

Weekly Progress Report

Text-to-Speech Web Extension using Python

Week: 1	Duration: 20/06/22 to 27/06/22	Group ID: M4	Guide: Siddharth Patel
----------------	---	-------------------------------	---

Group Members:

1. Bhakti Chokshi
2. Nupur Khatri
3. Bhavya Adhiya
4. Hasti Sutaria
5. Jaahanava Joshi

Project Description:

Most of us have to go through tons of articles and snippets of books before we understand a topic. Reading sure is fun, but becomes very demanding after a point.

What if you had someone narrate these articles at your desired speed so that you could go through them faster or listen to them while working on something else?

Text-to-speech web extension can enable this and allow readers to go through the content better.

Date	Work done
20/06/22	Introduction and had a discussion about project details with Slddharth sir.
21/06/22	Had a meeting with group members and guide to decide the project domain
22/06/22	Collected resources to learn Javascript.
23/06/22	Learnt about console logs, errors, warnings, etc. in Javascript
24/06/22	Learnt about variables and data types in Javascript
25/06/22	Learnt about type conversion and coercion, functions, for while and do while loops in Javascript
26/06/22	Learnt about arrays, objects, strings, methods and template literals, If Else conditionals in Javascript
27/06/22	Learnt how to use Javascript to make a pop-up

Research References:

Code with Harry - Javascript playlist:

https://www.youtube.com/playlist?list=PLu0W_9lII9ajyk081To1Cbt2eI5913SsL

W3 School - Javascript :

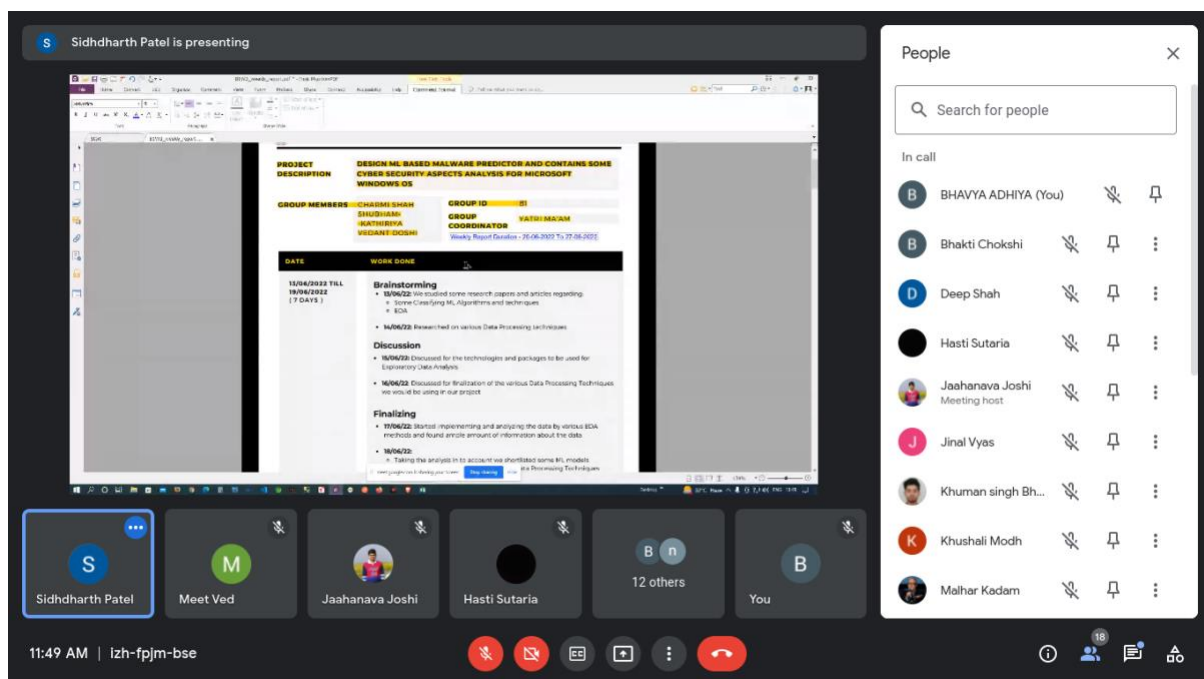
<https://www.w3schools.com/js/>

Free code camp - Javascript :

