

1) 배열을 순회하고 싶다면? forEach()

```
let array = [1, 2, 3, 4, 5, 6];

array.forEach((item, index) => {
  console.log(item);
});
```

2) 공백을 기준으로 분할하여 배열로 담는다면? split(" ")

```
let string = "hello world";

const str = string.split(" ");
console.log(str);
```

3) 배열을 다시 문자열로 바꾸고 싶다면? join(" ")

```
5
6 console.log(str.join(""));
7 console.log(str.join(" "));
8 console.log(str.join(" my "));
9
```

문제 출력 디버그 콘솔 터미널 포트

```
[ 'hello', 'world' ]  
helloworld  
hello world  
hello my world
```

4) 오름차순 정렬 $\text{sort}((a,b) \Rightarrow (a-b))$

```
2  
3    ascSortNumber = numbers.sort((a, b) => a - b);  
4    console.log(ascSortNumber);  
5
```

문제 출력 디버그 콘솔 터미널 포트

```
[  
  1,    1, 1, 2, 3,  
  3,    4, 5, 5, 9,  
  9, 222  
]
```

5) 내림차순 정렬 $\text{sort}((a,b) \Rightarrow (a-b) - 1)$

```
2  
3    descSortNumber = numbers.sort((a, b) => (a - b) * -1);  
4    console.log(descSortNumber);  
5
```

문제 출력 디버그 콘솔 터미널 포트

```
khch5@BOOK-8A3U060ROI MINGW64 /c/Project/HTML/test01
$ node 11.js
[
  222, 9, 9, 5, 5,
    4, 3, 3, 2, 1,
    1, 1
]
```

6) 짝수만 꺼내고 싶다면? filter()

```
1 let numbers = [1, 2, 3, 4, 5, 6];
2
3 const filteredNumbers = numbers.filter((i) => {
4   | return i % 2 == 0;
5   | });
6
7 console.log(filteredNumbers);
8
```

7) 요소에 x 2를 해서 반환하고 싶다면? map()

```
1 let numbers = [1, 2, 3, 4, 5, 6];
2
3 const mapNumber = numbers.map((i) => {
4   | return i * 2;
5   | });
6
```

8) to be continued ...

<https://www.youtube.com/watch?v=MlvZ2lufTFI>

