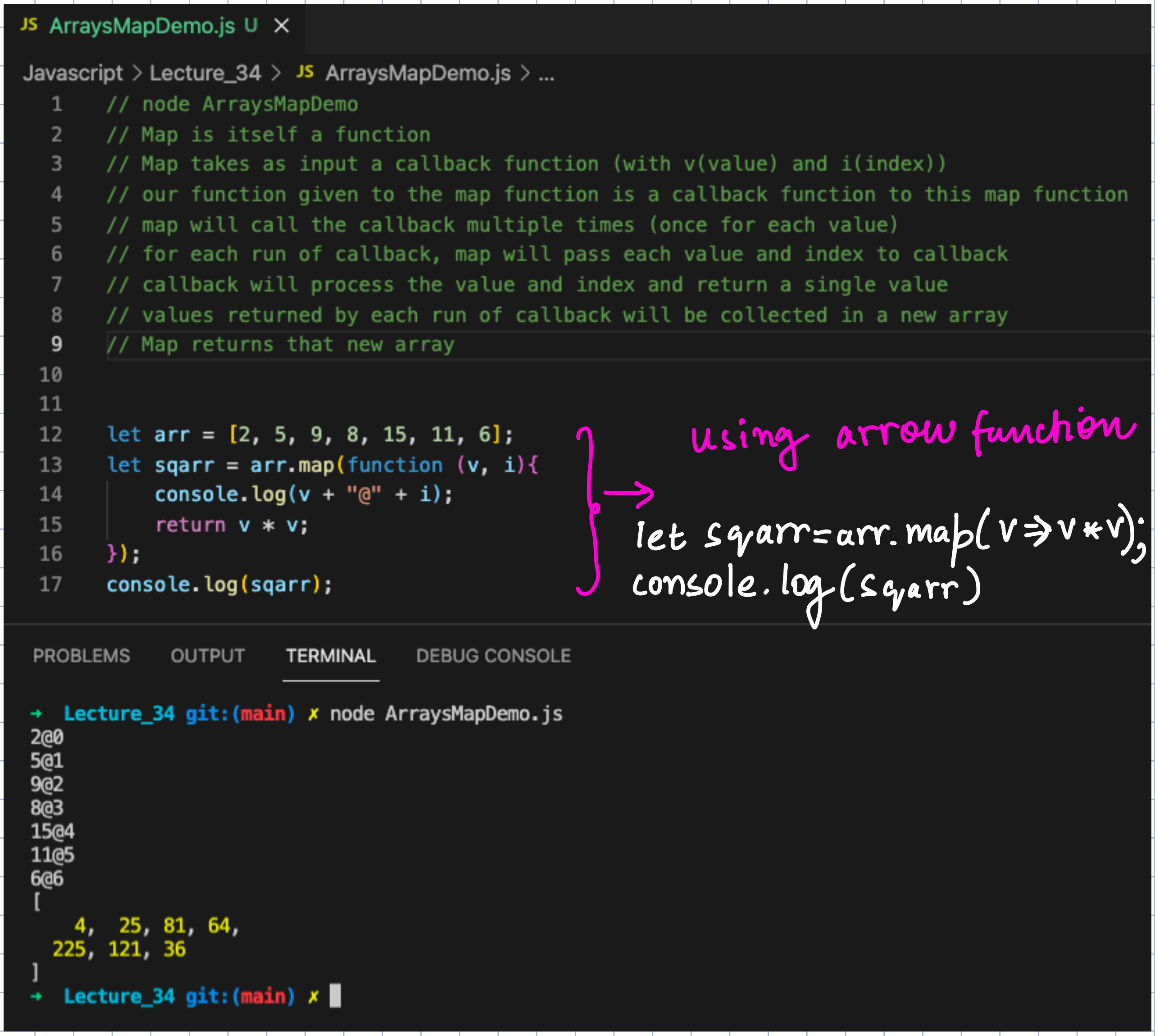
ARRAYS MAP

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array



```javascript
// node ArraysMapDemo
// Map is itself a function
// Map takes as input a callback function (with v(value) and i(index))
// our function given to the map function is a callback function to this map function
// map will call the callback multiple times (once for each value)
// for each run of callback, map will pass each value and index to callback
// callback will process the value and index and return a single value
// values returned by each run of callback will be collected in a new array
// Map returns that new array


let arr = [2, 5, 9, 8, 15, 11, 6];
let sqarr = arr.map(function (v, i){
    console.log(v + "@" + i);
    return v * v;
});
console.log(sqarr);
```