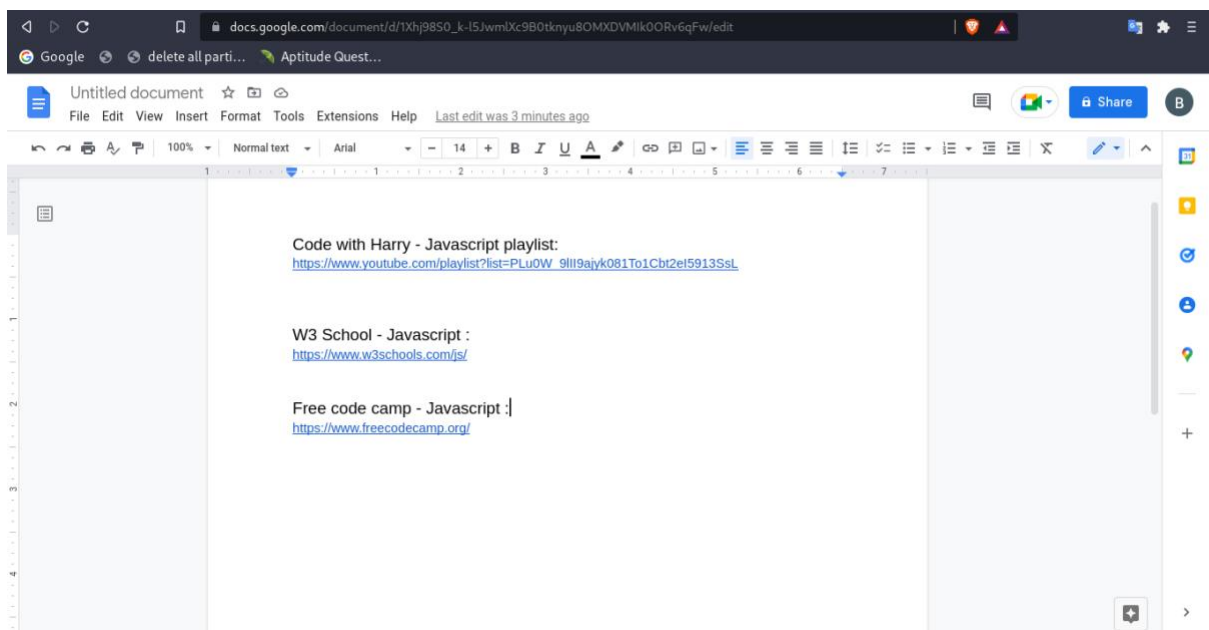
<https://www.freecodecamp.org/>

Proof of Work:

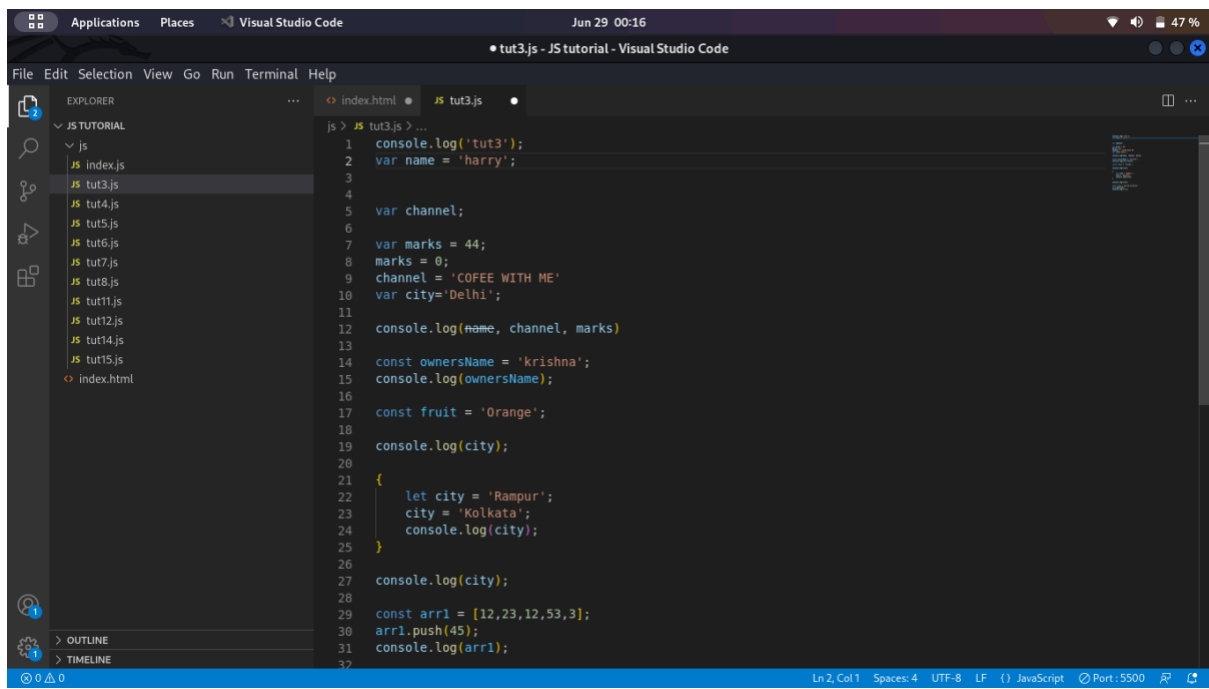
Screenshot 1: Introduction google meet



Screenshot 2: Javascript learning resources

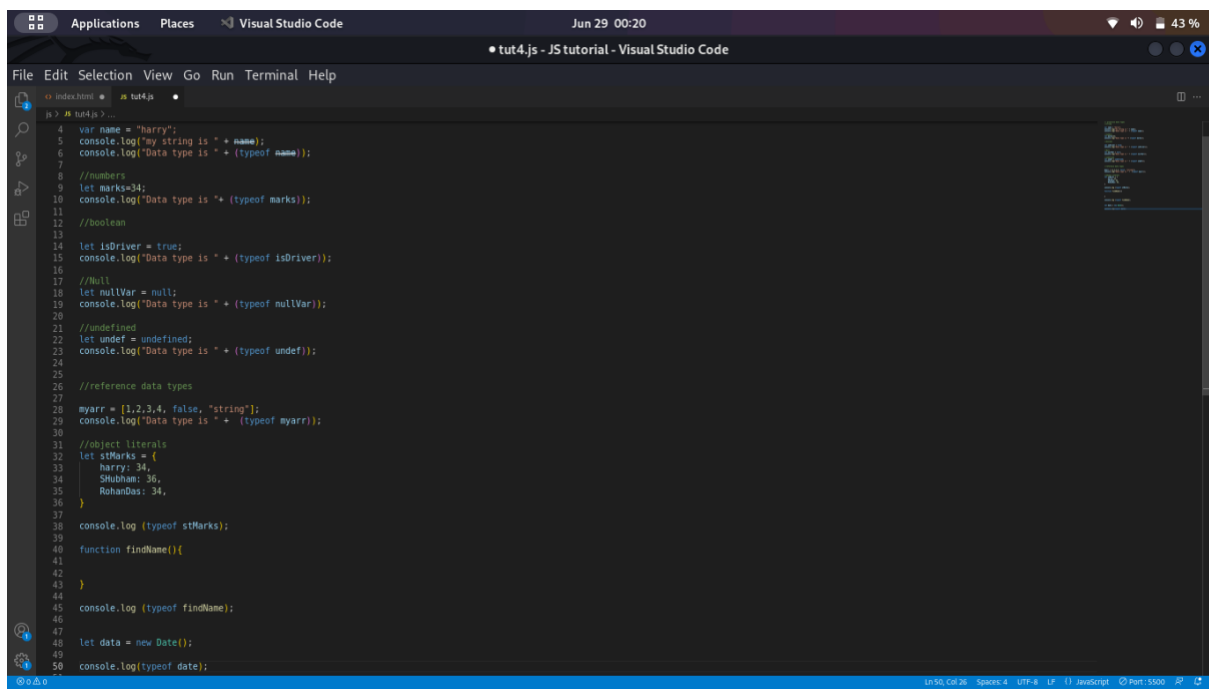


Screenshot 3: Learning variables and console logging in Javascript



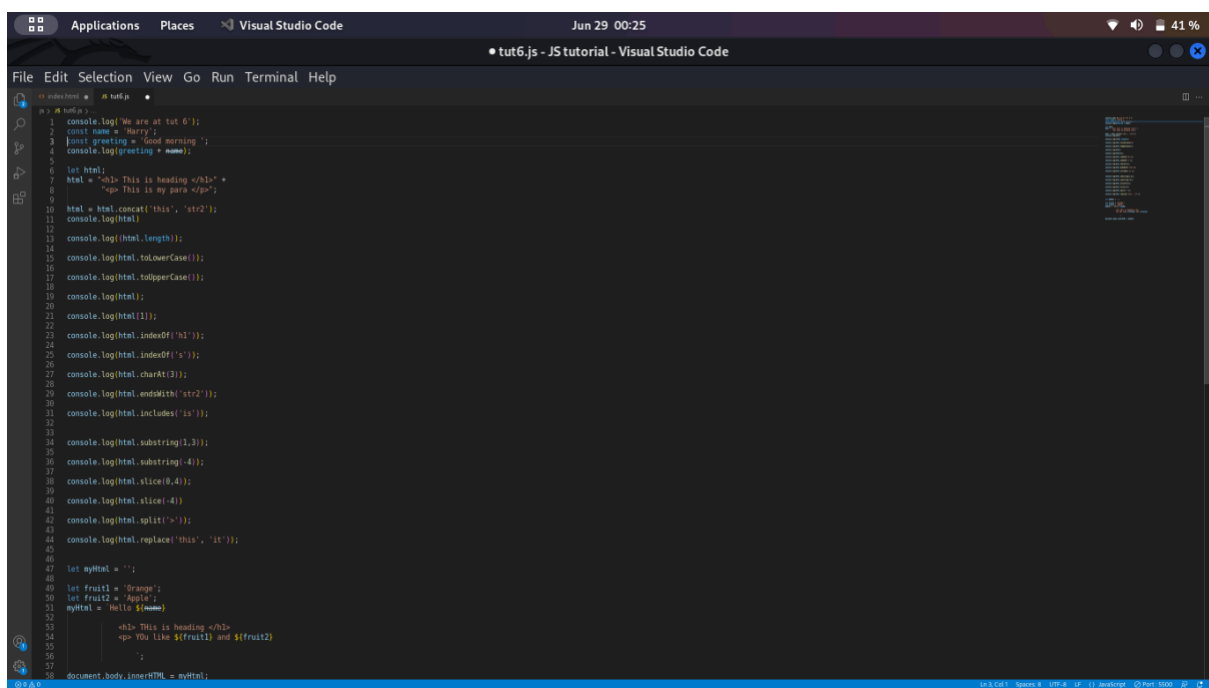
```
1 console.log('tut3');
2 var name = 'harry';
3
4
5 var channel;
6
7 var marks = 44;
8 marks = 0;
9 channel = 'COFFEE WITH ME';
10 var city='Delhi';
11
12 console.log(name, channel, marks)
13
14 const ownersName = 'krishna';
15 console.log(ownersName);
16
17 const fruit = 'Orange';
18
19 console.log(city);
20
21 {
22   let city = 'Rampur';
23   city = 'Kolkata';
24   console.log(city);
25 }
26
27 console.log(city);
28
29 const arr1 = [12,23,12,53,3];
30 arr1.push(45);
31 console.log(arr1);
32
```

