Wels

C Embebido

Booleanos

- El tipo de dato lógico también se conoce como **booleano**.

- Representa valores de lógica binaria es decir datos de lógica que sólo tienen dos estados: **verdadero o falso**.

True	False
1	0

Wels

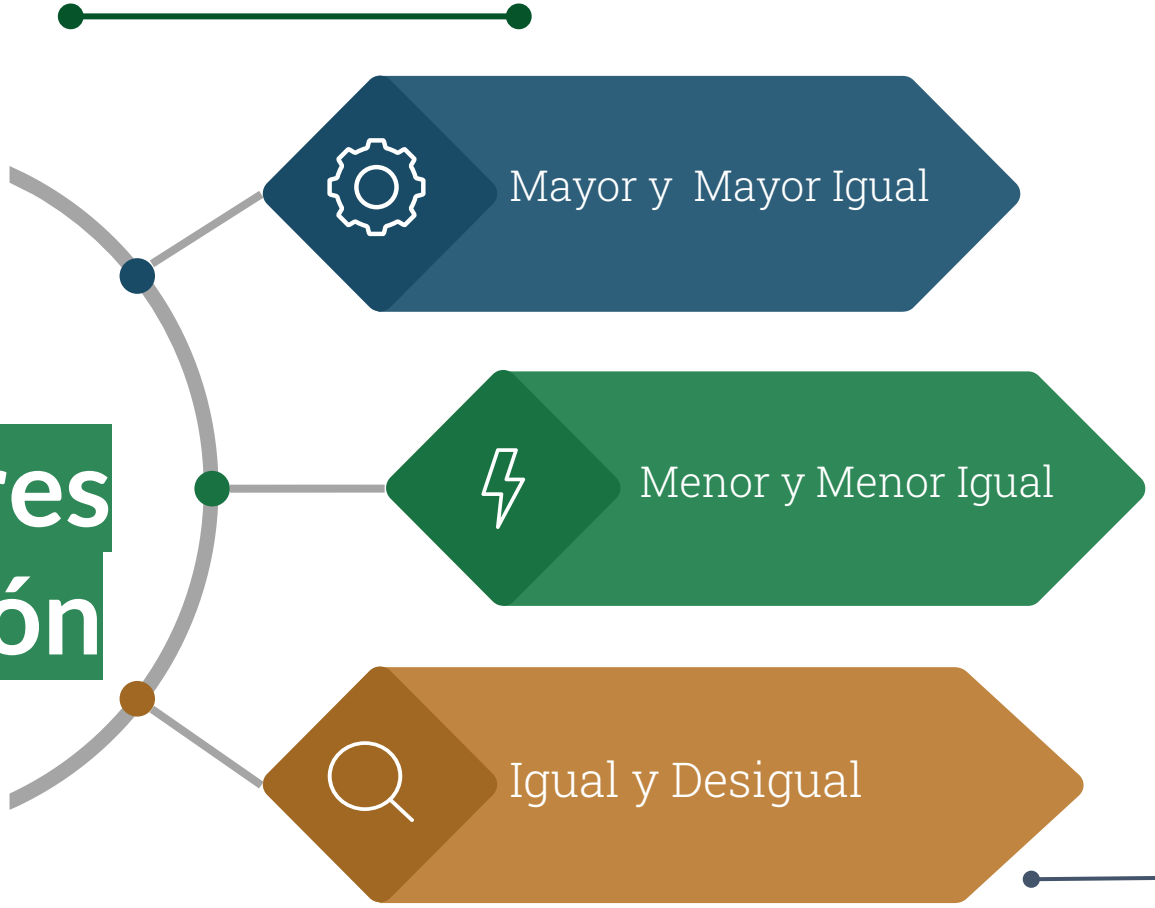
Stdbool.h

Librería estándar contiene 4 macros para para un tipo de datos booleanos.

```
#define bool    _Bool
#define true    1
#define false   0
```

```
/* Signal that all the definitions are present.  */
#define __bool_true_false_are_defined   1
```

Operadores de Relación



Mayor y Mayor Igual

Retorna un 1 (true) si es mayor o mayor igual.

> y >=

```
int main() {
   int a = 3, b = 5;
   printf("Es mayor: %d \r\n",a>b);
   int d = 8, c = 8;
   printf("Es mayor igual: %d \r\n",c>=d);

   return 0;
}
```

Menor y Menor Igual

Retorna un 1 (true) si es menor o menor igual.

