[ ].map()

[ ].filter()

[ ].reduce()

→ map() , filter() & reduce() are higher order functions in JS.

① [ ].map() :—

input

function

→ we Can use map function to Transform an array

Ex:- Const arr = [5,1,3,2,6]

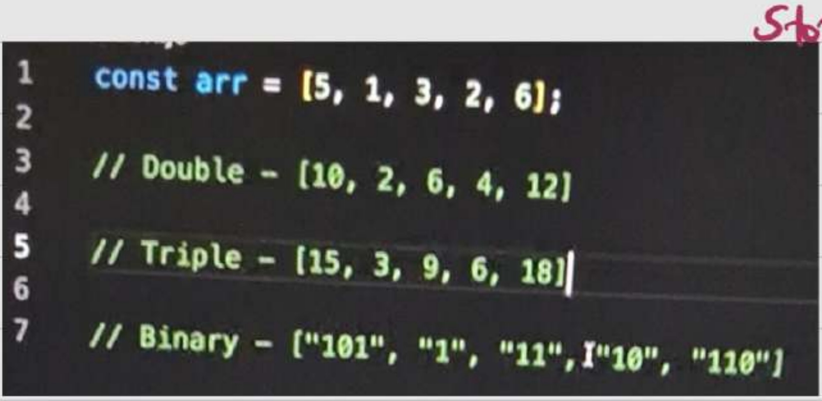
You want to double The value of arr & Store it in another array.

or

Triple The values & Store in another array.

