Date : 26 - 06 - 2020
Morning Session : 9am – 11 PM
By ~ Sundeep Charan Ramkumar Today

Topics: ES6 Day 5 (FP and Higher order functions)

Higher order methods:

- **forEach**
- **Map**
- **Filter**
- **Reduce**

forEach: straight forward function which will allows you to traverse each value inside an array as well as index .

```
const users = [
    {
      id: 1,
      email: "george.bluth@reqres.in",
      first_name: "George",
      last_name: "Bluth",
      avatar: "https://s3.amazonaws.com/uifaces/faces/twitter/calebogden/128.jpg"
    },
    {
      id: 2,
      email: "janet.weaver@reqres.in",
      first_name: "Janet",
      last_name: "Weaver",
      avatar: "https://s3.amazonaws.com/uifaces/faces/twitter/josephstein/128.jpg"
    },
    {
      id: 3,
      email: "emma.wong@reqres.in",
      first_name: "Emma",
      last_name: "Wong",
      avatar:
        "https://s3.amazonaws.com/uifaces/faces/twitter/olegpogodaev/128.jpg"
    },
```

```
  {
    id: 4,
    email: "eve.holt@reqres.in",
    first_name: "Eve",
    last_name: "Holt",
    avatar:
      "https://s3.amazonaws.com/uifaces/faces/twitter/marcoramires/128.jpg"
  },
  {
    id: 5,
    email: "charles.morris@reqres.in",
    first_name: "Charles",
    last_name: "Morris",
    avatar: "https://s3.amazonaws.com/uifaces/faces/twitter/stephenmoon/128.jpg"
  },
  {
    id: 6,
    email: "tracey.ramos@reqres.in",
    first_name: "Tracey",
    last_name: "Ramos",
    avatar: "https://s3.amazonaws.com/uifaces/faces/twitter/bigmancho/128.jpg"
  }
];
```

```
users.forEach(function (element, index) {
  console.log(element, index);
});
```

app.js:49

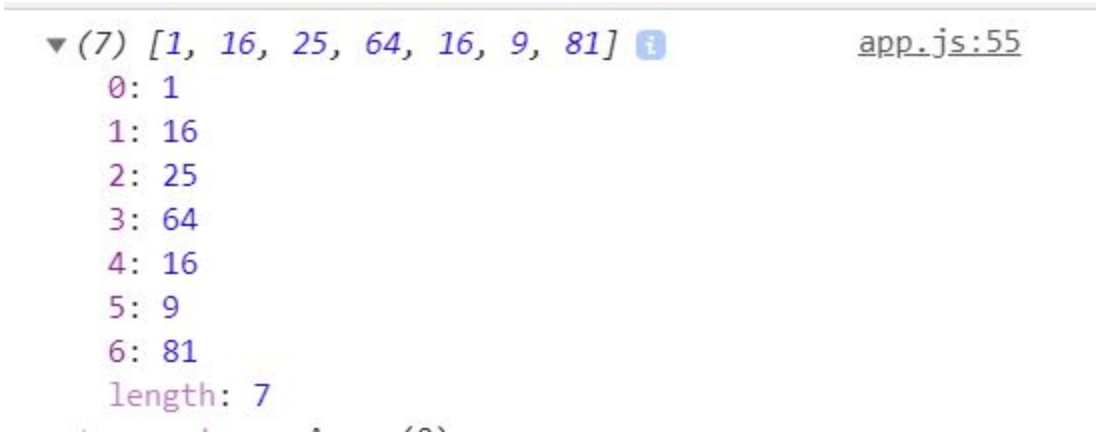
```
▼ Object
    avatar: "https://s3.amazonaws.com/uifaces/faces/…
    email: "george.bluth@reqres.in"
    first_name: "George"
    id: 1
    last_name: "Bluth"
  ▶ __proto__: Object
0
```

app.js:49

```
▼ Object
    avatar: "https://s3.amazonaws.com/uifaces/faces/…
    email: "janet.weaver@reqres.in"
    first_name: "Janet"
    id: 2
    last_name: "Weaver"
  ▶ __proto__: Object
1
```

Map: The **map()** method is used for creating a new array from an existing one, applying a function to each one of the elements of the first array.

```
const inputArray = [1, 4, 5, 8, 4, 3, 9];
const result = inputArray.map(element => element ** 2);
console.log(result);
```



Filter : The **filter()** method takes each element in an array and it applies a conditional statement against it.

```
const numbers = [1, 2, 3, 4];
const evens = numbers.filter(item => item % 2 === 0);
console.log(evens); // [2, 4]
```

Reduce

The `reduce()` method reduces an array of values down to just one value. To get the output value, it runs a reducer function on each element of the array.

```
arr.reduce(callback[, initialValue])
```

```
const numbers = [1, 2, 3, 4];
const sum = numbers.reduce(function (result, item) {
  return result + item;
}, 0);
console.log(sum); // 10
```

The callback argument is a function that will be called once for every item in the array. This function takes four arguments, but often only the first two are used.

- *accumulator* - the returned value of the previous iteration
- *currentValue* - the current item in the array
- *index* - the index of the current item
- *array* - the original array on which reduce was called
- The initialValue argument is optional. If provided, it will be used as the initial accumulator value in the first call to the callback function.

https://www.freecodecamp.org/news/javascript-map-reduce-and-filter-explained-with-examples/