ES 6 (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`.

A.

```
let array=[3,62,234,7,23,74,23,76,92];
let greaterThanArray=array.filter((item)=> item>70);
console.log(greaterThanArray);
```

Code here

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 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 data-time="3:31">Flexbox Video
**

**
**

1. **Select all the list items on the page and convert to array.**
   A.

```
var items=document.querySelectorAll("li");

var arrayOfItems=Array.from(items);
```

Code here

2. **Filter for only the elements that contain the word 'flexbox'.**

A.

```
let items = document.querySelectorAll('li');

let itemsArray=Array.from(items);

let flexItems= itemsArray.filter( (item) =>
item.innerText.includes('Flexbox'));
```

Code here

3. **map down to a list of time strings**

A.

```
let arr=document.querySelectorAll('li');

let array = Array.from(arr);

let timeArray=array.map((a)=>{

 return a.getAttribute('data-time');

});

 console.log(timeArray);
```

Code here

4. **map to an array of seconds**
    A.

```
 let arr=document.querySelectorAll('li');

let array = Array.from(arr);

let timeArray=array.map((a)=>{

 var secArray = a.getAttribute('data-time').split(':');

  var sec=secArray[0]*60+Number(secArray[1]);

  return sec;

});

console.log(timeArray);
```

Code here

5. **reduce to get total using .filter and .map**

A.

```
let arr = document.querySelectorAll('li');

let array = Array.from(arr);

let timeArray = array.map((a) => {

   var secArray = a.getAttribute('data-time').split(':');

   var sec = secArray[0] * 60 + Number(secArray[1]);

   return sec;

});

var total = timeArray.reduce((a, b) => {

   return a + b;

});

console.log(total);
```

Code here

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>
"
```

A. Code here

```
var p=document.querySelector(".song p");
```

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

```
p.innerText=`${song.name} - ${song.artist}
(Featuring ${song.featuring})`;
```

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
}
```

```
A. var {Line1 ,Line2, State, Pin, Country, City}=user.Address;
```

Code here