

HTML5 웹 프로그래밍 입문(교수용)

9장. DOM과 자바스크립트 활용

목차

- 9.1 자바스크립트로 HTML문서 다루기
- 9.2 이벤트 처리하기
- 9.3 동적 웹 문서 만들기



9.1 자바스크립트로 HTML문서 다루기

9.1.1 DOM의 정의 및 문서 구조

9.1.2 DOM을 통한 HTML 문서
접근

9.1.3 브라우저 제공 내장 객체

DOM과 자바스크립트

■ 자바스크립트 활용

- 자바스크립트를 이용하여 HTML 문서의 내용을 변경
- 사용자 입력을 받아 처리

■ DOM

- 자바스크립트는 HTML 문서를 객체(Object)로 바라보고 다룬다
 - ▶ 이 객체를 DOM(Document Object Model)이라고 부름
- HTML 문서 뿐만 아니라 CSS 속성도 변경 가능
- 자바스크립트 관점에서 HTML 문서나 웹 브라우저를 바라보고 처리
 - ▶ 웹 문서를 자바스크립트 입장에서 구조적 객체의 형태로 바라보고 처리하는 모델

■ DOM 표준

- 2000년에 DOM2 제정
 - ▶ 대부분의 웹브라우저가 지원
- 2004년에 DOM3 까지 제정된 상태



DOM과 HTML 웹 문서의 관계

- DOM을 이용해 웹문서를 프로그램 언어에서의 변수나 구조체와 같은 데이터로 처리할 수 있도록 한다

자바스크립트를 이용해 DOM의 내용을 추가/변경

HTML 문서의 태그/콘텐츠가 변경되는 효과

화면에 디스플레이 되는 내용도 변경

HTML 문서의 내용을 동적으로 변경

트리 구조의 DOM

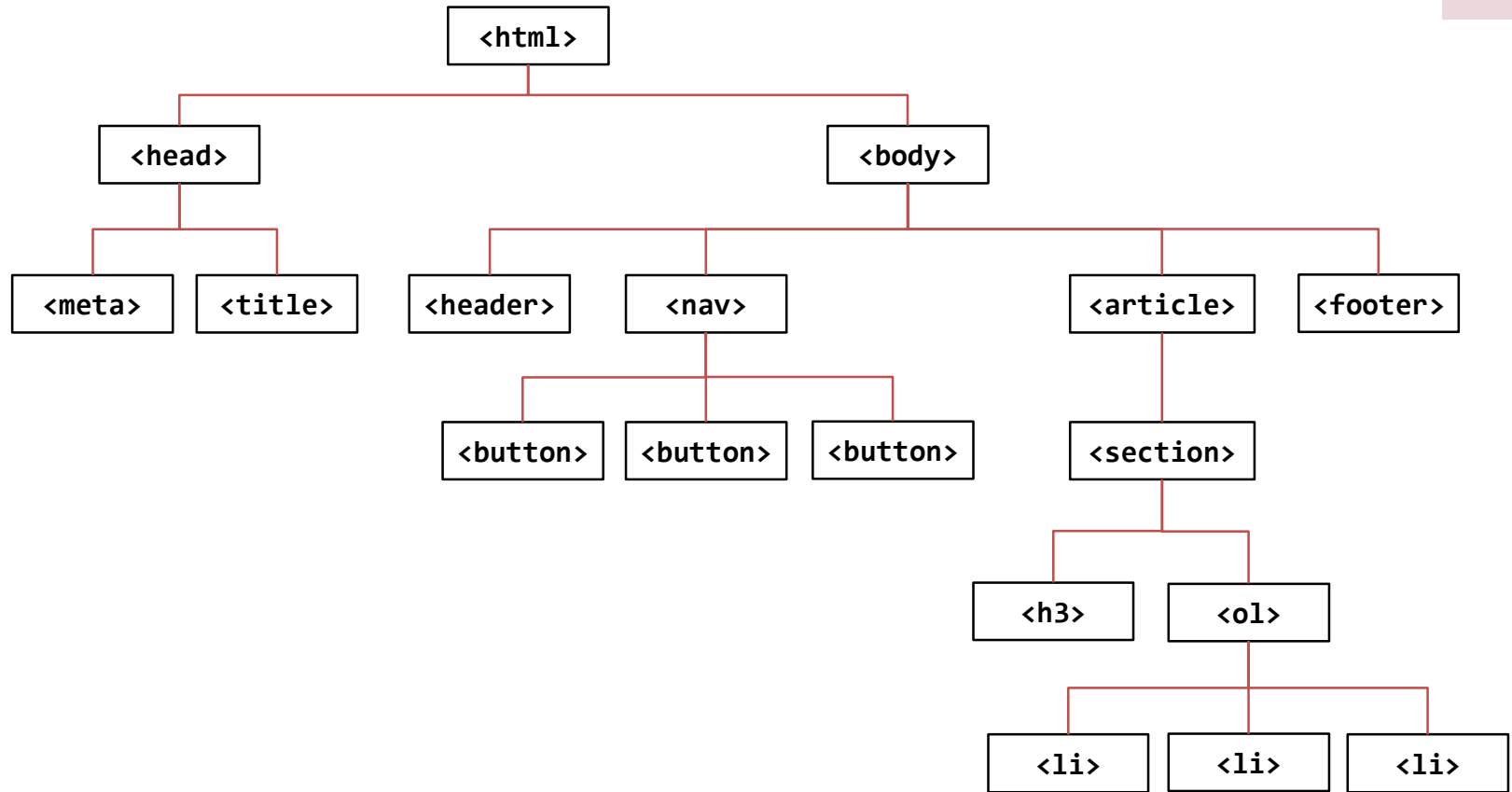
- HTML 문서는 태그 요소의 계층적 구조
 - DOM도 트리 구조의 형태
 - 트리의 노드: HTML 태그 요소
 - 노드는 속성과 속성값을 가짐
- DOM 구조는 웹 브라우저에서 확인 가능

