2020-2

웹 시스템 설계

학습과제

Week 4

JavaScript & DOM

Composed by:

WISE Research Lab Ajou University

학습목표

1. 기초적인 JavaScript 활용 방법에 대해 학습한다.
2. JavaScript DOM API 를 이용해서 HTML 문서의 DOM 을 동적으로 변경하는 방법을 학습한다.

I. JavaScript

1. JavaScript 정의

- Web Browser에서 HTML 문서의 동적 스타일, 사용자 이벤트 처리 등을 위해 사용되는 스크립트 언어 (최근 Node.js과 같은 서버 사이드 플랫폼에서도 사용되고 있음)
- Interpret 언어이며, JavaScript 엔진에 의해 해석(실행)됨
- JavaScript의 문법과 기능은 ECMAScript라는 표준 언어 명세를 기반으로 개발됨
  - ✓ JavaScript 개요에 대한 추가 정보는 아래 문서를 참고한다.
    https://developer.mozilla.org/ko/docs/Web/JavaScript/JavaScript_technologies_overview
  - ✓ Airbnb에서 제안하는 JavaScript Style Guide (코딩 스타일 가이드)
    https://github.com/airbnb/JavaScript

2. JavaScript 와 HTML 문서의 연동방법

- HTML 문서에 직접 작성

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Lab3 JS Example 1</title>
    <script>
        "use strict";
        window.onload(function() {
            // 변수 선언
            var number = 0;
            // JavaScript는 semicolon(;) 생략 가능
            var btnNode = document.getElementById("btn-count")

            // 함수 선언
            function onClickBtn() {
                ++number;
                document.getElementById("state").innerHTML = number;
            }
            btnNode.addEventListener("click", onClickBtn);
        })
    </script>
</head>
<body>
    <p>Counter example</p>
    <button id="btn-count" type="button">Count!</button>
    <p id="state">0</p>
</body>
</html>
```

- 외부 스크립트 파일 불러오기

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Lab3 JS Example 2</title>
    <script src="Lab3-201824417.js"></script>
</head>
<body>
    <p>Counter example</p>
    <button id="btn-count" type="button">Count!</button>
    <p id="state">0</p>
</body>
</html>
```

II. Document Object Model(DOM)

1. Document Object Model(DOM) 정의

- 문서(HTML, XML)의 각 element들이 브라우저에 로딩 되면서 객체화 된 것
- DOM은 스크립트가 문서(HTML)의 내용, 구조, 스타일 등에 접근/변경을 하기 위해 사용하는 프로그래밍 인터페이스이다.
  - ✓ DOM은 JavaScript의 일부분이 아닌, 브라우저에서 언어/플랫폼 중립적으로 제공하는 DOM API를 JavaScript에서 호출하여 사용
- DOM은 문서의 구조화된 표현을 제공해준다.

2. DOM API examples

```
<section id="my-root">
        <article>Article 1</article>
        <article>Article 2</article>
        <article id="article-mid">Article 3</article>
        <article class="special one">Article 4</article>
        <article class="special two">Article 5</article>
</section>
```

예제에서 사용될 <body> 내부 요소들

- API 설명에서 "node"란 DOM tree에 계층적 구조로 생성되어 있는 메모리 객체를 의미

  (즉, node 는 DOM tree 에 저장되어 있는 Element 의 객체)

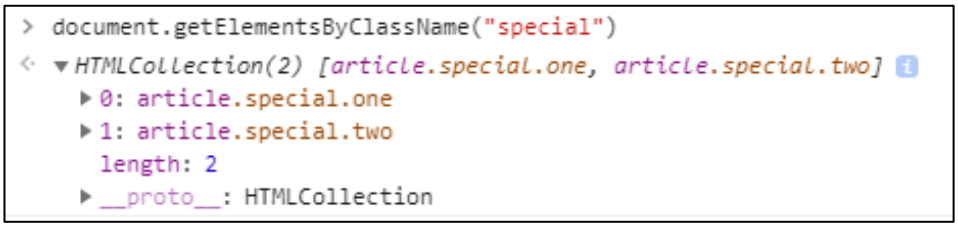
- document.getElementById(id) – 해당 id와 일치하는 node 를 반환함

```
> document.getElementById("my-root")
<- ▶<section id="my-root">…</section>
```

  - ✓ 다수의 Node가 아닌 하나의 node만 반환하는 이유
    - 원칙적으로 id는 고유(Unique)한 값이며, id가 중복되는 일은 없어야 한다.

