

# IndexOf

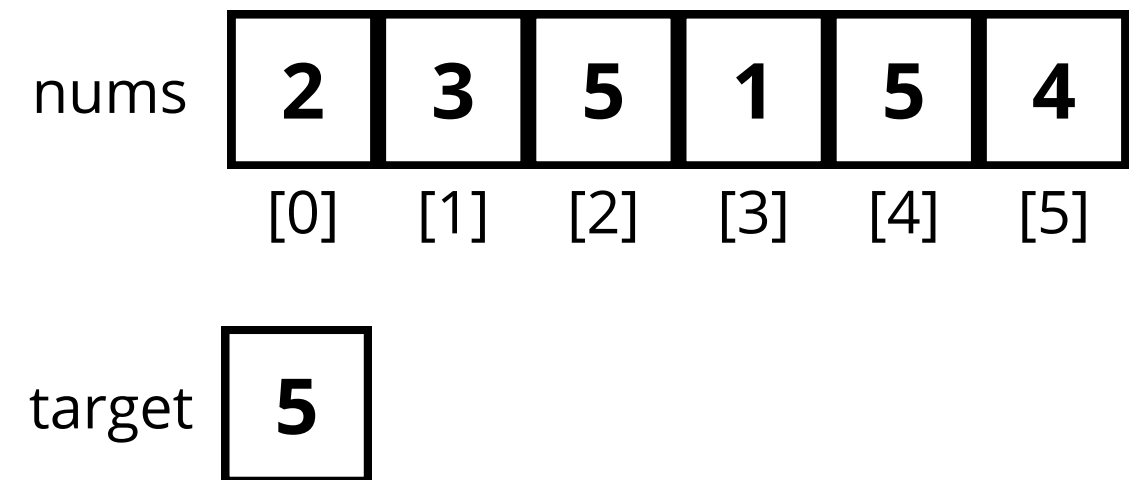
Donat un array d'enters volem saber la primera posició on es troba el número 5.

El programa ha de mostrar la posició del número a dins de l'array.

Si no es troba el número, s'ha de mostrar un -1.

```
int[] nums = new int[6]{2,3,5,1,5,4};
```

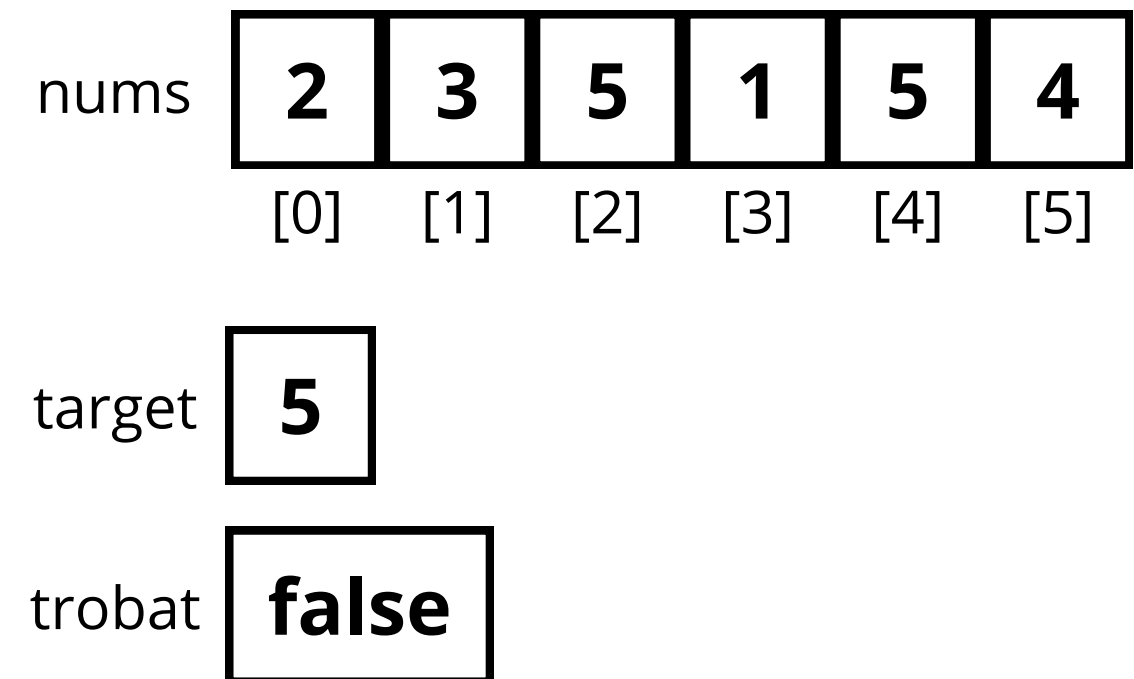
```
int target = 5;
```



# IndexOf

Com que es tracta d'una cerca necessitem una variable booleana que inicialitzem a false.

