

ES6(Session 1) Exercise

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`.

```
const numbers = [3, 62, 234, 7, 23, 74, 23, 76, 92];
const largeNumbers = numbers.filter(number => number > 70);
console.log(largeNumbers);
```

Output:

```
[234, 74, 76, 92]
```

Q2.

<li data-time="5:17">Flexbox Video

<li data-time="8:22">Flexbox Video

<li data-time="3:34">Redux Video

<li data-time="5:23">Flexbox Video

<li data-time="7:12">Flexbox Video

<li data-time="7:24">Redux Video

<li data-time="6:46">Flexbox Video

<li data-time="4:45">Flexbox Video

<li data-time="4:40">Flexbox Video

<li data-time="7:58">Redux Video

<li data-time="11:51">Flexbox Video

<li data-time="9:13">Flexbox Video

<li data-time="5:50">Flexbox Video

```
<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.

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

```
<!DOCTYPE html>

<html>

<head>

  <title>Javascript</title>

</head>

<body>

  <ul>
```

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

```
<li data-time="5:59">Redux Video</li>
```

```
<li data-time="3:31">Flexbox Video</li>
```

```
</ul>
```

```
<p></p>
```

```
<script type="text/javascript">
```

```
// Select all the list items on the page and convert to array
```

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

```
// Filter for only the elements that contain the word 'Flexbox'
```

```
const filtered = items
```

```
.filter(item => item.textContent.includes('Flexbox'))
```

```
// map down to a list of time strings
```

```
.map(item => item.dataset.time)
```

```
// map to an array of seconds
```

```
.map(timecode => {
```

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

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

```
})
```

```
// reduce to get total
```

```
.reduce((Total, seconds) => Total + seconds);

console.log(filtered);

document.querySelector('p').textContent = `Total time for all the Flexbox
videos: ${filtered}s`;

</script>

</body>

</html>
```

Output:

- Flexbox Video
- Flexbox Video
- Redux Video
- Flexbox Video
- Flexbox Video
- Redux Video
- Flexbox Video
- Flexbox Video
- Flexbox Video
- Redux Video
- Flexbox Video
- Flexbox Video
- Flexbox Video
- Redux Video
- Flexbox Video
- Flexbox Video
- Flexbox Video
- Redux Video

- Flexbox Video

Total time for all the Flexbox videos: 6132s

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>  
“
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title></title>
```

```
</head>

<body>

  <script type="text/javascript">

    const song =

      {

        name: 'Dying to live',

        artist: 'Tupac',

        featuring: 'Biggie Smalls'

      };

    const markup = `

      <div class="song">

        <p>

          ${song.name} - ${song.artist} <br>

          ${song.featuring ? `(Featuring ${song.featuring})` : ""}

        </p>

      </div>

    `;

    document.body.innerHTML = markup;

  </script>

</body>

</html>
```

Output:

Dying to live - Tupac

(Featuring Biggie Smalls)

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 {firstName, lastName, Address:{Line1, Line2, State, Pin, Country, City},  
phoneNo} = user;  
  
document.write(firstName, lastName, Line1, Line2, State, Pin, Country, City,  
phoneNo);
```


Output:

Sahil Dua address line 1 address line 2 Delhi 110085 India New Delhi 9999999999