CONCEPTO ARRAY,STRING

# Definicion:

Array (Arreglo): te permite almacenar **múltiples valores** del mismo tipo en una sola variable .Ademas puedes acceder y manipular utilizando **índices**.

# Características de un array:

* Todos los elementos en un array deben ser del **mismo tipo** de datos (por ejemplo, todos enteros, todos flotantes, etc.).
* Cada elemento dentro del array está almacenado en una posición específica, llamada **índice**.
* Los índices en un array empiezan desde **0**. Es decir, el primer elemento está en la posición 0, el segundo en la posición 1, y así sucesivamente.
* En un array de tipo byte en Arduino, cada elemento puede almacenar un valor entero sin signo de 8 bits. Esto significa que cada índice del array puede contener un número entre **0** y **255**.Como se ve en el ejemplo de abajo un array o arreglo de tipo byte llamado miArray Con seis posiciónes o seis índices y por lo que se ve el numero máximo en cada índice o posición es de 255

byte miArray[6] = {255, 0, 128, 50, 100, 200};

# Sintaxis en Arduino:

tipo nombreDelArray[tamaño];

* tipo: El tipo de datos del array (por ejemplo, int, float, char, etc.).
* nombreDelArray: El nombre que le das al array.
* tamaño: El número de elementos que tendrá el array.

# EJEMPLOS DE SUS USOS:

1-Declarando array especificando cuantas posiciones puede almacenar.

**int numeros[5];**

*// Declara un array de 5 enteros*

*//Este array puede almacenar 5 números enteros.*

2-Asignacion de valores y acceso de cada índice o posición :

**int numeros[5] = {10, 20, 30, 40, 50};**

Aquí a la variable llamada numero de tipo entero se le ha asignado un valor a cada indice o posición.

 El **índice 5** no existe en un array declarado con 5 elementos.

 Los índices válidos van de **0** a **4** en este caso.

 Acceder a un índice fuera de este rango puede generar errores o resultados inesperados.

*// Acceder a los elementos del array*

*int primerNumero = numeros[0]; // primerNumero será 10*

*int segundoNumero = numeros[1]; // segundoNumero será 20*

*// Modificar un valor del array*

*numeros[2] = 35; // Ahora el tercer elemento es 35*

3- Uso de arrays en bucles:

muy útiles cuando necesitas trabajar con múltiples valores en un bucle. Por ejemplo, puedes recorrer un array usando un bucle for:

**for (int i = 0; i < 5; i++) {**

**Serial.println(numeros[i]); // Imprime todos los elementos del array**

**}**

# STRING(CADENA DE CARACTERES)

Un **string** es una **secuencia de caracteres** (letras, números, símbolos) que se tratan como un solo valor.Un string es un array de carcteres asi de sencillo

# MANERAS DE TRABAJAR CON EL ARRAY

1. Cadenas de caracteres:

**char saludo[] = "Hola";**

Este código crea un array de caracteres que contiene las letras 'H', 'o', 'l', 'a', seguidas de un carácter nulo (\0). Los arrays de caracteres requieren ese carácter nulo para marcar el final de la cadena.El carácter nulo no es visible .

1. Acceso y modificación de caracteres en una cadena:

**saludo[0] = 'h'; // Cambia la 'H' por 'h'**

**Serial.println(saludo); // Imprime "hola"**

1. String como objeto:

utilizar el tipo de datos **String** (con mayúscula inicial), que es más flexible y proporciona varias funciones para manipular cadenas de texto. Este tipo de **String** es un **objeto**, y te permite hacer operaciones más fácilmente como concatenar, comparar, buscar dentro de cadenas, etc.

**String nombre = "Arduino"; // Declara un objeto String**

**Serial.println(nombre); // Imprime "Arduino"**

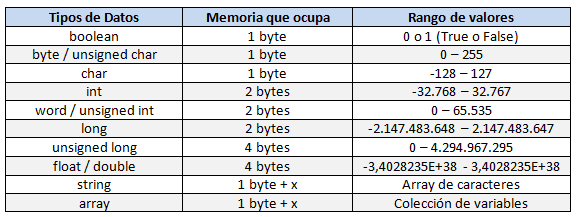
**// Concatenación de Strings**

**nombre = nombre + " Uno"; // Ahora nombre contiene "Arduino Uno"**

**Serial.println(nombre);**

Los **String (objeto)** son más fáciles de usar porque vienen con muchas funciones integradas, pero consumen más memoria y, en algunos casos, pueden causar fragmentación de la memoria, lo que es algo a tener en cuenta si tu programa es grande.

# TIPOS DE DATOS ARDUINO:



**Tamaño del array**: Depende del tipo de dato y la cantidad de elementos que se declaren. Un array de int con 5 elementos ocupará 5 \* 2 bytes = 10 bytes en un Arduino Uno.

Int Nombre[5];

# EXPLICACION IMAGEN:

1-El primero es un array llamado MIARRAY de tipo byte .Tiene 6 indices o posiciones,Aquí se especifican las posiciones para almacenar números.El índice 6 no se usa y se empieza siempre por el índice 0.

2-En segundo array las 6 posiciones tienen valores asignados mientras que en el anterior no se especificaron y están para almacenar valores.

3-El tercero accedes al índice[0] y almacenas su valor en una variable llamada Valor .Ademas declaras VALOR.

4- accedes al índice[4] que es igual a 5 y almacenas su valor en una variable llamada Valor .Ademas declaras VALOR.

