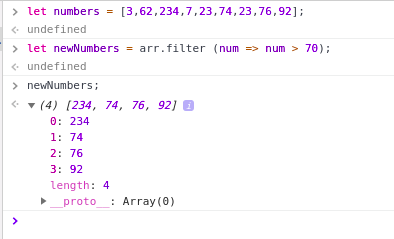
ES 6 session 1 assignment

**Question 1: Given this array: `[3,62,234,7,23,74,23,76,92]`, Using arrow function, create an array of the numbers greater than `70`**

**Solution:**

****

**Question 2:**

|  |  |
| --- | --- |
| **<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>** |

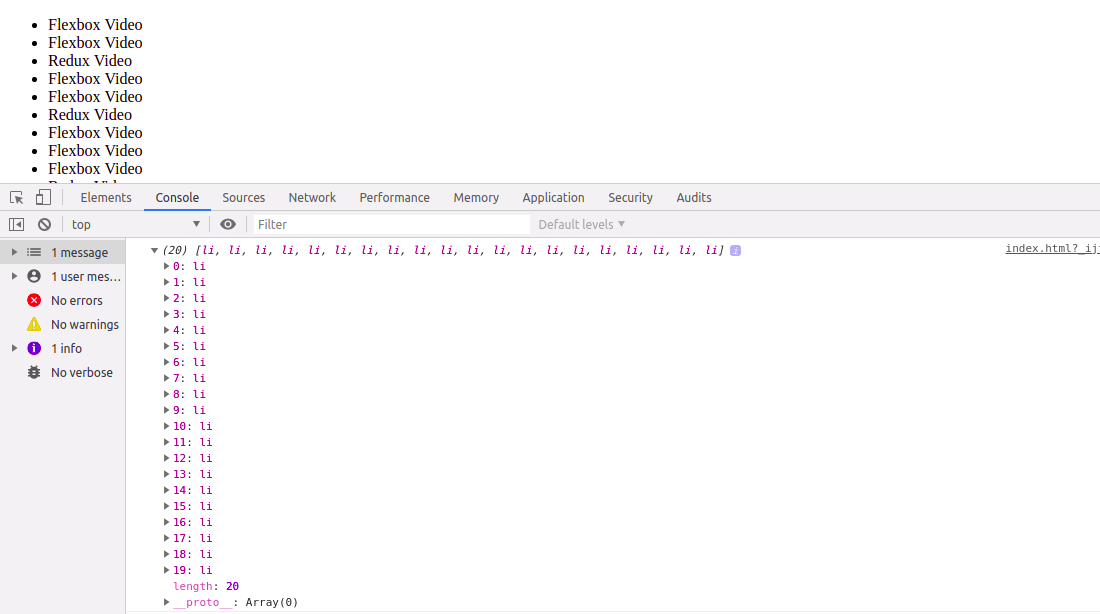
1. **Select all the list items on the page and convert to array.**
2. **Filter for only the elements that contain the word 'flexbox'**
3. **map down to a list of time strings**
4. **map to an array of seconds**
5. **reduce to get total using .filter and .map**

**Solution:**

**1)**

let items = Array.from(document.querySelectorAll('[data-time]'));

console.log(items);

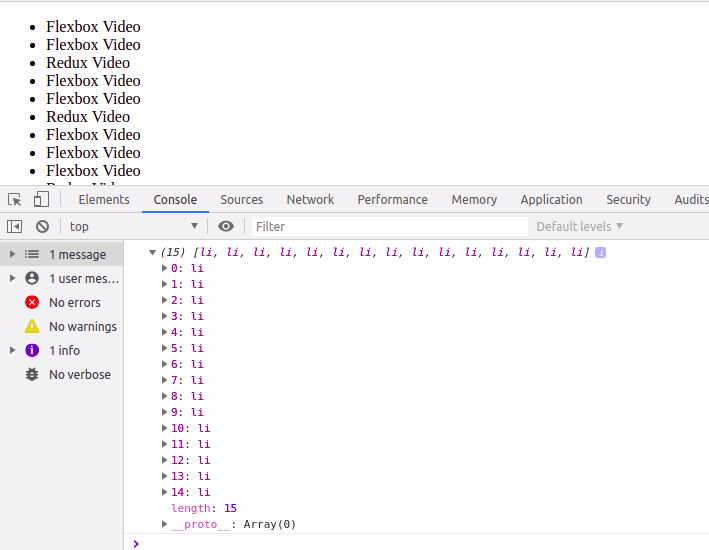


2)

