L8-Document

예제 실습 및 보고서 작성



학 과 : 컴퓨터공학전공

과 목 : 웹프로그래밍

교수님 : 한영선 교수님

학 번 : 202030494

이 름 : 윤원재

제출일 : 2023년 05월 29일

1. **예제 8-1 DOM 객체의 구조 출력 : p 객체 사례**

**<!DOCTYPE html>**

**<html>**

**<head><title>**HTML DOM 트리**</title></head>**

**<body>**

**<h3>**DOM 객체 p의 프로퍼티, 스타일, 이벤트 리스너**</h3>**

**<hr>**

**<p id="firstP"**

**style="color:blue; background: yellow"**

**onclick="this.style.color='teal'">**

이것은 <span style="color:red">문장입니다.**</span>**

**</p>**

**<script>**

var p = document.getElementById("firstP");

var text = "p.id = " + p.id + "\n";

text += "p.tagname = " + p.tagname + "\n";

text += "p.innerHTML = " + p.innerHTML + "\n";

text += "p.style.color = " + p.style.color + "\n";

text += "p.onclick = " + p.onclick + "\n";

text += "p.childElementCount = " + p.childElementCount + "\n";

text += "너비 = " + p.offsetWidth + "\n";

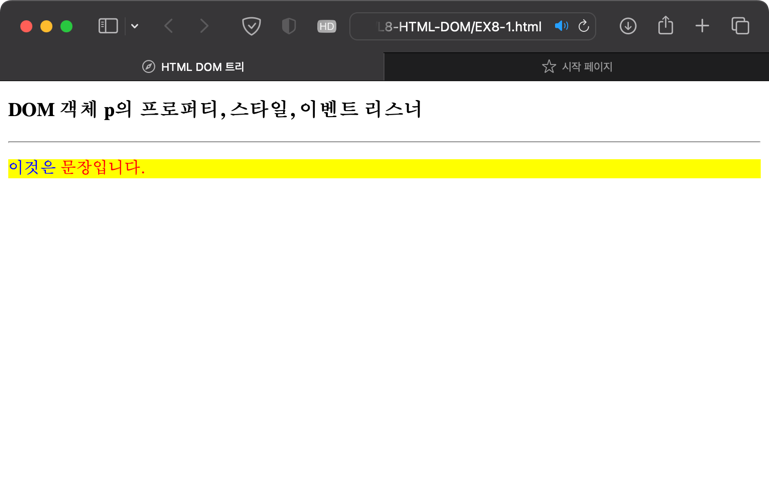
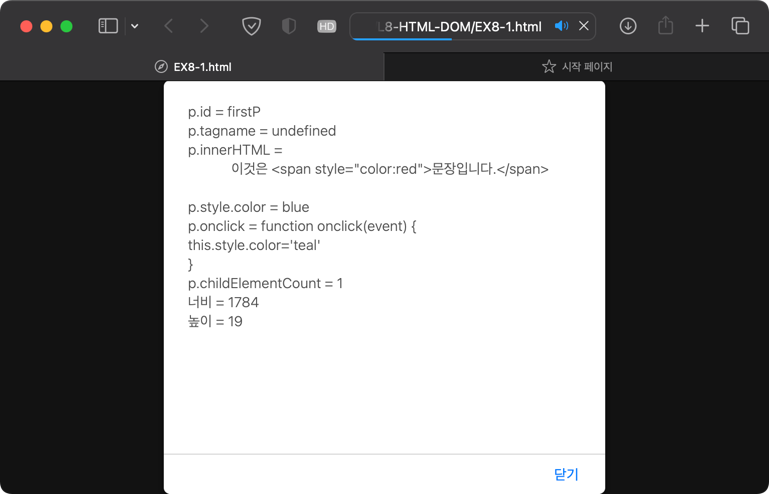
text += "높이 = " + p.offsetHeight + "\n";

alert(text);

**</script>**

**</body>**

**</html>**

****

<그림 1-1> 예제 8-1 결과

DOM 객체를 사용하여 태그의 구조를 출력할 수 있다. <그림 1-1>에서 “document.getElementById ("firstP")” 코드로 id가 firstP인 <p> 태그를 DOM 객체로 하여 내부 속성 값을 불러온 모습을 확인할 수 있다. 객체에는 프로퍼티, 스타일, 이벤트 리스너 등이 포함된다.

