



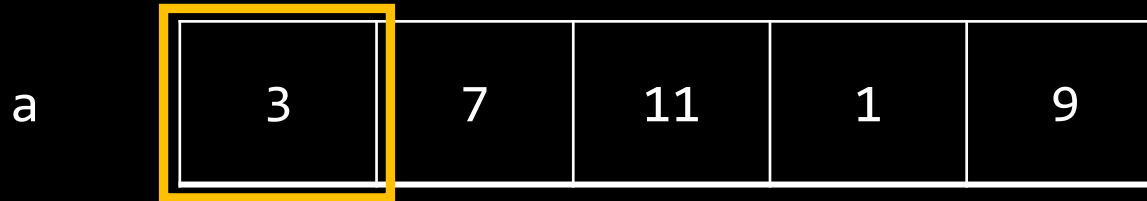
# ARRAYS

# DECLARACIÓN

Oscar Paniagua @2023

# array de una dimensión 1D

El array mas simple que podemos imaginar es de una dimensión....



- un array es un bloque grande de memoria
- podemos verlo como una “fila”, dividida en pequeños bloques
- donde cada bloque es capaz de almacenar un valor
- y todos los valores tienen que ser del mismo tipo
- nos podemos imaginar que un bloque del array es una variable
- y el array es una colección de variables

# array@declaración

**tipo** nombre[tamaño];

- El tamaño es cantidad de elementos de “tipo”
- El tamaño puede ser cualquier entero positivo
- El tamaño se define en tiempo de compilación  
(\*excepción C99, no recomendada, no lo consideramos una buena practica de programación)

- Ejemplo: un array de enteros  
int valores[5];



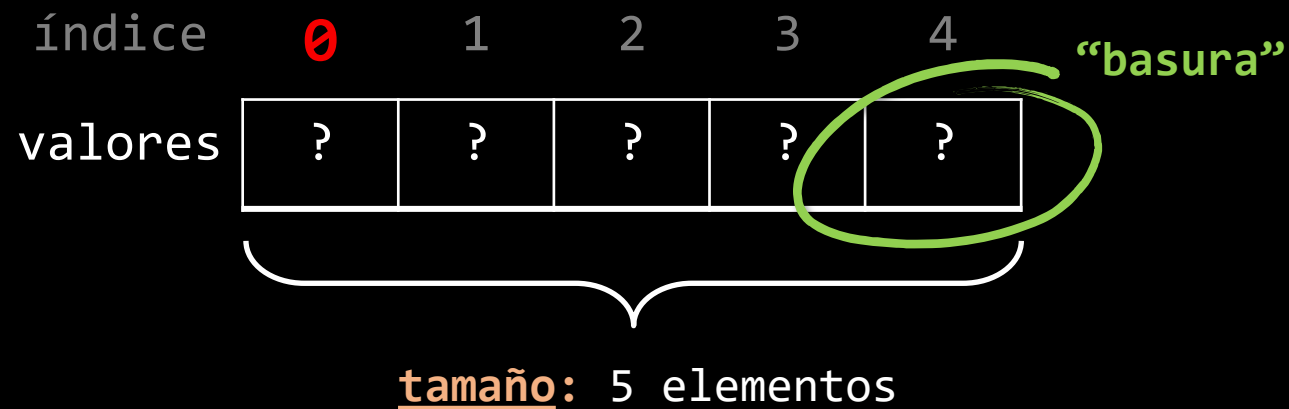
El compilador reservará un bloque continuo de memoria de tamaño = 5\*sizeof(int)

# array@declaración

**tipo** nombre[tamaño];

- El tamaño es cantidad de elementos de “tipo”
- El tamaño puede ser cualquier entero positivo
- El tamaño se define en tiempo de compilación  
(\*excepción C99, no recomendada, no lo consideramos una buena practica de programación)

- Ejemplo: un array de enteros  
`int valores[5];`



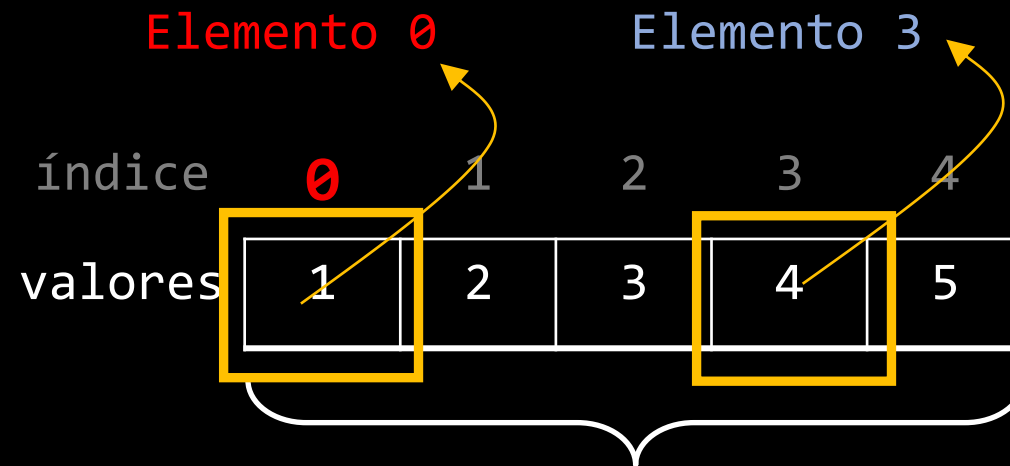
El compilador reservará un bloque continuo de memoria de tamaño =  $5 * \text{sizeof}(\text{int})$

# array@declaración e inicialización

```
tipo nombre[tamaño] = { valor, valor, ...} ;
```

- Ejemplo:

```
int valores[5] = { 1, 2, 3, 4, 5};
```



\* índice "0-based int"

# array@declaración e inicialización

```
tipo nombre[] = { valor, valor, ...} ;
```

- Ejemplo:

```
int valores[] = { 1, 2, 3, 4, 5};
```

