



Usman Institute of Technology

Department of Computer Science

Course Code: SE319

Course Title: Web Engineering

Fall 2022

Lab 02

Objective:

This experiment is to become familiar with DOM and BOM & the APIs that may be used to manipulate them.

Student Information

Student Name	
Student ID	
Date	

Assessment

Marks Obtained	
Remarks	
Signature	

Usman Institute of Technology
Department of Computer Science
CS335 - Computer Graphics

Lab 02

Instructions

- Come to the lab in time. Students who are late more than 15 minutes, will not be allowed to attend the lab.
 - Students have to perform the examples and exercises by themselves.
 - Raise your hand if you face any difficulty in understanding and solving the examples or exercises.
 - Lab work must be submitted on or before the submission date.
-

1. Objective

This experiment is to become familiar with DOM and BOM & the APIs that may be used to manipulate them.

2. Theory

DOM stands for Dynamic Object Model. The HTML DOM defines a standard way for accessing and manipulating HTML documents. In an HTML document, it is the object presentation.

When HTML is parsed and rendered by a browser, a hierarchical memory representation of all HTML elements is generated and made available for manipulation known as the DOM. Every time a change is made to the DOM via API, a similar change is made to the actual HTML document and the document is re-rendered. The same applies if some changes are made directly to the HTML elements, it affects the counterpart DOM. The picture below depicts that how DOM is used to interact with HTML elements of the document via DOM Tree.

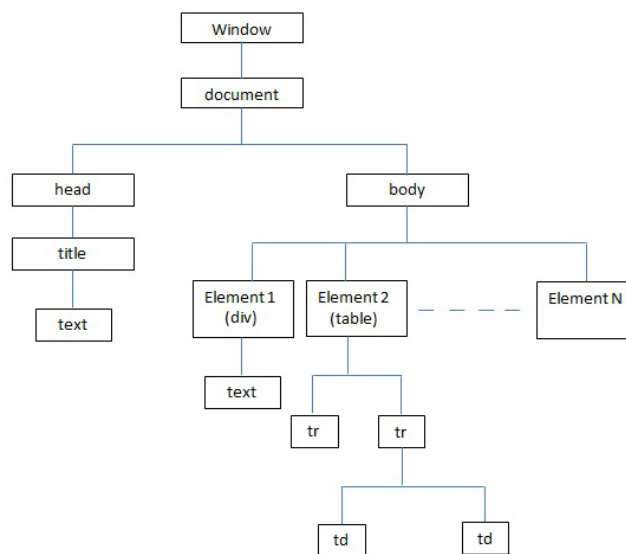


Figure 1: Simple HTML DOM Tree (Source: <http://www.authorcode.com/html-document-dom-tree/>)

The elements can be modified, added to, removed from, searched for, and appended as part of DOM manipulation. It can be used to alter the style, set the parent element's inner HTML, attach events to the DOM elements, etc. Document object provide access to the DOM API.

BOM (Browser Object Model) is used when we need to manipulate the browser features. BOM provide different ways to manipulate alerts, windows, timers, navigation, history, location, cookies etc. also their APIs to manipulate them.

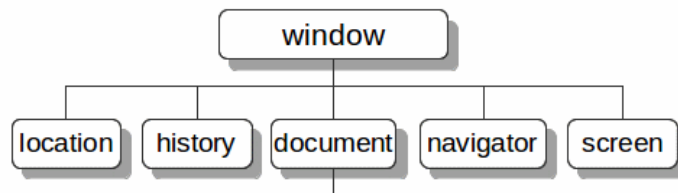


Figure 2: Browser Objects (Source: <https://www.4stud.info/web-programming/browser-object-model.html>)

3. DOM – Selecting Elements

For selection and searching, below are the following DOM APIs are available that are used frequently;

1. *document.getElementById*
This is used to select any element with in the DOM using element Id.
2. *document.getElementsByName*
This is used to select elements within the DOM using element by Name.
3. *document.getElementsByTagName*
This is used to select elements within the DOM using element by Tag Name.
4. *document.getElementsByClassName*
This is used to select elements within the DOM using element by one or more class names.
5. *document.querySelector*
This is used to select elements within the DOM using element by CSS selectors.