1. **예제 8–2 <span>의 CSS3 스타일 동적 변경**

**<!DOCTYPE html>**

**<html>**

**<head><title>**CSS 스타일 동적 변경**</title>**

**<script>**

function change() {

var span = document.getElementById("mySpan");

span.style.color = "green";

span.style.fontSize = "30px";

span.style.display = "block";

span.style.width = "6em";

span.style.border = "3px dotted magenta";

span.style.margin = "20px";

**}**

**</script>**

**</head>**

**<body>**

**<h3>**CSS 스타일 동적 변경**</h3>**

**<hr>**

**<p style="color:blue">**

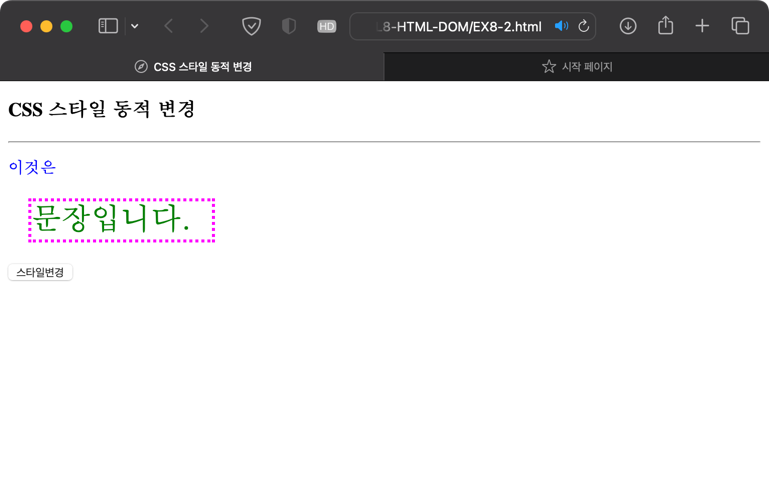
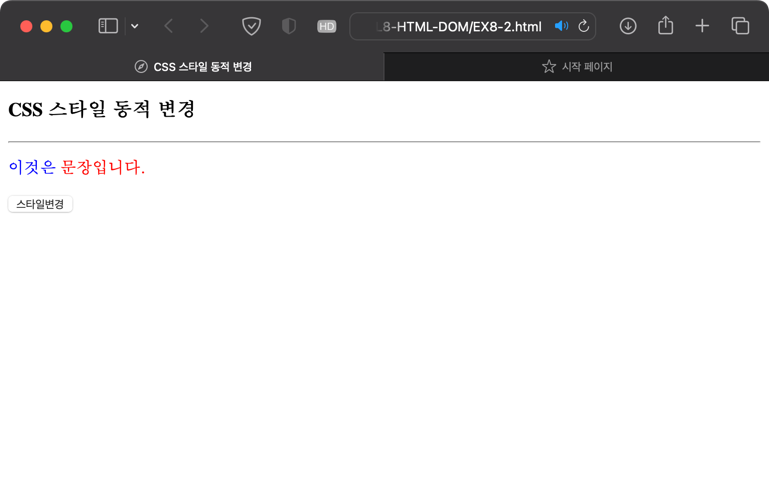
이것은 <span id="mySpan" style="color:red">문장입니다.**</span>**

**</p>**

**<input type="button" value="스타일변경" onclick="change()">**

**</body>**

**</html>**

****

<그림 2-1> 예제 8-2 결과

DOM 객체의 CSS 스타일을 변경하는 예제이다. <그림 2-1>에서 id가 mySpan인 <span> 태그를 “span.style.\*”로 스타일을 변경하는 것을 확인할 수 있다. 해당 변경은 change() 함수를 통해 실행되며 change()함수는 ‘스타일변경’ 버튼을 눌렀을 때 실행된다.