let items = Array.from(document.querySelectorAll('[data-time]'));

let filtered = items.filter(item => item.textContent.includes('Flexbox'));

console.log(filtered);

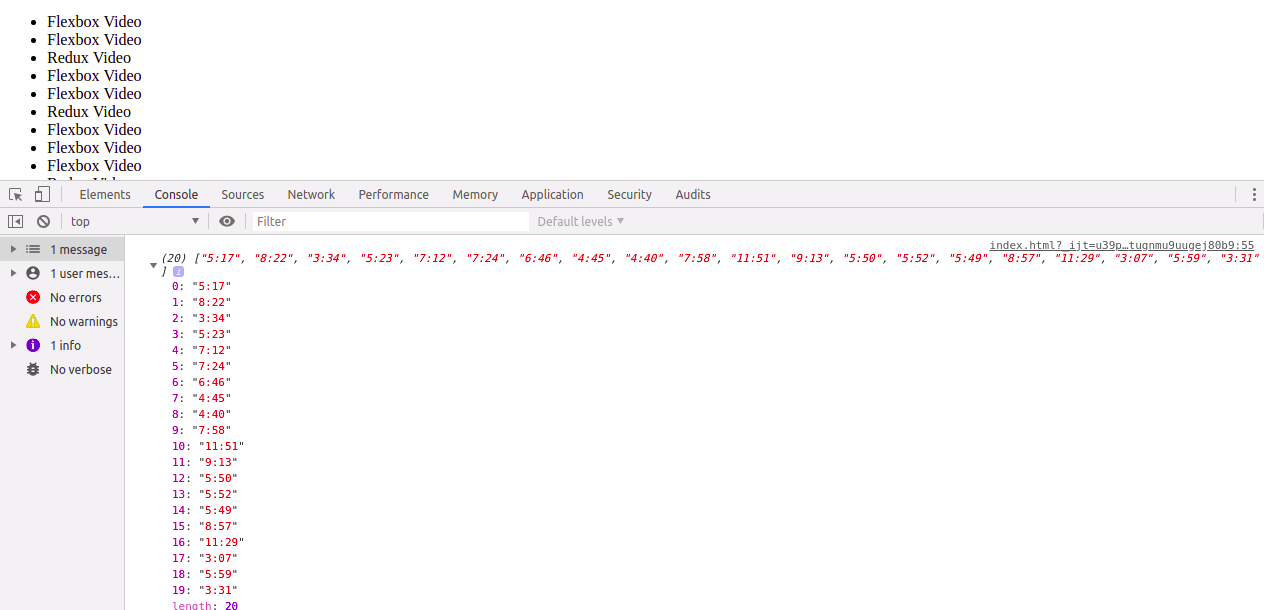


3)

let items = Array.from(document.querySelectorAll('[data-time]'));

let mapArr = items.map(item => item.dataset.time);

console.log(mapArr);



4)

let items = Array.from(document.querySelectorAll('[data-time]'));

let mapArr = items.map(item => item.dataset.time);