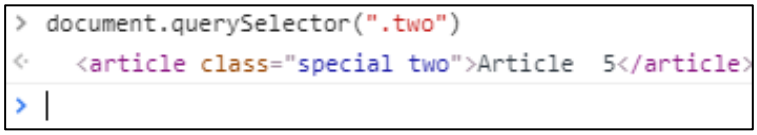
- document.getElementsByTagName(tagname) – Element의 태그가 매개변수 tagname과 일치하는 모든 node를 배열로 반환함

- document.getElementsByClassName(classname) – 매개변수 classname과 일치하는 Class를 가지고 있는 모든 node를 배열로 반환함

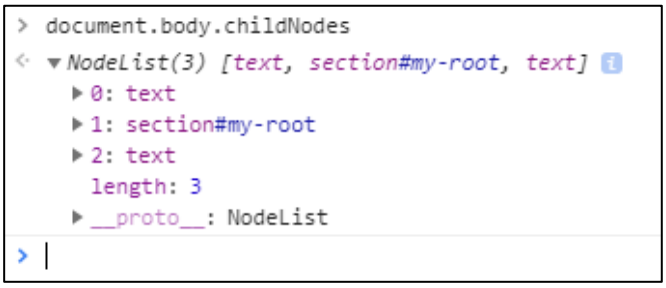


- document.querySelector(selector) – id 또는 class가 일치하는 node를 찾아 반환

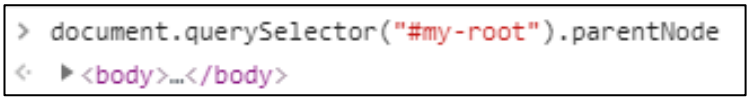
  ✓ 매개변수는 CSS의 selector 문법 형태로 지정함 ("#id", ".class")



- element.childNodes – 해당 element 객체의 하위에 존재하는 모든 자식 node를 반환



- element.parentNode – 해당 element 객체의 부모인 node객체를 반환



- document.createElement(tagname) – 매개변수 tagname의 tag를 가지는 새로운 node 객체를 생성하여 반환함

```
> let newDiv = document.createElement("div")
< undefined
> newDiv
<   <div></div>
```

- element.innerHTML – 해당 node의 html 내용을 가져오거나 수정함
  - ✓ innerHTML 데이터 타입은 문자열이다.

```
> let newDiv = document.createElement("div")
< undefined
> newDiv.innerHTML = "new div"
< "new div"
> newDiv
<   <div>new div</div>
```

- appendChild(selector) – 해당 node의 하위에 새로운 node를 추가함

```
> let newDiv = document.createElement("div")
< undefined
> document.querySelector("#my-root").appendChild(newDiv)
<   <div></div>
> document.querySelector("#my-root")
< ▼<section id="my-root">
    <article>Article  1</article>
    <article>Article  2</article>
    <article id="article-mid">Article  3</article>
    <article class="special one">Article  4</article>
    <article class="special two">Article  5</article>
    <div></div>
  </section>
```

3. Event Listener

- 특정 이벤트에 대해 대기하고 있다가, 해당 이벤트 발생 시에 실행되는 함수
  - ✓ 즉, 사용자의 특수한 행위(버튼 클릭, 텍스트 입력, 특정 위치에 커서 올리기 등)에 따라 사용자와 상호작용을 할 수 있다.
- Event Listener의 사용 예제

HTML

```html
<!-- 버튼 -->
<button id="btn-count"> Button1 </button>
```

JavaScript

```javascript
// id가 "btn-count"인 node를 가져옴
let buttonNode = document.querySelector( selectors: '#btn-count');

// 버튼의 클릭 이벤트에 listener 등록
buttonNode.addEventListener( type: "click", listener: function () {
    // 버튼의 class 속성값에 새로운 class를 추가
    buttonNode.classList.add("bg=black");

    //id가 "some-id"인 node의 하위내용 수정
    document.querySelector( selectors: '#some-id').innerHTML = "Like this";
})
```