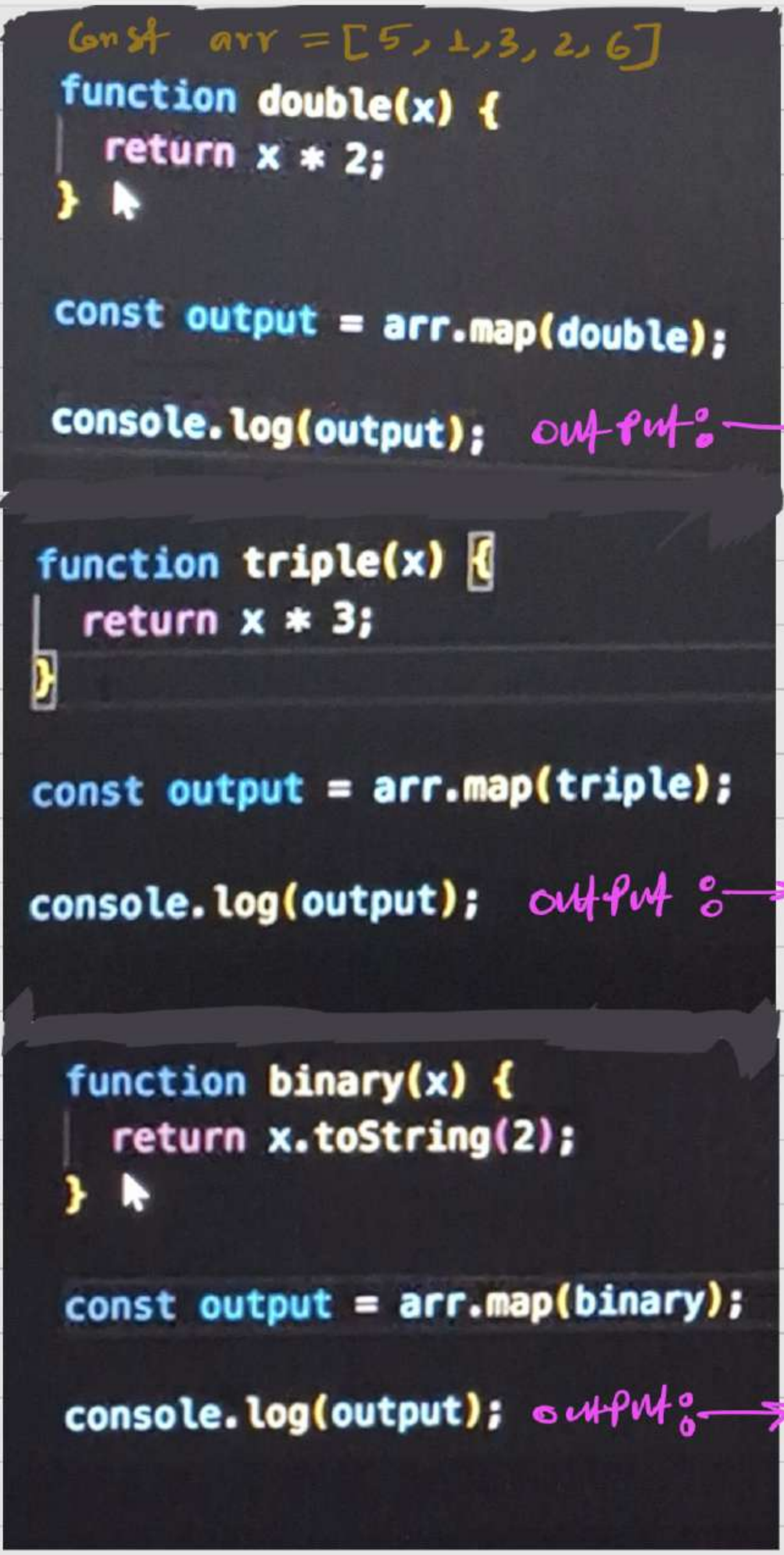
or

Convert The values into binary form then Store in another array



```
const arr = [5, 1, 3, 2, 6];

// Double - [10, 2, 6, 4, 12]

// Triple - [15, 3, 9, 6, 18]

// Binary - ["101", "1", "11", "10", "110"]
```

we Can do These by using map() function.



Const arr = [5, 1, 3, 2, 6]

```
function double(x) {
  return x * 2;
}

const output = arr.map(double);

console.log(output);
```

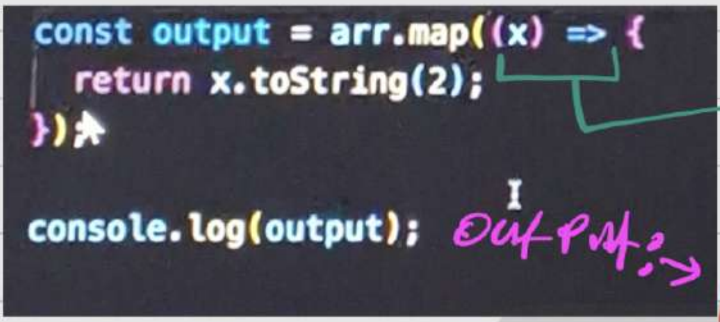
output: → [10, 2, 6, 4, 12]

```
function triple(x) {
  return x * 3;
}

const output = arr.map(triple);

console.log(output);
```

output: → [15, 3, 9, 6, 18]

```
function binary(x) {
  return x.toString(2);
}

const output = arr.map(binary);

console.log(output);
```

output: → [“101, “1”, “11”, “10”, “110”]



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

console.log(output);
```

Output: → [“101, “1”, “11”, “10”, “110”]

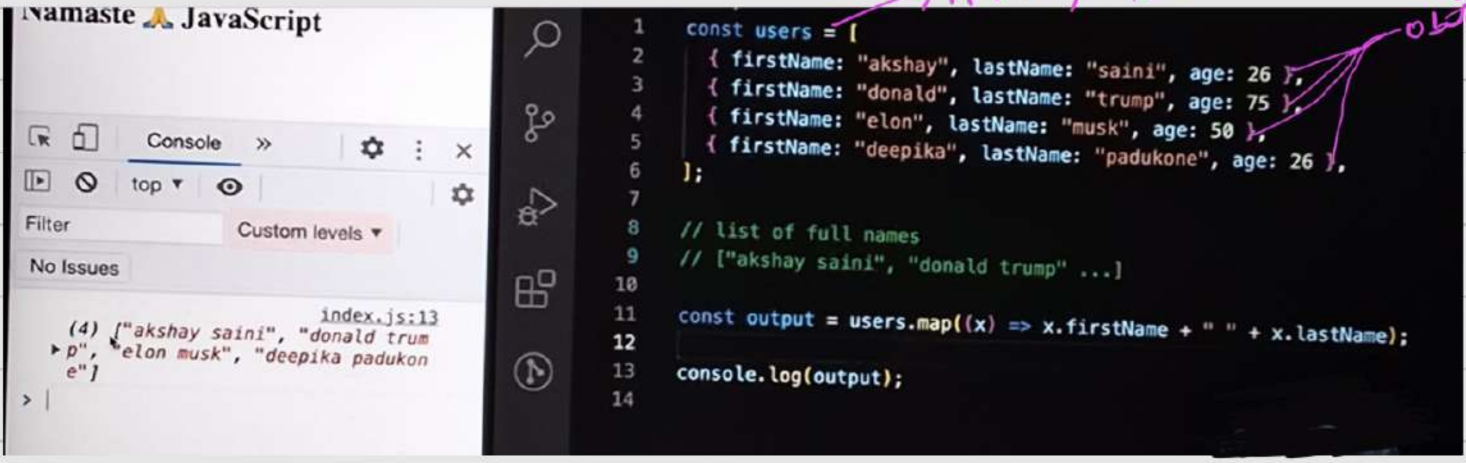
You can also pass an Arrow function to map.

Questions :-

① Print The full name of users.

→ Array of objects

object



