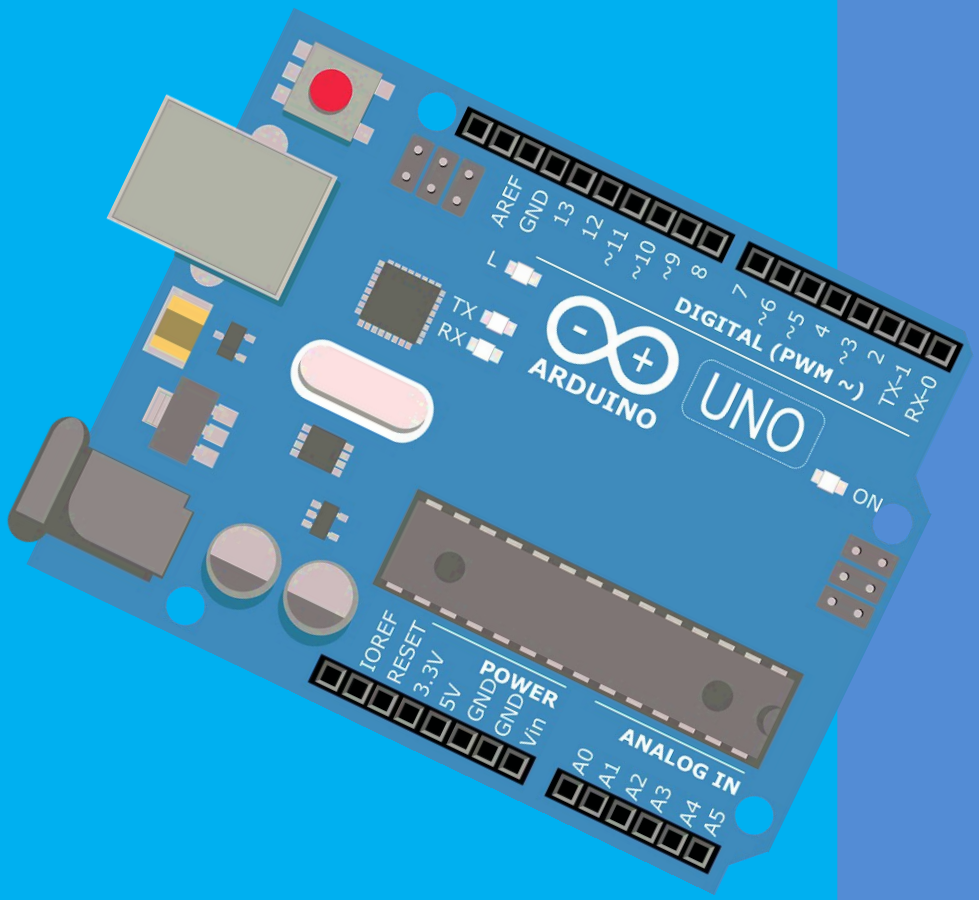
ARDUINO NIVEL 1

El truco de las variables para los nombres

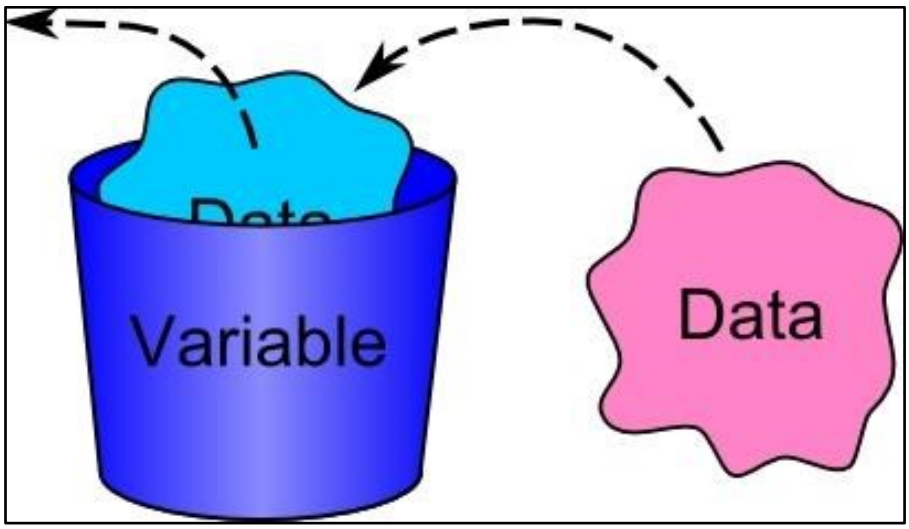
Instructor: Konrad Peschka



El truco de las variables para los nombres

Hemos ido jugando con variables de una forma más o menos informal, por una decisión deliberada. Hemos usado los diferentes tipos de variables y hemos operado con ellas, pero sin entrar en muchos detalles.

¿Qué es una variable? Una variable es un lugar donde almacenar un dato, tiene un nombre, un valor y un tipo.



Variable

Los nombres de variables pueden tener letras, números y el símbolo '_'. Deben empezar por una letra (pueden empezar por '_ ' pero no es recomendable pues es el criterio que usan las rutinas de la biblioteca). Pueden llevar mayúsculas y minúsculas. La convención es que las variables van en minúscula y las constantes en mayúscula

Usa las mismas reglas dentro del código para el nombramiento de variables, ya sea en minúscula con palabras separadas con guiones bajos, tantos como sea necesario para mejorar la legibilidad

Las palabras reservadas if, else,etc . . . no pueden usarse como nombres de variables. Nombres para evitar: Nunca uses los caracteres ‘l’ (letra ele en minúscula), ‘O’ (letra o mayúscula), o ‘I’ (letra i mayúscula) como simples caracteres para nombres de variables, para evitar confusiones a la hora de leer el código.

Tipos de variables

Los datos que guardamos en las variables pueden ser de diferentes tipos, vamos a listar algunos de ellos.

- **char**, se utilizan para almacenar caracteres, ocupan un byte.
- **byte**, pueden almacenar un número entre 0 y 255.
- **int**, ocupan 2 bytes (16 bits), y por lo tanto almacenan número entre 2-15 y 215-1, es decir, entre -32,768 y 32,767.
