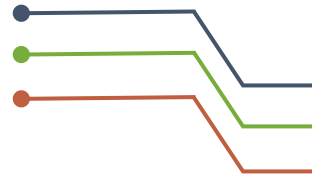
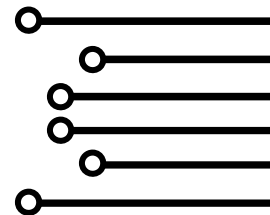
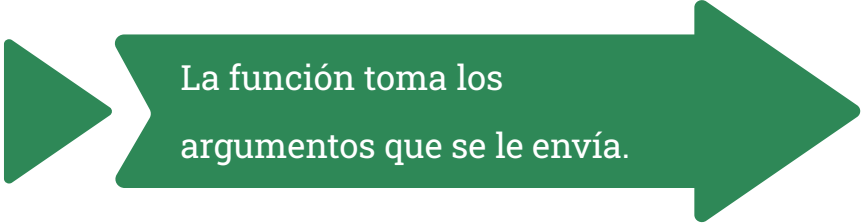




C Embebido



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 argumentos

```
void funcion1(int varX)
{
    varX++;
    printf("Funcion varX: %d \n\r",varX);
}

void funcion2(int varX)
{
    varX++;
    printf("Funcion 2 varX: %d \n\r",varX);
}
```

Cualquier parámetro declarado en una función sólo existe en el contexto de la función

El resto de programa no tiene el acceso de leer o modificar el parámetro

Variable local

```
void Funcion1(int varX)
{
    int x = 5 ,y;
    y = x + varX;
    printf("Valor Y: %d \r\n",y);
}
```

Variables
local

Las variables **x**, **y** y **varX** son sólo accesibles cuando llamas a la función "Funcion1"

Variables locales

```
void Funcion1(int varX)
{
    int x = 5 ,y;
    y = x + varX;
    printf("Valor Y: %d \r\n",y);
}
```

- **Cuando se termina de ejecutar la función(})** la variable local se destruye y no se podrá acceder.
- **Nunca retorne la dirección** de la variable local desde la función.
- La pila contiene las variables locales no estáticas de funciones.

Variable global

```
int varX;
```

Las variables globales son declaradas ***fuera de las funciones.***

Puede accederse por cualquier función o fuera del archivo .c

Variables globales

```
int varX;
```

- A diferencia de las variables locales, las globales **no se destruye** cuando finaliza la función.
- Las variables globales **se inicializan con 0** si no se menciona explícitamente
- Las variables globales se pueden acceder entre archivos .c



STATIC

Variables



Static



Extern

Static Variables

Una variable estática conserva su valor anterior.

static int variable

```
void Funcion1(void)
{
    static int varX = 0;
    varX++;
    printf("Funcion1 varX: %d \r\n", varX);
}
```

Static

```
void Funcion1(void)
{
    static int varX =0;
    varX++;
    printf("Funcion1 varX: %d \r\n",varX);
}
```

- Variables static **dentro de funciones** sólo se pueden acceder dentro de la función.
- Existe **durante toda** la vida del programa.
- Las variables globales **son siempre del tipo estático.**
- Las variables globales static sólo son **accesibles** por este archivo .c

Static Funciones

Una función estática son locales para el archivo .c en el que se encuentra.

```
static int Funcion1(void)
```

```
static void Funcion1(void)
{
    static int varX =0;
    varX++;
    printf("Funcion1 varX: %d \r\n",varX);
}
```

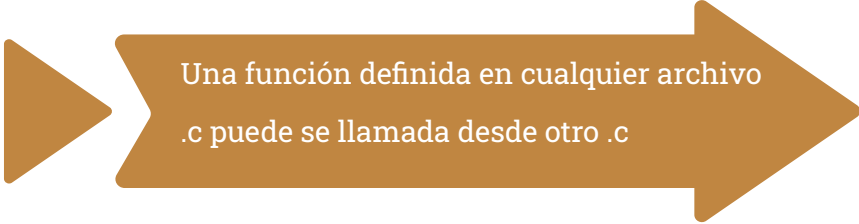
Extern variable

Extern se utiliza para declarar variables que son definidas en otro archivo .c

```
extern int variable
```

```
extern int variable;  
  
void Funcion1(void)  
{  
    variable++;  
    printf("Funcion1 varX: %d \r\n",variable);  
}
```

Extern Funciones



Una función definida en cualquier archivo
.c puede ser llamada desde otro .c

```
extern int Funcion2(void);
```

```
extern int Funcion2(void);
```



Gracias

@welstheory

hola@welstheory.com

+51 918 899 684