웹 브라우저에서 DOM 구조 확인

```
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title> Title </title>
6 </head>
7 <body>
8 <header>
9   Web Page Logo
10 </header>
11 <nav>
12   <button> Navigation Menu1 </button>
13   <button> Navigation Menu2 </button>
14   <button> Navigation Menu3 </button>
15 </nav>
16 <article>
17   <section>
18     <h3>베스트셀러</h3>
19     <ol>
20       <li>멀티미디어 배움터 2.0 </li>
21       <li>모바일 멀티미디어 </li>
22       <li>자바입문: 이론과 실습 </li>
23     </ol>
24   </section>
25 </article>
26 <footer>
27   Footer Information
28 </footer>
29 </body>
30 </html>
```



DOM 트리 구조



HTML 태그 요소와 DOM

- 태그 요소는 DOM의 객체로 표현 됨
 - 태그 요소에 포함된 다른 요소는 객체내에 소속된 객체 형태로 표현 (하위객체)
- 태그 속성은 DOM 객체의 속성으로 표현 됨
- 요소 전체가 하나의 객체

```
<input type = "text" name = "username"/>
```

- type과 name은 속성
- "text"와 "username"은 type과 name의 속성값

DOM을 통한 HTML 문서 접근

■ 자바스크립트 입장에서는 웹 브라우저 환경과 HTML 문서를 모두 객체로 바라보고 처리

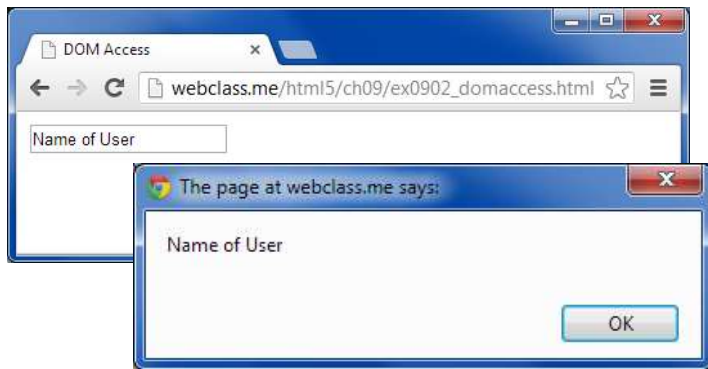
- 일반 프로그래밍에서 처럼 객체에 접근해서 값을 읽어내거나, 저장하고 수정하는 작업을 수행함으로써 웹 브라우저나 HTML 문서에 대한 처리를 수행

■ DOM 접근 방법

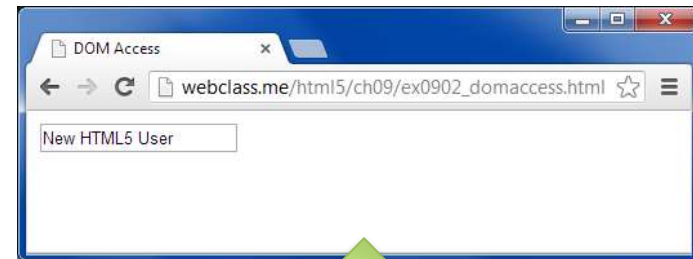
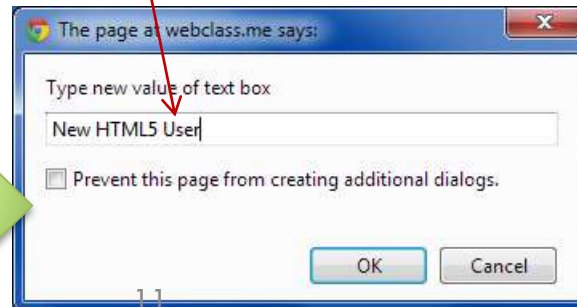
1. document의 forms 속성을 이용해서 접근하는 방법
2. 요소 이름을 이용해 접근하는 방법
3. document 객체가 제공하는 getElementById 등의 메소드를 이용해서 접근하는 방법
 - ▶ 가장 사용이 쉽고 많이 사용되고 방법인 getElementById 메소드 방법을 중심으로 설명