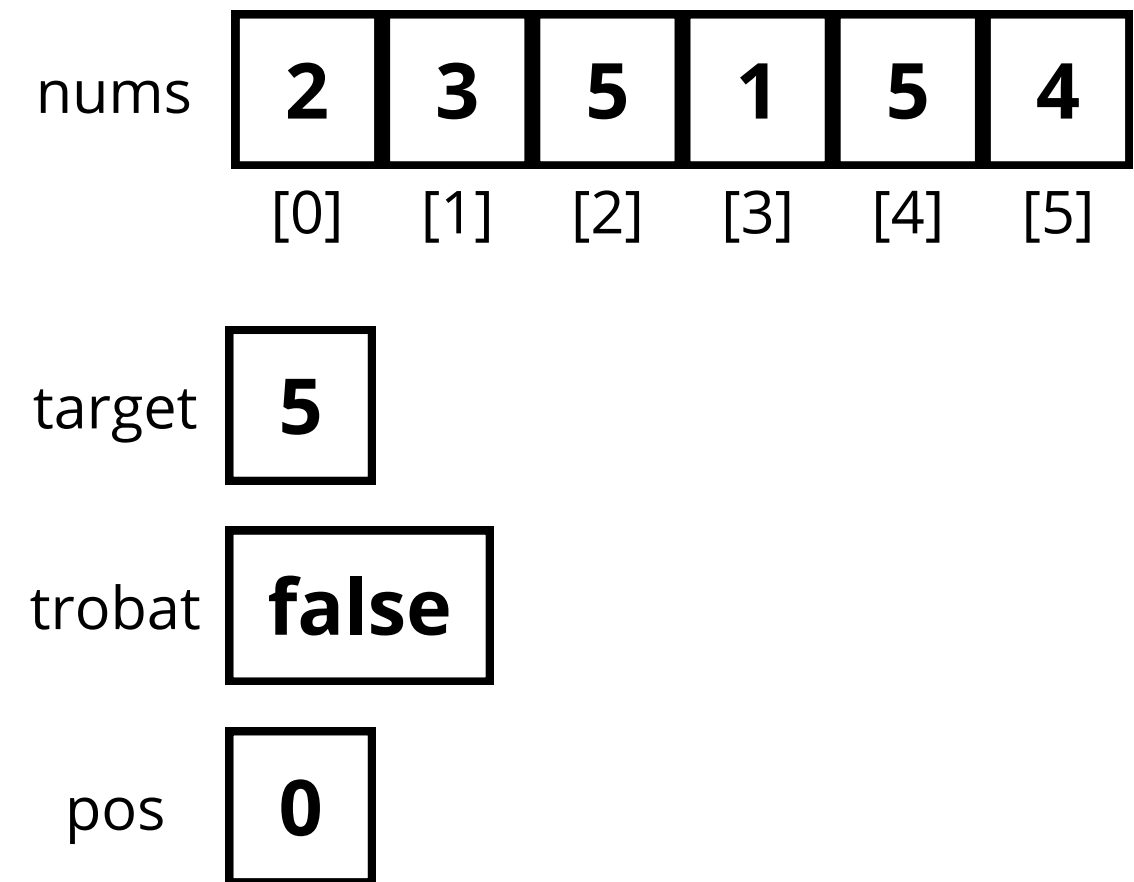
```
int[] nums = new int[6]{2,3,5,1,5,4};  
int target = 5;  
bool trobat = false;
```



# IndexOf

També necessitem una variable que ens indiqui la posició de la cerca.