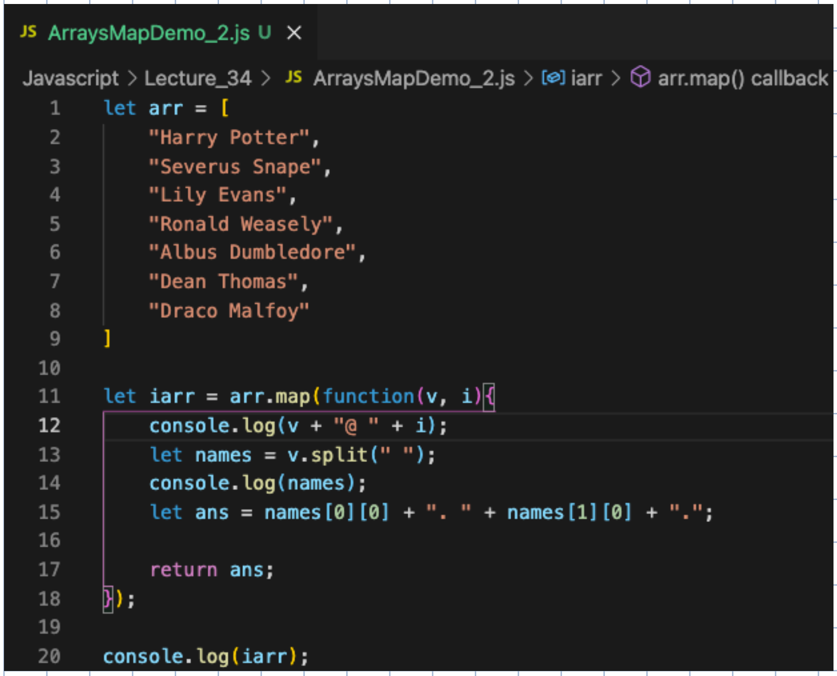
using arrow function

→

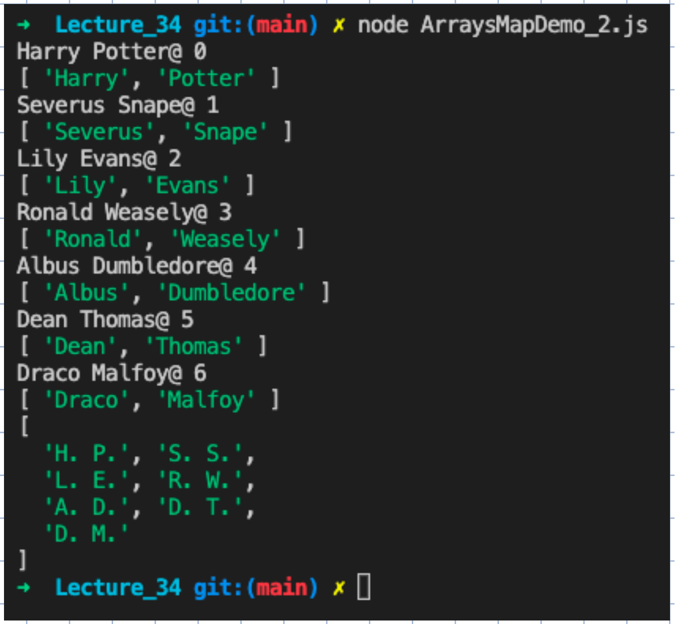
```
let sqarr=arr.map(v=>v*v);
console.log(sqarr)
```

```
→ Lecture_34 git:(main) ✗ node ArraysMapDemo.js
2@0
5@1
9@2
8@3
15@4
11@5
6@6
[
    4,  25, 81, 64,
  225, 121, 36
]
→ Lecture_34 git:(main) ✗ ▌
```