DOM 접근 방법 예제

```
1 <form action="">
2   <input id="username" type="text" value="Name of User" />
3 </form>
4
5 <script type="text/javascript">
6   var dom = document.getElementById("username");
7   alert(dom.value);
8   var newValue = prompt("Type new value of text box", "");
9   dom.value = newValue;
10 </script>
```



사용자 입력



브라우저 제공 **내장 객체**

- 브라우저에서 제공하는 자바스크립트 객체
 - 대표적으로 navigator, window, document 객체가 있다
 - document 객체
 - ▶ HTML 문서를 DOM을 통해 접근하기 위한 최상위 객체
 - 본 절에서는 navigator와 window 객체에 대해서만 설명

window 객체

- 웹 브라우저에 열려 있는 창을 뜻하며 브라우저 창에 관련된 다양한 속성과 메소드를 제공
 - `open()`
 - ▶ 새로운 창을 연다
 - `close()`
 - ▶ 열려 있는 창을 닫는다
 - `alert()`, `confirm()`, `prompt()`
 - ▶ 브라우저에서 사용자에게 경고창을 띄우거나 키보드 입력을 받아들이는 메소드

window 객체 예제

- URL 주소를 입력받아 새로운 윈도우에 표시
- close() 메소드로 그 윈도우를 닫는 예제

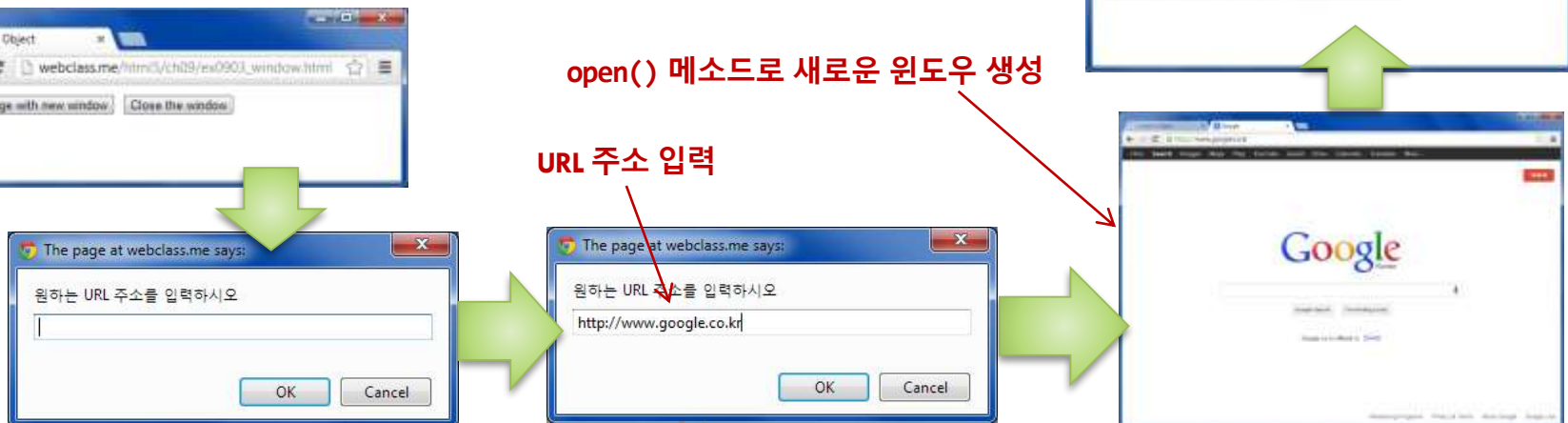
```
1 <script type="text/javascript" >
2   var win;
3   function open_window() {
4       var url = prompt ("원하는 URL 주소를 입력하시오");
5       if (url) win = window.open(url);
6   }
7 </script>
8
9 <button onclick="open_window();">Open a page with new window</button>
10 <button onclick="win.close();">Close the window</button>
```