Screenshot 4: Learning data types, object literals in Javascript



```
4 var name = "harry";
5 console.log("my string is " + name);
6 console.log("Data type is " + (typeof name));
7
8 //numbers
9 let marks=34;
10 console.log("Data type is " + (typeof marks));
11
12 //boolean
13
14 let isDriver = true;
15 console.log("Data type is " + (typeof isDriver));
16
17 //Null
18 let nullVar = null;
19 console.log("Data type is " + (typeof nullVar));
20
21 //undefined
22 let undef = undefined;
23 console.log("Data type is " + (typeof undef));
24
25 //reference data types
26
27
28 myarr = [1,2,3,4, false, "string"];
29 console.log("Data type is " + (typeof myarr));
30
31 //object literals
32 let stMarks = {
33   harry: 34,
34   Shubham: 36,
35   RohanDas: 34,
36 }
37
38 console.log (typeof stMarks);
39
40 function findName(){
41
42
43 }
44
45 console.log (typeof findName);
46
47
48 let data = new Date();
49
50 console.log(typeof date);
```

Screenshot 5 : String methods in Javascript



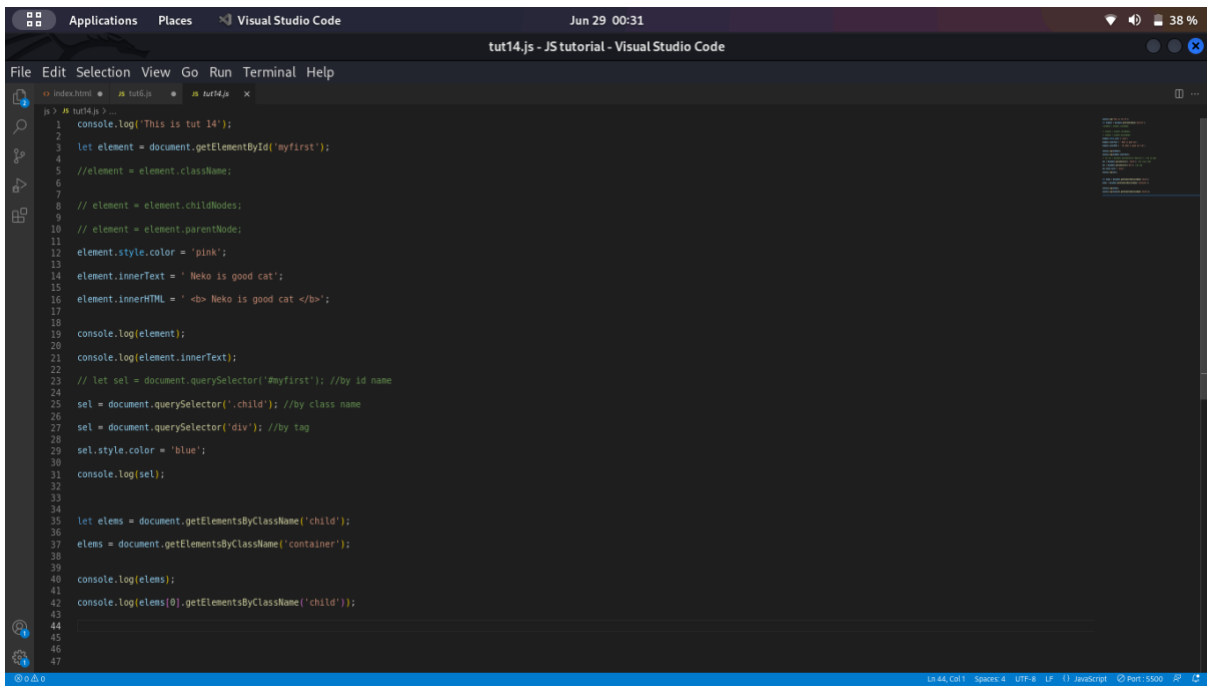
The screenshot shows the Visual Studio Code editor with a JavaScript file named 'tut6.js'. The code demonstrates various string methods and properties. The console output on the right shows the results of these operations.

```
1 console.log('We are at tut 6');
2 const name = 'Harry';
3 const greeting = 'Good morning';
4 console.log(greeting + name);
5
6 let html;
7 html = `<h1> This is heading </h1>` +
8       `<p> This is my para </p>`;
9
10 html = html.concat('this', 'str2');
11 console.log(html);
12 console.log(html.length);
13
14 console.log(html.toLowerCase());
15 console.log(html.toUpperCase());
16 console.log(html);
17 console.log(html[1]);
18 console.log(html.indexOf('h1'));
19 console.log(html.indexOf('s'));
20 console.log(html.charAt(3));
21 console.log(html.endsWith('str2'));
22 console.log(html.includes('s'));
23
24 console.log(html.substring(1,3));
25 console.log(html.substring(-4));
26 console.log(html.slice(0,4));
27 console.log(html.slice(-4));
28 console.log(html.split('>'));
29 console.log(html.replace('this', 'it'));
30
31 let myhtml = '';
32 let fruit1 = 'Orange';
33 let fruit2 = 'Apple';
34 myhtml = `Hello ${name}
35
36 <h1> This is heading </h1>
37 <p> You like ${fruit1} and ${fruit2}
38
39 `;
40 document.body.innerHTML = myhtml;
```

