# 자바스크립트 메모리 모델

원시 타입과 참조 타입 데이터의 저장 방식 이해하기

참고자료

https://medium.com/@ethannam/javascripts-memory-model-7c972cd2c239

# 물음

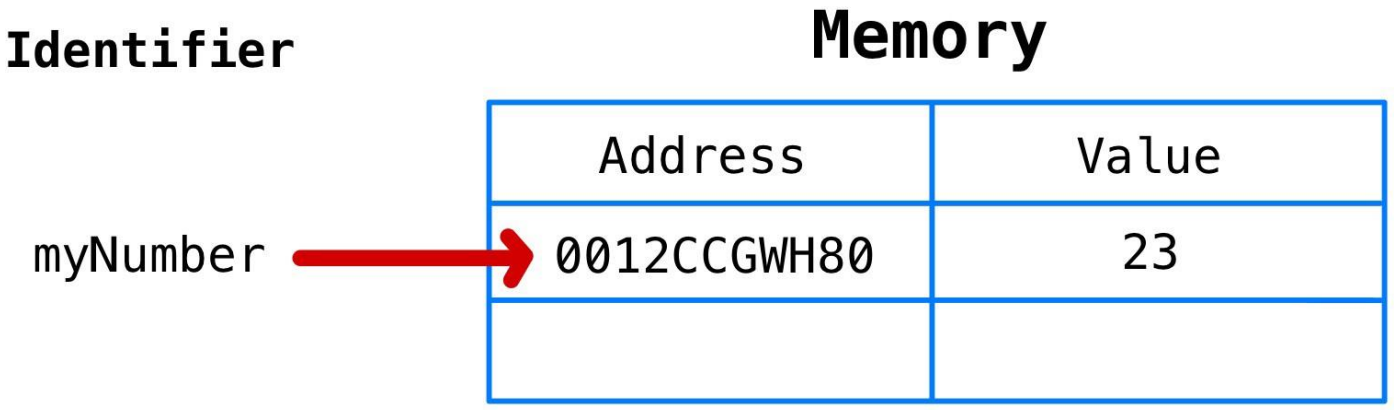
(var, let, const) myArray = {}

- const로 생각하고 변경될 가능성이 있는 경우 let으로 정의

# 원시타입

```
let myNumber = 23
```

- myNumber라는 식별자를 생성
- 메모리의 주소를 할당(런타임 시)
- 23이라는 숫자를 할당된 메모리 공간에 저장




"myNumber equals 23"
more technically → " myNumber equals the memory address that holds the value 23"

# 원시타입

```
let newVar = myNumber
```

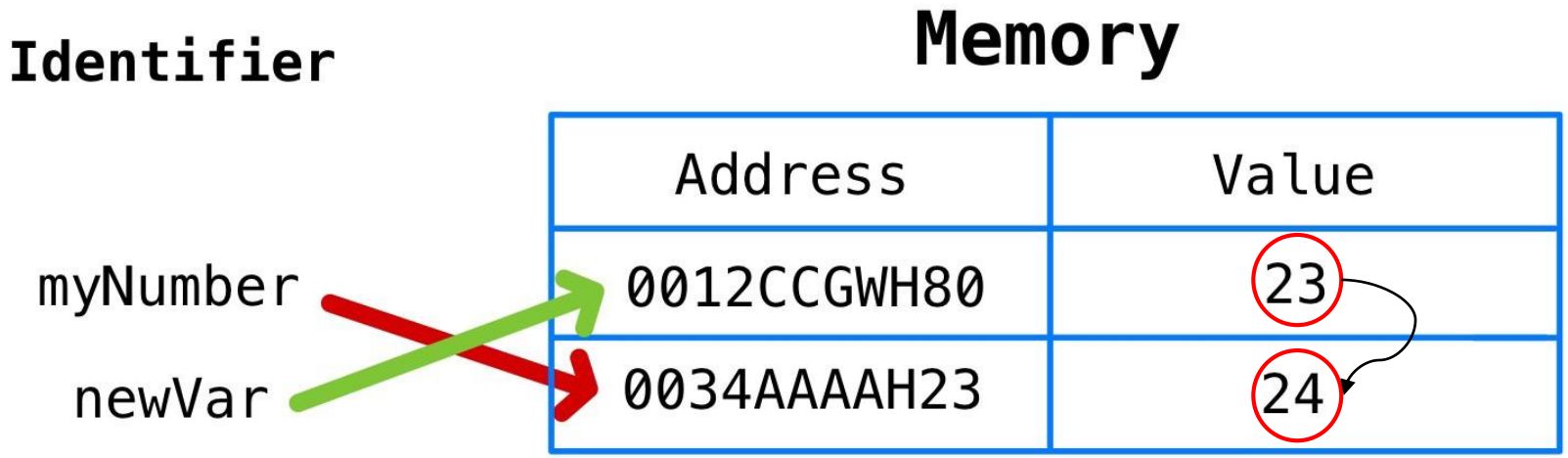
- myNumber는 "0012CCGWH80"와 같기 때문에 새로운 변수 newVar도 23이라는 변수를 저장하는 메모리 공간의 주소 "0012CCGWH80"와 같음