< y <=

```
int main() {
   int a = 3, b = 5;
   printf("Es menor: %d \r\n",a<b);
   int d = 8, c = 8;
   printf("Es menor igual: %d \r\n",c<=d);

   return 0;
}
```

Igual y Desigual

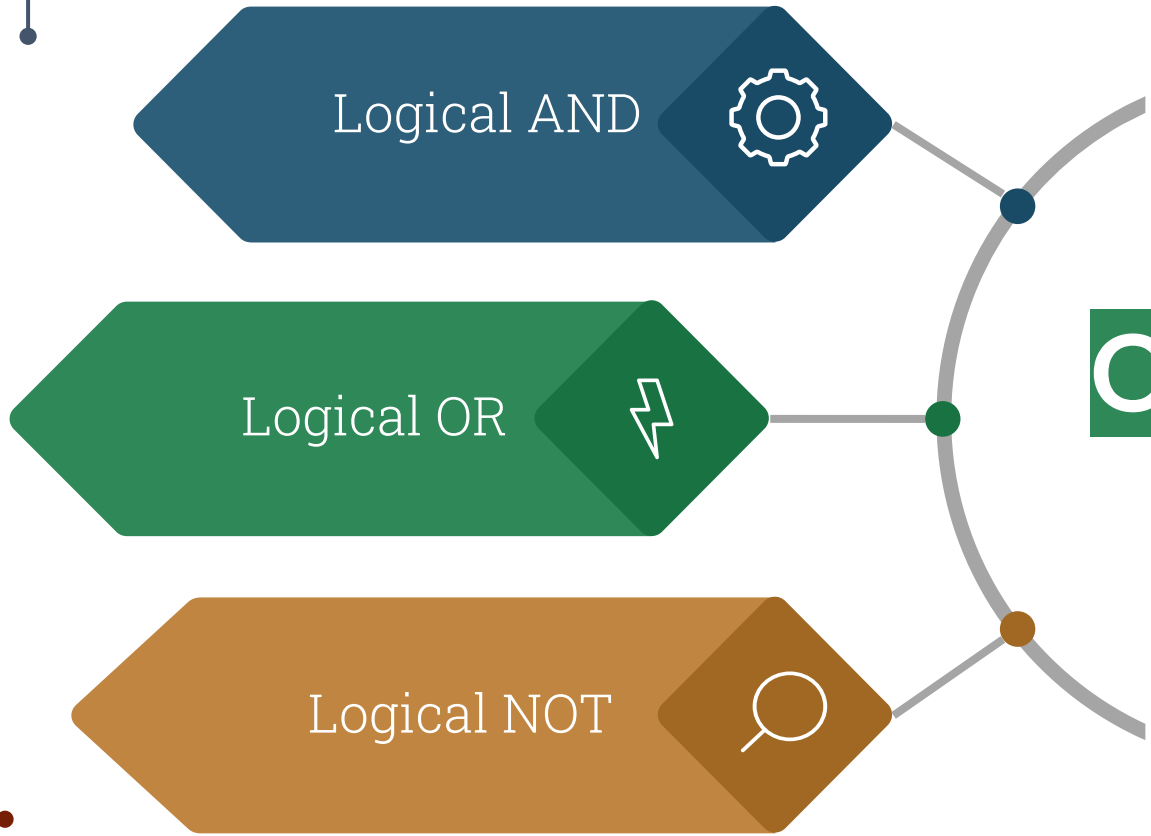
Retorna un 1 (true) si es igual.

Retorna un 1 (true) si es desigual.

== y !=

```c
int main() {
   int a = 3, b = 5;
   printf("Es desigual: %d \r\n",a!=b);
   int d = 8, c = 8;
   printf("Es igual: %d \r\n",c==d);

   return 0;
}
```



Operadores Lógicos

Logical AND

Si los dos operandos tienen valores de true (1), el resultado es true (1)

&&

```
int main()
{
    int a = 3, b = 5;
    printf("Logical AND: %d \r\n", a>0 && b>0);
    int c = -5, d = 2;
    printf("Logical AND: %d \r\n", c>0 && d>0);

    return 0;
}
```

Logical OR

Si uno de los operandos tienen valor de true (1), el resultado es true (1)

||

```
int main()
{
    int a = 3, b = 5;
    printf("Logical OR: %d \r\n", a>0 || b>0);
    int c = -5, d = 2;
    printf("Logical OR: %d \r\n", c>0 ||| d>0);

    return 0;
}
```

Logical NOT

Retorna un 1 (true) si el operando es 0.

