



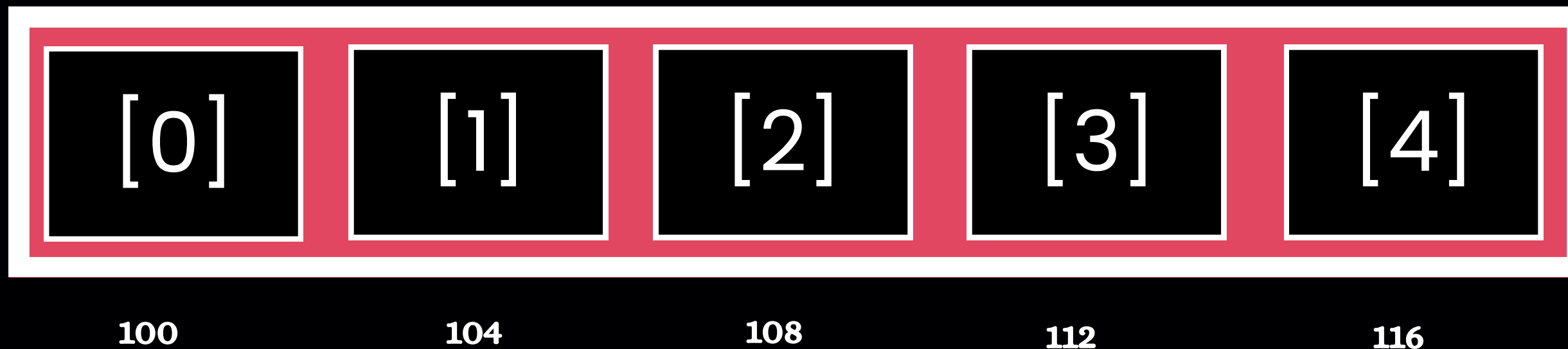
LISTAS Y ARREGLOS



ARREGLOS ESTATICOS (STATIC ARRAYS)

Los arreglos con colecciones de datos que nos permiten guardar multiples datos del mismo tipo en una misma variable.

```
int miArreglo[5]
```