- 따라서 우리는 일반적으로 " newVar는 23과 같다"라고 말할 수 있음

# 원시타입

```
myNumber = myNumber + 1
```

- myNumber는24라는 값을 가짐
- 24는 원래 공간에 있는 23이 24로 변한 것인가? 아니면 새로운 공간에 24라는 값이 생긴 것인가?

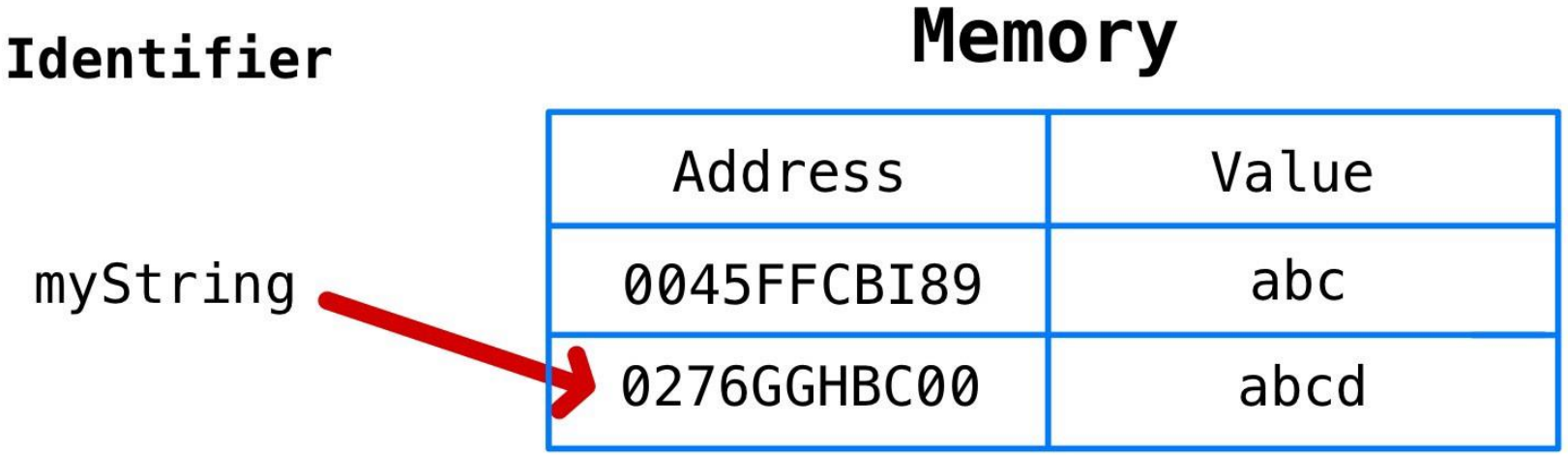



원시(primitive)타입은
값이 변하지 않는다라는 의미
(immutable)

# 원시타입

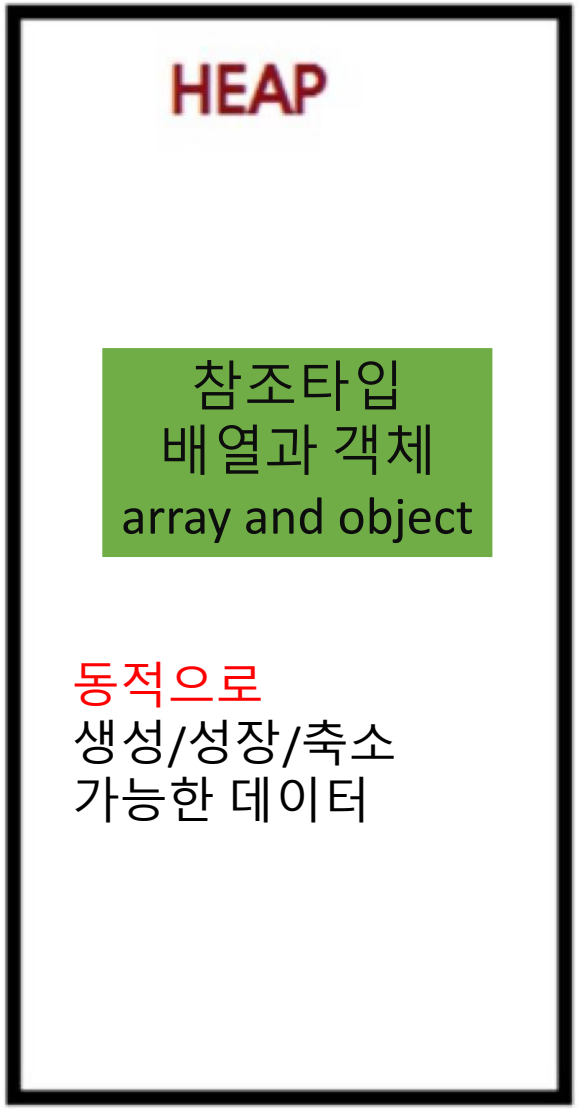
```
let myString = 'abc'
myString = myString + 'd'
```