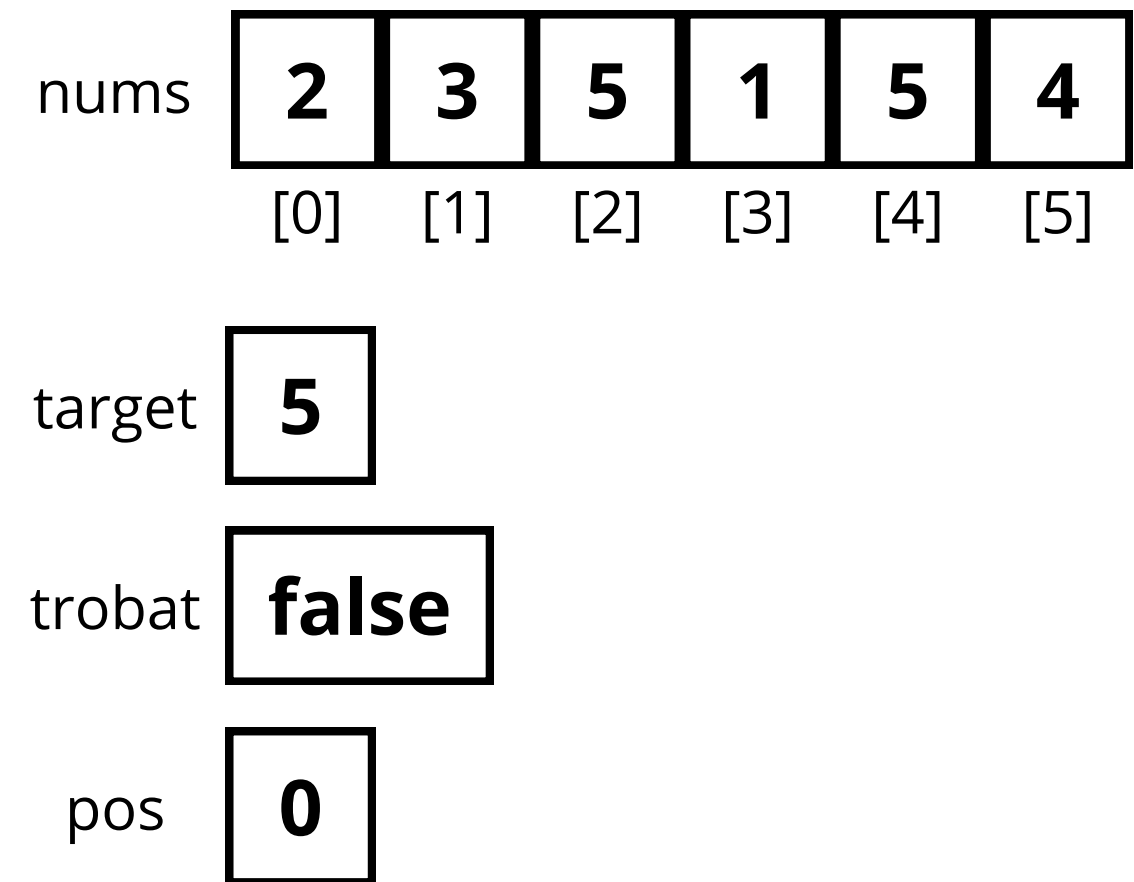
```
int[] nums = new int[6]{2,3,5,1,5,4};  
int target = 5;  
bool trobat = false;  
int pos = 0;
```



# IndexOf

Apliquem l'esquema de cerca.

