



# ARRAYS

# MULTIDIMENSIONAL

Oscar Paniagua @2023

# array@multidimensional

Arrays multidimensionales, se definen como **arrays de arrays**

```
tipo nombre[tamaño1][tamaño2]... tamañoN;
```

Ejemplos:

```
int a[2][3];    // array de dos dimensiones (2D)  
int b[2][3][4]; // array de tres dimensiones (3D)
```

# array@multidimensional

El tamaño de un array, se calcula multiplicando el tamaño de todas las dimensiones.

Ejemplos:

tamaño de `a[2][3]`  $\Rightarrow 2 \times 3 = 6$  en cantidad de elementos  
 $\Rightarrow 6 \times 4 = 24$  en bytes

tamaño de un elemento

`sizeof(a[0][0]);` // tamaño de un elemento (tipo)

```
printf("Cantidad de elementos %lu\n", sizeof(arr) / sizeof(arr[0][0]));
```

# array@multidimensional

Ejemplos de declaración de arrays:

1D                    `int   arr[5];`

2D                    `int   arr[3][5];`

	col 0	col 1	col 2	col 3	col 4
fila 0					

	col 0	col 1	col 2	col 3	col 4
fila 0					
fila 1					
fila 2					

# array@multidimensional

```
int arr[2][3];
```

Se puede visualizar como una matriz de 2\*3  
(filas\*columnas)

Podemos Visualizar:

	col 0	col 1	col 2
fila 0			
fila 1			

es una Matriz 2 x 3

Se accede con dos subíndices:

- el primer subíndice es la **fila**
- el segundo subíndice es la **columna**

arr[0][0] - primer elemento  
arr[0][1] - segundo elemento  
arr[0][2] - tercer elemento  
arr[1][0] - cuarto elemento  
arr[1][1] - quinto elemento  
arr[1][2] - sexto elemento

```
printf("El valor del cuarto elemento es %d\n", arr[1][0]);
```

# array@multidimensional

```
int arr[2][3];
```

```
// Si intentas acceder a un elemento más allá de ROW y COL válidos,  
// el compilador de C no mostrará ningún tipo de mensaje de error;  
// en su lugar, se imprimirá un valor basura o se producirá un  
// excepción de memoria segmentation fault  
// Es responsabilidad del programador manejar los límites de acceso  
// del array
```

```
printf("El valor del cuarto elemento es %d\n", arr[1][3]);
```

# array@multidimensional

```
#define ROW 2
```

```
#define COL 3
```

```
// es una buena practica de programacion utilizar  
// constantes para declarar arrays
```

```
int arr_ok[ROW][COL];
```

```
// ERROR. no esta permitido
```

```
int i = 4, j = 6;
```

```
int arr_error[i][j];
```