1. **예제 8-3 innerHTML을 이용하여 HTML 콘텐츠 동적 변경**

**<!DOCTYPE html>**

**<html>**

**<head><title>**innerHTML 활용**</title>**

**<script>**

function change() {

var p = document.getElementById("firstP");

p.innerHTML = "나의 <img src='puppy.jpg' width='150'> 강아지";

}

**</script>**

**</head>**

**<body>**

**<h3>**innerHTML 활용 : 클릭하면 강아지가 보입니다.**</h3><hr>**

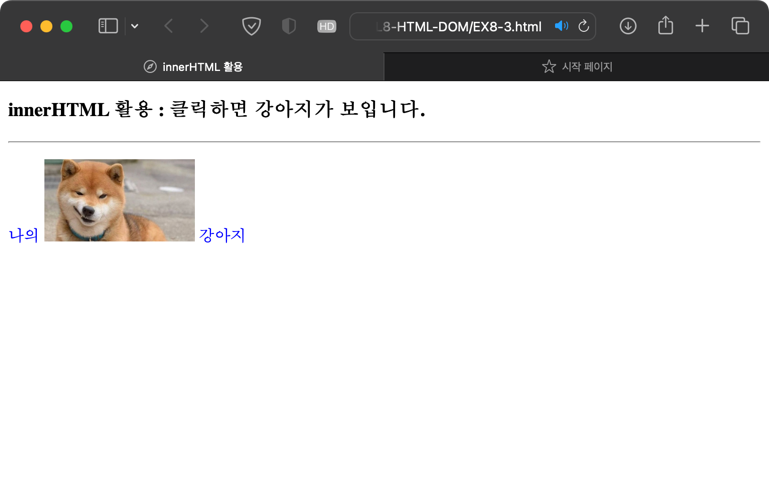
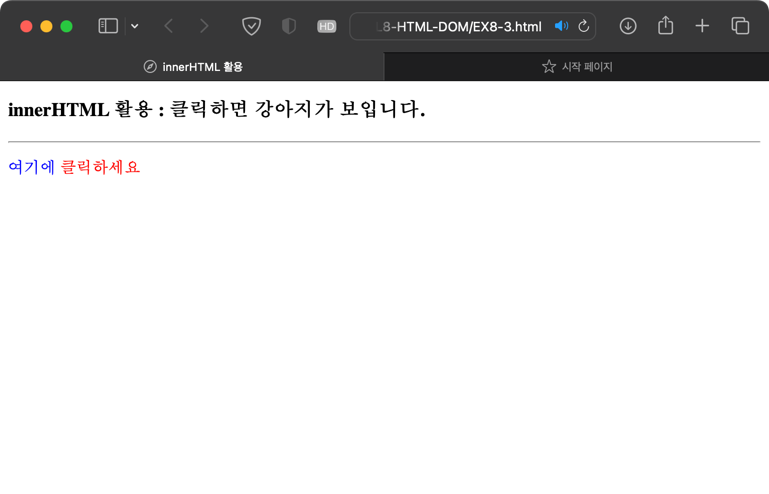
**<p id="firstP" style="color: blue;" onclick="change()">**

여기에 **<span style="color: red;">** 클릭하세요**</span>**

**</p>**

**</body>**

**</html>**

****

<그림 3-1> 예제 8-3 결과

innerHTML을 이용하면 HTML의 콘텐츠, 즉, 내용을 변경할 수 있다. <그림 3-1>에서 innerHTML의 사용으로 ‘여기에 클릭하세요’ 라는 문구가 <img> 태그를 포함한 ‘나의 강아지’로 변경된 것을 확인할 수 있다.

1. **예제 8-4 this 활용**

**<!DOCTYPE html>**

**<html>**

**<head><title>**this 활용**</title>**

**<script>**

function change(obj, size, color) {

obj.style.color = color;

obj.style.fontSize = size;

