
&= (compound bitwise and)	& = (compuesto bit a bit y)
= (compound bitwise or)	= (compuesto bit a bit o)

Variables – Variables

Constants – Constantes

Variables	
Constants	Constantes
HIGH LOW	ALTO BAJO
INPUT OUTPUT INPUT_PULLUP	INPUT SALIDA INPUT_PULLUP
LED_BUILTIN	LED_INCORPORADO
true false	verdadero falso
integer constants	constantes enteras
floating point constants	constantes de punto flotante

Data Types - Tipos de datos

Variables	
Data Types	Tipos de datos
void	vacío
boolean	booleano
char	carbonizarse
unsigned char	carácter sin signo
byte	byte
int	int
unsigned int	unsigned int
word	palabra
long	largo
unsigned long	sin firmar
short	corto
float	flotador
double	doble
string - char array	cuerda - char array
String - object	cuerda - objeto
array	formación
void	vacío

Variable Scope & Qualifiers - Alcance variable y calificadores

Variables	
Variable Scope & Qualifiers	Alcance variable y calificadores
variable scope	alcance variable
Static	estático
Volatile	volátil
Const	const

Utilities – Utilidades

Variables	
Utilities	Utilidades
sizeof()	sizeof ()
PROGMEM	PROGMEM

Functions – Funciones

<https://www.arduino.cc/en/Reference/FunctionDeclaration>

Digital I/O - E / S digital

Functions – Funciones	
Digital I/O	E / S digital
pinMode()	pinMode ()
digitalWrite()	digitalWrite ()
digitalRead()	digitalRead ()

Analog I/O - E / S analógica

Functions – Funciones	
Analog I/O	E / S analógica
analogReference()	analogReference ()
analogRead()	analogRead ()
analogWrite() - PWM	analogWrite () - PWM

Due & Zero only - Sólo para Due & Zero

Functions – Funciones	
Due & Zero only	Sólo para Due & Zero
analogReadResolution()	analogReadResolution ()
analogWriteResolution()	analogWriteResolution ()

Advanced I/O - E / S avanzada

Functions – Funciones	
Advanced I/O	E / S avanzada
tone()	tono ()
noTone()	noTone ()
shiftOut()	shiftOut ()
shiftIn()	shiftIn ()
pulseIn()	pulseIn ()

Time – Hora

Functions – Funciones	
Time	Hora
millis()	millis ()
micros()	micros ()
delay()	retraso ()
delayMicroseconds()	delayMicroseconds ()
millis()	millis ()

Math - Mates

Functions – Funciones	
Math	Mates
min()	min ()
max()	max ()
abs()	abs ()
constrain()	limitar ()
map()	mapa ()
pow()	pow ()
sqrt()	sqrt ()

Trigonometry - Trigonometría

Functions – Funciones	
Trigonometry	Trigonometría
sin()	sin()
cos()	cos()
tan()	tan()

Characters – Caracteres

Functions – Funciones	
Characters	Caracteres
isAlphaNumeric()	isAlphaNumeric ()
isAlpha()	isAlpha ()
isAscii()	isAscii ()
isWhitespace()	isWhitespace ()
isControl()	isControl ()
isDigit()	isDigit ()
isGraph()	isGraph ()
isLowerCase()	isLowerCase ()
isPrintable()	esPrintable ()
isPunct()	isPunct ()
isSpace()	isSpace ()
isUpperCase()	isUpperCase ()
isHexadecimalDigit()	isHexadecimalDigit ()

Random Numbers - Números al azar

Functions – Funciones	
Random Numbers	Números al azar
randomSeed()	randomSeed ()
random()	azar ()

Bits and Bytes - Bits y bytes

Functions – Funciones	
Bits and Bytes	Bits y bytes
lowByte()	bajoByte ()
highByte()	highByte ()
bitRead()	bitRead ()
bitWrite()	bitWrite ()
bitSet()	bitSet ()
bitClear()	bitClear ()
bit()	bit ()
lowByte()	bajoByte ()

External Interrupts - Interrupciones externas

Functions – Funciones	
External Interrupts	Interrupciones externas
attachInterrupt()	attachInterrupt ()
detachInterrupt()	detachInterrupt ()

Interrupts – Interrupciones

Functions – Funciones	
Interrupts	Interrupciones
interrupts()	interrupciones ()
noInterrupts()	noInterrupts ()

Communication – Comunicación

Functions – Funciones	
Communication	Comunicación
Serial	De serie
Stream	Corriente

USB (32u4 based boards and Due/Zero only) - USB (tablas basadas en 32u4 y Due / Zero solamente)

Functions – Funciones	
USB (32u4 based boards and Due/Zero only)	USB (tablas basadas en 32u4 y Due / Zero solamente)
Keyboard	Teclado
Mouse	Ratón

<https://www.arduino.cc/en/Reference/Libraries>

Bibliotecas

El entorno Arduino puede ampliarse mediante el uso de bibliotecas, al igual que la mayoría de las plataformas de programación. Las bibliotecas proporcionan funcionalidades adicionales para su uso en bocetos, por ejemplo, trabajar con hardware o manipular datos. Para utilizar una biblioteca en un boceto, selecciónela en Boceto> Importar biblioteca .

Varias bibliotecas vienen instaladas con el IDE, pero también puedes descargar o crear las tuyas propias. Consulte estas instrucciones para obtener detalles sobre la instalación de bibliotecas. También hay un tutorial sobre cómo escribir sus propias bibliotecas . Consulte la Guía de estilo API para obtener información sobre cómo crear una API de estilo Arduino adecuada para su biblioteca.

Bibliotecas estándar

EEPROM - lectura y escritura en almacenamiento permanente

Ethernet / Ethernet 2 - para conectarse a Internet mediante el Arduino Ethernet Shield, Arduino Ethernet Shield 2 y Arduino Leonardo ETH

Firmata - para comunicarse con aplicaciones en el equipo mediante un protocolo serie estándar.

GSM - para la conexión a una red GSM / GRPS con el blindaje GSM.

LiquidCrystal - para controlar las pantallas de cristal líquido (LCD)

SD - para leer y escribir tarjetas SD

Servo - para el control de servomotores

SPI - para comunicarse con dispositivos que utilizan el bus de interfaz periférica serie (SPI)

SoftwareSerial - para la comunicación serial en cualquier pines digitales. La versión 1.0 y posterior de Arduino incorpora la biblioteca NewSoftSerial de Mikal Hart como SoftwareSerial .

Stepper - para el control de motores paso a paso

TFT - para dibujar texto, imágenes y formas en la pantalla TFT de Arduino

WiFi - para conectarse a Internet mediante el escudo WiFi de Arduino

Wire - Interfaz de dos hilos (TWI / I2C) para enviar y recibir datos a través de una red de dispositivos o sensores.

Las bibliotecas Matrix y Sprite ya no forman parte de la distribución central.