- Event Listener을 등록할 수 있는 이벤트 타입 종류
  - ✓ "click" : Element가 클릭 되었을 때
  - ✓ "dblclick" : Element가 연속으로 두 번 클릭 되었을 때
  - ✓ "mouseover" : 커서가 Element의 위에 올라왔을 때 (해당 Element의 하위 자식들의 범위까지 포함)
  - ✓ 그 외에도 다양하게 발생할 수 있는 이벤트들에 대한 상호작용을 정의할 수 있다.
- DOM의 설명과 API 예제는 아래 문서의 내용을 참고하여 작성되었으며, 추가 정보는 해당 문서들을 참고하도록 한다.
  - ✓ DOM: https://developer.mozilla.org/ko/docs/Web/API/Document_Object_Model
  - ✓ DOM API: https://www.w3schools.com/js/js_htmldom_document.asp
  - ✓ Event : https://developer.mozilla.org/ko/docs/Web/Events

III. 학습과제

1. 과제 개요

- DOM API 로 웹 페이지 내용을 수정하여 사용자가 쓴 글(즉, 메모)이 추가될 수 있으며, 메모의 현황을 보여주는 메모장을 만들어 본다.
- 메모장이 하는 기능은 아래와 같다.
  - ✓ 글을 쓰고 Append Text를 누르면 내가 작성한 글의 메모가 메모장에 기록된다.
  - ✓ 내가 작성한 글의 개수를 보여준다.

2. 학습과제 파일 구성 및 과제 제출 양식

- Lab4.html : HTML 구조를 만들고, Lab4.css와 Lab4.js를 불러와서 적용시킨 HTML 문서
- Lab4.css : Desktop화면과 Mobile 화면에 대응하는 각 Element의 크기, 색상, 폰트 등의 style 속성값을 정의한 파일
- Lab4.js : 웹 페이지를 수정하는데 필요한 변수와 함수 등의 script를 정의한 JavaScript 파일
- 위의 파일을 하나의 폴더 내에서 생성하고, 폴더(프로젝트)의 이름을 "Lab4_자신의학번"으로 생성한다.
- 압축 파일의 이름을 "Lab4_자신의학번.zip"으로 지정하여 폴더(프로젝트)를 압축한다.

3. 제출 방법 및 주의 사항

- 학습과제와 같이 업로드 되는 학습보고서의 답안을 작성하여 프로젝트(폴더)의 압축파일과 함께 제출한다.
- 지각 제출 시, 0점으로 처리 한다.
- 채점 시 완성도의 평가는 Google의 **Chrome** 브라우저에서 렌더링 된 화면을 기준으로 함

※ **"4 프로그램 설계"**, **"5 프로그램 실행 예시"**를 참고하여 프로그램이 올바르게 동작하도록 구현한다.