```
index.js:13
(4) ["akshay saini", "donald trum
p", "elon musk", "deepika padukon
e"]
```

```
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 },
];

// list of full names
// ["akshay saini", "donald trump" ...]

const output = users.map((x) => x.firstName + " " + x.lastName);

console.log(output);
```

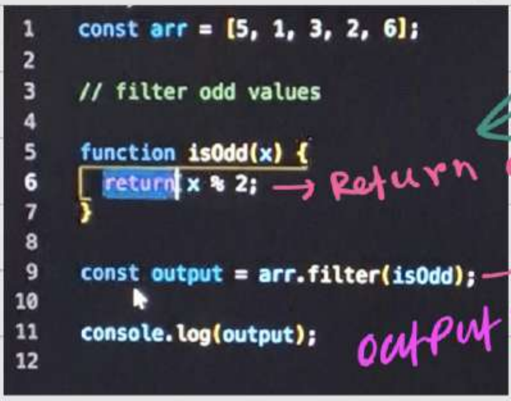
② [ ].filter( ) :—

→ input

→ logic to filter

→ is used to filter values inside an array.

Ex. — → lets say you want to filter all The values in an array which are odd.



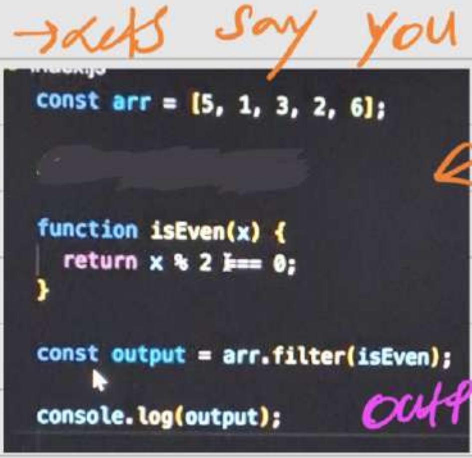
```
const arr = [5, 1, 3, 2, 6];

// filter odd values

function isOdd(x) {
  return x % 2;
}

const output = arr.filter(isOdd);

console.log(output);
```

→ Return 0(false) or 1 (True)

→ reciеve 0 as False & 1 as True.

output:— [5, 1, 3]

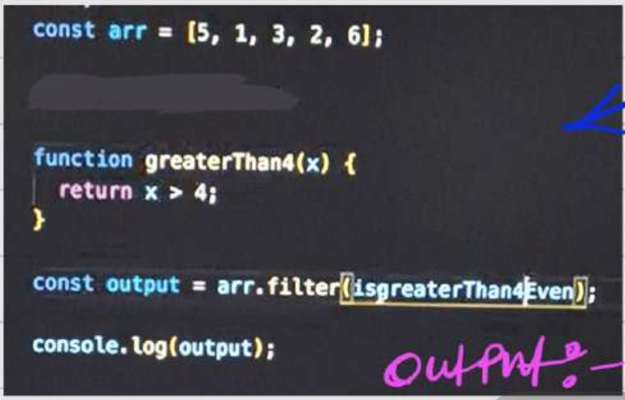
→ lets say you want to filter all The even values



