

Array(<https://velog.io/@undefined/%EB%B0%B0%EC%97%B4%EC%9D%98-API>)

선언

```
let arr=["one","two","three"]
```

```
console.log(arr)
```

```
console.log(arr.length)
```

```
console.log(arr.toString())
```

```
function ArrayApiTest () {  
  console.log("JavaScript Array test")  
  
  let arr = ["one", "two", "three"]  
  
  console.log(arr)  
  console.log(arr.toString())  
}
```

```
JavaScript Array test  
▶ (3) ['one', 'two', 'three']  
one,two,three  
>
```

.pop() 요소빼기(뒷자리부터)

.shift() 요소빼기(앞자리부터)

.push() 요소추가(뒤부터)

.unshift() 요소추가 (앞부터)

.slice() 앞에서부터 지정수만큼 자르기

.concat() 붙이기 (데이터 타입을 가리지 않음)

.splice() 요소수정 (게시판 수정에서 자주 씀)

.join() 구분자 삽입

console(arr.join(" * "))->결과: one * two * three

console(arr.join(" , "))->결과: one , two , three

//각 원소와 원소사이에 join()에 있는 정보를 붙이는 역할

.pop() 요소빼기(뒷자리부터)

```
let elem=arr.pop()
```

```
console.log(arr) -> 결과 one, two
```

```
console.log(elem)-> three
```

젤 마지막으로 입력된 정보가 빠짐

```
console.log("pop(요소 빼기) 이전: " + arr)  
let elem = arr.pop()  
  
console.log("pop(요소 빼기) 이후: " + arr)  
console.log("빼낸 요소: " + elem)
```

.shift() 요소빼기(앞자리부터)

```
let elem=arr.shift()
```

console.log(arr) -> two

console.log(elem)-> one

```
// 위의 pop 케이스는 나중에 들어온 놈이 먼저 나간다: Stack  
// 먼저 들어온 놈이 먼저 나간다: Queue  
elem = arr.shift()  
console.log("shift() 이후: " + arr)  
console.log("shift 값(elem): " + elem)
```

```
shift() 이후: two  
shift 값(elem): one
```

.push() 요소추가(뒤부터)

let len=arr.push("seven")

설명:

len은 길이 값. 따라서, push()는 seven을 뒤에서부터 더한 배열의 길이 값을 return한다.

console.log(arr) -> 결과 two, seven

console.log(len)-> 결과 2 //길이 값

```
console.log("push(집어 넣기) 이전: " + arr)  
let len = arr.push("seven")  
  
console.log("push(집어 넣기) 이후: " + arr)  
console.log("java의 array.length 역할: " + len)  
console.log(arr.length)
```

```
push(집어 넣기) 이전: two  
push(집어 넣기) 이후: two,seven  
java의 array.length 역할: 2
```

.unshift() 요소추가 (앞부터)

let len = arr.unshift("333")

console.log(arr)-> 333, two, seven

console.log(len)-> 3

설명:

len은 길이 값. 따라서, unshift()는 333을 앞에서부터 더한 배열의 길이 값을 return한다.

```
len = arr.unshift("333")  
console.log("unshift() 이후: " + arr)  
console.log("len 값: " + len)
```

```
let
```

.splice() 요소수정 (게시판 수정에서 자주 씬)

splice(x,y,element1,element2)->x자리에 y만큼 지우고 정보들을 추가해라

splice(x,y)->x자리부터 y만큼 지워라

```
// 2번째 인덱스에서 0개를 지우고 해당 위치에 정보를 입력
arr.splice(2, 0, "elem1", "elem2")
console.log("splice 이후: " + arr)

// 2번째 인덱스에서 2개를 지우고 해당 위치에 정보 입력
arr.splice(2, 2, "elem3", "elem4")
console.log("splice 이후: " + arr)

arr.splice(1, 2)
console.log("splice 이후: " + arr)
```

```
splice 이후: 333,two,elem1,elem2,seven
splice 이후: 333,two,elem3,elem4,seven
splice 이후: 333,elem4,seven
```

.concat() 붙이기 (데이터 타입을 가리지 않음)

```
let test = ["four", "five", "six"]

arr = arr.concat(test)
console.log("concat 이후: " + arr)

splice 이후: 333,elem4,seven
concat 이후: 333,elem4,seven,four,five,six
```

.concat 응용

```
arr = arr.concat("I can add string")
console.log("concat 이후: " + arr)

let testArr1 = [1, 3]
let testArr2 = [3, 7]

arr = arr.concat(testArr1, testArr2)
console.log("concat 이후: " + arr)

concat 이후: 333,elem4,seven,four,five,six,I can add string
concat 이후: 333,elem4,seven,four,five,six,I can add string,1,3,3,7
```

.slice() 앞에서부터 지정수만큼 자르기

.slice(x,y)->x~y까지의 인덱스 추출 옆에

```
arr = [1, 2, 3, 4, 5, 6, 7]
let slicedArr = arr.slice(1)

console.log("slice 이후: " + arr)
console.log("slice의 결과: " + slicedArr)

arr = [1, 2, 3, 4, 5, 6, 7]
slicedArr = arr.slice(3)

console.log("slice 이후: " + arr)
console.log("slice의 결과: " + slicedArr)
```

```
slice 이후: 1,2,3,4,5,6,7
slice의 결과: 2,3,4,5,6,7
slice 이후: 1,2,3,4,5,6,7
slice의 결과: 4,5,6,7
```