- "abc"라는 문자열에 "d"라는 문자열이 붙여졌다고 생각할 수 있으나 "abcd"라는 새로운 원시타입 값이 할당된 것



# 메모리 모델

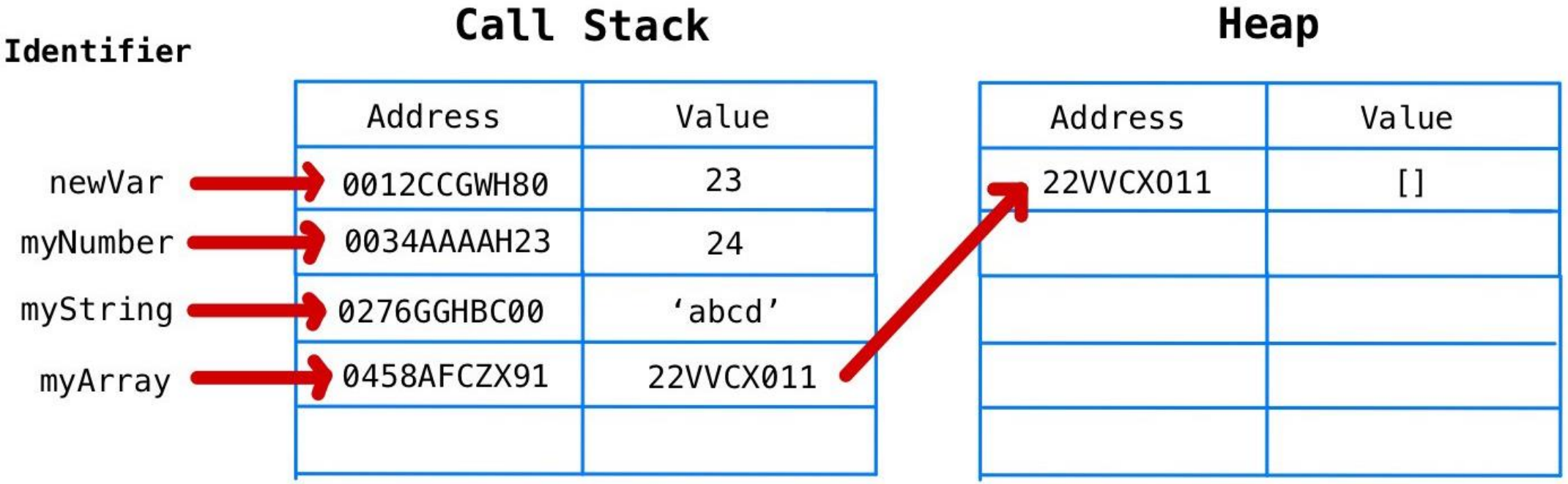
CALL STACK은 스택영역으로 생각하면 됨