```
const arr = [5, 1, 3, 2, 6];

[redacted]

function isEven(x) {
  return x % 2 === 0;
}

const output = arr.filter(isEven);

console.log(output);
```

output:— [2, 6]

→ You want to filter the values in an array which are > 4.

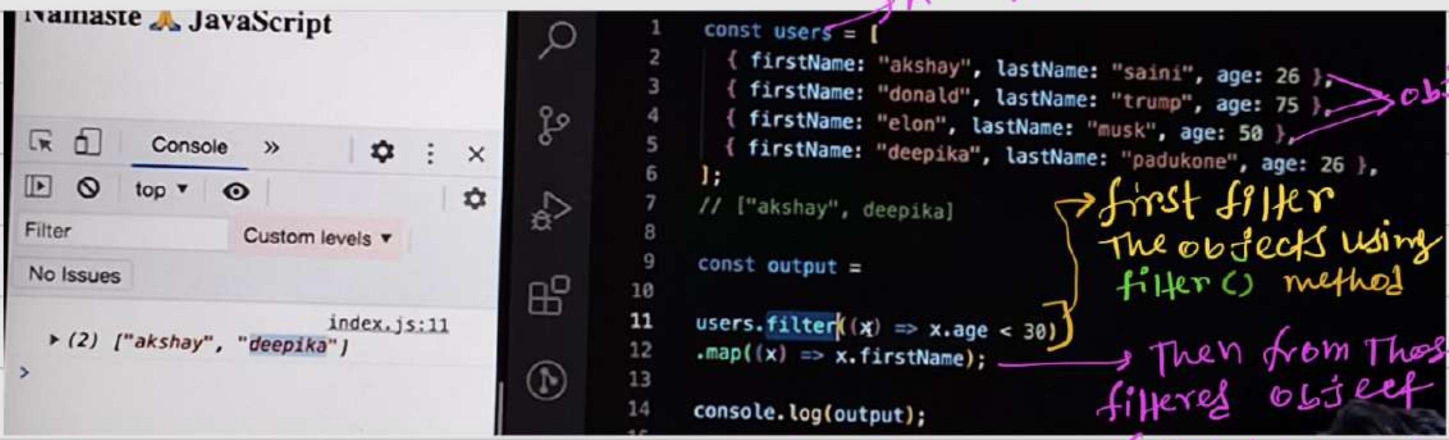


```
const arr = [5, 1, 3, 2, 6];

[redacted]

function greaterThan4(x) {
  return x > 4;
}

const output = arr.filter(isgreaterThan4Even);

console.log(output);
```

output:— [5, 6]

Saved in Gallery.

⑧ we need to find The first name of those user whose age is less then 30.



```
▸ (2) ["akshay", "deepika"]
>
```

```
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 },
];
// ["akshay", deepika]

const output =

users.filter((x) => x.age < 30)
.map((x) => x.firstName);