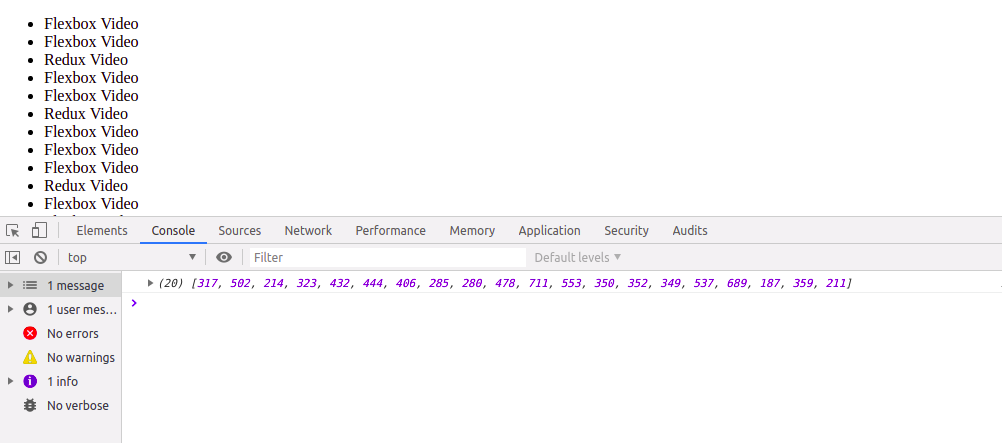
let arrSec = mapArr.map(timecode => {

let parts = timecode.split(':').map(part => parseFloat(part));

return (parts[0] \* 60) + parts[1];

})

console.log(arrSec);



5)

let items = Array.from(document.querySelectorAll('[data-time]'));

let mapArr = items.map(item => item.dataset.time);

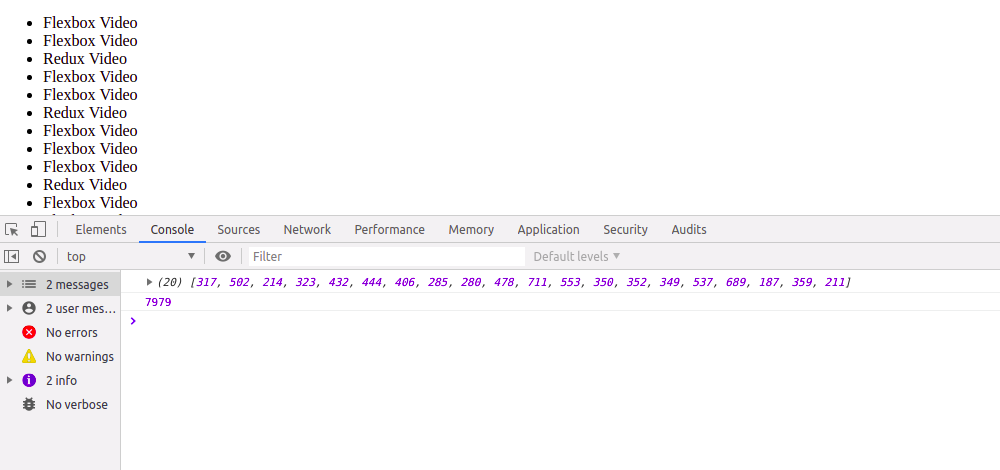
let arrSec = mapArr.map(timecode => {

let parts = timecode.split(':').map(part => parseFloat(part));

return (parts[0] \* 60) + parts[1]; })

console.log(arrSec);

let reduceArr = arrSec.reduce((runningTotal, seconds) => runningTotal + seconds); console.log(reduceArr);

****

**Question 3: Create a markup template using string literal**

**const song = {**

**name: 'Dying to live',**

**artist: 'Tupac',**

**featuring: 'Biggie Smalls'**

**};**

**Solution:**

const song = {

name: 'Dying to live',

artist: 'Tupac',

featuring: 'Biggie Smalls'

};

const markup = `<div class="song">

<p>

${song.name} - ${song.artist}

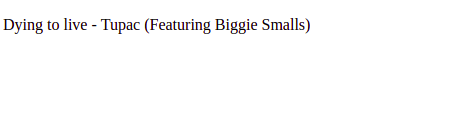
(Featuring ${song.featuring})

</p>

</div>

`

document.body.innerHTML = markup;



**Question 4: 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**

**}**

**Solution:**

****

**Problem in Question no . 2 part iv and v**