close() 메소드
실행시 윈도우 닫힘

open() 메소드로 새로운 윈도우 생성

URL 주소 입력

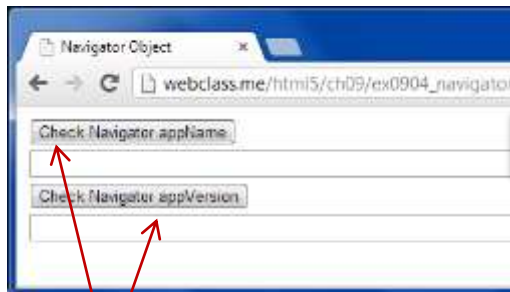
버튼 클릭



navigator 객체

- 현재 사용하는 웹 브라우저의 종류와 버전을 알아내기 위해 사용
- 대표적인 속성: appName, appVersion, userAgent

```
1 <form>
2   <input type = "button" value = "Check Navigator appName"
3     onclick = "document.getElementById('appName').value = navigator.appName;"/>
4   <br/>
5   <input id = "appName" type = "text" size = "110"/>
6   <br/>
7   <input type = "button" value = "Check Navigator appVersion"
8     onclick = "document.getElementById('appVersion').value = navigator.appVersion;"/>
9   <br/>
10  <input id = "appVersion" type = "text" size = "110"/>
11 </form>
```



9.2 이벤트 처리하기

9.2.1 이벤트 처리 개요

9.2.2 이벤트의 정의와 종류

9.2.3 이벤트 핸들링 및 이벤트
등록

9.2.4 폼 다루기

이벤트 처리 개요

■ 이벤트

- 사용자가 웹 브라우저를 사용하는 중에 발생시키는 키보드, 마우스 등의 입력

■ 이벤트 처리

- 이벤트가 입력 되었을때 미리 구현된 자바스크립트 코드를 수행
 1. 이벤트의 정의
 2. 이벤트 핸들러
 - ▶ 이벤트가 발생할때마다 호출되는 자바스크립트 코드
 3. 이벤트 등록 (registration)
 - ▶ 이벤트와 이벤트 핸들러를 연결시키는 과정

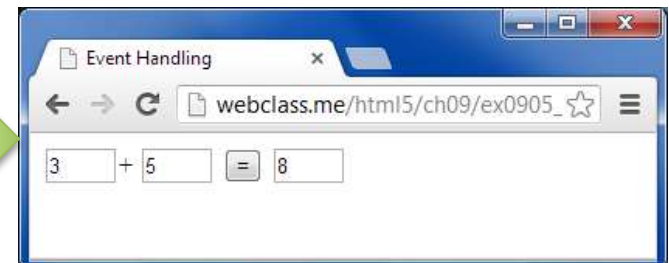
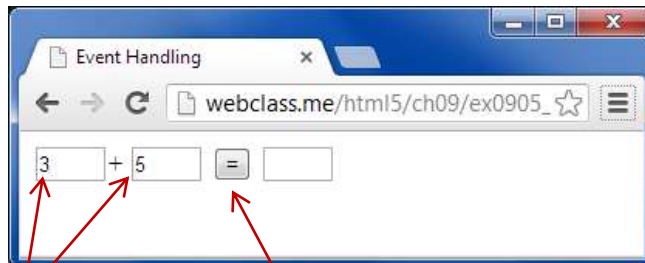
이벤트 처리 예제

```
1 <script type="text/javascript">
2     function add() {
3         var a = document.getElementById("op1").value;
4         var b = document.getElementById("op2").value;
5         document.getElementById("result").value = parseInt(a) + parseInt(b);
6     }
7 </script>
8
9 <form>
10     <input id="op1" type="text" size="2"/>+
11     <input id="op2" type="text" size="2"/>
12     <input type="button" value="=" onclick="add();" />
13     <input id="result" type="text" size="2"/>
14 </form>
```

이벤트 타입

이벤트 등록

이벤트 핸들러 함수



이벤트의 종류

■ 마우스 이벤트

이벤트 이름	태그 속성	설명
click	onclick	HTML 문서내의 요소를 클릭했을때 발생한다.
dblclick	ondblclick	HTML 문서내의 요소를 더블클릭했을때 발생한다.
mousedown	onmousedown	마우스 커서를 HTML 문서내의 요소 위에 위치시키고 마우스 버튼을 누를때 발생한다.
mousemove	onmousemove	마우스 커서를 HTML 문서내의 요소 위에서 이동시킬때 발생한다. 마우스 커서를 움직이는 동안에는 계속해서 이벤트가 발생하게 된다.
mouseover	onmouseover	마우스 커서가 해당 요소 위에 위치할 경우에 발생한다. 요소 위에 위치할때 1회만 발생하며 연속해서 발생하지 않는다.
mouseout	onmouseout	마우스 커서가 해당 요소 위를 벗어날때 발생하는 이벤트이다.
mouseup	onmouseup	사용자가 마우스 커서를 HTML 문서내의 요소 위에 위치시키고 마우스 버튼을 땔때 발생한다.

