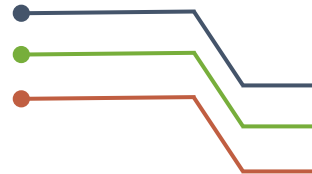
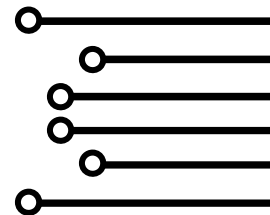




C Embebido



Funciones

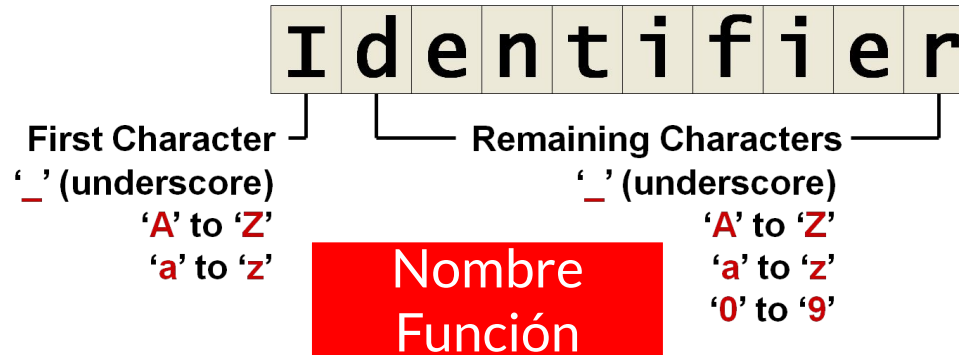
- Son **bloques de código** que realizan una tarea en específico.
- Su propósito principal es **dividir el código** en partes más pequeñas y manejables, lo que facilita la comprensión y el mantenimiento del programa.

```
funcion1()
```

```
void funcion1(void)
```

Estructura de Función

```
return_data_type funcion_name(arguments)
{
    // Cuerpo de la funcion
}
```





Función Delay

Funciones



Función sin retorno

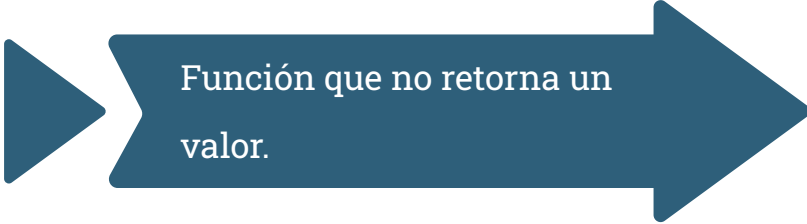


Función con argumentos



Función con retorno

Función sin retorno

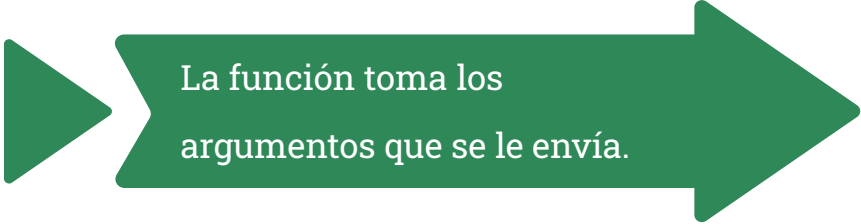


Función que no retorna un valor.

```
void funcion(void)
```

```
void funcion_prueba(void)
{
    printf("Prueba \r\n");
}
```

Función con argumentos



La función toma los
argumentos que se le envía.

```
void funcion_prueba(uint32_t valor)
```

```
void funcion_prueba(uint32_t valor)
{
    printf("El valor es %d \r\n", valor);
}
```

Función con retorno

La función retorna un tipo de variable y siempre se utiliza return.

```
int funcion_prueba(uint32_t valor)
```

```
uint32_t funcion_prueba(uint32_t valor, uint32_t valor2)
{
    uint32_t result;
    result = valor * valor2;
    return result;
}
```