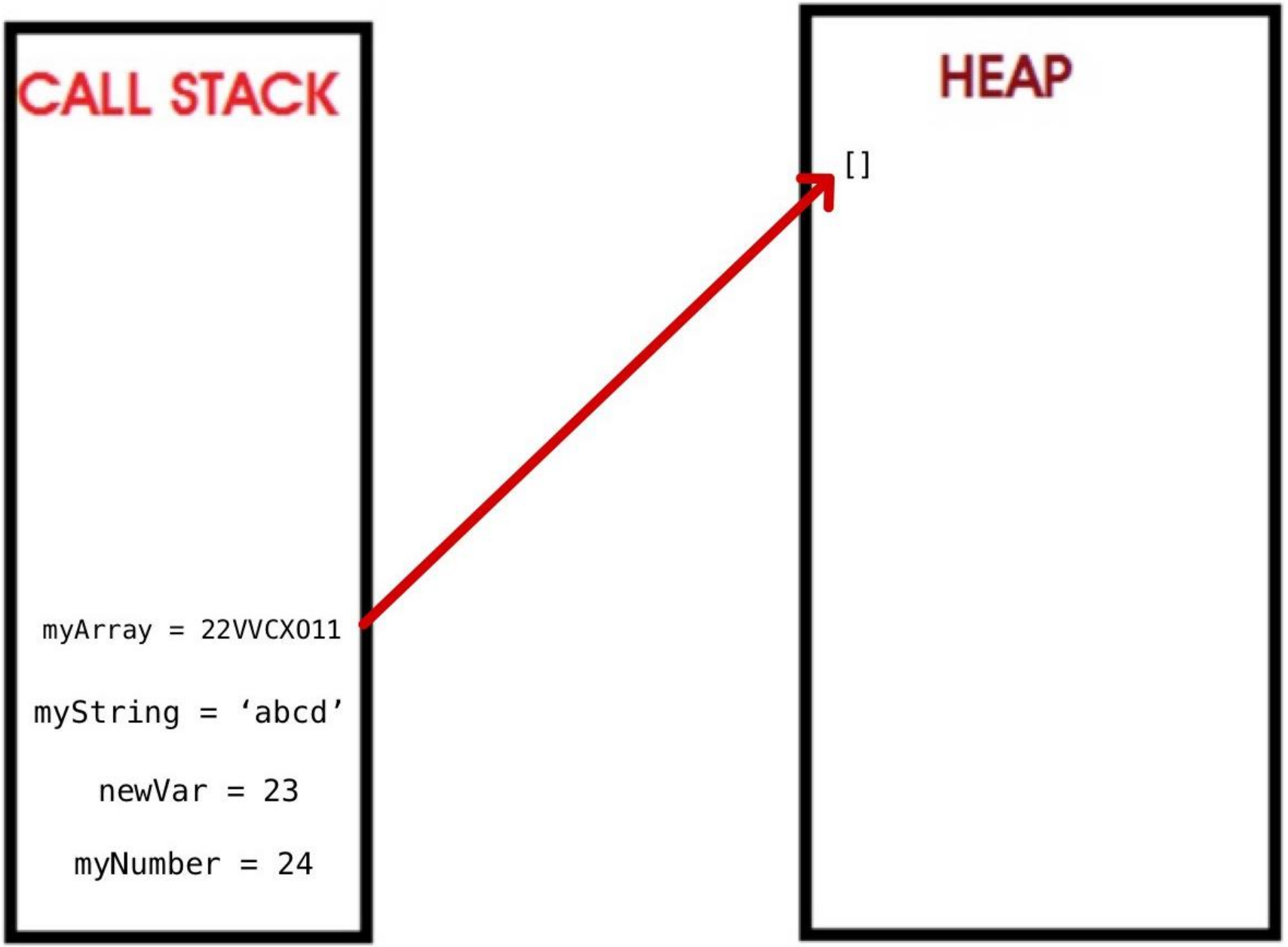
# 메모리 모델

```
let myArray = []
```

- myArray라는 식별자를 선언
- 메모리의 주소를 할당(런타임 시)
- 힙 메모에 할당된 주소를 값으로 저장(⇔위 예에서는 23을 값으로 저장)
- 힙 공간에 값, 즉 []을 할당