The console output on the right shows the following results:

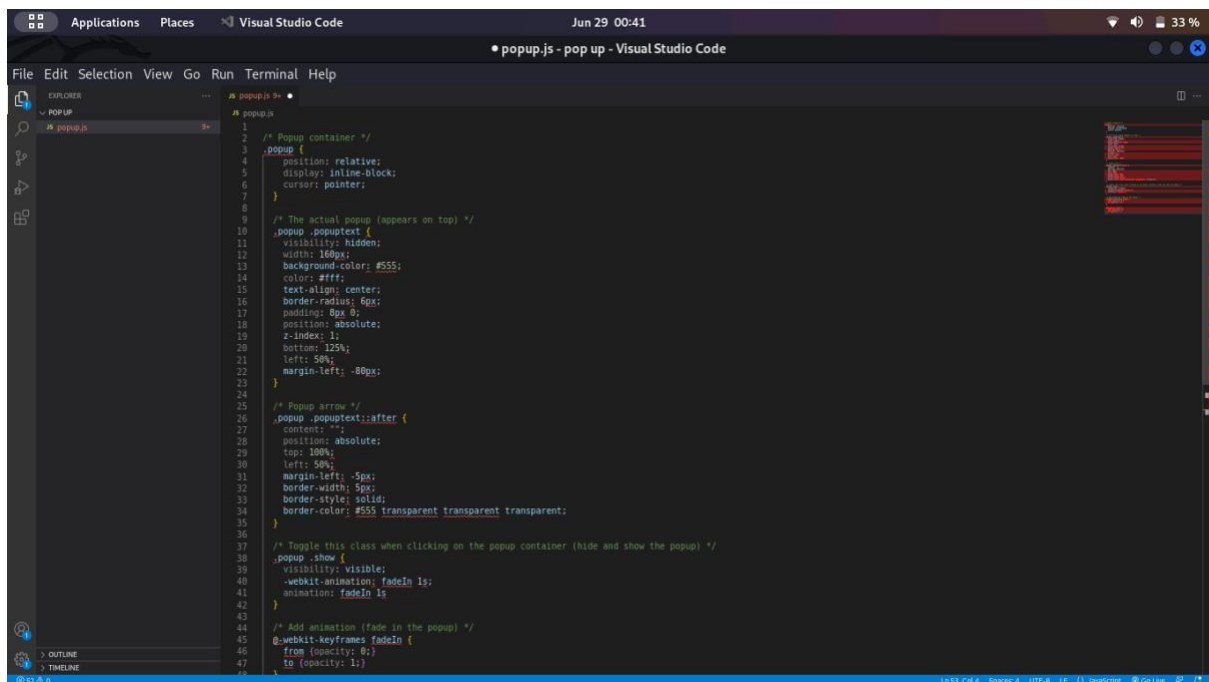
```
1 We are at tut 6
2 Good morningHarry
3
4 <h1> This is heading </h1> <p> This is my para </p>
5
6 <h1> This is heading </h1> <p> This is my para </p>thisstr2
7 41
8
9 <h1> this is heading </h1> <p> this is my para </p>
10 <h1> THIS IS HEADING </H1> <P> THIS IS MY PARA </P>
11 <h1> This is heading </h1> <p> This is my para </p>
12 41
13
14 <h1> this is heading </h1> <p> this is my para </p>
15 <h1> THIS IS HEADING </H1> <P> THIS IS MY PARA </P>
16 <h1> This is heading </h1> <p> This is my para </p>
17 h
18 1
19 4
20 11
21 h
22 true
23
24 h
25 1
26 0
27 4
28 ["<h1>", " ", "p>", " ", "this", "str2"]
29 <h1> it is heading </h1> <p> this is my para </p>
30
31 Hello Harry
32
33 <h1> This is heading </h1>
34 <p> You like Orange and Apple
35
36 </p>
37 </h1>
```