이벤트의 종류

■ 키보드 이벤트

이벤트 이름	태그 속성	설명
keydown	onkeydown	키보드를 누를때 발생하는 이벤트이다. 키보드를 눌러서 내려갈때 1회 발생한다.
keypress	onkeypress	키보드를 타이핑할때 발생하는 이벤트이다. 키보드를 누른 후 떼때 1회 발생한다.
keyup	onkeyup	키보드를 누른 후 떼때 발생하는 이벤트이다. 키보드에서 손을 떼때 키보드가 올라올때 1회 발생한다.

이벤트의 종류

■ 프레임/객체 이벤트

이벤트 이름	태그 속성	설명
abort	onabort	이미지가 웹 브라우저상에 완전히 로드되기 전에 정지되었을 때 발생하는 이벤트이다. 예외 처리를 하고자 할 때 사용할 수 있다.
error	onerror	웹 브라우저상에 이미지를 로드할 때 문제가 발생하여 로드가 정상적으로 이루어지지 않았을 경우 발생한다. 오류가 발생했을 때 예외처리를 위해 사용할 수 있다.
load	onload	문서, 프레임, 객체 등이 웹 브라우저상에 로드가 완료 되었을 때 발생하는 이벤트이다.
resize	onresize	문서 창, 문서 뷰의 크기가 리사이즈(resize) 되었을 경우 발생한다.
scroll	onscroll	문서 창, 문서 뷰가 스크롤 되었을 경우 발생한다.
unload	onunload	윈도우 창, 프레임으로부터 문서가 제거되었을 경우 발생하는 이벤트이다. 창을 종료시킬 때 종료하기 직전에 처리해야 할 사항이 있을 때 이용하면 된다.

이벤트의 종류

■ 폼 이벤트

이벤트 이름	태그 속성	설명
abort	onabort	이미지가 웹 브라우저상에 완전히 로드되기 전에 정지되었을 때 발생하는 이벤트이다. 예외 처리를 하고자 할 때 사용할 수 있다.
error	onerror	웹 브라우저상에 이미지를 로드할 때 문제가 발생하여 로드가 정상적으로 이루어지지 않았을 경우 발생한다. 오류가 발생했을 때 예외처리를 위해 사용할 수 있다.
load	onload	문서, 프레임, 객체 등이 웹 브라우저상에 로드가 완료 되었을 때 발생하는 이벤트이다.
resize	onresize	문서 창, 문서 뷰의 크기가 리사이즈(resize) 되었을 경우 발생한다.
scroll	onscroll	문서 창, 문서 뷰가 스크롤 되었을 경우 발생한다.
unload	onunload	윈도우 창, 프레임으로부터 문서가 제거되었을 경우 발생하는 이벤트이다. 창을 종료시킬 때 종료하기 직전에 처리해야 할 사항이 있을 때 이용하면 된다.

이벤트 핸들링 및 이벤트 등록

■ 이벤트 핸들러(handler)

- 이벤트가 발생시 실행하고자 하는 자바스크립트 함수나 코드
- 사용자가 입력한 내용이 맞는지 검사하거나 입력한 내용에 따라 웹 문서를 수정하는 등의 작업을 통해 동적 웹 문서를 만든다

■ 이벤트 등록

- 이벤트의 종류와 이를 처리할 이벤트 핸들러를 연결시키는 작업
- 두가지 등록 방법
 1. 태그 속성에 직접 이벤트 핸들러 기술
 2. 객체의 이벤트 속성 값에 이벤트 핸들러 함수 기술

태그 속성에 이벤트 핸들러 기술

- 이벤트 태그 속성에 직접 이벤트 핸들러 기술
 - 이벤트 핸들러: 자바스크립트 코드 혹은 함수 이름

```
<form action="">
  <input id="username" type="text" value="Name of User"
    onclick="alert('Please type your full name');" />
</form>
```

```
<form action="">
  <input id="username" type="text" value="Name of User"
    onclick="myEventHandler();" />
</form>

<script type="text/javascript">
  function myEventHandler() {
    alert("Please type your full name");
  }
</script>
```


객체의 이벤트 속성에 이벤트 핸들러 함수 기술

- 요소 객체의 해당 이벤트 속성에 이벤트 핸들러 함수를 지정
 - 이벤트 핸들러는 반드시 함수 형태로 미리 구현

```
<form action="">
  <input id="username" type="text" value="Name of User" />
</form>

<script type="text/javascript">
  function myEventHandler() {
    alert("Please type your full name");
  }

  var dom = document.getElementById("username");
  dom.onclick = myEventHandler;
</script>
```

- 주의: 객체의 이벤트 속성에 ()없이 함수 이름만 적어야 함

폼 다루기

- 자바스크립트를 이용해 폼의 값을 읽어내거나 계산하여 수정하는 것이 가능

Form handling with JavaScript x

webclass.me/html5/ch09/ex0906_formhandling.html

서적 주문 양식

주문서

책 제목	가격	수량	합계
멀티미디어 배움터2.0	25,000원	<input type="text" value="0"/>	<input type="text" value="0"/> 원
모바일 멀티미디어	27,000원	<input type="text" value="0"/>	<input type="text" value="0"/> 원
자바입문: 이론과 실습	25,000원	<input type="text" value="0"/>	<input type="text" value="0"/> 원
합계		<input type="text" value="0"/>	<input type="text" value="0"/> 원