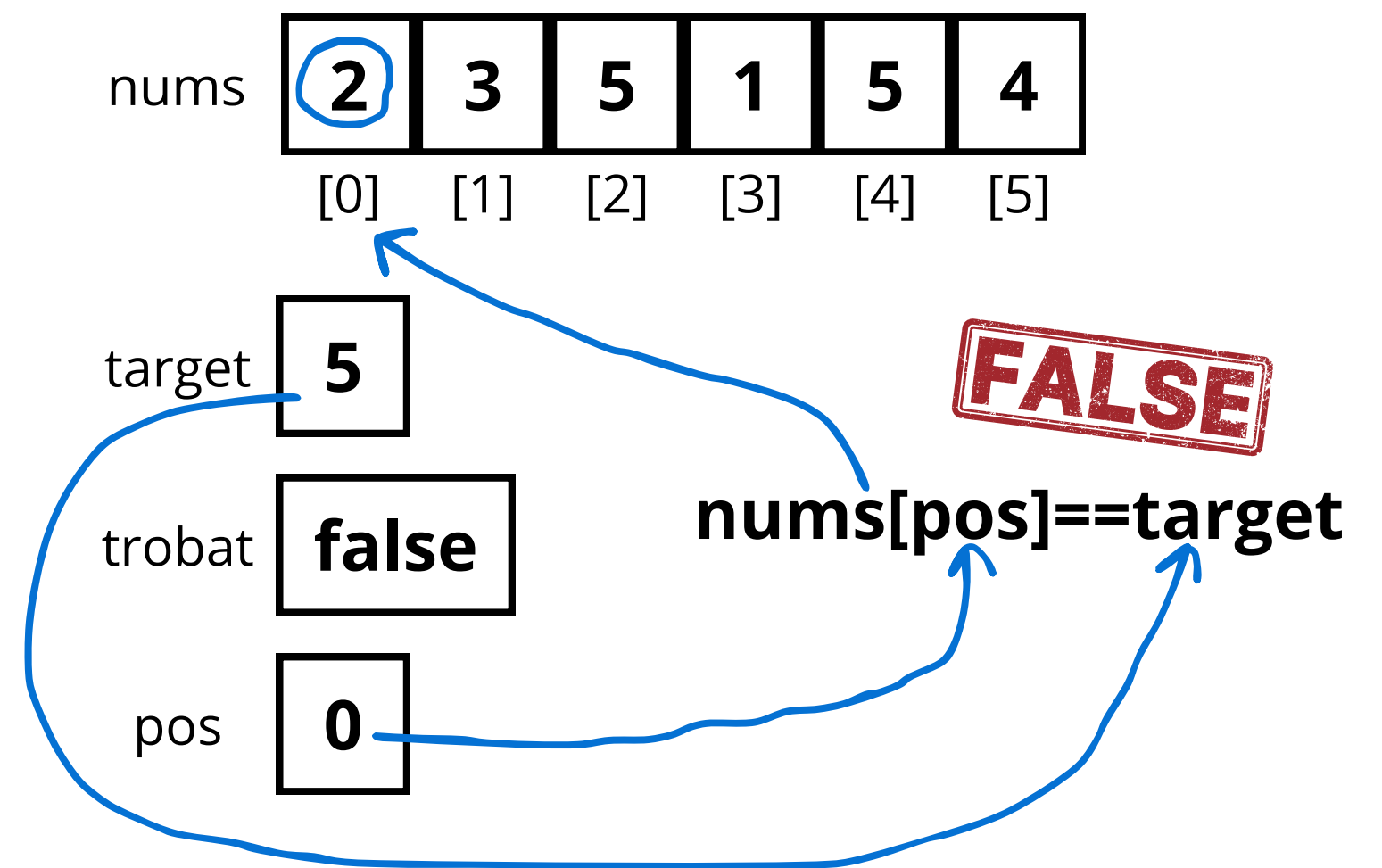
```
int[] nums = new int[6]{2,3,5,1,5,4};  
int target = 5;  
bool trobat = false;  
int pos = 0;  
while(!trobat && pos<nums.Length){  
    ...  
}
```



# IndexOf

Primera iteració del bucle while amb pos igual a zero.  
De moment no s'ha trobat l'element.