Task 1

```
<html>
  <head>
    <title>Lab 2 - DOM & BOM</title>
  </head>
  <body>
    <p id="message">A paragraph</p>

    <script>
      const p = document.getElementById('message');
      console.log(p);
    </script>
  </body>
</html>
```

Task 2

```

<!DOCTYPE html>
<html>
<head>
  <title>Lab 2 - DOM & BOM</title>
</head>
<body>
  <p>Please rate the service:</p>
  <p>
    <label for="very-poor">
      <input type="radio" name="rate" value="Very poor"
id="very-poor"> Very poor
    </label>
    <label for="poor">
      <input type="radio" name="rate" value="Poor" id="poor">
Poor
    </label>
    <label for="ok">
      <input type="radio" name="rate" value="OK" id="ok"> OK
    </label>
    <label for="good">
      <input type="radio" name="rate" value="Good"> Good
    </label>
    <label for="very-good">
      <input type="radio" name="rate" value="Very Good"
id="very-good"> Very Good
    </label>
  </p>
  <p>
    <button id="btnRate">Submit</button>
  </p>
  <p id="output"></p>
  <script>
    let btn = document.getElementById('btnRate');
    let output = document.getElementById('output');

    btn.addEventListener('click', () => {
      let rates = document.getElementsByName('rate');
      rates.forEach((rate) => {
        if (rate.checked) {
          output.innerHTML = `You selected:
${rate.value}`;
        }
      });
    });
  </script>

```

```
    });  
  </script>  
</body>  
  
</html>
```

Task 3

```
<!DOCTYPE html>  
<html>  
  <head>  
    <title>Lab 2 - DOM & BOM</title>  
  </head>  
  <body>  
    <h1>JavaScript getElementsByTagName() Demo</h1>  
    <h2>First heading</h2>  
    <p>This is the first paragraph.</p>  
    <h2>Second heading</h2>  
    <p>This is the second paragraph.</p>  
    <h2>Third heading</h2>  
    <p>This is the third paragraph.</p>  
  
    <button id="btnCount">Count H2</button>  
  
    <script>  
      let btn = document.getElementById('btnCount');  
      btn.addEventListener('click', () => {  
        let headings = document.getElementsByTagName('h2');  
        alert(`The number of H2 tags: ${headings.length}`);  
      });  
    </script>  
  </body>  
  
</html>
```

Task 4

```
<!DOCTYPE html>  
<html lang="en">  
  <head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>Lab 2 - DOM & BOM</title>  
  </head>  
  <body>
```

```

<header>
  <nav>
    <ul id="menu">
      <li class="item">HTML</li>
      <li class="item">CSS</li>
      <li class="item highlight">JavaScript</li>
      <li class="item">TypeScript</li>
    </ul>
  </nav>
  <h1>getElementsByClassName Demo</h1>
</header>
<section>
  <article>
    <h2 class="secondary">Example 1</h2>
  </article>
  <article>
    <h2 class="secondary">Example 2</h2>
  </article>
</section>
<script>
  let menu = document.getElementById('menu');
  let items = menu.getElementsByClassName('item');

  let data = [].map.call(items, item => item.textContent);

  console.log(data);
</script>
</body>
</html>

```

Task 5

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Lab 2 - DOM & BOM</title>
</head>
<body>
  <div id="main">
    <p class="note">This is a note!</p>
  </div>

  <script>
    let note = document.querySelector('.note');

```

```

        console.log(note.parentNode);
    </script>
</body>
</html>

```

Task 6

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Lab 2 - DOM & BOM</title>
</head>
<body>
    <ul id="menu">
        <li>Home</li>
        <li>Products</li>
        <li class="current">Customer Support</li>
        <li>Careers</li>
        <li>Investors</li>
        <li>News</li>
        <li>About Us</li>
    </ul>

    <script>
        let getSiblings = function (e) {
            debugger;
            // for collecting siblings
            let siblings = [];
            // if no parent, return no sibling
            if(!e.parentNode) {
                return siblings;
            }
            // first child of the parent node
            let sibling = e.parentNode.firstChild;
            // collecting siblings
            while (sibling) {
                if (sibling.nodeType === 1 && sibling !== e) {
                    siblings.push(sibling);
                }
                sibling = sibling.nextSibling;
            }
            return siblings;
        };
        debugger;
    </script>