4 프로그램 설계

4.1 HTML(Lab4.html) 문서

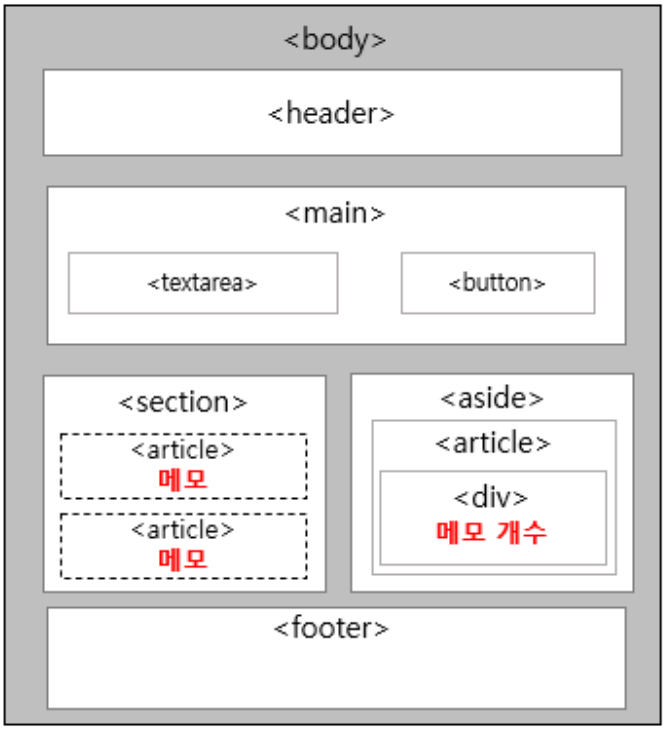


그림 1 HTML문서의 레이아웃

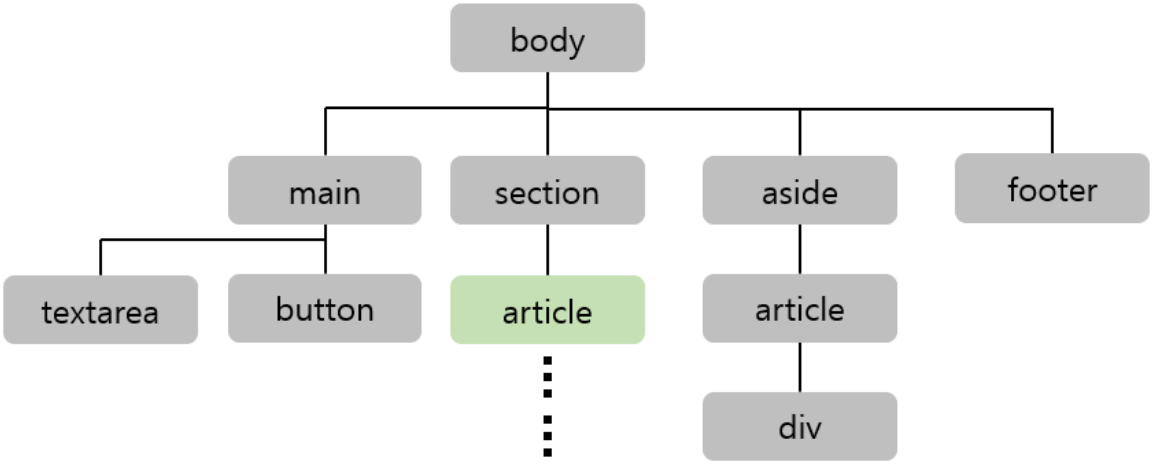


그림 2 HTML의 DOM 객체 트리

- Element의 구조는 그림 2와 동일하게 구성하도록 한다.
- HTML 문서의 제목은 'Lab4-자신의 학번'으로 지정한다.

- **<head>** 태그
  - ✓ 태그 내에 <title>Lab4-201912345</title> 와 같이 작성
  - ✓ Lab4.css를 불러온다.
- **<main>** 태그
  - ✓ 메모에 들어갈 글을 작성하는 **<textarea>** element를 가진다.
    - cols 속성값은 "50", rows 속성은 "3" 값을 가진다.
  - ✓ 메모 추가 기능을 수행하는 **<button>** element를 가진다.
    - 버튼을 누르면 **<textarea>** 내에 작성한 글을 포함하는 node 객체를 생성하여 **<section>** 에 추가한다.
    - 이 때, 생성된 node 객체의 tag name는 ** 혹은 <article>**로 한다.
    - 생성된 node 객체의 class attribute값은 "5.2의 작성한 메모를 위한 CSS 설정"을 참조하여 작성한다.
    - 단, 작성한 글이 없으면 (즉, textarea가 비어 있으면) 메모가 추가되어서는 안된다.
- **<section>** 태그
  - ✓ 생성된 메모 element를 자식 노드로 가진다.
- **<aside>** 태그
  - ✓ **<article>** element를 가진다.
    - 'Number of Text Memo'가 되도록 작성한다.
    - 메모의 개수를 나타내는 ** ** element를 가진다.
- **<footer>** 태그
  - ✓ 'Implemented by [자신의 이름], [자신의 학번]'가 되도록 작성한다.
- **<section>** 태그 내에 메모 외에는 다른 element가 없도록 한다.
  - ✓ 생성된 node 객체(메모)를 자식 노드로 추가한다.
- **<body>** 태그 내의 가장 마지막에 **<script>**태그를 사용하여 Lab4.js 파일을 불러온다.
  - ✓ <head>가 아닌 <body> 내의 가장 마지막에서 JavaScript 파일을 불러오는 이유는 HTML DOM load가 완료되어 element들을 저장하는 변수들의 값이 결정된 후에

script가 그 변수들을 사용할 수 있도록 하기 위함이다.

4.2 CSS(Lab4.css) 문서

- lab4.css 는 "**3주차 학습 과제"**의 "**CSS 요구조건**"을 유지하되 다음의 요소를 추가로 적용하도록 한다.
- <body>, <section>과 작성한 메모가 서로 다른 background-color를 가지도록 한다.
- 작성한 메모에 대한 CSS 설정
  - ✓ 모든 메모가 margin-bottom 값을 가지도록 하여 서로 구분될 수 있도록 한다.
  - ✓ 메모의 width를 지정하여 <section>의 범위를 벗어나지 않도록 한다.
  - ✓ 작성한 메모들의 height가 <section>의 height 보다 커지면 scroll 하여 모두 볼 수 있도록 한다.
- 요구사항에 따라 레이아웃을 구성하되 padding, margin 및 폰트 크기와 관련된 속성값은 자유롭게 지정한다.