}

**</script>**

**</head>**

**<body>**

**<h3>**this 활용**</h3><hr>**

**<button onclick="change(this, '30px', 'red')">**버튼**</button>**

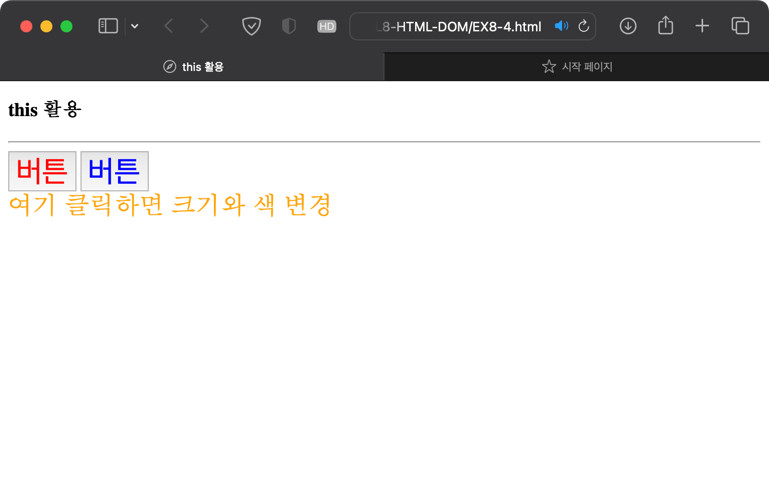
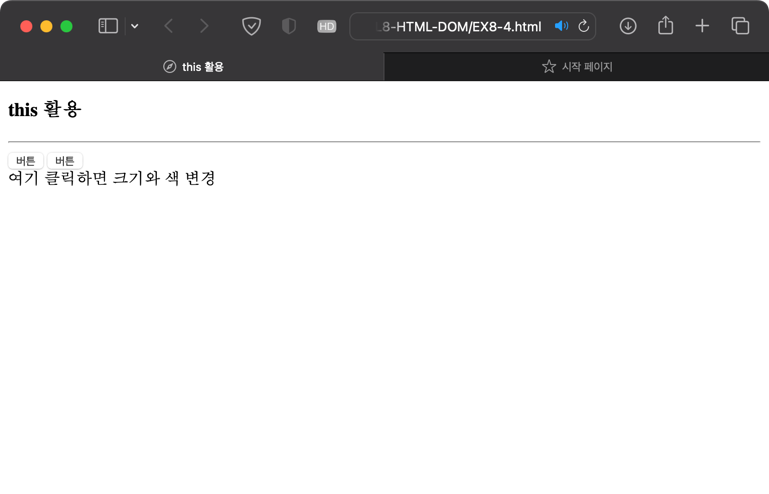
**<button onclick="change(this, '30px', 'blue')">**버튼**</button>**

**<div onclick="change(this, '25px', 'orange')">**

여기 클릭하면 크기와 색 변경**</div>**

**</body>**

**</html>**

****

<그림 4-1> 예제 8-4 결과

자바스크립트에서 this는 객체 자신을 가리키는 키워드이다. <그림 4-1>에서 change() 함수의 파라미터로 this를 받았는데 이는 객체 자신을 의미한다. 다시말해, 첫번째 버튼(빨간색)의 경우 change() 함수로 ‘this’, ‘30px’, ‘red’를 보내서 버튼의 글자 크기가 ‘30px’, 글자 사이즈가 ‘red’로 바뀐 것을 확인할 수 있다.

1. **예제 8-5 document 객체의 프로퍼티 출력**

**<!DOCTYPE html>**

**<html>**

**<head id="myHead">**

**<title>**document 객체의 주요 프로퍼티**</title>**

**<script>**

var text = '문서 로딩 중일 때 readyState = ' + document.readyState + '\n';

**</script>**

**</head>**

**<body style="background-color: yellow; color: blue; direction: rtl;" onload="printProperties()">**

**<h3>**document의 주요 프로퍼티**</h3><hr>**

**<a href="http://www.naver.com">**네이버 홈페이지**</a>**

**<div>**이곳은 div 영역입니다.</**div>**

**<input id="input" type="text" value="여기 포커스가 있습니다"/>**

**<script>**

// 문서가 완전히 로드(출력)되었을 때, 현재 document의 프로퍼티 출력

function printProperties() {

document.getElementById('input').focus(); // <input> 태그에 포커스를 줌

text += '1. location =' + document.location + '\n';

text += '2. URL =' + document.URL + '\n';

text += '3. title =' + document.title + '\n';

text += '4. head의 id =' + document.head.id + '\n';

text += '5. body color =' + document.body.style.color + '\n';

text += '6. domain =' + document.domain + '\n';

text += '7. lastModified =' + document.lastModified + '\n';

text += '8. defaultView = ' + document.defaultView + '\n';

text += '9. 문서의 로드 완료 후 readyState = ' + document.readyState + '\n';

text += '10. referrer = ' + document.referrer + '\n';

text += '11. activeElement = ' + document.activeElement.value + '\n';

text += '12. documentElement의 태그 이름 = ' + document.documentElement.tagName + '\n';

