

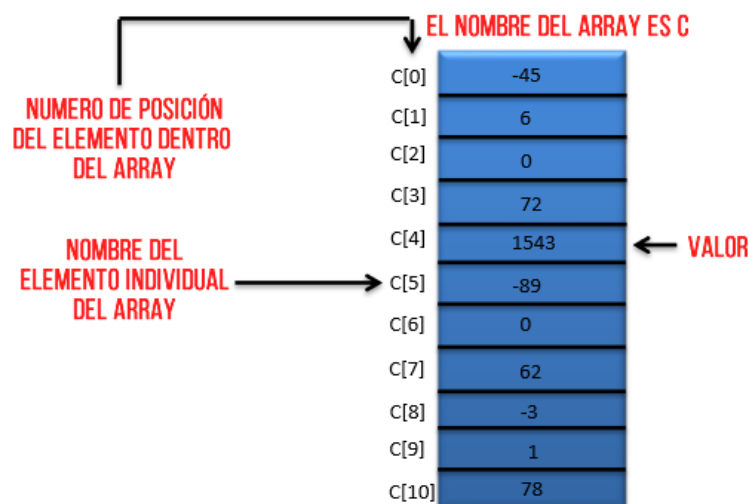
## IOT – INTERNET OF THINGS

### ARDUINO – ARRAYS (ARREGLOS)

Un arreglo es un grupo consecutivo de ubicaciones de memoria que **son del mismo tipo**. Para hacer referencia a una ubicación o elemento en particular en el arreglo, especificamos el nombre del arreglo y el número de posición del elemento particular en el arreglo.

La ilustración que se muestra a continuación muestra un arreglo de enteros denominada C que contiene 11 elementos. Puedes hacer referencia a cualquiera de estos elementos dando el nombre del arreglo seguido del número de posición del elemento en particular entre corchetes ([ ]). El número de posición se denomina más formalmente subíndice o índice (este número especifica el número de elementos desde el principio del arreglo). El primer elemento tiene un subíndice 0 (cero) y, a veces, se le llama elemento cero.

Por lo tanto, los elementos del arreglo C son C [0] (pronunciado "C sub cero"), C [1], C [2] y así sucesivamente. El subíndice más alto en la matriz C es 10, que es 1 menos que el número de elementos en el arreglo (11). Los nombres de los arreglos siguen las mismas convenciones que otros nombres de variables.



Un subíndice debe ser una expresión entera o entera (usando cualquier tipo integer). Si un programa usa una expresión como subíndice, entonces el programa evalúa la expresión para determinar el subíndice. Por ejemplo, si asumimos que la variable a es igual a 5 y que la variable b es igual a 6, entonces la declaración agrega 2 al elemento de matriz C [11].

Un nombre de arreglo con subíndice es un valor l, se puede usar en el lado izquierdo de una asignación, al igual que los nombres de variables que no son de arreglo.

Examinemos el arreglo C en la figura dada, más de cerca. El nombre de este vector completo es C. Sus 11 elementos se denominan C [0] a C [10]. El valor de C [0] es -45, el valor de C [1] es 6, el valor de C [2] es 0, el valor de C [7] es 62 y el valor de C [10] es 78.

Para imprimir en Arduino (en el monitor Serie) la suma de los valores contenidos en los primeros tres elementos de la matriz C, escribiríamos –

```
Serial.print (C[ 0 ] + C[ 1 ] + C[ 2 ] );
```

Para dividir el valor de C [6] por 2 y asignar el resultado a la variable x, escribiríamos –



`x = C[ 6 ] / 2;`

## Declaración de Arrays

Los arreglos ocupan espacio en la memoria. Para especificar el tipo de elementos y el número de elementos requeridos por un array, use una declaración de la forma -

`tipo nombreArray [ tamañoArray ] ;`

El compilador reserva la cantidad adecuada de memoria. (Recuerde que una declaración, que reserva memoria, se conoce más propiamente como definición). `tamañoArray` debe ser una constante entera mayor que cero. Por ejemplo, para decirle al compilador que reserve 11 elementos para la matriz de enteros C, use la declaración -

`int C[ 12 ]; // C es un arreglo de 12 enteros`

Se puede declarar que los arrays contienen valores de cualquier tipo de datos que no sean de referencia. Por ejemplo, se puede utilizar una matriz de tipo cadena para almacenar cadenas de caracteres.

## Ejemplos que utilizan arrays

Esta sección ofrece muchos ejemplos que demuestran cómo declarar, inicializar y manipular arrays.

### Ejemplo 1: Declaración de un array y uso de un bucle para inicializar los elementos de un array

El programa declara un array `n` de números enteros de 10 elementos. Las líneas `a – b` usan una instrucción `For` para inicializar los elementos de la matriz a ceros. Como otras variables automáticas, las matrices automáticas no se inicializan implícitamente a cero. La primera declaración de salida (línea `c`) muestra los encabezados de las columnas impresas en la declaración `for` subsiguiente (líneas `d-e`), que imprime la matriz en formato tabular.

**Veámoslo en código:**

```
int n[ 10 ] ; // n es un array de 10 enteros sin inicializar

void setup () {

}

void loop () {
  // inicializar los índices del array n a 0
  for ( int i = 0; i < 10; ++i ) {
    n[ i ] = 0; // determinar que el elemento en la posición i sea 0
    Serial.print (i); // imprimo el resultado en pantalla del monitor Serial
    Serial.print ('\r');
  }
  // imprimir en pantalla cada elemento del array
  for ( int j = 0; j < 10; ++j ) {
    Serial.print (n[j]); // imprimo n[j] en pantalla del monitor Serial
    Serial.print ('\r');
  }
}
```

### Ejemplo 2: Inicialización de un array en una declaración con una lista de inicializadores

Los elementos de un array también se pueden inicializar en la declaración del array siguiendo el nombre del arreglo con un signo igual a y una lista de inicializadores separados por comas y delimitados por llaves. El



programa usa una lista de inicializadores para inicializar un array de enteros con 10 valores (línea a) e imprime la matriz en formato tabular (líneas b – c).

**Veámoslo en código:**

```
// n es un arreglo de 10 enteros inicializados
int n[ 10 ] = { 32, 27, 64, 18, 95, 14, 90, 70, 60, 37 } ;

void setup () {

}

void loop () {
    for ( int i = 0; i < 10; ++i ) {
        Serial.print (i) ;
        Serial.print ('\r') ;
    }
    // imprimir cada elemento del array
    for ( int j = 0; j < 10; ++j ) {
        Serial.print (n[j]) ;
        Serial.print ('\r') ;
    }
}
```

### Ejemplo 3: Suma de los elementos de un array

A menudo, los elementos de un array representan una serie de valores que se utilizarán en un cálculo. Por ejemplo, si los elementos de una matriz representan calificaciones de exámenes, un profesor puede desear sumar los elementos de la matriz y usar esa suma para calcular el promedio de la clase para el examen. El programa suma los valores contenidos en la matriz de números enteros de 10 elementos a.

**Veámoslo en código:**

```
const int TAMANIO = 10; // variable constante que indica el tamaño de la matriz
int a[ TAMANIO ] = { 87, 68, 94, 100, 83, 78, 85, 91, 76, 87 };
int total = 0;

void setup () {

}

void loop () {
    // suma del contenido de la matriz a
    for ( int i = 0; i < TAMANIO; ++i ){
        total += a[ i ];
    }
    Serial.print ("Total de la suma de elementos : ") ;
    Serial.print(total) ;
}
```