Form handling with JavaScript x

webclass.me/html5/ch09/ex0906_formhandling.html

서적 주문 양식

주문서

책 제목	가격	수량	합계
멀티미디어 배움터2.0	25,000원	<input type="text" value="2"/>	<input type="text" value="50000"/> 원
모바일 멀티미디어	27,000원	<input type="text" value="3"/>	<input type="text" value="81000"/> 원
자바입문: 이론과 실습	25,000원	<input type="text" value="5"/>	<input type="text" value="125000"/> 원
합계		<input type="text" value="10"/>	<input type="text" value="256000"/> 원

폼의 위젯 값 접근하기

■ DOM을 통해 <input> 텍스트 박스 위젯에 입력한 값은 읽어 낸다

- 요소 객체의 value라는 속성을 통해 읽거나 수정

```
(8) <input id="book1" type="text" size="2" value="0"  
(9)      onclick = "this.select();" />
```

Just to get attention
of user: Highlight the cell

```
(40) <input type="button" value="합계계산" onclick="updateAll();" />
```

```
var n1 = document.getElementById("book1").value;  
var p1 = 25000 * n1;  
document.getElementById("book1Total").value = p1;
```

9.3 동적 웹 문서 만들기

9.3.1 동적 문서 정의

9.3.2 콘텐츠 변경을 통한 동적 문서
만들기

9.3.3 스타일 변경을 통한 동적 문서
만들기

동적 문서 정의

■ 동적 문서

- 웹 문서가 브라우저상에 처음에 표시된 이후에 콘텐츠나 스타일이 변경되어 화면에 표시되는 내용이나 표현 형태가 변경되는 문서
- 웹 문서의 콘텐츠나 스타일이 변경되면 즉시 변경된 값을 바탕으로 화면의 문서를 갱신

■ 웹 문서의 콘텐츠나 스타일의 변경

- 태그 요소, 태그 속성, 태그 콘텐츠, 요소의 CSS 스타일 등의 값을 자바스크립트를 이용해서 변경
- 태그 요소의 화면내 표시 위치 변경, 애니메이션, 색상 및 글씨체 변경, 인터랙티브 사용자 인터페이스 등 포함

■ 동적 문서 구현 방식

1. 웹 문서의 콘텐츠를 변경시키는 방법
2. CSS 스타일을 변경시키는 방법

콘텐츠 변경을 통한 동적 문서

■ 웹 문서 콘텐츠 변경

- 폼 위젯 요소의 **value** 값을 변경
- **태그** 요소의 콘텐츠를 변경

■ HTML 태그 콘텐츠 속성

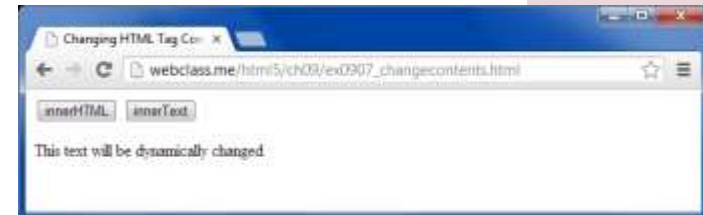
```
<p id = "example"> This is an example content </p>
```

- “**This is an example content**” 부분이 **<p>** 태그의 **콘텐츠**
- **innerHTML** 속성
 - ▶ 속성에 저장된 값을 HTML 태그로 해석
- **innerText** 속성
 - ▶ 속성에 저장된 값을 단순히 문자열로 해석

콘텐츠 변경을 통한 동적 문서 예제

```
1 <button id="b1">innerHTML</button>
2 <button id="b2">innerText</button>
3 <br/>
4 <p id= text">This text will be dynamically changed</p>
5
6 <script type="text/javascript" >
7     document.getElementById("b1").onmouseover = mouseover_innerHTML;
8     document.getElementById("b2").onmouseover = mouseover_innerText;
9
10    document.getElementById("b1").onmouseout = reset_text;
11    document.getElementById("b2").onmouseout = reset_text;
12
13    function mouseover_innerHTML() {
14        document.getElementById("text").innerHTML =
15            "<h1>Mouse cursor is over the innerHTML button</h1>";
16    }
17
18    function mouseover_innerText() {
19        document.getElementById("text").innerText =
20            "<h1>Mouse cursor is over the innerHTML button</h1>";
21    }
22
23    function reset_text() {
24        document.getElementById("text").innerHTML =
25            "This text will be dynamically changed";
26    }
27 </script>
```