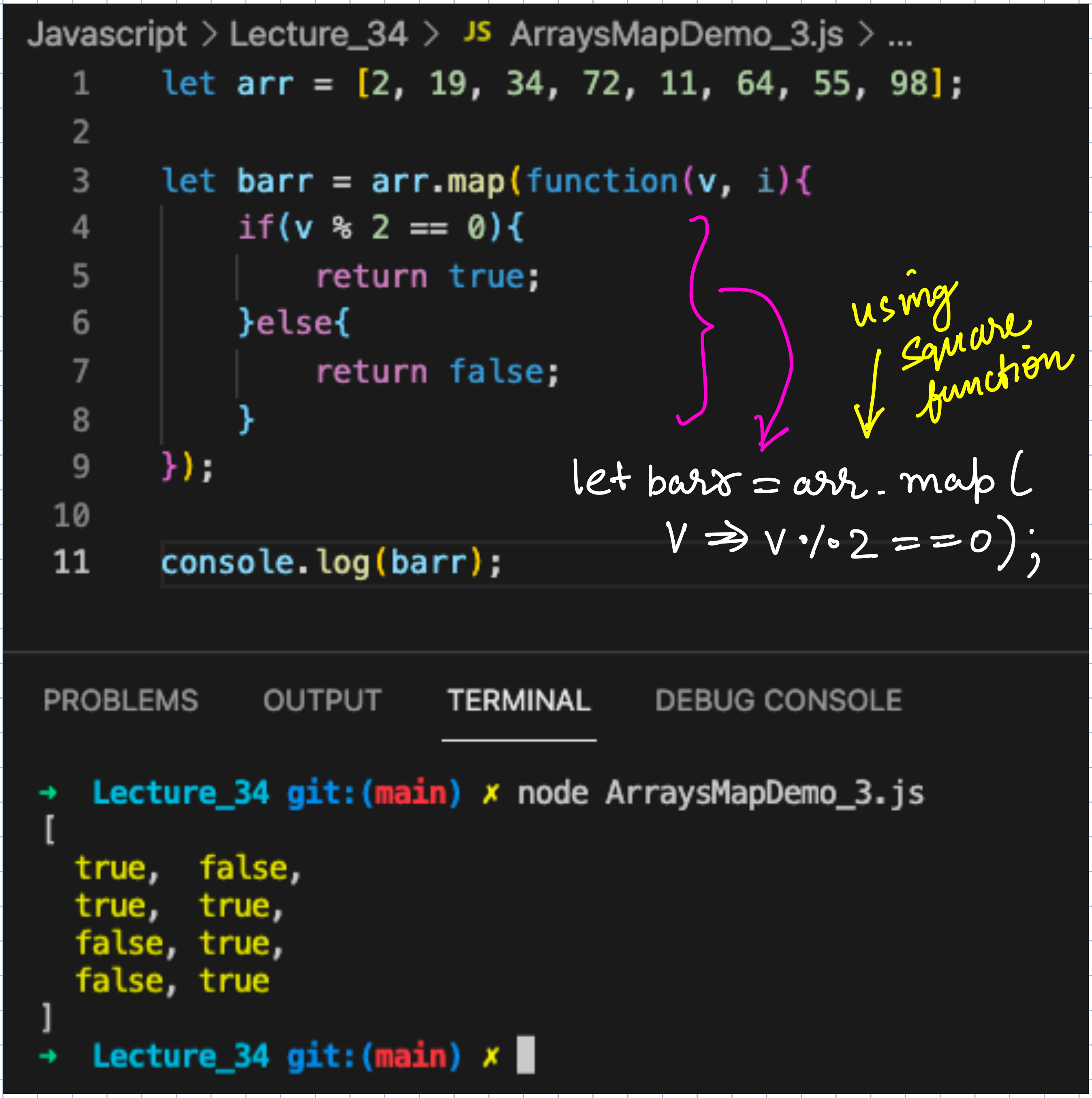
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/map


JS ArraysMapDemo_2.js U ×
Javascript > Lecture_34 > JS ArraysMapDemo_2.js > iarr > arr.map() callback

```javascript
let arr = [
    "Harry Potter",
    "Severus Snape",
    "Lily Evans",
    "Ronald Weasely",
    "Albus Dumbledore",
    "Dean Thomas",
    "Draco Malfoy"
]

let iarr = arr.map(function(v, i){
    console.log(v + "@ " + i);
    let names = v.split(" ");
    console.log(names);
    let ans = names[0][0] + ". " + names[1][0] + ".";

    return ans;
});

console.log(iarr);
```



```
→ Lecture_34 git:(main) ✗ node ArraysMapDemo_2.js
Harry Potter@ 0
[ 'Harry', 'Potter' ]
Severus Snape@ 1
[ 'Severus', 'Snape' ]
Lily Evans@ 2
[ 'Lily', 'Evans' ]
Ronald Weasely@ 3
[ 'Ronald', 'Weasely' ]
Albus Dumbledore@ 4
[ 'Albus', 'Dumbledore' ]
Dean Thomas@ 5
[ 'Dean', 'Thomas' ]
Draco Malfoy@ 6
[ 'Draco', 'Malfoy' ]
[
  'H. P.', 'S. S.',
  'L. E.', 'R. W.',
  'A. D.', 'D. T.',
  'D. M.'
]
→ Lecture_34 git:(main) ✗ ▯
```