Screenshot 6 : element selection in Javascript



```
js > # tut14.js > ...  
1 console.log('This is tut 14');  
2  
3 let element = document.getElementById('myfirst');  
4  
5 //element = element.className;  
6  
7  
8 // element = element.childNodes;  
9  
10 // element = element.parentNode;  
11  
12 element.style.color = 'pink';  
13  
14 element.innerText = ' Neko is good cat';  
15  
16 element.innerHTML = ' <b> Neko is good cat </b>';  
17  
18  
19 console.log(element);  
20  
21 console.log(element.innerText);  
22  
23 // let sel = document.querySelector('#myfirst'); //by id name  
24  
25 sel = document.querySelector('.child'); //by class name  
26  
27 sel = document.querySelector('div'); //by tag  
28  
29 sel.style.color = 'blue';  
30  
31 console.log(sel);  
32  
33  
34 let elems = document.getElementsByClassName('child');  
35  
36 elems = document.getElementsByClassName('container');  
37  
38  
39  
40 console.log(elems);  
41  
42 console.log(elems[0].getElementsByClassName('child'));  
43  
44  
45  
46  
47
```


Screenshot 7: Using javascript to make pop up :

A screenshot of the Visual Studio Code editor interface. The top status bar shows 'Jun 29 00:41' and '33 %'. The title bar indicates the active file is 'popup.js - pop up - Visual Studio Code'. The editor is displaying JavaScript code for a popup. The code includes CSS styles for a container and a popup, and JavaScript functions to toggle the popup's visibility and add a fade-in animation. The code is as follows:

```
1 2
3  /* Popup container */
4  .popup {
5    position: relative;
6    display: inline-block;
7    cursor: pointer;
8  }
9
10 /* The actual popup (appears on top) */
11 .popup .popup-text {
12   visibility: hidden;
13   width: 160px;
14   background-color: #555;
15   color: #fff;
16   text-align: center;
17   border-radius: 6px;
18   padding: 8px 0;
19   position: absolute;
20   z-index: 1;
21   bottom: 125%;
22   left: 50%;
23   margin-left: -80px;
24 }
25
26 /* Popup arrow */
27 .popup .popup-text::after {
28   content: "";
29   position: absolute;
30   top: 100%;
31   left: 50%;
32   margin-left: -5px;
33   border-width: 5px;
34   border-style: solid;
35   border-color: #555 transparent transparent transparent;
36 }
37
38 /* Toggle this class when clicking on the popup container (hide and show the popup) */
39 .popup.show {
40   visibility: visible;
41   -webkit-animation: fadeIn 1s;
42   animation: fadeIn 1s;
43 }
44
45 /* Add animation (fade in the popup) */
46 @-webkit-keyframes fadeIn {
47   from {opacity: 0;}
48   to {opacity: 1;}
49 }
```

Plan for Week 2:

Making of pop up using java script for our web extension.

