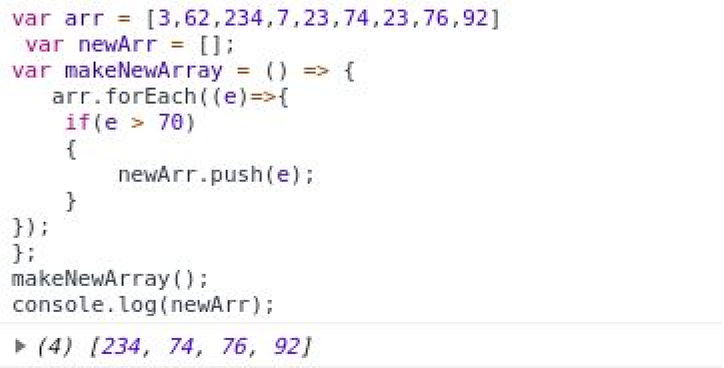
ES6 Javascript Assessment

Q1. Given this array: `[3,62,234,7,23,74,23,76,92]`, Using arrow function, create an array of the numbers greater than `70`.

```
var arr = [3,62,234,7,23,74,23,76,92]

var newArr = [];

var makeNewArray = () => {

   arr.forEach((e)=>{

    if(e > 70)

    {

        newArr.push(e);

    }

});

};

makeNewArray();

console.log(newArr);
```

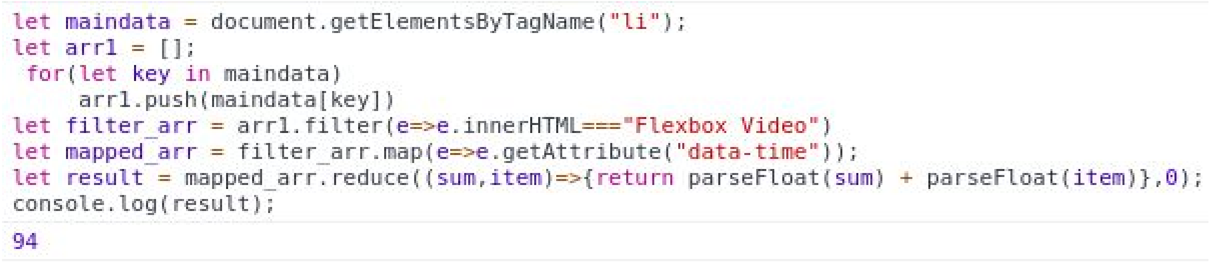


```
var arr = [3,62,234,7,23,74,23,76,92]
 var newArr = [];
var makeNewArray = () => {
   arr.forEach((e)=>{
    if(e > 70)
    {
        newArr.push(e);
    }
});
};
makeNewArray();
console.log(newArr);
▶ (4) [234, 74, 76, 92]
```

Q2. <ul> <li data-time="5:17">Flexbox Video</li> <li data-time="8:22">Flexbox Video</li> <li data-time="3:34">Redux Video</li> <li data-time="5:23">Flexbox Video</li> <li data-time="7:12">Flexbox Video</li> <li data-time="7:24">Redux Video</li> <li data-time="6:46">Flexbox Video</li> <li data-time="4:45">Flexbox Video</li> <li data-time="4:40">Flexbox Video</li> <li data-time="7:58">Redux Video</li> <li data-time="11:51">Flexbox Video</li> <li data-time="9:13">Flexbox Video</li><li data-time="5:50">Flexbox Video</li> <li data-time="5:52">Redux Video</li> <li data-time="5:49">Flexbox Video</li> <li data-time="8:57">Flexbox Video</li> <li data-time="11:29">Flexbox Video</li> <li data-time="3:07">Flexbox Video</li> <li data-time="5:59">Redux Video</li> <li data-time="3:31">Flexbox Video</li></ul>

Select all the list items on the page and convert to array. Filter for only the elements that contain the word 'flexbox' map down to a list of time strings map to an array of seconds reduce to get total using .filter and .map