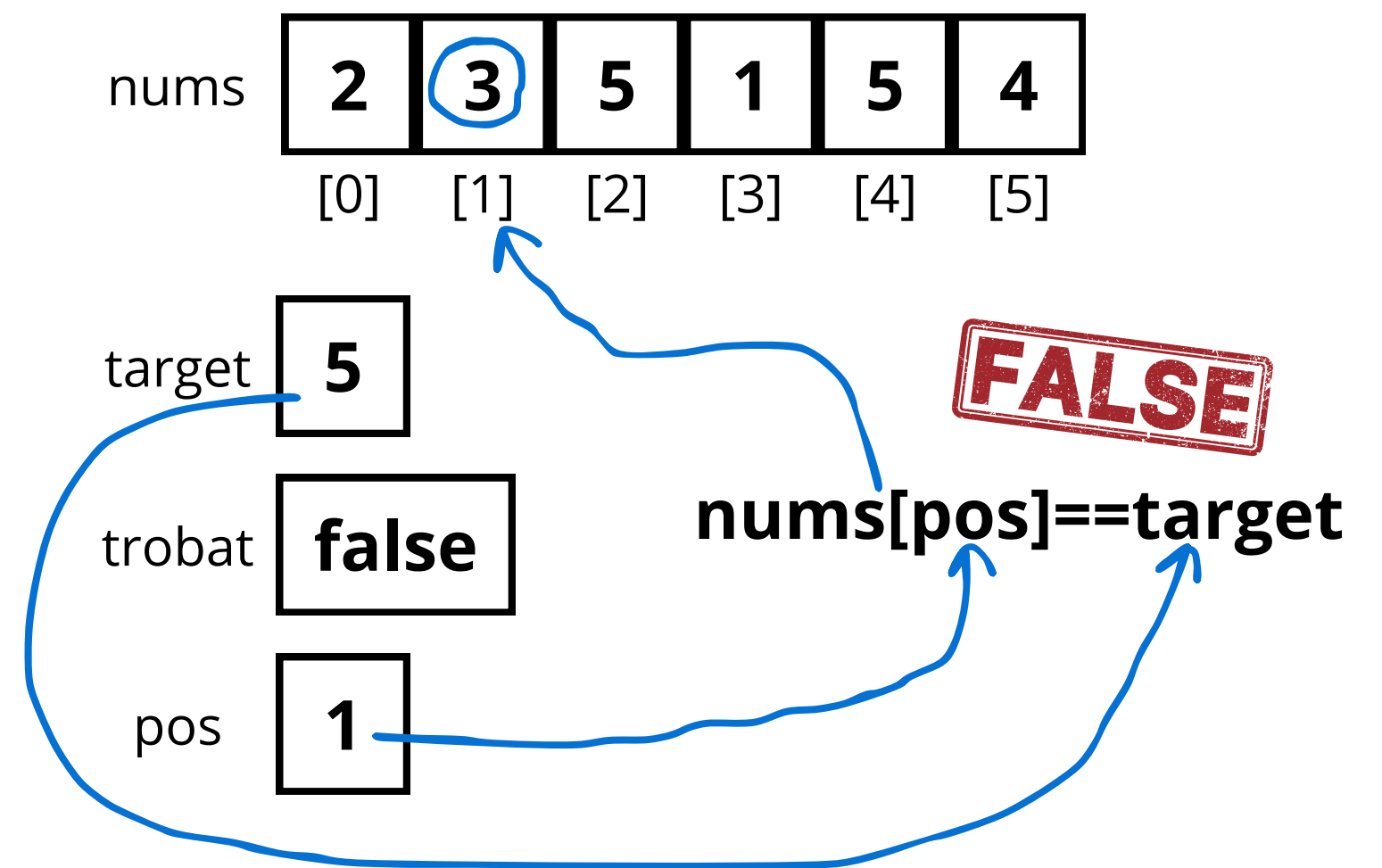
```
int[] nums = new int[6]{2,3,5,1,5,4};  
int target = 5;  
bool trobat = false;  
int pos = 0;  
while(!trobat && pos<nums.Length){  
    if(nums[pos]==target) trobat=true;  
    else pos++;  
}
```



# IndexOf

Segona iteració del bucle while amb pos igual a zero.  
De moment no s'ha trobat l'element.

