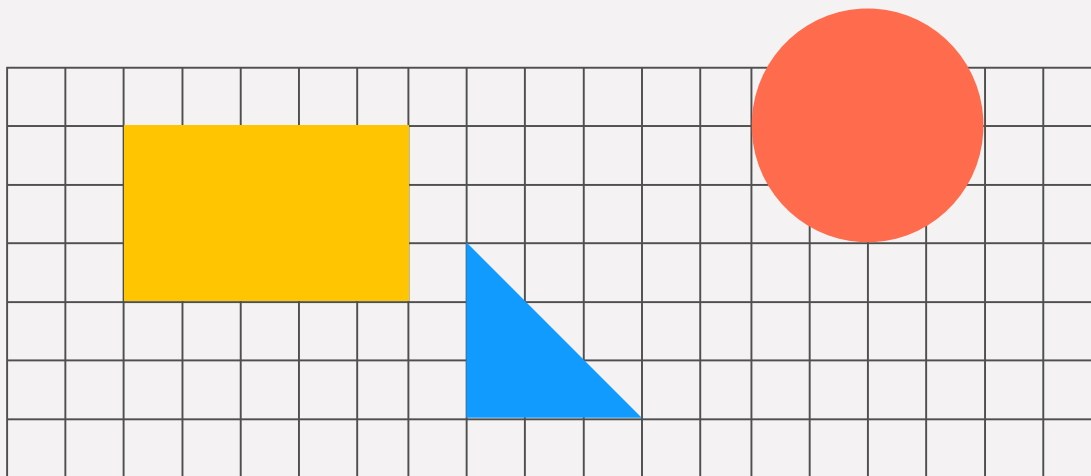


Unidad 5

Funciones y Procedimientos



Funciones

Una **función** es un bloque de código que **devuelve un valor**.

```
tipo_de_dato nombre_funcion(parámetros) {  
    // código  
    return valor;  
}
```



```
#include <stdio.h>
// Función que suma dos números enteros
int sumar(int a, int b) {
    return a + b;
}

int main() {
    int resultado = sumar(5, 3);
    printf("La suma es: %d\n", resultado);
    return 0;
}
```


Procedimiento

Un **procedimiento** es una función que **NO devuelve valor**. En C, se usa el tipo `void`.

```
#include <stdio.h>

// Procedimiento sin valor de retorno
void saludar() {
    printf("¡Hola, bienvenido!\n");
}

int main() {
    saludar(); // Llamamos al procedimiento
    return 0;
}
```



```
#include <stdio.h>
```

```
int cuadrado(int x) {  
    return x * x;  
}
```

```
void mostrarResultado(int numero, int resultado) {  
    printf("El cuadrado de %d es %d\n", numero, resultado);  
}
```

```
int main() {  
    int n = 4;  
    int res = cuadrado(n);  
    mostrarResultado(n, res);  
    return 0;  
}
```

Pasaje de Parámetros




1. Pasaje por valor (el más común en C)

La función **recibe una copia** del valor original. Si la función modifica el valor, **no afecta** al original.

2. Pasaje por referencia (usando punteros)

La función **recibe la dirección** de la variable, por lo tanto, **puede modificar el valor original**.



```
/*Pasaje por Valor*/
```

```
#include <stdio.h>
```

```
void duplicar(int x) {
```

```
    x = x * 2;
```

```
    printf("Dentro de la funcion: %d\n", x);
```

```
}
```

```
int main() {
```

```
    int numero = 5;
```

```
    duplicar(numero);
```

```
    printf("Fuera de la funcion: %d\n", numero);
```

```
    return 0;
```

```
}
```

*/*Pasaje por Referencia*/*

`#include <stdio.h>`

`void duplicar(int *x) {`

`*x = *x * 2;`

`printf("Dentro de la funcion: %d\n", *x);`

`}`

`int main() {`

`int numero = 5;`

`duplicar(&numero);`

`printf("Fuera de la funcion: %d\n", numero);`

`return 0;`

`}`