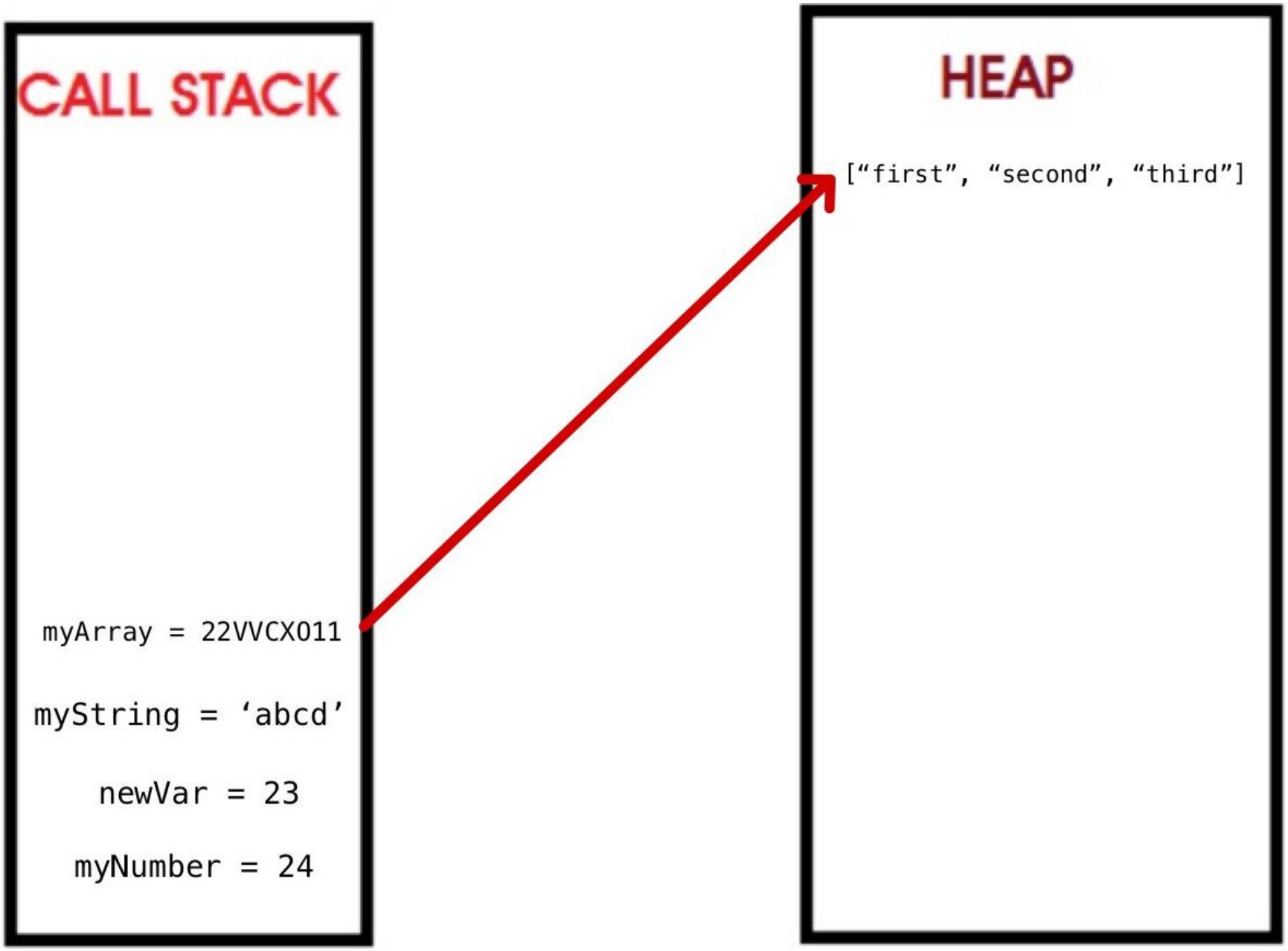


# 메모리 모델


CALL STACK
myArray = 22VVCX011
myString = ‘abcd’
newVar = 23
myNumber = 24
HEAP
[]

# 메모리 모델

```
myArray.push("first")
myArray.push("second")
myArray.push("third")
myArray.push("fourth")
myArray.pop()
```



# Let vs. const

- 일반적으로 되도록이면 **const**를 사용하고, 변수가 변경될 가능성이 있다면 **let**을 사용
- 변경된다?

```
let sum = 0
sum = 1 + 2 + 3 + 4 + 5
let numbers = []
numbers.push(1)
numbers.push(2)
numbers.push(3)
numbers.push(4)
numbers.push(5)
```

sum은 변경될 가능성이 있으므로 let으로 선언

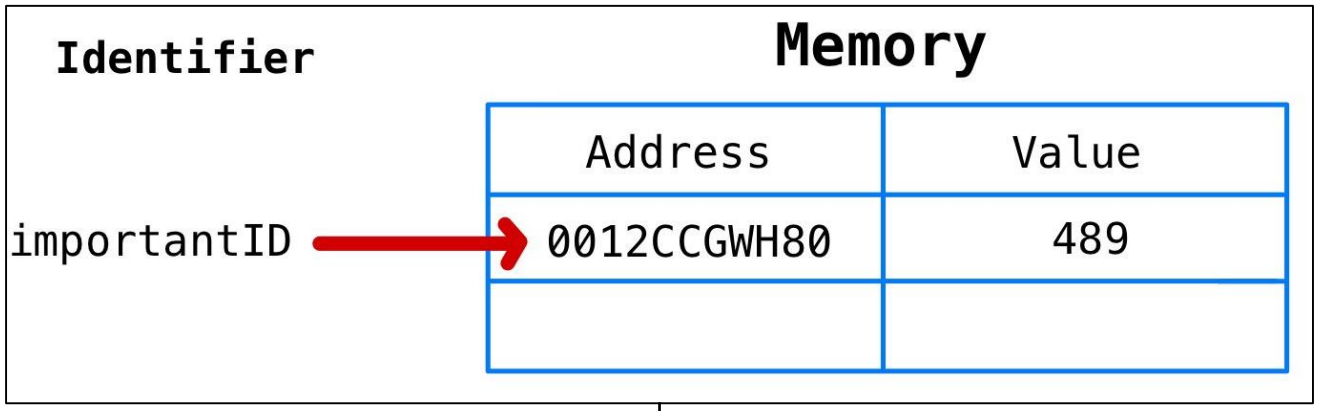
numbers는 변경될 가능성이 있으므로 let으로 선언?