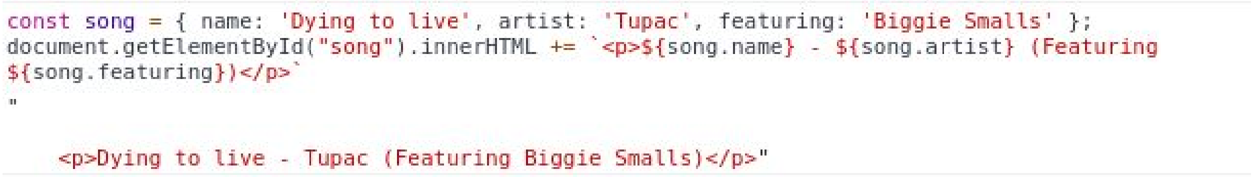
```
let maindata = document.getElementsByTagName("li");

let arr1 = [];

 for(let key in maindata)

     arr1.push(maindata[key])

let filter_arr = arr1.filter(e=>e.innerHTML==="Flexbox Video")

let mapped_arr = filter_arr.map(e=>e.getAttribute("data-time"));

let result = mapped_arr.reduce((sum,item)=>{return parseFloat(sum) + parseFloat(item)},0);

console.log(result);
```



```
let maindata = document.getElementsByTagName("li");
let arr1 = [];
 for(let key in maindata)
     arr1.push(maindata[key])
let filter_arr = arr1.filter(e=>e.innerHTML==="Flexbox Video")
let mapped_arr = filter_arr.map(e=>e.getAttribute("data-time"));
let result = mapped_arr.reduce((sum,item)=>{return parseFloat(sum) + parseFloat(item)},0);
console.log(result);
94
```

Q3. Create a markup template using string literal const song = { name: 'Dying to live', artist: 'Tupac', featuring: 'Biggie Smalls' }; Result: "<div class="song"> <p> Dying to live — Tupac (Featuring Biggie Smalls) </p> </div> "

```
const song = { name: 'Dying to live', artist: 'Tupac', featuring: 'Biggie Smalls' };

document.getElementById("song").innerHTML += `<p>${song.name} - ${song.artist} (Featuring ${song.featuring})</p>`
```

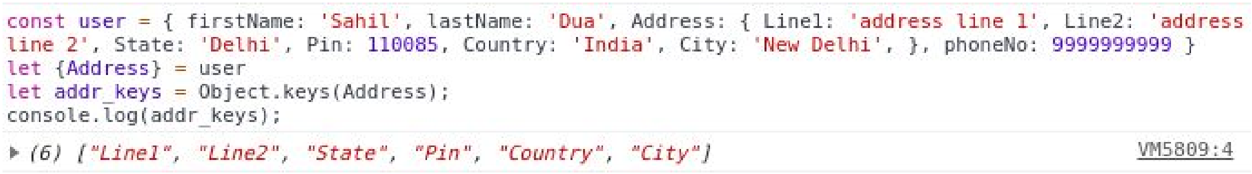


```
const song = { name: 'Dying to live', artist: 'Tupac', featuring: 'Biggie Smalls' };
document.getElementById("song").innerHTML += `<p>${song.name} - ${song.artist} (Featuring
${song.featuring})</p>`
"

    <p>Dying to live - Tupac (Featuring Biggie Smalls)</p>"
```

Q4. Extract all keys inside address object from user object using destructuring ?

const user = { firstName: ‘Sahil’, lastName: ‘Dua’, Address: { Line1: ‘address line 1’, Line2: ‘address line 2’, State: ‘Delhi’, Pin: 110085, Country: ‘India’, City: ‘New Delhi’, }, phoneNo: 9999999999 }

```
const user = { firstName: 'Sahil', lastName: 'Dua', Address: {
Line1: 'address line 1', Line2: 'address line 2', State: 'Delhi',
Pin: 110085, Country: 'India', City: 'New Delhi', }, phoneNo:
9999999999 }

let {Address} = user

let addr_keys = Object.keys(Address);

console.log(addr_keys);
```



```
const user = { firstName: 'Sahil', lastName: 'Dua', Address: { Line1: 'address line 1', Line2: 'address
line 2', State: 'Delhi', Pin: 110085, Country: 'India', City: 'New Delhi', }, phoneNo: 9999999999 }
let {Address} = user
let addr_keys = Object.keys(Address);
console.log(addr_keys);
▸ (6) ["Line1", "Line2", "State", "Pin", "Country", "City"]                               VM5809:4
```