!

```c
int main()
{
    int a = 3, b = 5;
    printf("Logical NOT: %d \r\n", !(a>0));
    int c = -5, d = 2;
    printf("Logical NOT: %d \r\n", !(c>0));

    return 0;
}
```

Wels

IF

Wels

Declaración IF

Conozcamos mejor cómo utilizar el IF

Wels

IF

- Si la expresión se evalúa como verdadera, se ejecuta lo que está dentro del IF.

IF

```
int valor = 28;

if( valor < 30)
{
    // Código dentro del IF
}
// Código fuera del IF
```

Wels

ELSE

- Ejecuta un bloque de código si la condición es falsa.

ELSE

```
int valor = 28;

if( valor < 30)
{
    // Si la condición del IF es true
}
else
{
    // Si la condición del IF es false
}
```

Wels

ELSE IF

- Usa **Else IF** si la primera condición del **IF** es false.

Else If

```
int valor = 28;

if( valor < 30)
{
    // Si la condición del IF es true
}
else if( valor > 30)
{
    // Si la 1° condición es false
}
else
{
    // Si la 2° condición es false
}
```

IF Anidado

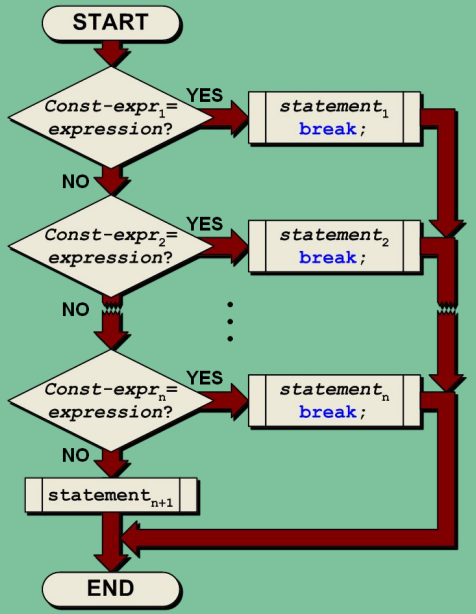
Wels

Se puede utilizar un **IF** o **ELSE IF** dentro de otro **IF** o **ELSE IF**.

```c
if( valor < 30) // 1° IF
{
    if ( valor2 == 5) // 2° IF
    {
        if ( valor3 > 0 ) // 3° IF
        {
            printf("Todo es verdad \r\n");
        }
    }
}
```

SWITCH

- Es una manera más elegante de manejar código que tiene múltiples IF.
- Las condiciones se evalúan cómo tipos de enteros (int y char).



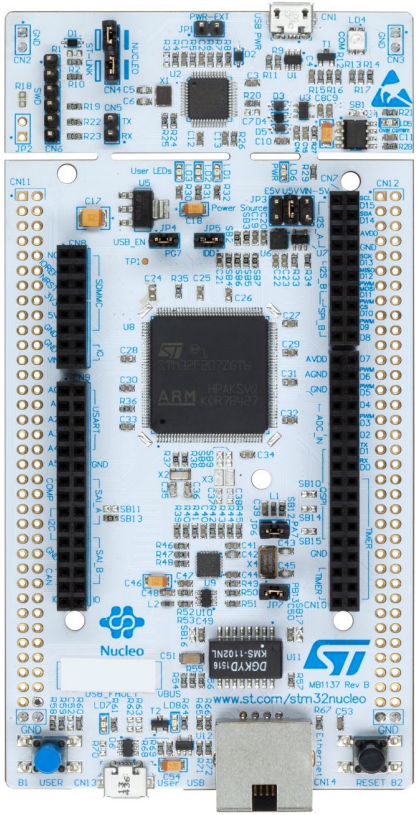
Wels

SWITCH

```
switch(channel)
{
    case 1:
        printf("Canal 1 \r\n");
        break;
    case 2:
        printf("Canal 2 \r\n");
        break;
    case 3:
        printf("Canal 3 \r\n");
        break;
    default:
        printf("No es ninguno \r\n");
        break;
}
```

¡Veamos un ejemplo!

Wels



Gracias

@welstheory
hola@welstheory.com
+51 918 899 684