초기화면



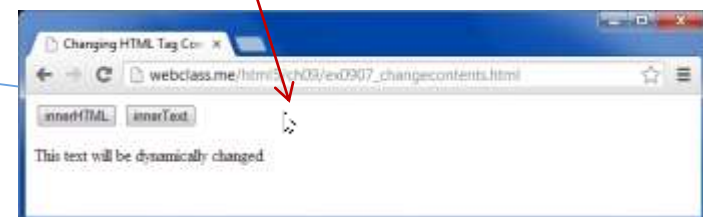
마우스 커서가 innerHTML 버튼 위에 위치



마우스 커서가 innerText 버튼 위에 위치



마우스 커서가 버튼 바깥으로 이동



스타일 변경을 통한 동적 문서

- DOM을 이용하면 CSS 스타일에 접근 가능
 - 일반적인 DOM에 접근하는 방법과 동일
 - CSS 스타일 값 변경으로 웹 문서를 보다 동적으로 제작 가능
- 예제 HTML 문서

```
1 <div id="outerBox"
2   style="background: blue; width: 550px; height:300px;" />
3
4 <div id="innerBox"
5   style="background: yellow; width: 450px; height:200px;
6         position: absolute; left: 50px; top: 50px;" />
7
8   <img id="img1" src = "image1.png" width = "100px"
9     style="position: absolute; left: 50px; top: 25px; visibility: visible;" />
10
11   <img id="img2" src = "image2.png" width = "100px"
12     style="position: absolute; left: 175px; top: 25px; visibility: visible;" />
13
14   <img id="img3" src = "image3.png" width = "100px"
15     style="position: absolute; left: 300px; top: 25px; visibility: visible;" />
16 </div>
17
18 </div>
```

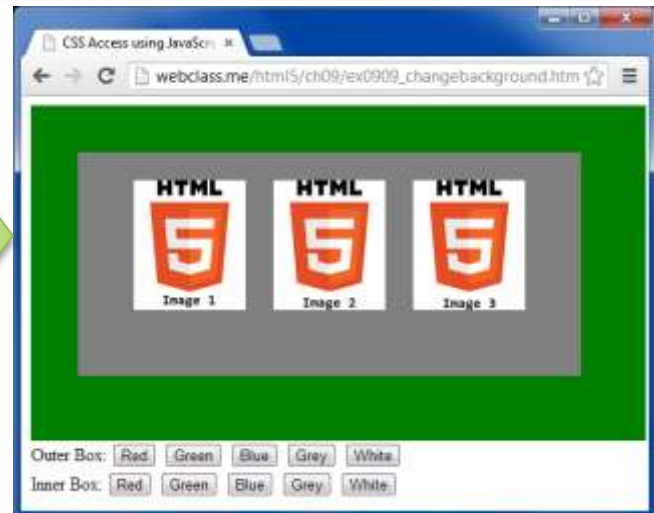
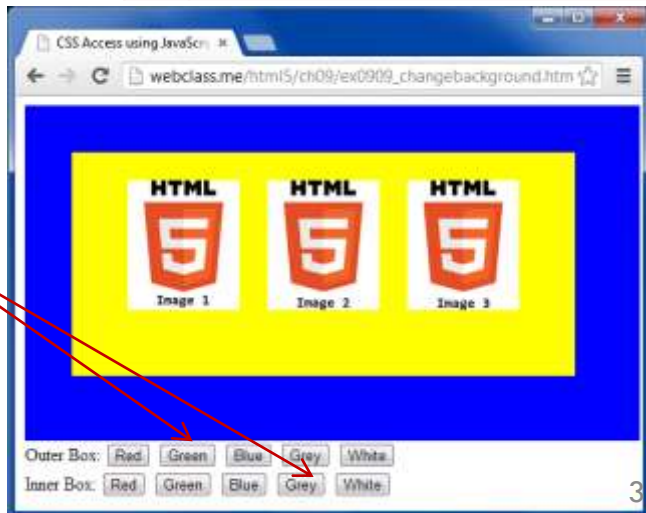


배경색 스타일 속성 접근

style.background
속성 접근

```
1 <script type = "text/javascript">
2 function changeColor(id, color) {
3     document.getElementById(id).style.background = color;
4 }
5 </script>
6
7 Outer Box:
8 <button onclick = "changeColor('outerBox', 'red');"> Red </button>
9 <button onclick = "changeColor('outerBox', 'green');"> Green </button>
10 <button onclick = "changeColor('outerBox', 'blue');"> Blue </button>
11 <button onclick = "changeColor('outerBox', 'grey');"> Grey </button>
12 <button onclick = "changeColor('outerBox', 'white');"> White </button>
13 <br />
14
15 Inner Box:
16 <button onclick = "changeColor('innerBox', 'red');"> Red </button>
17 <button onclick = "changeColor('innerBox', 'green');"> Green </button>
18 <button onclick = "changeColor('innerBox', 'blue');"> Blue </button>
19 <button onclick = "changeColor('innerBox', 'grey');"> Grey </button>
20 <button onclick = "changeColor('innerBox', 'white');"> White </button>
```