console.log(output);
```

Array of objects

objects

first filter The objects using filter () method

Then from Those filtered object Select The names using map () & Store inside output

③ [ ].reduce ( ) :-

a function which has two parameter

arr = [1, 2, 3, 4, 5];

Const output = arr.reduce(function(acc, curr){...logic to Accumulate}, initial value of "acc")

→ reduce function have 2 parameter,

① A function:-

The Function which will have 2 parameter, "acc" and "curr".

→ reduce iterates the function inside it to the each and every element of the array. & the value of curr will be the value of element where the function currently in.

"acc":- will Accumulate the values comming to the curr one by one and store it.

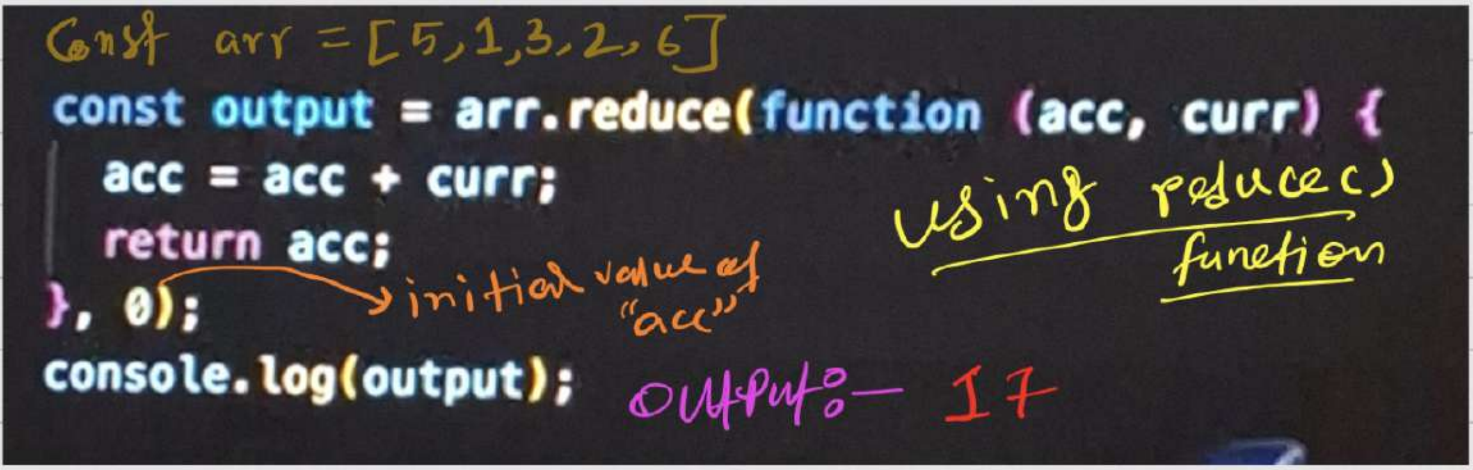
means doing any operations & combine the results. (ex:- sum all the element, subtract the element etc)

② Initial value of "ACC" inside the function

→ reduce function is used where you have to take all the elements of an array and output single value out of them.

Ex:-1

→ Lets say you have an array and you have to find the sum of elements in that array we can use reduce function to do the task.

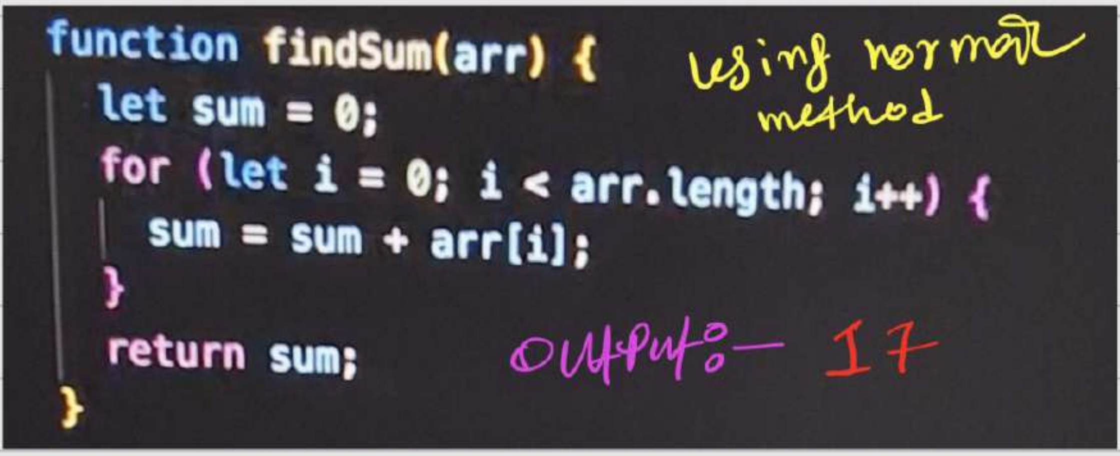