```

```

    let siblings =
getSiblings(document.querySelector('.current'));
    siblingText = siblings.map(e => e.innerHTML);
    console.log(siblingText);
  </script>
</body>
</html>

```

Task 7

```

<!DOCTYPE html>
<html>
  <head>
    <title>Lab 2 - DOM and BOM</title>
  </head>
  <body>
    <table id="studentNames" class="students">
      <thead>
        <tr>
          <th>Student #</th>
          <th>Student Name</th>
          <th>Batch</th>
        </tr>
      </thead>
      <tbody>
        <tr id="row1" class="row">
          <td>Student 1</td>
          <td>Abdul Hafeez</td>
          <td>2021</td>
        </tr>
        <tr id="row2" class="row">
          <td>Student 2</td>
          <td>Arsalan Latif</td>
          <td>2019</td>
        </tr>
        <tr id="row3" class="row">
          <td>Student 3</td>
          <td>Muzammil Ali</td>
          <td>2018</td>
        </tr>
      </tbody>
    </table>
    <script>
      //document.getElementById - Search by element's ID
      console.log(document.getElementById('studentNames'));
    </script>
  </body>
</html>

```



```
//document.getElementsByClassName - Search by element's CSS
class name
console.log(document.getElementsByClassName('row').length);

//document.getElementsByTagName - Search by element's tag
name
console.log(document.getElementsByTagName('row').length);

//document.querySelector - Generic search using any of CSS
selector
console.log(document.querySelector('#row1'));

//Search element using ID
console.log(document.querySelector('#row3'));

//Search element using ID
console.log(document.querySelector('.row'));

//Search first element from document top using CSS class
name
console.log(document.querySelectorAll('tr').length);

</script>
</body>
</html>
```

4. DOM – Manipulating Elements

To manipulate DOM, we have the following APIs that are used frequently;

1. *createElement*

This method is used to create a new HTML element and attach it to the DOM tree.

2. *appendChild*

This method is used to add a node to the end of list of child nodes of a specified parent node.

3. *removeChild*

This method is used to remove a child node from a parent node.

4. *innerHTML*

This is the property of an element to get or set HTML markup contained in the element. Using this, either text or HTML can be updated as its content.

Task 8

```
<!DOCTYPE html>
<html>
  <head>
    <title>Lab 2 – DOM and BOM</title>
  </head>
```

```

<body>
  <table id="studentNames" class="students">
    <thead>
      <tr>
        <th>Student #</th>
        <th>Student Name</th>
        <th>Batch</th>
      </tr>
    </thead>
    <tbody>
      <tr id="row1" class="row">
        <td>Student 1</td>
        <td>Abdul Hafeez</td>
        <td>2021</td>
      </tr>
      <tr id="row2" class="row">
        <td>Student 2</td>
        <td>Arsalan Latif</td>
        <td>2019</td>
      </tr>
      <tr id="row3" class="row">
        <td>Student 3</td>
        <td>Muzammil Ali</td>
        <td>2018</td>
      </tr>
    </tbody>
  </table>

  <br>

  <input type="button" value="Add New Student"
onclick="addNewStudent();" />
  <input type="button" value="Delete Student"
onclick="deleteStudent();" />
  <input type="button" value="Update Student"
onclick="updateStudent();" />

  <script>
    //This declarative function will be called on 'Add New
    Row' button click and add new row
    function addNewStudent(){
      //1 - Find the 'body' element
      const body = document.querySelector('tbody');
      //2 - Create elements using document.createElement
API (method)

```

```

        const row = document.createElement('tr');
        //Create 'tr' element
        const tdStudentID = document.createElement('td');
        //Create 'td' element
        const tdStudentName = document.createElement('td');
        //Create 'td' element
        const tdBatch = document.createElement('td');
        //Create 'td' element
        const studentId = ((Math.round(Math.random() *
100)) + 4);
        const studentName = ((Math.round(Math.random() *
500)) + 50);
        const batch = (Math.round(Math.random() * 2020)) +
2020;

        row.id = 'row' + studentId;
        //3 - Update td's 'innerHTML' (basically element
content - anything within the angle brackets)
        tdStudentID.innerHTML = 'Student ' + studentId;
        tdStudentName.innerHTML = 'Student - DOM - ' +
studentName;
        tdBatch.innerHTML = batch;

        //4 - Insert newly created 'td's into newly created
'tr' using element's appendChild API
        row.appendChild(tdStudentID);
        row.appendChild(tdStudentName);
        row.appendChild(tdBatch);

        //5 - Finally insert newly created 'row' into
'body' element
        body.appendChild(row);
    }

    function deleteStudent() {
        //Ask for course# using prompt
        const studentId = prompt('Please enter course#');
        //If user has entered course#
        if(studentId) {
            //Find the row containing the 'course #'
entered by the user
            const row = document.getElementById('row' +
studentId);