4.3 JavaScript(Lab4.js) 문서

- 메모 추가 **<button>** 클릭 시 실행되는 **Event Listener**
  - ✓ 혹은 <article> tag name을 가진 객체를 생성한다.
  - ✓ 생성된 객체의 값을 <textarea> element의 값으로 지정한다.
    - document.getElementById(id).value을 사용하여 Input Text의 값을 가져온다.
    - element.innerHTML 사용하여 값을 지정한다.
  - ✓ 생성된 객체를 <section> 에 추가한다.
    - element.appendChild(selector)를 사용하여 추가한다.
  - ✓ 생성된 객체의 수를 <aside>내에 <article>내에 element의 값으로 지정한다.
    - element.innerHTML 사용하여 값을 지정한다.
  - ✓ 단, 작성한 글이 없으면 (즉, textarea가 비어 있으면) 메모가 추가되어서는 안된다.

※ EventListener 등록하는 방법

- Inline Style
  - ✓ 클릭 이벤트가 발생하면 호출할 함수 지정

```
<button id="my-btn" onclick="doSomething()">
```

- Add EventListener (Recommended)

  ✓ JavaScript 에서 해당 element 에 EventListener 등록

```
document.querySelector("#my-btn").addEventListener("click", function () {

        // Do Something…
})
```

5. 프로그램 실행 예시

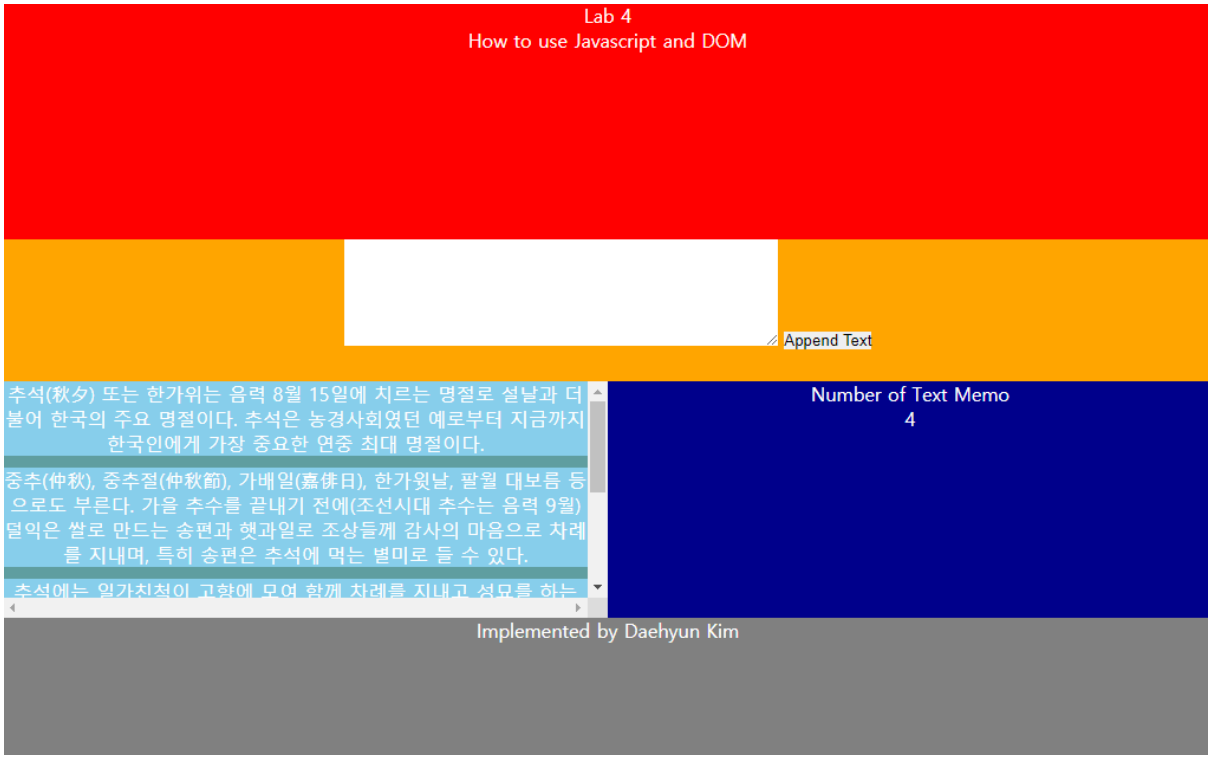


그림 2 메모 추가 후 실행결과(768px 이상, 메모 4개 추가)

