

Recitation 12

Solving problems using `map()`, `reduce()`, `filter()` in JS

Example 1 – Removing Duplicates

- Consider the following array

```
var data = [  
  {  
    name: 'Kyle',  
    occupation: 'Fashion Designer'  
  },  
  {  
    name: 'Liza',  
    occupation: 'Web Developer'  
  },  
  {  
    name: 'Emily',  
    occupation: 'Web Designer'  
  },  
]
```

```
{  
  name: 'Melissa',  
  occupation: 'Fashion Designer'  
},  
{  
  name: 'Tom',  
  occupation: 'Web Developer'  
}  
];
```

Example 1 – Removing Duplicates

- Getting the list of occupations using map()

```
var jobs = data.map(function (item) {  
    return item.occupation;  
});  
  
// Logs ["Fashion Designer", "Web Developer", "Web  
Designer", "Fashion Designer", "Web Developer"]  
console.log(jobs);
```

Example 1 – Removing Duplicates

- Removing duplicates using filter()

```
var jobsUnique = jobs.filter(function(item, index){  
    return jobs.indexOf(item) >= index;  
});  
  
// Logs ["Fashion Designer", "Web Developer", "Web  
Designer"]  
console.log(jobsUnique);
```

On each iteration, we use `Array.indexOf()` to see if our item already exists. If the returned index is smaller than the current index, that means an instance of item already exists. Otherwise, we return it to add it to the new array.

Example 2: Using map()

- Consider the following array of objects:

```
const songs = [  
  { id: 1, name: "Curl of the burl", artist: "Mastodon"},  
  { id: 2, name: "Oblivion", artist: "Mastodon"},  
  { id: 3, name: "Flying Whales", artist: "Gojira"},  
  { id: 4, name: "L'Enfant Sauvage", artist: "Gojira"},  
];
```

Example 2: Using map()

- Now, add and remove properties from and to each object

```
var mapped = songs.map(function (song) {  
  var newSong = _.omit(song, "artist");  
  
  return Object.assign(newSong, {  
    scrobbleCount: 0,  
    spotifyUrl: "let's just imagine these properties make sense for now",  
  });  
});
```

Example 3: Using filter()

- Filter strings containing the substring 'at'

```
var strings = ["hello", "Matt", "Mastodon", "Cat", "Dog"];

var filtered = strings.filter(function (str) {
  return str.includes("at");
});

// ES6
const filtered = strings.filter(str => {
  return str.includes("at");
});

console.log(filtered); // ["Matt", "Cat"];
```

Example 4: Using reduce()

- Consider the array *songs* in slide 5. Reduce the array into an object containing elements {artist: song count}

```
var obj = songs.reduce(function (acc, currValue) {  
  var artist = currValue.artist;  
  var artistCount = acc[artist] ? acc[artist] + 1 : 1;  
  
  var newObj = {};  
  newObj[artist] = artistCount;  
  
  return Object.assign(acc, newObj);  
}, {});
```


Example 5: Using Reduce

- Turn an array of arrays into a single array.

```
const mult = [songs, [{id: 112, name: "Chop Suey", artist: "System of a Down" }]];

var flatMult = mult.reduce(function (acc, currValue) {
  return acc.concat(currValue);
}, []);
```

Example 6: Using map(), reduce(), filter() together

- Consider the two arrays:

```
const songs = [  
  { id: 1, name: "Curl of the burl", artist: "Mastodon", duration: 204}  
  { id: 2, name: "Oblivion", artist: "Mastodon", duration: 306},  
  { id: 3, name: "Flying Whales", artist: "Gojira", duration: 444}  
  { id: 4, name: "L'Enfant Sauvage", artist: "Gojira", duration: 246}  
];  
const lastFmSongs = [  
  { id: 5, name: "Abc", artist: "X", duration: 192}  
  { id: 6, name: "DEF", artist: "Y", duration: 186}  
  { id: 7, name: "GHI", artist: "Z", duration: 132}  
  { id: 8, name: "JKL", artist: "A", duration: 241}  
];
```

Example 6: Using map(), reduce(), filter() together

- Get a string separated by commas of all songs whose duration is superior to 4 minutes

```
const songNames = allSongs.reduce((acc, currValue) => {  
    return acc.concat(currValue);  
}, [])  
    .map(song => {  
        return { ...song, duration: Math.floor(song.duration / 60) };  
    })  
    .filter(song => {  
        return song.duration > 4;  
    })  
    .map(song => song.name).join(" , ");  
console.log(songNames); // Oblivion , Flying Whales , L'Enfant Sauvage, JKL
```

References

- <https://gomakethings.com/removing-duplicates-from-an-array-with-vanilla-javascript/>
- <https://medium.com/@joomiguelcunha/learn-map-filter-and-reduce-in-javascript-ea59009593c4>