```
Const arr =[5,1,3,2,6]
const output = arr.reduce(function (acc, curr) {
  acc = acc + curr;
  return acc;
}, 0);
console.log(output);
```

using reduce() function

initial value of "acc"

output:- 17



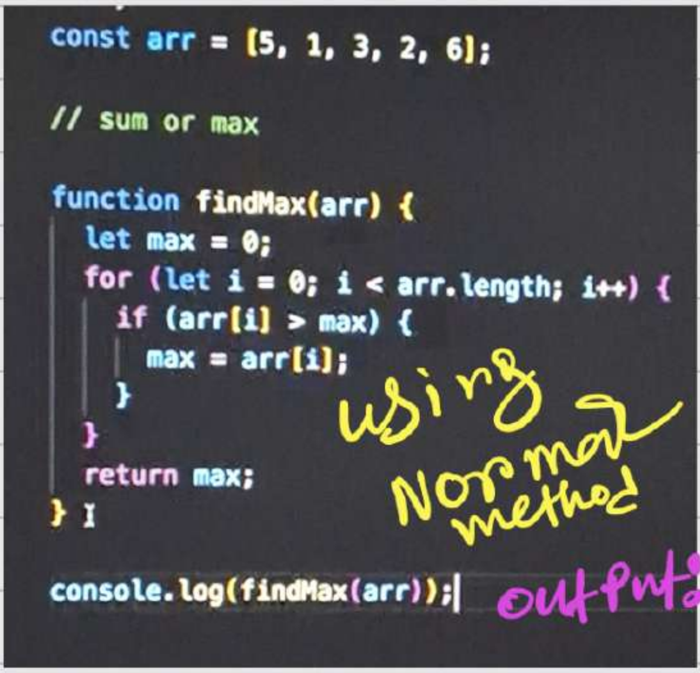
```
function findSum(arr) {
  let sum = 0;
  for (let i = 0; i < arr.length; i++) {
    sum = sum + arr[i];
  }
  return sum;
}
```

using normal method

output:- 17

Ex.-2

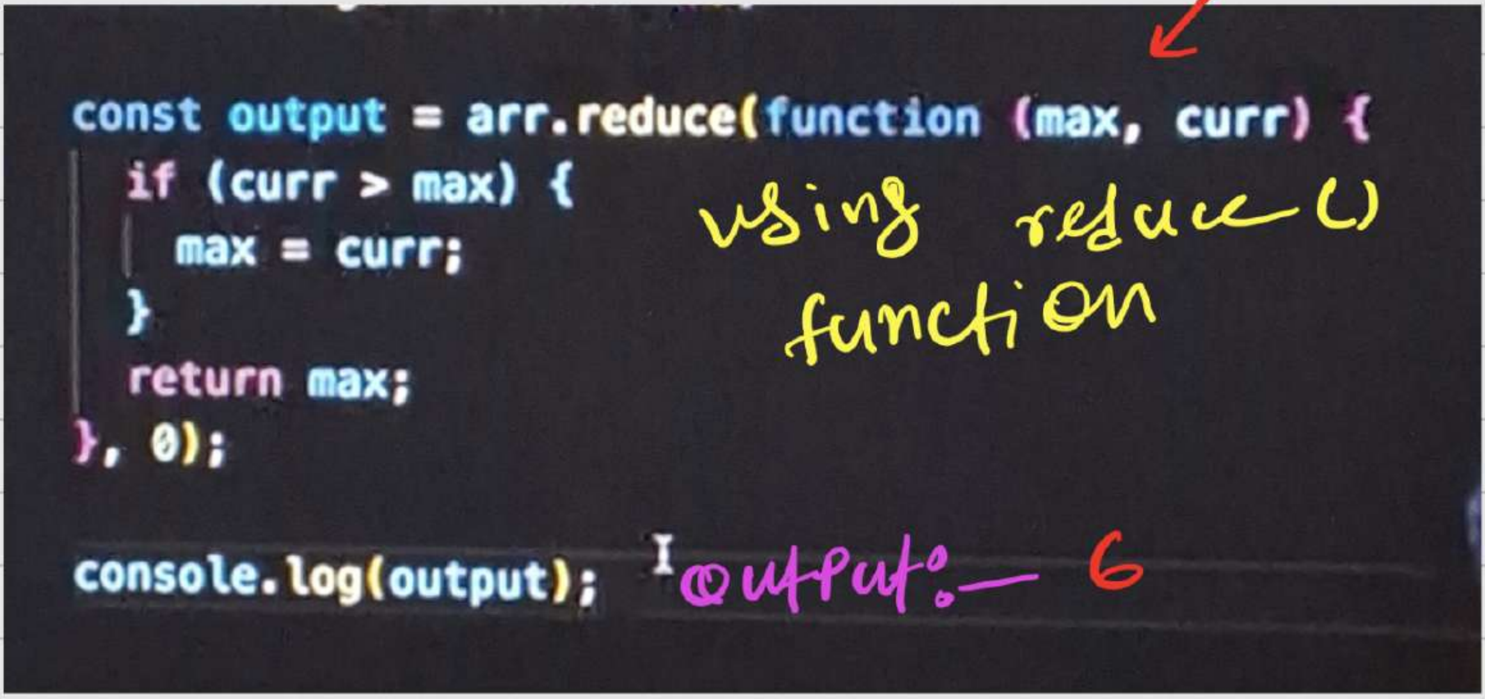
→ Now lets Say You have to find The max inside an array.