```

```

        if(row) {
            //Find the parent of the 'row' just found
            //so that it can be removed from its child list
            const body =
document.querySelector('tbody');
            //Finally remove it from 'tbody' element
            body.removeChild(row);
        }
    }
}

</script>
</body>
</html>

```

5. BOM – Manipulating Browser Objects

Task 9

```

<!DOCTYPE html>
<html>
  <head>
    <title>Lab 2 - DOM and BOM</title>
  </head>
  <body>
    <div>
      <h1>Window</h1>
      <input type="button" value="Open window"
onclick="openWindow()" />
    </div>

    <div>
      <h1>Location</h1>
      <input type="button" value="Goto Location"
onclick="gotoLocation()" />
    </div>

    <div>
      <h1>Navigator</h1>
      <input type="button" value="Browser Name"
onclick="showBrowserName()" />
      <input type="button" value="Browser Language"
onclick="showBrowserLanguage()" />
    </div>
  </body>
</html>

```

```

    <div>
        <h1>Screen</h1>
        <input type="button" value="Screen Size"
onclick="showScreenSize()" />
        <input type="button" value="Screen Orientation"
onclick="showScreenOrientation()" />
    </div>

    <div>
        <h1>History</h1>
        <h3 style="color:red">Note: To move backward, first
visit any site and then load your page and click Backward button.
To move forward, first load your page and then visit any site and
comeback to your page and click Forward button</h3>
        <input type="button" value="Forward"
onclick="moveForward()" />
        <input type="button" value="Backward"
onclick="moveBackward()" />
    </div>

    <script>
        //Open a new window and point it to google.com
        function openWindow(){
            //Ask from the user to enter delay before opening the
window using window.prompt
            const delay = window.prompt('Enter the delay in seconds
to wait before opening the window');
            //Wait 2 seconds (2000 milliseconds) using
'window.setTimeout' before opening the window
            window.setTimeout(function(){
                //Show the window using 'window.open'
                window.open('https://www.google.com');
                //Finally, show the message using 'window.alert'
for success
                window.alert('New window has been opened'); },
parseInt(delay) * 1000);
        }

        //Goto to location
        function gotoLocation(){
            //Ask the user to enter the url she wants to replace
current page with
            const url = window.prompt('Enter the url to visit to
(start with http(s))?');

```

```
        if(url) {
            //Visit the url provided by the user within the
            same page using window.location
            window.location = url;
        }
    }

    //Show browser name
    function showBrowserName() {
        window.alert(window.navigator.appName);
    }

    //Show browser language
    function showBrowserLanguage(){
        window.alert(window.navigator.language);
    }

    //Show screen size
    function showScreenSize(){
        window.alert(`screen width: ${window.screen.width},
screen height: ${window.screen.height}`);
    }

    //Show screen orientation
    function showScreenOrientation(){
        window.alert(window.screen.orientation.type);
    }

    //Move forward in browser history from the current position
    function moveForward(){
        window.history.forward();
    }

    //Move backward in browser history from the current
    position
    function moveBackward(){
        window.history.back();
    }
    </script>
</body>
</html>
```

6. Lab Task

1. Extend the Task 8 code and write a function to update a student batch year in the studentTable.
 - a. Ask the Student # and Batch from the user using prompt alert. (You can ask from user in comma separated form)
 - b. If users enter the data correctly, split the comma separated input.
 - c. Find the row containing the 'student #' entered by the user.
 - d. Now, find the 'batch' td element with in the 'row' entered by the user.
 - e. Finally update the batch entered by the user using 'innerHTML' property of element.

7. Homework Tasks