

# Punteros

## Qué es un puntero?

Un puntero en C es una variable que almacena la dirección de memoria de otra variable. En lugar de contener un valor directo, un puntero contiene la dirección donde está guardado ese valor.

### ¿Qué significa esto?

Imagina que las variables son como cajas, cada caja tiene una dirección única. Un puntero es como la ficha con la dirección de una de esas cajas.

## Cómo se declara un puntero?

Se usa el operador “ \* ”.

**Ejemplo:** int

```
*puntero_a_entero;
```

## Cómo asignamos la dirección a un puntero?

Se usa el operador “ & ”.

**Ejemplo:**

```
int a = 5;
```

```
int *puntero_a_entero;
```

```
puntero_a_entero = &a;
```

## Cómo accedemos al valor apuntado por un puntero?

Usando **desferencia** = usando el operador “ \* ”.

**Ejemplo:**

```
int a = 5;
```

```
int *puntero_a_entero;
```

```
puntero_a_entero = &a;
```

```
printf("%d\n", *puntero_a_entero); // Imprime 5
```

## Paso por Valor

En el paso por valor, la función recibe una copia del valor de la variable, por lo que cualquier modificación dentro de la función no afecta a la variable original.

### Ejemplo:

```
#include <stdio.h>

void duplicar(int n) {
    n = n * 2;
}

int main() {
    int numero = 5;
    duplicar(numero);
    printf("Valor después de duplicar: %d\n", numero); // Imprime 5
    return 0;
}
```

## Paso por referencia (con Punteros)

En el paso por referencia, se pasa la dirección de memoria de la variable, permitiendo modificar su valor dentro de la función.



### Ejemplo:

```
#include <stdio.h>

void duplicar(int *n) {
    *n = *n * 2;
}

int main() {
    int numero = 5;
    duplicar(&numero);
    printf("Valor después de duplicar: %d\n", numero); // Imprime 10
    return 0;
}
```

### Diferencias Clave

Método	Se pasa...	Se modifica el valor original
<b>Por valor</b>	Una copia de la variable	 No
<b>Por referencia</b>	La dirección de memoria	 Sí