```
const arr = [5, 1, 3, 2, 6];

// sum or max

function findMax(arr) {
  let max = 0;
  for (let i = 0; i < arr.length; i++) {
    if (arr[i] > max) {
      max = arr[i];
    }
  }
  return max;
}

console.log(findMax(arr));
```

using Normal method

output:- 6



```
const output = arr.reduce(function (max, curr) {
  if (curr > max) {
    max = curr;
  }
  return max;
}, 0);

console.log(output);
```

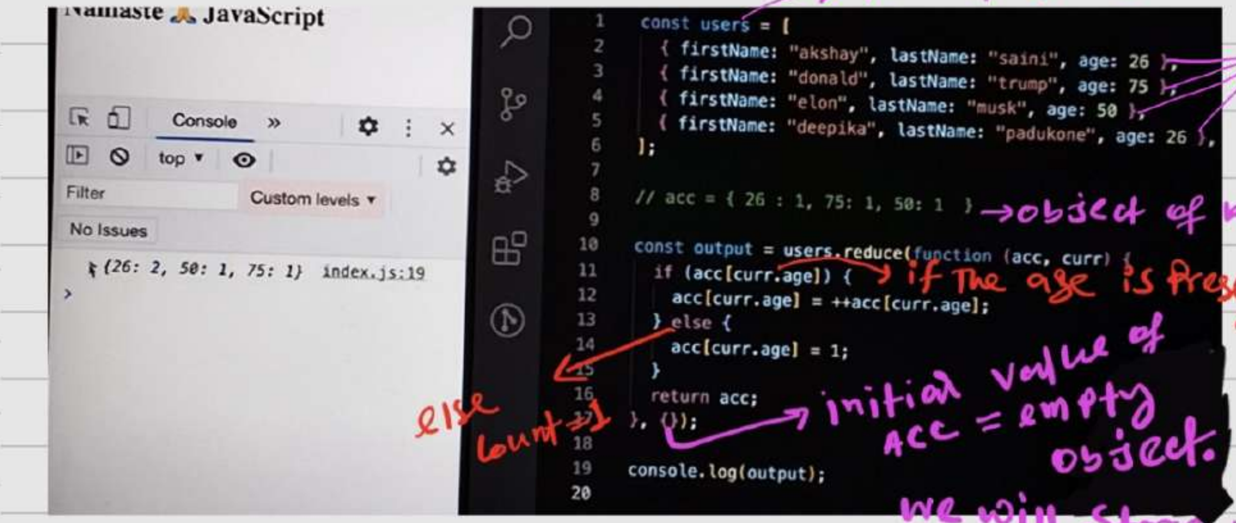
using reduce() function

output:- 6

Question :-

② Lets Say we have to find how many people have a Particular age.

→ Array of objects



```
{26: 2, 50: 1, 75: 1}  index.js:19
```

```
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 },
];

// acc = { 26 : 1, 75: 1, 50: 1 }

const output = users.reduce(function (acc, curr) {
  if (acc[curr.age]) {
    acc[curr.age] = ++acc[curr.age];
  } else {
    acc[curr.age] = 1;
  }
  return acc;
}, {});

console.log(output);
```

→ objects

→ object of key: value Pair

→ if The age is Present inside acc already then Count++

else Count = 1

→ initial value of ACC = empty object.

we will store key: value Pairs here.

key = age, value = Count.