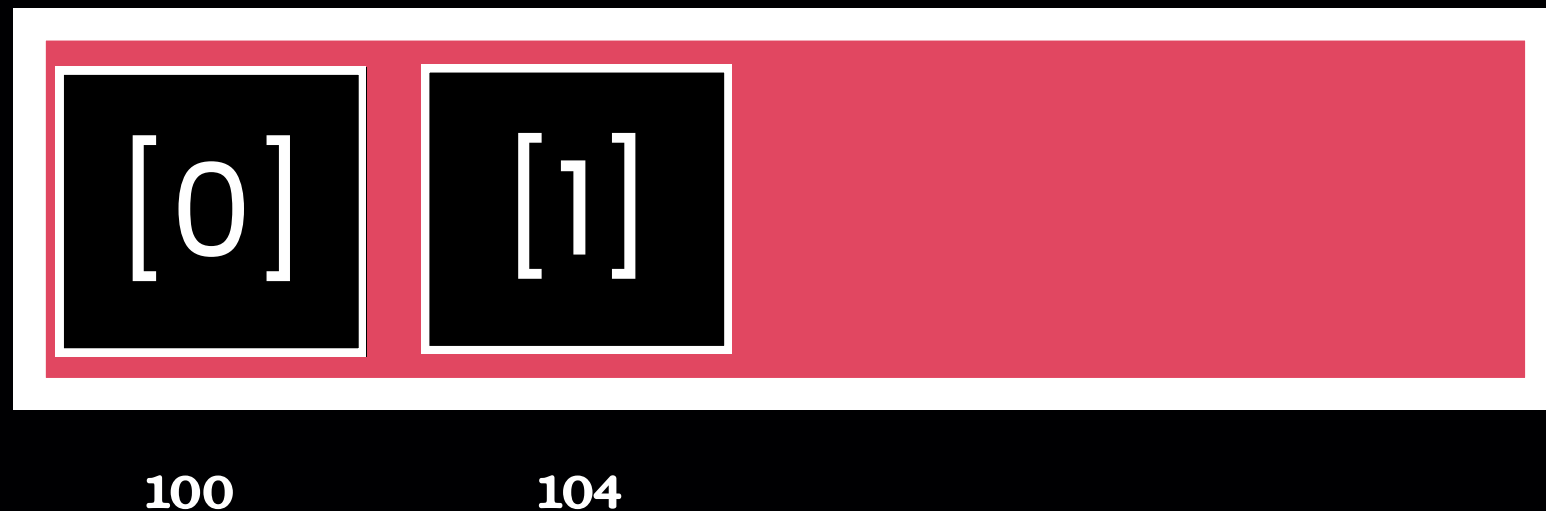


- Cada elemento en el arreglo es guardado en un bloque de memoria contigua.
- El tamaño de un arreglo permanece constante luego de su declaración. (Arreglos estaticos)
- Podemos acceder a los elementos del arreglo a través de sus indices.

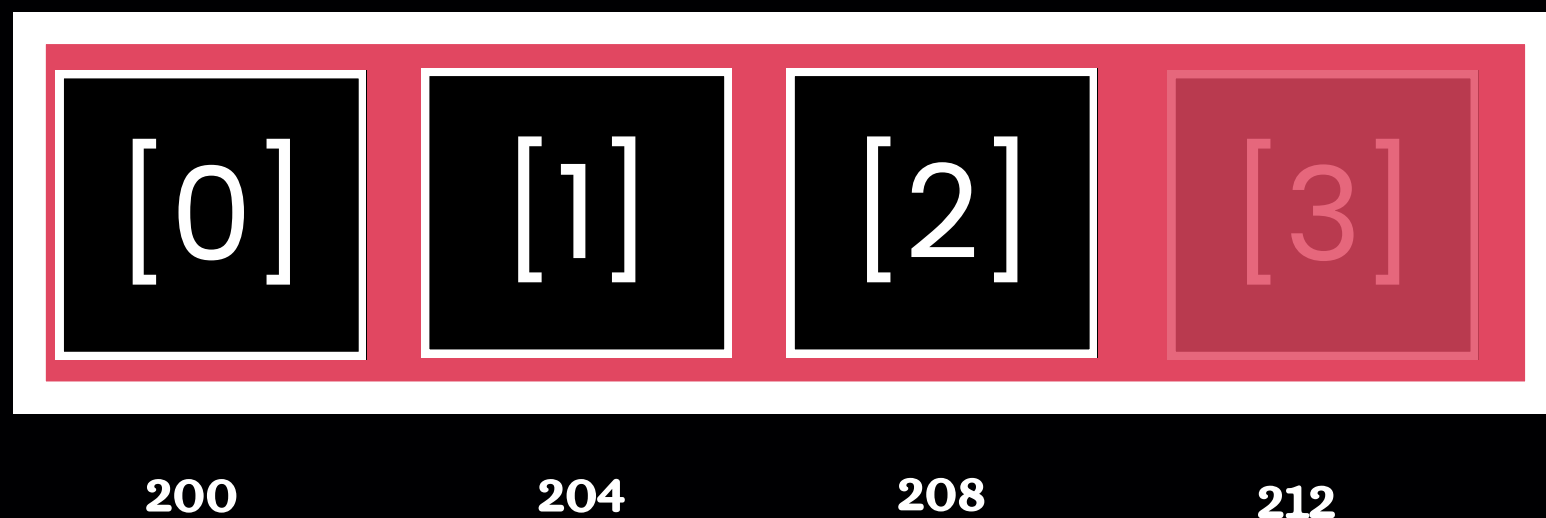
ARREGLOS DINAMICOS (DYNAMIC ARRAYS)

Los arreglos dinámicos son estructuras similares a los arreglos ordinarios, con la diferencia de que estos pueden ajustar su tamaño durante la ejecución del programa.

```
int* miArreglo = (int*)malloc(2*sizeof(int))
```