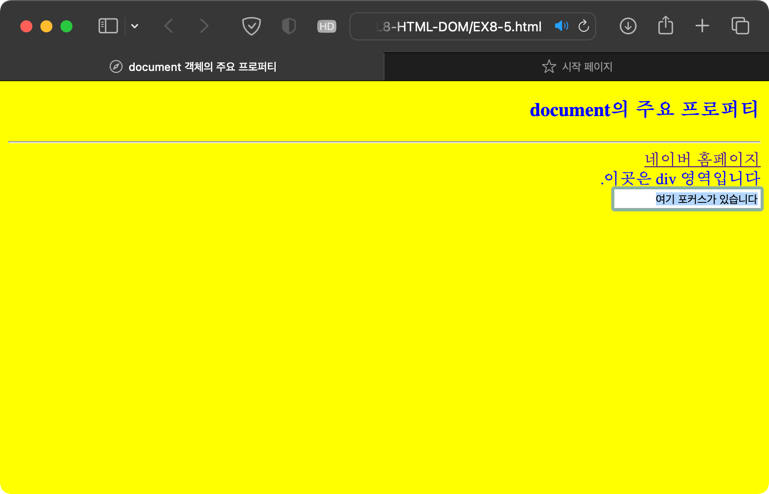
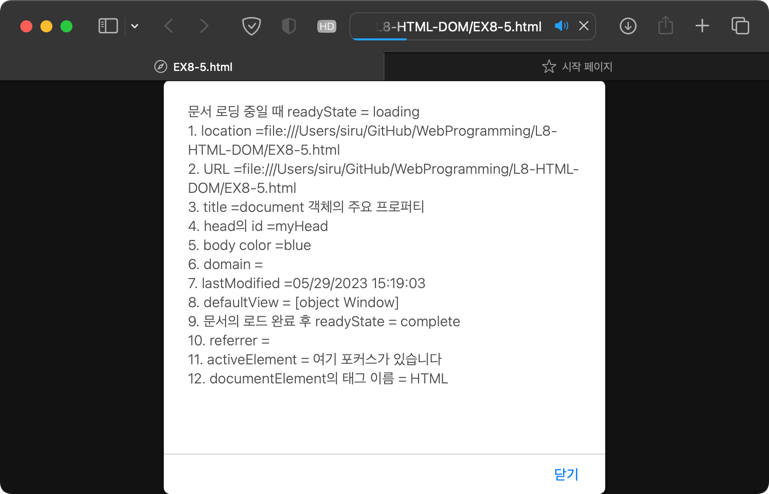
alert(text);

}

**</script>**

**</body>**

**</html>**

****

<그림 5-1> 예제 8-5 결과

<그림 5-1>에서 <body>태그가 실행되기 전에 ‘printproperties()’ 메소드가 실행된다. 이 메소드는 이HTML문서의 document객체에 대한 프로퍼티를 출력하는 메소드이다. 그림에서 오른쪽의 HTML에 대한 프로퍼티를 해당 문서가 실행되기 전인 왼쪽 그림에서 alert()를 통해 띄우는 것을 확인할 수 있다.

1. **예제 8-6 태그 이름으로 DOM 객체 찾기**

**<!DOCTYPE html>**

**<html>**

**<head><title>**document.getElementsByTagName()</**title>**

**<script>**

function change() {

var spanArray = document.getElementsByTagName('span');

for (var i = 0; i < spanArray.length; i++) {

var span = spanArray[i];

span.style.color = 'orchid';

span.style.fontSize = '20px';