- **unsigned int**, ocupa también 2 bytes, pero al no tener signo puede tomar valores entre 0 y 216-1, es decir entre 0 y 65,535.
- **long**, ocupa 32 bits (4 bytes), desde -2,147,483,648 a 2,147,483,647.
- **float**, son números decimales que ocupan 32 bits (4 bytes). Pueden tomar valores entre -3.4028235E+38 y +3.4028235E+38.
- **double**, también almacena números decimales, pero disponen de 8-bytes (64 bit).

Siempre que elegimos un tipo de dato debemos escoger el que menos tamaño necesite y que cubra nuestras necesidades, ya que ocupan espacio en la memoria de nuestra placa y podría darse el caso de que nuestro programa requiere más memoria de la disponible.

Pero, ¿cómo “hacemos” variables en nuestro código? Es muy sencillo: indicando el tipo y el nombre de la variable. Además, podemos darle o no un valor inicial

Declaración de variables

Una variable tiene un nombre, un valor y un tipo. Con la asignación, se puede cambiar el valor de la variable.

Todas las variables deben ser declaradas antes de su uso. Las declaraciones deben aparecer al principio de cada función o bloque de sentencias. Al declarar una variable se debe indicar primero el tipo de variable y luego su nombre, opcionalmente se le puede dar un valor, lo que se llama inicializar la variable.

La declaración consta de un tipo de variable y una lista de variables separadas por coma.

- int i,j;
- float x,pi;
- unsigned long longitud, contador;

Las variables pueden inicializarse en la declaración

- float pi=3.1416;
- unsigned long contador=0;

Puede utilizarse el modificador const para indicar que la variable no puede ser cambiada en tiempo de ejecución.

- const float e=2.7182;

La declaración de una variable sólo debe hacerse una vez en un programa, pero el valor de la variable se puede cambiar en cualquier momento usando aritmética y reasignaciones diversas.