→ push를 5번 수행했다고 numbers가 변할까?

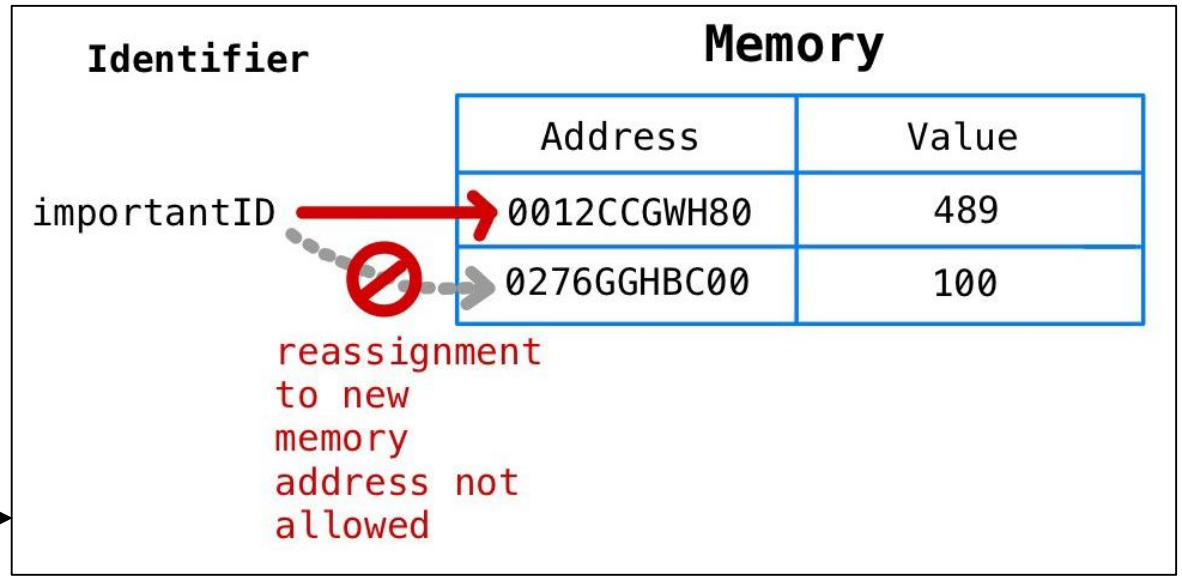
→ change의 의미는 "메모리 주소가 변하느냐?"라는 관점에서 해석 되어야 함

