

Apuntadores

- Es una variable que contiene una dirección de memoria.
- Generalmente, esa dirección es la dirección de otra variable de memoria.
- Si una variable contiene la dirección de otra variable, entonces se dice que la primera apunta a la segunda.

Apuntadores

Declaración:

```
tipo_dato *nombre_variable;
```

Operadores

- & Devuelve la dirección de memoria de su operando. Por ejemplo, en la línea $m = \& \text{cuenta}$; m guarda la dirección de la variable *cuenta*.
- * Devuelve el valor de la variable localizada en la dirección de memoria a la que se apunta. Por ejemplo: si a la variable *cuenta* le asignamos el valor de 1200, entonces $*m$ tendrá el valor de 1200.

Apuntadores

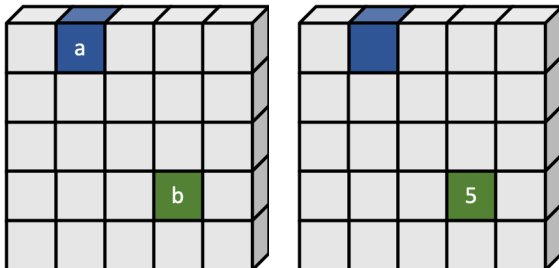
- El **tipo_dato** del apuntador determina el tipo de variables a las que puede apuntar el apuntador.

Apuntadores

Por ejemplo:

- Supongamos que existen el apuntador `a` y la variable `b`.

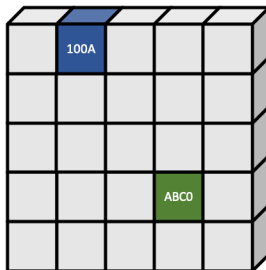
```
int *a;  
int b=5;
```



Apuntadores

Por ejemplo:

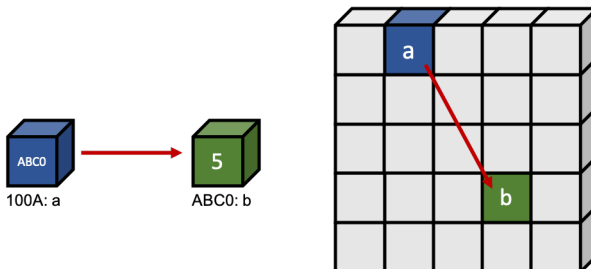
- Ahora digamos que las variables `a` y `b` están ubicadas en las direcciones `100A` y `ABCO`, respectivamente.



Apuntadores

Por ejemplo:

- Si se ejecuta la línea $a = \&b$; entonces la celda 100A contendrá la dirección de b.



Apuntadores

Los apuntadores se utilizan en situaciones en las que pasar valores es difícil o indeseable, como podrían ser los siguientes casos:

- Para regresar más de un valor de una función.
- Pasar arreglos fácilmente de una función a otra.
- Manipular arreglos más fácilmente, manipulando punteros a ellos, o a partes de ellos, en vez de mover los arreglos en sí.
- Crear estructuras de datos complejas, como son las listas encadenadas a y árboles binarios, donde una estructura de datos debe tener referencias a otras estructuras de datos.
- Para comunicar información acerca de la memoria.

Ejemplo: manejo de apuntadores

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

int main()
{
    int *direccion;
    int edad=0;
    printf("Cuantos años tienes");
    scanf("%d", &edad);
    direccion = &edad;
    printf("\nTu edad esta guardada en la posicion de memoria: %p",
           direccion);
    printf("\nTu edad es: %d", *direccion);
    printf("\nEl apuntador a tu edad esta guardado en la posicion de memoria
           : %p", &direccion);
    getch();
}
```


Apuntadores y vectores

- Un vector es un apuntador que contiene la dirección de la primera (0-ésima) posición del vector.

```
int v[5] = {5, 10, 15, 20, 25};  
int *dir;
```

1000	1004	1008	1012	1016
5	10	15	20	25
v[0]	v[1]	v[2]	v[3]	v[4]

- $v = 1000$

Apuntadores y vectores

- Por lo tanto las siguientes expresiones son equivalente:
 - $\text{dir} = \&\text{v}[0] \leftrightarrow \text{dir} = \text{v}$
 - $\text{x} = *\text{dir} \leftrightarrow \text{x} = \text{v}[0]$
 - $*(\text{v} + 1) \leftrightarrow \text{v}[1]$
 - $*(\text{v} + \text{i}) \leftrightarrow \text{v}[\text{i}]$

Aritmética de apuntadores

```
int v[5], *dir;  
dir = v;
```

- $\text{dir} \rightarrow$ Apunta a la posición inicial del vector.
- $\text{dir} + 0 \rightarrow$ Apunto a la posición inicial del vector.
- $\text{dir} + 1 \rightarrow$ Apunta a la segunda posición del vector.
- $\text{dir} + i \rightarrow$ Apunta a la posición $i + 1$ del vector.

```
dir = &v[4];
```

- $\text{dir} \rightarrow$ Apunta a la última posición del vector.
- $\text{dir} - 1 \rightarrow$ Apunta a la penúltima posición del vector.
- $\text{dir} - i \rightarrow$ se refiere a la posición $4 - i$ en v .

Ejemplo: manejo de vectores con apuntadores

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

void imprimir(int *v)
{
    int i=0;
    do
    {
        printf("%d \n", *(v+i));
        i++;
    }while(i<5);
}

void main()
{
    int vector[5] = {3,0,2,2,4};
    imprimir(vector);
    getch();
}
```