}

}

**</script>**

**</head>**

**<body>**

**<h3>**내가 좋아하는 과일

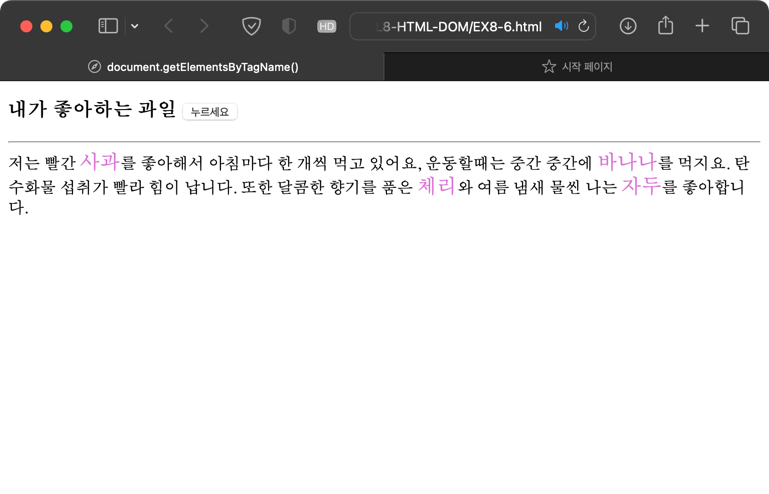
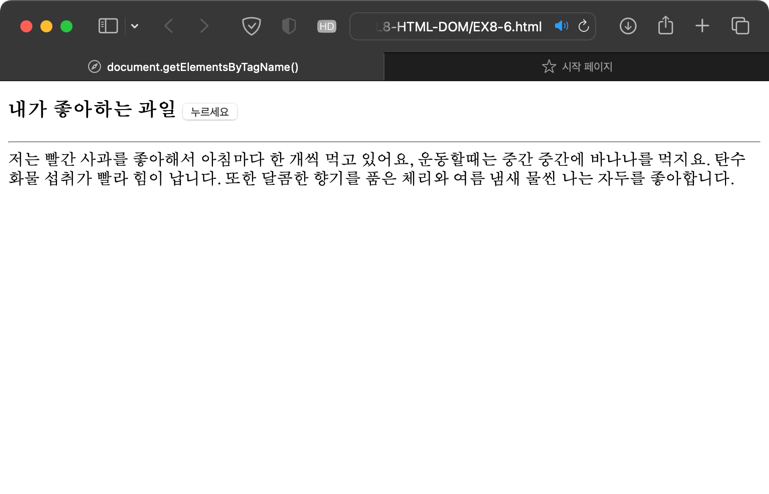
**<button onclick="change()">**누르세요**</button>**

**</h3><hr>**

저는 빨간 **<span>**사과**</span>**를 좋아해서 아침마다 한 개씩 먹고 있어요, 운동할때는 중간 중간에 **<span>**바나나**</span>**를 먹지요. 탄수화물 섭취가 빨라 힘이 납니다. 또한 달콤한 향기를 품은 **<span>**체리**</span>**와 여름 냄새 물씬 나는 **<span>**자두**</span>**를 좋아합니다.

**</body>**

**</html>**

****

<그림 6-1> 예제 8-6 결과

document.getElementsByTegName()을 사용하면 태그 이름으로 DOM 객체를 찾을 수 있다. <그림 6-1>에서 <span>태그를 찾아 해당 태그 CSS 스타일을 ‘orchid’ 색과 ‘20px’의 글자 크기로 바꾸는 것을 확인할 수 있다.

1. **예제 8-7 write()와 writeln() 메소드 활용**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>**write()와 writeln() 활용**</title>**

**</head>**

**<body>**

**<h3>**write()와 writeln() 활용**</h3>**

**<hr>**

**<script>**

document.write('<h3>동물원에 소풍갑시다</h3>');

document.write("<p style='color:blue'>날씨가 좋아 ");

document.write("소풍갑시다</p>");

document.write(2+3);

document.write('명입니다.<br>');

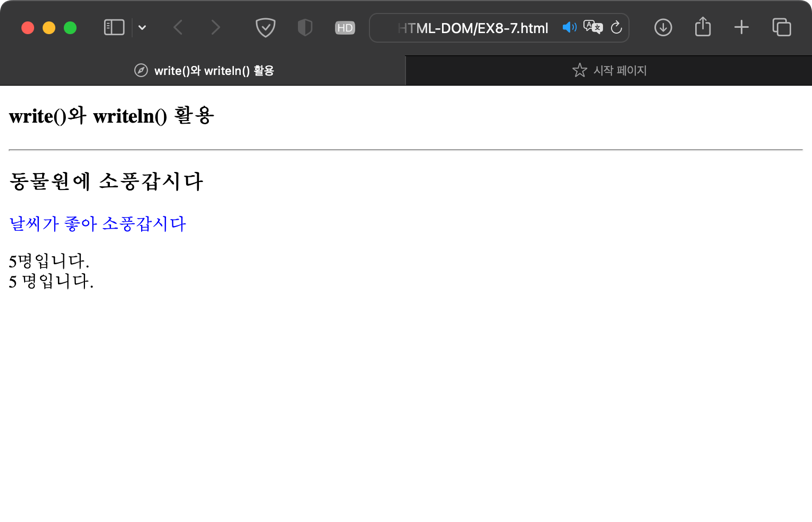
document.writeln(5);

document.write('명입니다.<br>');