# Let vs. const

```
const importantID = 489
importantID = 100 // TypeError: Assignment to constant variable
```



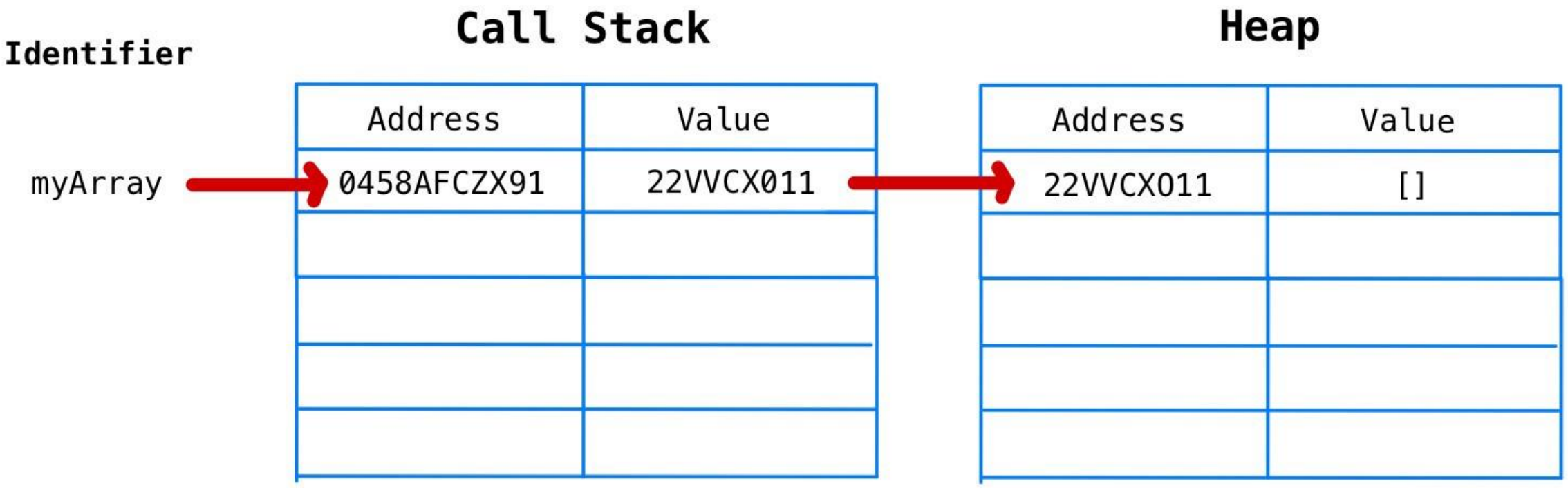

importantID에 100을 할당
100은 원시 타입이라
새로운 메모리 할당하려고 시도



# Let vs. const

- 따라서 array를 let으로 선언하는 것이 아닌 const로 선언해야 함

```
const myArray = []
```



# Let vs. const

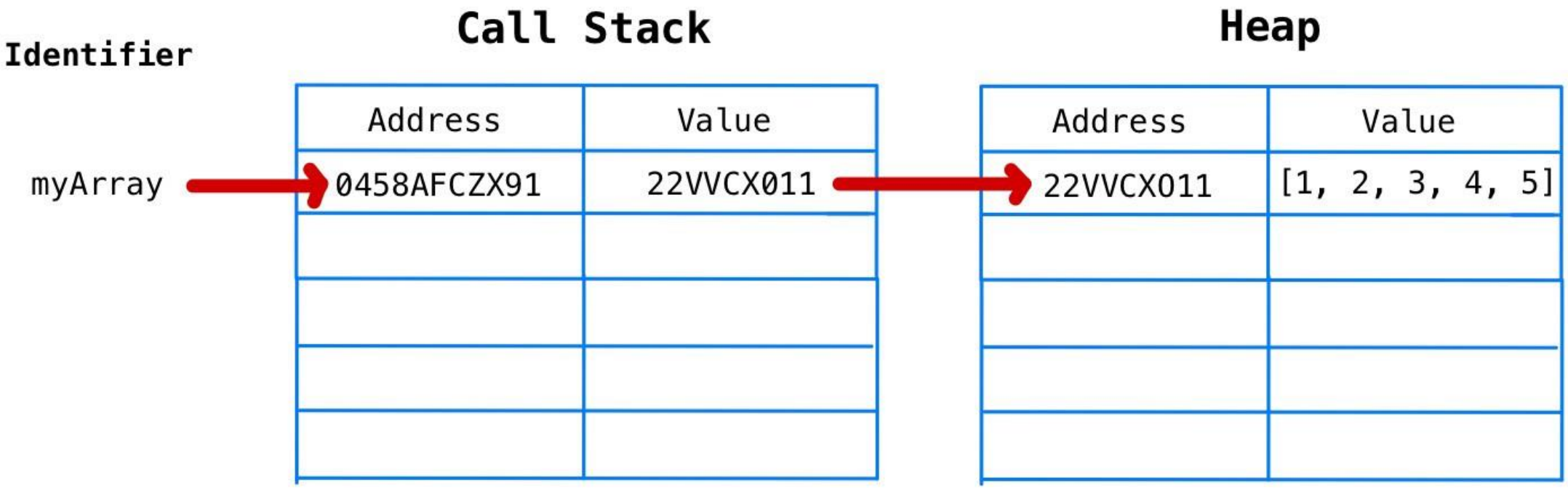
- 따라서 array를 let으로 선언하는 것이 아닌 const로 선언해야 함(하고 싶은 말)

```
myArray.push(1)
myArray.push(2)
myArray.push(3)
myArray.push(4)
myArray.push(5)
```