Constantes

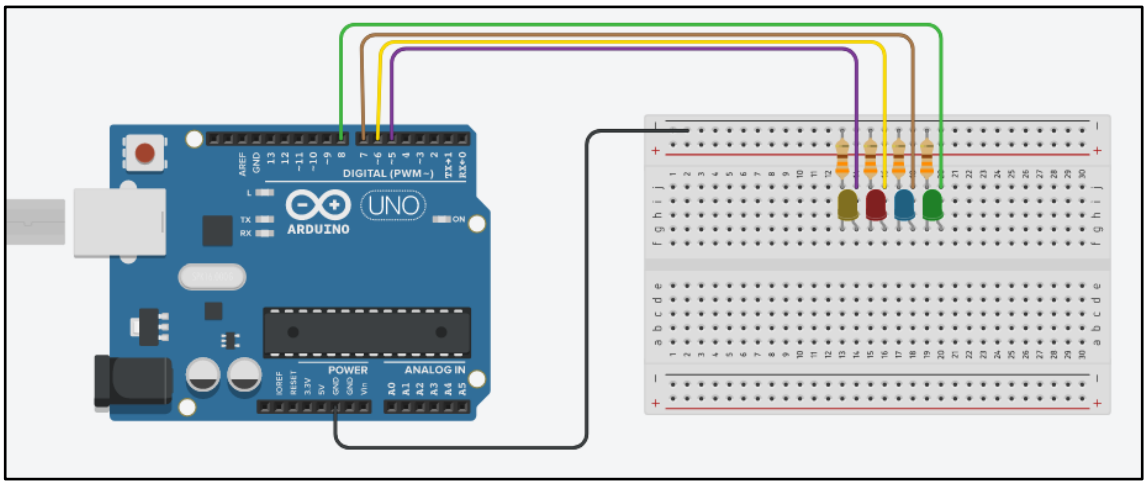
En programación, una constante es un valor que no puede ser alterado/modificado durante la ejecución de un programa, únicamente puede ser leído. Una constante corresponde a una longitud fija de un área reservada en la memoria principal del ordenador, donde el programa almacena valores fijos. Por ejemplo, el valor de PI = 3.1416.

El modificador const, modifica el comportamiento de una variable haciéndola “read-only”, esto significa que puede usarse como cualquier otra variable, pero su valor no puede ser cambiado.

Circuito Propuesto

El circuito propuesto consta de un Arduino y 4 leds de color amarillo, rojo, azul, y verde que irán encendiendo secuencialmente para después apagarse secuencialmente, para ello usaremos:

- 1 Arduino uno
- 4 Diodos Leds color rojo, amaillo, azul, verde
- 4 Resistencia 330Ω
- 1 Protoboard



Circuito: Secuencia de leds usando variables

Solución de la Programación

Programa: Secuencia de leds usando variables

```
const int amarillo=5;                    //Se declaran variables globales
const int rojo=6;                         //de tipo int
const int azul=7;
const int verde=8;

void setup ()
{

  pinMode (amarillo, OUTPUT);
  pinMode (rojo, OUTPUT);                //Definir los puertos 5, 6, 7, 8 como salida
  pinMode (azul, OUTPUT);
  pinMode (verde, OUTPUT);

}

void loop ()
{
    digitalWrite (amarillo, HIGH);          //se asigna la variable amarillo al estado alto
    delay(1000);                            //Se ejecuta un retardo de 1 seg
    digitalWrite (rojo, HIGH);
    delay(1000);
    digitalWrite (azul, HIGH);
    delay(1000);
    digitalWrite (verde, HIGH);
  delay(1000);
    digitalWrite (verde, LOW);            // se asigna la variable amarillo al estado bajo
  delay(1000);                              // Se ejecuta un retardo de 1 seg
    digitalWrite (azul, LOW);
  delay(1000);
    digitalWrite (rojo, LOW);
  delay(1000);
    digitalWrite (amarillo, LOW);
   delay(1000);
}
```