- github: divyaranjan10

DATE

CHAP 19: Map, filter & Reduce

- Map function: It is used to transform the array, perform operation on each value of array.

- Example 1: (Double the value)

```
let arr = [2, 3, 4, 5];
const output = arr.map ((x) => {
      return x * 2;
})
console.log (output);
```

- Output

```
[4, 6, 8, 10]
```

- Example 2: (Convert to binary)

```
const binary  = arr.map((x) => {
      return x.toString (2);
})
console.log (binary);
```

- Output

```
['10', '11', '100', '101']
```

- **filter function**: Filter fn is used to filter out the desired values from an array.
- Example 1: (Filter the odd one out):

```
const arr = [2,3,4,5];

const oddValues = arr.filter((x)=> {
    return x % 2;
})
console.log(oddValues);
```

- Output:

```
[3, 5]
```

- **Reduce function**: Reduce function takes all the elements of the array and comes out with a single value.

⇒ Sum of all numbers in array (using normal approach and then using reduce function)

- Example 1: (Normal approach)

```
let arr = [2, 3, 4, 5, 6, 8];
function sumAll (arr) {
    let sum = 0;
    for (let i=0; i<arr.length; i++){
        sum = sum + arr[i];
    }
    return sum;
}
console.log(sumAll(arr));
```

· Output

28

· Example 2 : (using reduce function)

```
const sumArray = arr.reduce((acc, curr) => {
    acc = acc + curr;       Accumulator     Current
    return acc;
}, 0)  ——> accumulator initial value (acc)
console.log(sumArray);
```

· Output

28

· Tricky Map() question :

=> Given an array of objects, return full-name combining first and last name.

```
=> const users = [
   {firstName : "akshay", lastName : "saini", age : 26},
   {firstName : "donald", lastName : "trump", age : 75},
   {firstName : "Elon", lastName : "Musk", age : 50},
   {firstName : "deepika", lastName : "padukone", age : 26}
];

const userName = users.map((x) => {
      return x.firstName + " " + x.lastName;
}
console.log(userName);
```

- Output

```
['akshay saini', 'donald trump', "Elon Musk',
 'deepika padukone']
```

- Tricky Reduce() question:

=> Count the unique age of person in the users array.

```
=> const users = [same as above]
```

```
const ageGroup = users. reduce ((acc, curr) => {
    if (acc [curr. age]) {
        acc [curr. age] = ++acc [curr. age];
    }
    else {
        acc[ curr. age] = 1;
    }
    return acc;            • Output
}, {})                     {26 : 2, 75 : 1, 50 : 1}

console. log (ageGroup);
```

• Chaining map, filter and reduce:

=> Print out the firstName whose age is less than 30:

=> const users = [same as in last ex]

```
const output = users. filter((x) => {
    return x.age <30;
}).map ((x) => {                     • Output
    return x. firstName;  => ['akshay', 'deepika'].
})
console. log (output);
```

- Doing previous question (using reduce function)

```
=> const users = [used in previous example]

const output = users.reduce((acc, curr) => {
        if(curr.age < 30){
            acc.push(curr.firstName);
        }
        return acc;
  }, []);
console.log(output);
```

- Output

=> ['akshay', 'deepika'];