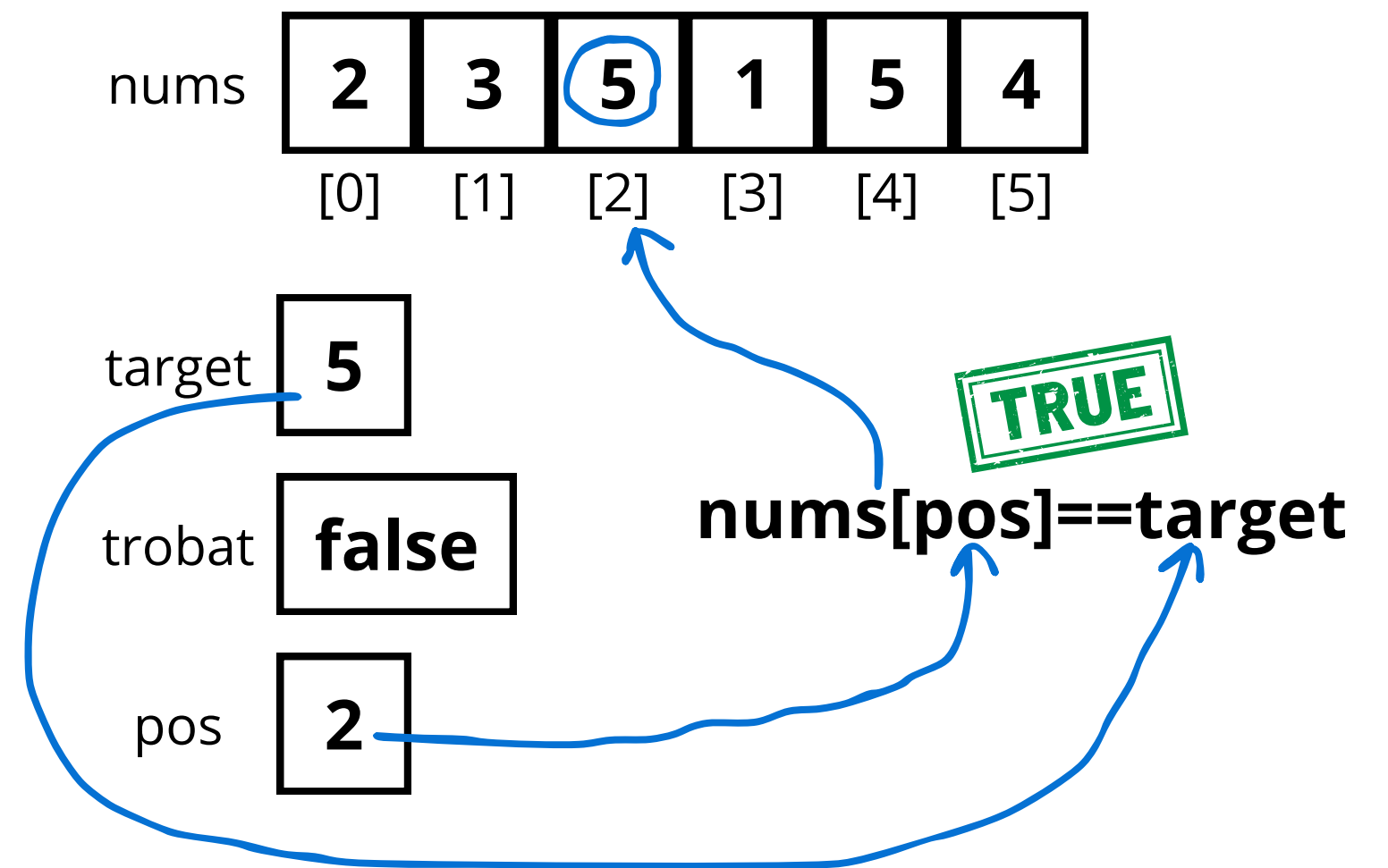
```
int[] nums = new int[6]{2,3,5,1,5,4};  
int target = 5;  
bool trobat = false;  
int pos = 0;  
while(!trobat && pos<nums.Length){  
    if(nums[pos]==target) trobat=true;  
    else pos++;  
}
```



# IndexOf

Tercera iteració del bucle while amb pos igual a zero.  
En aquest cas s'ha trobat l'element.

```
int[] nums = new int[6]{2,3,5,1,5,4};  
int target = 5;  
bool trobat = false;  
int pos = 0;  
while(!trobat && pos<nums.Length){  
    if(nums[pos]==target) trobat=true;  
    else pos++;  
}
```



# IndexOf

En finalitzar mostrem la posició de l'element.

```
int[] nums = new int[6]{2,3,5,1,5,4};  
int target = 5;  
bool trobat = false;  
int pos = 0;  
while(!trobat && pos<nums.Length){  
    if(nums[pos]==target) trobat=true;  
    else pos++;  
}  
if(!trobat) CWL(-1);  
else CWL(pos);
```

nums

2	3	5	1	5	4
[0]	[1]	[2]	[3]	[4]	[5]

target

5
---

trobat

true
------

pos

2
---

>_
2