MAIN

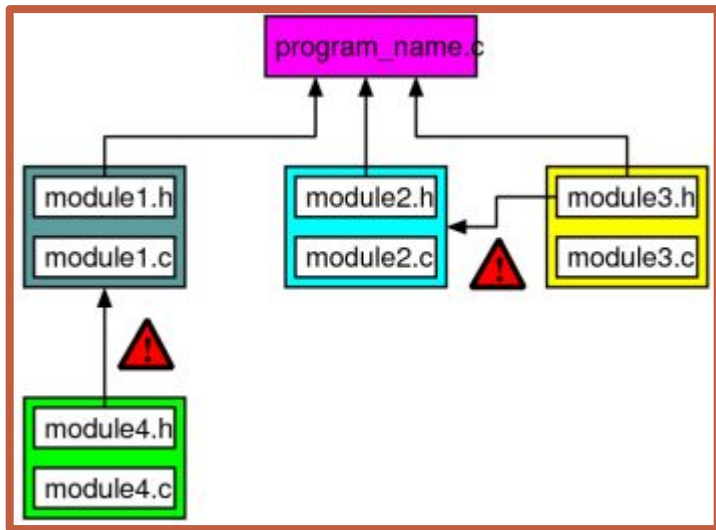
La función principal (main) es una palabra clave usada para ejecutar el inicio y final del programa.

```
int main(void)
{
    return 0;
}
```

Retorna un valor de 0 que significa exitoso.

Prototipos de Función

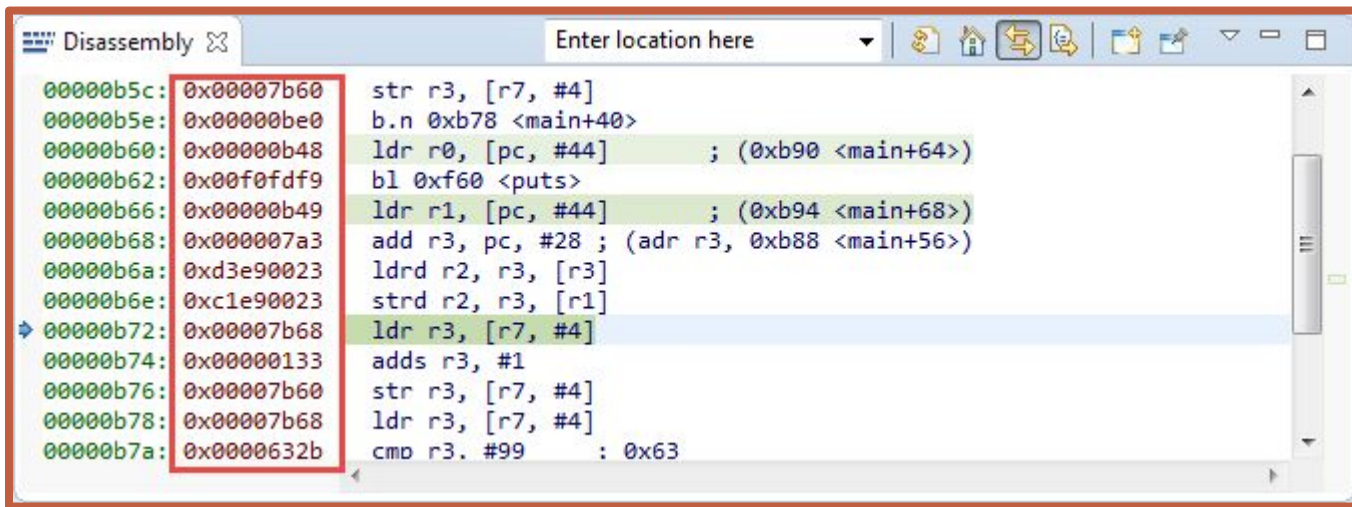
Wels



- Las funciones primero debe declararse.
- Muy parecido a una declaración de variables.
- Es por esto que es preferible usar módulos.

Depuración Disassembly

Wels



```
Disassembly  Enter location here
0000b5c: 0x00007b60 str r3, [r7, #4]
0000b5e: 0x00000be0 b.n 0xb78 <main+40>
0000b60: 0x00000b48 ldr r0, [pc, #44] ; (0xb90 <main+64>)
0000b62: 0x00f0fdf9 bl 0xf60 <puts>
0000b66: 0x00000b49 ldr r1, [pc, #44] ; (0xb94 <main+68>)
0000b68: 0x000007a3 add r3, pc, #28 ; (adr r3, 0xb88 <main+56>)
0000b6a: 0xd3e90023 ldrd r2, r3, [r3]
0000b6e: 0xc1e90023 strd r2, r3, [r1]
➔ 0000b72: 0x00007b68 ldr r3, [r7, #4]
0000b74: 0x00000133 adds r3, #1
0000b76: 0x00007b60 str r3, [r7, #4]
0000b78: 0x00007b68 ldr r3, [r7, #4]
0000b7a: 0x0000632b cmp r3, #99 : 0x63
```

Veamos cómo se ejecutan las funciones en C
Embebidos.



Gracias

@welstheory

hola@welstheory.com

+51 918 899 684