**Heap**

| Address | Value |
|---|---|
| 22VVCX011 | [] |
| | |
| | |
| | |
| | |

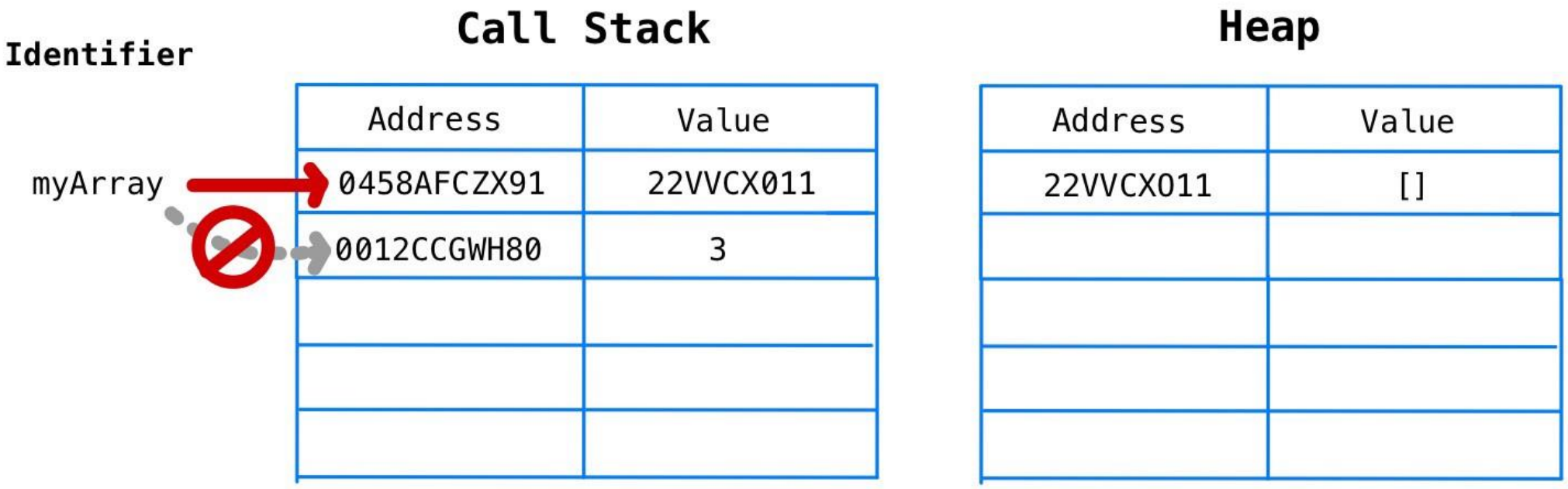
메모리 주소는 변하지 않았다



# Let vs. const

- 다음과 같이 재할당 할 경우 에러가 발생

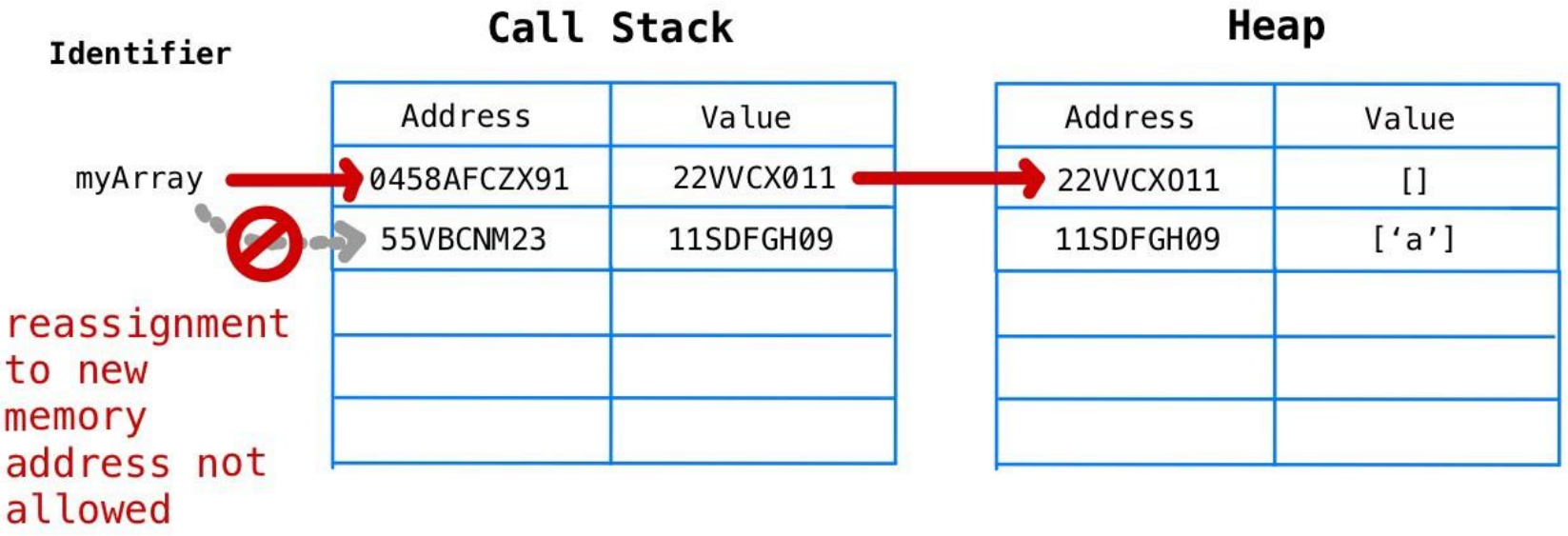
myArray = 3



# Let vs. const

- 다음과 같이 재할당 할 경우 에러가 발생

```
myArray = ['a']
```




- 다음은 정상적으로 동작

```
const myObj = {}
myObj['newKey'] = 'someValue' // this will not throw an error
```