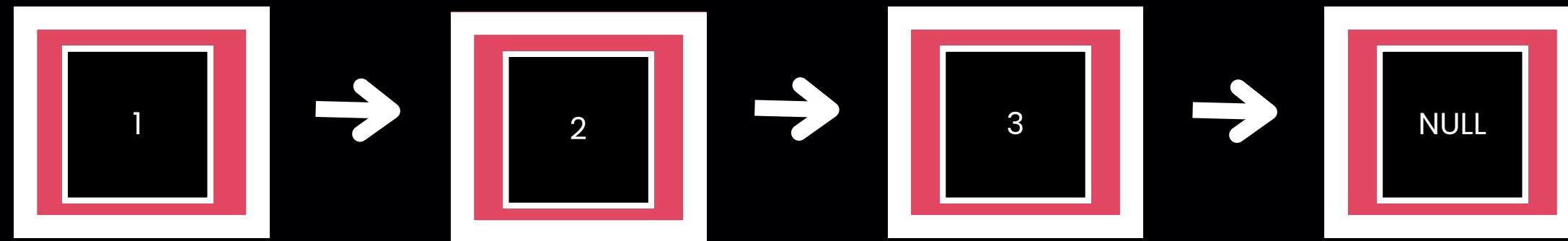
```
miArreglo = realloc(miArreglo,4*sizeof(int))
```



- Cada elemento en el arreglo es guardado en un bloque de memoria contigua.
- Las implementaciones de los arreglos dinámicos poseen un valor llamado capacidad.
- La capacidad es el tamaño de memoria asignada al arreglo. (Distinto del tamaño del arreglo)
- La capacidad se reajusta cuando el arreglo requiera mas elementos de lo asignado.

LISTAS

Las listas son un tipo de estructura de datos en el cual sus elementos no se encuentran asignados por espacios de memorias contiguas, sino que cada elemento es conectado a otro a traves de su direccion de memoria.



- Cada nodo contendrá un valor y un conjunto de variables que apunten a la dirección de memoria de sus vecinos
- La dirección de memoria de cada elemento es aleatoria.
- El tamaño de una lista puede ser reajustado sin tener que asignar grandes espacios de memoria.

LISTAS (STL LIST)

C++ posee la coleccion de listas de la libreria stl cuya implementacion se basa en listas doblemente enlazadas

```
• List<tipoDato> nombreLista{elementos};
```

ACCEDER POR INDICE

O(N)

Si se conoce el nodo:

INSERTAR ELEMENTO

o(1)

ELIMINAR ELEMENTO

o(1)

Operation	Time Complexity	Space Complexity
front()	O(1)	O(1)
back()	O(1)	O(1)
empty()	O(1)	O(1)
insert()	O(1)	O(1)
erase()	O(N)	O(1)
remove()	O(N)	O(1)
push_back()	O(1)	O(1)
pop_back()	O(1)	O(1)
push_front()	O(1)	O(1)
pop_front()	O(1)	O(1)
size()	O(1)	O(1)

ARREGLOS EN C/C++/PYTHON

ESTATICOS

Declaracion de un arreglo estatico en c/c++

```
tipoDato nombreArreglo[tamañoArreglo];
```

- `int edades[10];`
- `char caracteres[15];`
- `int DP[1e6];`

ACCEDER POR INDICE

$O(1)$

BUSQUEDA

$O(N)$

DINAMICOS

C++ posee la coleccion de vectores de la libreria stl cuya implementacion se basa en arreglos dinamicos al igual que las listas en python.

- `vector<tipoDato> nombreVector(tamañoArreglo);`
- `nombreLista = []`

INSERCIÓN AL FINAL

$O(1)$

INSERTAR/ELIMINAR

$O(N)$

REAJUSTAR TAMAÑO

$O(N)$

<https://wiki.python.org/moin/TimeComplexity>

- `vector<tipoDato> nombreVector(tamañoArreglo);`

Operation	Time Complexity	Space Complexity
<code>push_back()</code>	$O(1)$	$O(1)$
<code>pop_back()</code>	$O(1)$	$O(1)$
<code>size()</code>	$O(1)$	$O(1)$
<code>empty()</code>	$O(1)$	$O(1)$
<code>erase()</code>	$O(N)$	$O(1)$
<code>clear()</code>	$O(N)$	$O(1)$
<code>begin()</code>	$O(1)$	$O(1)$
<code>end()</code>	$O(1)$	$O(1)$