**</script>**

**</body>**

**</html>**

****

<그림 7-1> 예제 8-7 결과

document.write()를 사용하면 HTML 콘텐츠를 출력할 수 있다. 태그 역시 인식되며 <그림 7-1>에서 write()를 사용하여 HTML 콘텐츠를 출력한 것을 확인할 수 있다. writeln() 메소드는 한 칸을 띄우는 효과를 가지고 있으며 이 또한 <그림 7-1>에서 확인할 수 있다. 만일 한 줄을 띄우려면 ‘document.write(“<br>”);’로 <br>태그를 추가하면 된다.

1. **예제 8-8 HTML 문서 작성 연습 페이지 만들기**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>**HTML 문서 작성기 만들기**</title>**

**<script>**

var win = null;

function showHTML() {

if (win == null || win.closed)

win = window.open('', 'outWin', 'width=300, height=200');

var textArea = document.getElementById('srcText');

win.document.open();

win.document.write(textArea.value);

win.document.close();

}

**</script>**

**</head>**

**<body>**

**<h3>**HTML 문서 작성기 만들기**</h3>**

**<hr>**

**<p>**HTML 문서를 작성하고 버튼을 클릭해보세요.**</p>**

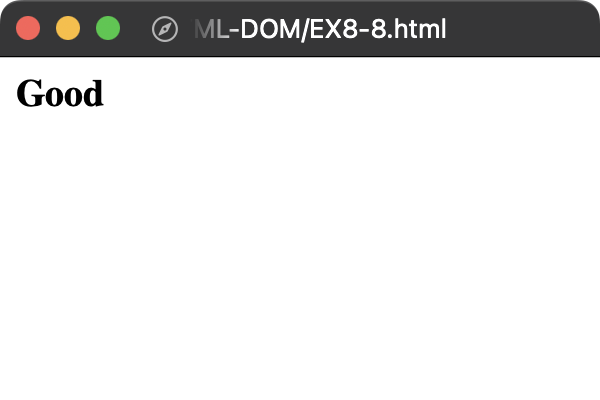
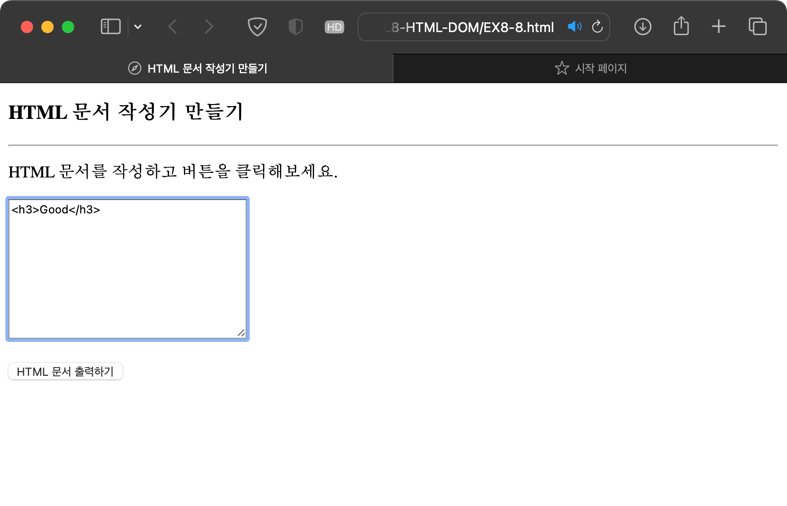
**<textarea id="srcText" rows="10" cols="30"></textarea>**

**<br><br>**

**<button onclick="showHTML()">**HTML 문서 출력하기**</button>**

**</body>**

**</html>**

****

<그림 8-1> 예제 8-8 결과

<textarea>태그 내에 작성한 HTML 문서가 실제 HTML 창으로 띄워지는 예제이다. <그림 8-1>에서 window.open()을 통해 새로운 창에 작성한 HTML 문서가 띄어지는 것을 확인할 수 있다. 또한 document.open()과 document.close()를 사용하여 해당 내용을 처리하는 것을 확인할 수 있다.

document.open()는 document객체에 담긴 DOM 트리를 지우고 새로 시작하며 document.close()는 출력된 HTML 페이지를 완성하는 기능을 한다.