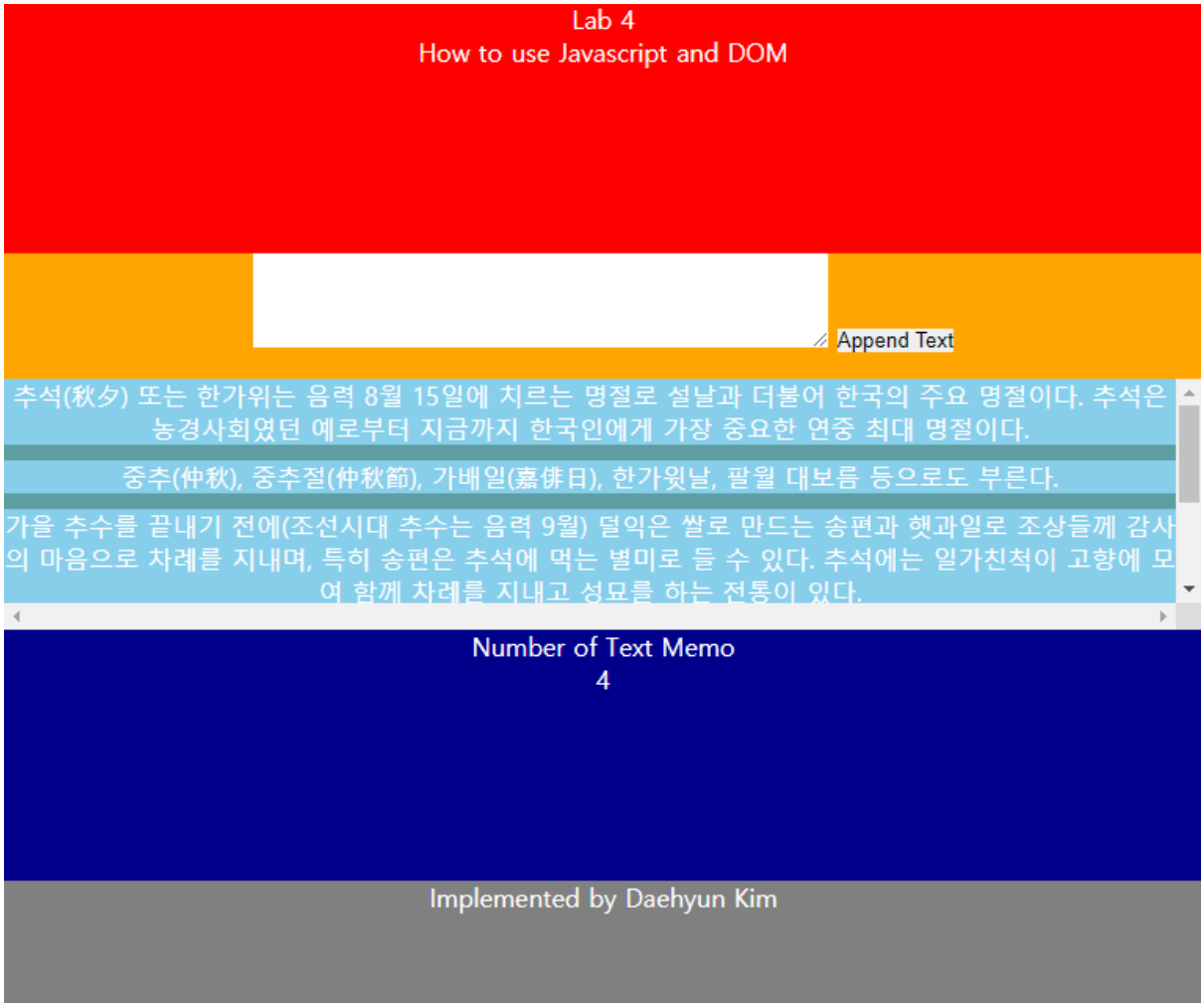


그림 3 메모 추가 후 실행결과(768px 미만, 메모 4개 추가)