

JS-중급 7

☰ 태그	
📅 날짜	@2023년 6월 1일

Array

`arr.splice(n, m)` : 특정 요소를 지움
n부터 m개를 지운다.

```
let arr = [1,2,3,4,5];  
arr.splice(1,2);
```

```
console.log(arr); // [3,4,5]
```

`arr.splice(n, m, x)` : 특정 요소 지우고 추가
n부터 m까지 지우고 x를 그자리에 추가

```
let arr = [1,2,3,4,5];  
arr.splice(1,3, 100, 200);
```

```
console.log(arr); // [1,100,200,5]
```

```
let arr = ["나는", "철수", "입니다"];
```

```
arr.splice(1,0,"대한민국", "소방관");
```

```
console.log(arr); // ["나는","대한민국", "소방관", "철수", "입니다"]
```

`arr.splice()` : 삭제된 요소 반환

```
let arr = [1,2,3,4,5];  
let result = arr.splice(1,2);
```

```
console.log(arr); // [1,4,5]  
console.log(result); // [2, 3]
```

`arr.slice(n,m)` : n 부터 m까지 반환

```
let arr = [1,2,3,4,5];
arr.slice(1,4); // [2,3,4]

let arr2 = arr.slice();
console.log(arr2); // [1,2,3,4,5]
```

`arr.concat (arr2, arr3 ...)` : 합쳐서 새배열 반환

```
let arr = [1,2];

arr.concat([3,4]); // [1,2,3,4]
arr.concat([3,4], [5,6]); // [1,2,3,4,5,6]
arr.concat([3,4],5,6); // [1,2,3,4,5,6]
```

`arr.forEach(fn)` : 배열 반복

```
let users= [ 'Mike' , 'Tom', 'Jane'];

user.forEach((item, index, arr) =>{
  //...
});
```

3개의 매개변수가 있는데 첫 번째는 해당요소이고 두번째는
인덱스이다.
세번째는 해당 배열 자체를 의미한다.

`arr.indexOf()` : 해당요소 위치 반환, 없으면 -1 반환
요소가 두개일 경우 첫번째 인수 부터 두번째 인수를 찾음
`arr.lastIndexOf()` : 끝에서 부터 찾아서 위치를 반환

```
let arr = [1,2,3,4,5,1,2,3];

arr.indexOf(3); //2
arr.lastIndexOf(3,3) // 7
```

`arr.includes()` : 포함하는지 확인

```
let arr = [1,2,3];

arr.includes(2); // true
arr.includes(8); // false
```

```
arr.find(fn)
arr.findIndex(fn)
```

indexOf처럼 찾는건 같지만 보다 복잡한 연산이 가능하도록
함수를 사용할 수 있다.

첫번째 true값만 반환하고 끝
만약 없으면 undefined를 반환

```
let arr = [1,2,3,4,5];
const result = arr.find((item) => {
  return item % 2 === 0;
})
console.log(result)
2
```

```
// find / findIndex
```

```
let userList = [
  {name : "Mike", age : 30},
  {name : "Mike1", age : 31},
  {name : "Mike2", age : 32}
];
```

```
const result = userList.find((user) => {
  if(user.age < 31){
    return true;
  }
  return false;
});
```

```
console.log(result);
.
```

```
// [object Object]
{
  "name": "Mike",
  "age": 30
}
```

```
// find / findIndex
```

```
let userList = [
  {name : "Mike", age : 30},
  {name : "Mike1", age : 31},
  {name : "Mike2", age : 32}
];
```

```
const result = userList.findIndex((user) => {
  if(user.age < 31){
    return true;
  }
  return false;
});

console.log(result);

0
```

arr.filter(fn) :
만족하는 모든 요소를 배열로 반환

```
let arr = [1,2,3,4,5];
const result = arr.filter((item) => {
  return item % 2 === 0;
})
console.log(result)
```

```
// [object Array] (2)
[2,4]
```

arr.reverse() : 역순으로 재정렬

```
let arr = [1,2,3,4,5];
arr.reverse(); // [5,4,3,2,1]
```

arr.map(fn) : 함수를 받아 특정 기능을 시행하고 새로운 배열을 반환

```
let userList = [
  {name : "Mike", age : 30},
  {name : "Jane", age : 27},
  {name : "Tom", age : 10}
];

let newUserList = userList.map((user, index) => {
  return Object.assign({}, user, {
    isAdult : user.age > 19,
  })
})
```

```

})

console.log(newUserList)
console.log(userList)

// [object Array] (3)
[// [object Object]
{
  "name": "Mike",
  "age": 30,
  "isAdult": true
},// [object Object]
{
  "name": "Jane",
  "age": 27,
  "isAdult": true
},// [object Object]
{
  "name": "Tom",
  "age": 10,
  "isAdult": false
}]
----
// [object Array] (3)
[// [object Object]
{
  "name": "Mike",
  "age": 30
},// [object Object]
{
  "name": "Jane",
  "age": 27
},// [object Object]
{
  "name": "Tom",
  "age": 10
}]

```

```

let user = {
  name : "Mike",
  age : 30,
};

let userList = ["Mike", "Tome", "Jane"];

console.log(typeof user);
console.log(typeof userList);

console.log(Array.isArray(user));

```

```
console.log(Array.isArray(userList));
```

```
"object"
```

```
"object"
```

```
false
```

```
true
```