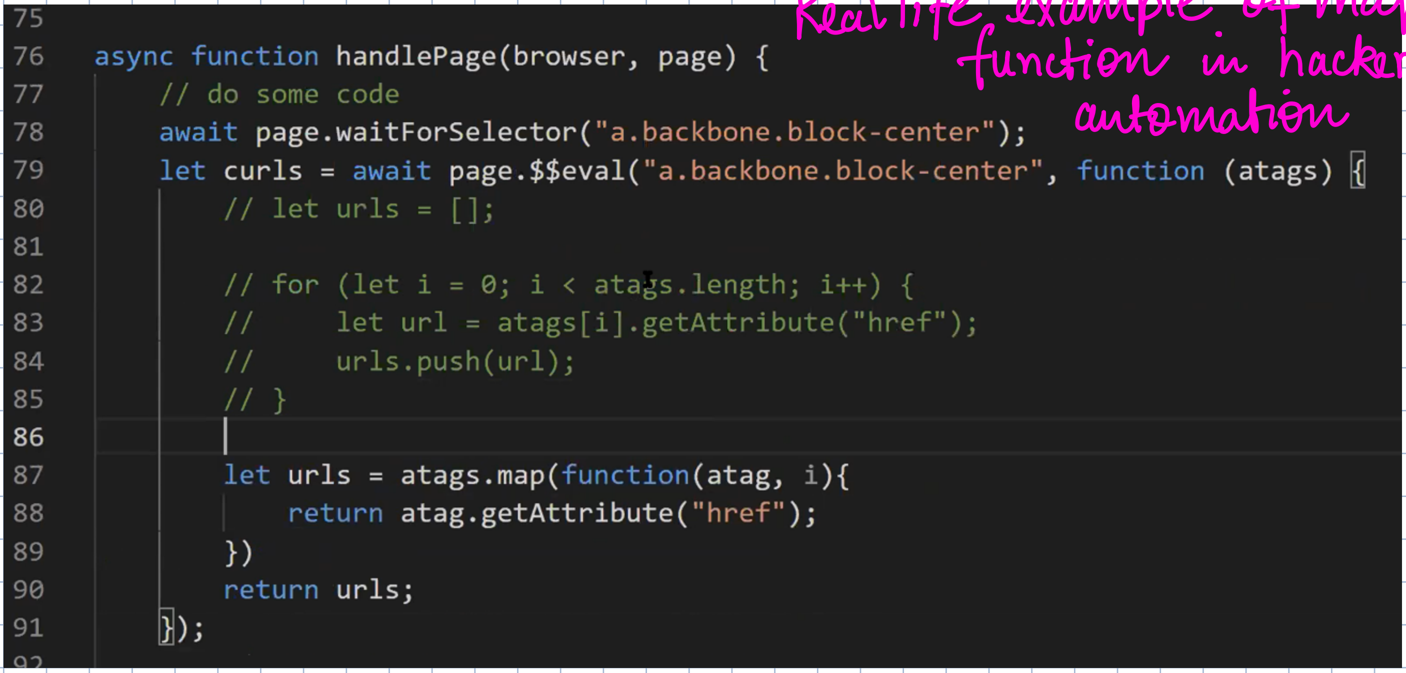
```
let arr = [2, 19, 34, 72, 11, 64, 55, 98];

let barr = arr.map(function(v, i){
    if(v % 2 == 0){
        return true;
    }else{
        return false;
    }
});

console.log(barr);
```

using square function

```
let barr = arr.map(
    v => v%2 == 0);
```

```
→ Lecture_34 git:(main) ✗ node ArraysMapDemo_3.js
[
  true,  false,
  true,  true,
  false, true,
  false, true
]
→ Lecture_34 git:(main) ✗ 
```

Real life example of map function in hackerrank automation



```
async function handlePage(browser, page) {
    // do some code
    await page.waitForSelector("a.backbone.block-center");
    let curls = await page.$$eval("a.backbone.block-center", function (atags) {
        // let urls = [];

        // for (let i = 0; i < atags.length; i++) {
        //     let url = atags[i].getAttribute("href");
        //     urls.push(url);
        // }

        let urls = atags.map(function(atag, i){
            return atag.getAttribute("href");
        })
        return urls;
    });
```