1. **예제 8-9 HTML 태그의 동적 추가 및 삭제**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>**문서의 동적 구성**</title>**

**<script>**

function createDIV() {

var obj = document.getElementById('parent');

var newDIV = document.createElement('div');

newDIV.innerHTML ='새로 생성된 DIV입니다.';

newDIV.setAttribute('id', 'myDiv');

newDIV.style.backgroundColor = 'yellow';

newDIV.onclick = function() {

var p = this.parentElement;

p.removeChild(this);

};

obj.appendChild(newDIV);

}

**</script>**

**</head>**

**<body id="parent">**

**<h3>**DIV 객체를 동적으로 생성, 삽입, 삭제하는 예제**</h3><hr>**

**<p>**

DOM 트리에 동적으로 객체를 삽입할 수 있습니다. createElement(), appendChild(), removeChild() 메소드를 이용하여 새로운 객체를 생성, 삽입, 삭제하는 예제입니다.

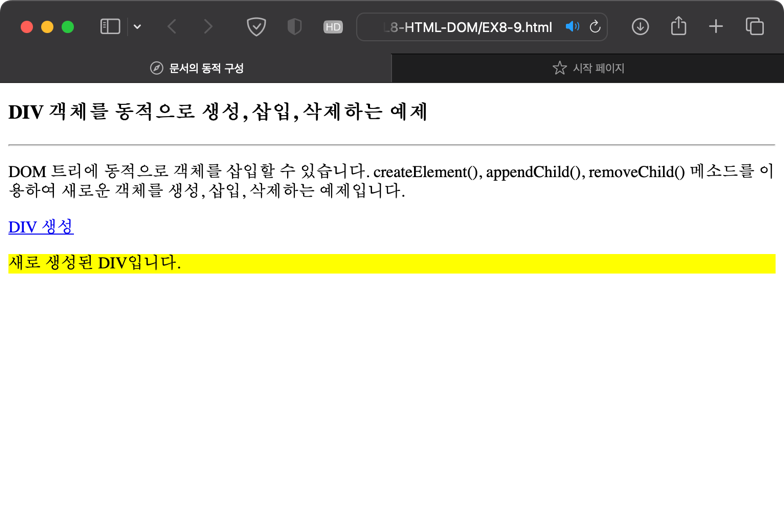
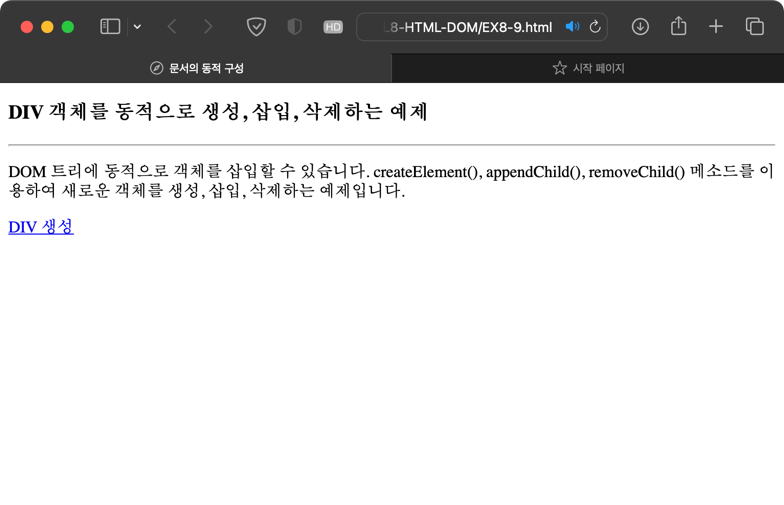
**</p>**

**<a href="javascript:createDIV()">**DIV 생성**</a>**

**<p><p>**

**</body>**

**</html>**

****

<그림 9-1> 예제 8-9 결과

문서를 동적으로 구성할 수 있다. document.createElement()를 사용하여 DOM 객체를 동적으로 생성하고 appendChild()를 사용하여 DOM 트리에 삽입, removeChile()를 사용하여 DOM 트리에서 삭제할 수 있다.

<그림 9-1>에서 <div>태그를 가진 HTML 콘텐츠가 document.createElement()를 통해 생성되었으며 ‘obj.appendChild(newDIV);’를 통해 obj 아래, 다시말해 body 태그 아래에 추가된 것을 확인할 수 있다.