

Javascript Array Util Functions

#TIL/dev/language/javascript

Remove certain element in array

```
["bar", "baz", "foo", "qux"]  
  
list.splice(list.indexOf('foo'), 1);
```

[JavaScript: Remove Element from an Array - HostingAdvice.com](#)

Copy on array

Shallow Copy

Array.slice()를 활용하면 아주 손쉽게 가능하다!
하지만 이건 shallow copy 인 것 같고,

```
var arrayOriginal = new Array();  
arrayOriginal.push("a");  
arrayOriginal.push("b");  
arrayOriginal.push("c");  
var arrayClone = arrayOriginal.slice(0); arrayOriginal.push("d");  
arrayClone.unshift("d");  
  
// arrayOriginal => a, b, c, d  
// arrayClone => d, a, b, c
```

javascript 배열복사(passing by value) 편하게 하기.

Deep Copy

배열에 Object등이 담겨있을때, 레퍼런스를 복사하는게 아니라, Value를 복사하고 싶다면, 객체를 직렬화 해서, 다시 파싱하여 객체를 만들어줘야한다.

```
var aaa = [0,1,2,3,4]

var bbb = JSON.parse(JSON.stringify(aaa));
```

JavaScript 레퍼런스 참조없는 배열 복사(Deep Copy) | 감성 프로그래밍

String copy in Javascript

Javascript에서는 기본적으로 String을 Reference로 처리한다. String 기반의 많은 메서드들 역시 reference 기반으로 동작하게 되는데, 이때 deep copy를 하게하는 꿀같은 팁이 있었다.

```
var string_copy = ( ' ' + original_string ).slice(1);
```

How to force JavaScript to deep copy a string?

How to groupBy. Array in Javascript

→ 잠깐 리서치해본 결과, kotlin이나 swift에서 제공하는 groupBy 유틸 함수는 없는 듯 했고, 직접 구현하는 방법 밖에 없어보였다. 아래의 레퍼런스를 참고했는데 굉장히 복잡하고 가독성이 안좋아보였고, 나에게 필요한 기능도 아닌 것 같아서(Array => section으로 구분된 Array) 그냥 내가 직접 구현하였다.

처음에는 한번의 루프로 원하는 array를 뽑아낼 수 있는 방법이 없을까 고민했는데, 굳이 그럴 필요가 없었다. 그냥 독립적으로 루프를 돌리면 성능상의 이슈가 발생하지도 않고($n + n = 2n$), 주석을 간단하게 달면 가독성까지 챙길 수 있었다!

reduce

→ Array의 element값들을 이용한 연산을 누적하여 계산하고 싶을때 사용하면 된다.

```
initExpand={item.item.contents.reduce<boolean>((pre, choice) => {  
    // 감정 키워드들중 하나라도 선택되어있다면 Cell은 Expand 시킵니다.  
    return pre || selectedEmotions.includes(choice);  
}, false)}
```

some

배열 안의 어떤 요소라도주어진 판별 함수를 통과하는지 테스트합니다.

[Array.prototype.some\(\) - JavaScript | MDN](#)