Green 과 Grey
버튼 클릭

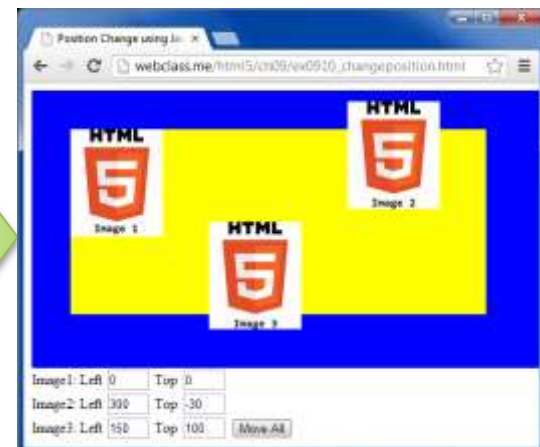


위치 스타일 속성 변경

style.left 속성 접근

style.top 속성 접근

```
1 <script type = "text/javascript">
2 function changePositions() {
3     for(i = 1; i <= 3; i++) {
4         var left = document.getElementById("left" + i).value;
5         var top = document.getElementById("top" + i).value;
6         document.getElementById("img" + i).style.left = left + "px";
7         document.getElementById("img" + i).style.top = top + "px";
8     }
9 }
10 </script>
11
12 <form>
13 Image1: Left <input id = "left1" size = "2" type = "text"/>
14 Top <input id = "top1" size = "2" type = "text"/> <br />
15
16 Image2: Left <input id = "left2" size = "2" type = "text"/>
17 Top <input id = "top2" size = "2" type = "text"/> <br />
18
19 Image3: Left <input id = "left3" size = "2" type = "text"/>
20 Top <input id = "top3" size = "2" type = "text"/>
21
22 <input type = "button" value = "Move All" onclick = "changePositions();" />
23 </form>
```



보이기/감추기 스타일 속성 변경

■ 화면 표시 여부를 결정

- 보이기/감추기 스타일 속성

- ▶ 스타일 속성: `visibility`
- ▶ 속성 값: `visible` 혹은 `hidden`

■ `hidden`으로 설정되도 웹 문서 내에 태그 요소로는 존재

- 화면에 표시만 안될 뿐이다

보이기/감추기 예제

```
1 <script type = "text/javascript">
2 function toggleVisibility(id) {
3     var dom = document.getElementById(id);
4
5     if (dom.style.visibility == "visible")
6         dom.style.visibility = "hidden";
7     else
8         dom.style.visibility = "visible";
9 }
10 </script>
11
12 Toggle Visibility:
13 <button onclick = "toggleVisibility('img1');">Image 1</button>
14 <button onclick = "toggleVisibility('img2');">Image 2</button>
15 <button onclick = "toggleVisibility('img3');">Image 3</button>
```

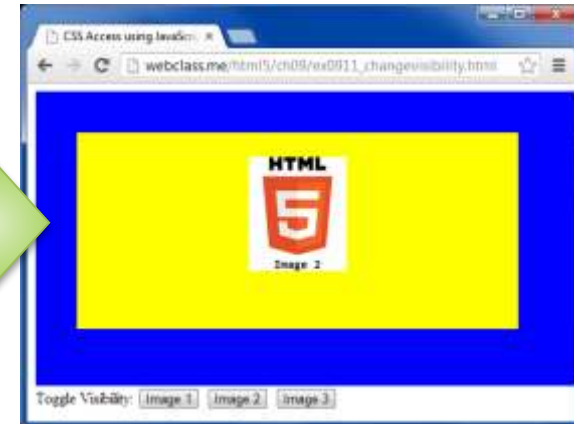
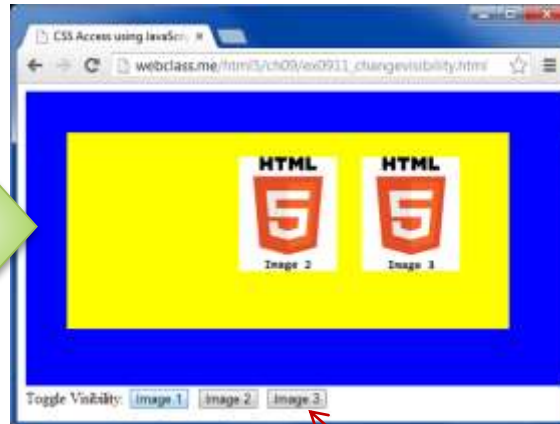


Image1 클릭

Image3 클릭