



CSS3+JavaScript와 함께 하는 HTML5 웹 프로그래밍

8장 HTML DOM과 동적 문서 작성

8.2.3 DOM 객체의 다양한 접근 방법

(1) `document.getElementById()` 이용 : 특정의 한 DOM 객체 접근

(2) 컬렉션 객체 반환 메서드 이용

(3) DOM 제공 컬렉션 객체 이용

(4) DOM 트리를 이용한 접근

(1) 컬렉션 객체 반환 메서드 이용

- 공통된 특성을 가진 요소객체들을 그룹핑해서 한꺼번에 접근할 수 있게 함
- 컬렉션(collection) 객체와 노드리스트(nodeList) 객체
 - ✓ 객체들의 모임을 나타내는 객체
 - ✓ 배열과 같은 리스트 구성, length 속성-리스트 크기
 - ✓ 컬렉션과 노드리스트 객체의 각 원소객체들은 배열원소처럼 접근할 수 있음
 - ✓ 그러나 배열객체는 아님(배열객체의 모든 속성과 메서드들을 지원하지는 않음)
- 관련 메서드들

[표 8.3] document 객체의 요소객체 그룹 메서드들

메서드	기능
getElement s ByName()	지정된 name 속성값을 가진 요소객체들을 반환함
getElement s ByClassName()	지정된 class 속성값을 가진 요소객체들을 반환함
getElement s ByTagName()	지정된 태그명을 요소객체들을 반환함
querySelectorAll()	지정된 CSS 선택자를 만족하는 모든 요소객체들을 반환함
querySelector()	지정된 CSS 선택자를 만족하는 첫 번째 요소객체들을 반환함

- 컬렉션 객체 obj의 요소객체의 처리

```
Var obj=document.getElementsByTagName("h1");  
For (i=0;i<obj.length;i++){  
    ....// obj[i]형식으로각요소객체접근함  
}
```

```
var obj = document.getElementsByTagName("h1");  
for (var i in obj){  
    .... //obj[i]형식으로각요소객체접근함  
}
```

- 컬렉션 객체와 노드리스트 객체의 표현 차이
 - ✓ 컬렉션 객체(초기 개념) : 모든 유형의 객체들에 대한 모임을 리스트 객체화함
 - ✓ 노드리스트 객체 : DOM 트리의 노드 객체들의 모임을 리스트 객체화함
 - ✓ DOM을 다룰 때 두 개념의 큰 차이는 없음

▪ **querySelectorAll()** 메서드를 이용한 요소 노드 접근

// 사용 방법 1

```
var tags = querySelectorAll('CSS-선택자'); // CSS-선택자를 만족하는 노드리스트 객체 반환
for (i=0; i<tags.length-1; i++) { // 컬렉션 객체 tags는 length 속성을 가지고 있음
    tags[i].innerHTML = "텍스트 (HTML 태그 포함)"; // tags[i] 객체의 요소내용 참조, 변경
    tags[i].속성명 = "속성값"; // tags[i] 요소객체의 속성값 참조 및 변경
    tags[i].style.CSS속성명 = "CSS-속성값"; // tags[i] 요소객체의 CSS 스타일 속성 지정
}
```

// 사용 방법 2

```
for var i in tags {
    tags[i].innerHTML = "텍스트 (HTML 태그 포함)"; // tags[i] 객체의 요소내용 참조, 변경
    tags[i].속성명 = "속성값"; // tags[i] 객체의 속성값 참조 및 변경
    tags[i].style.CSS속성명 = "CSS-속성값"; // tags[i] 객체의 CSS 스타일 속성 지정
}
```

[예 8.14] 요소 노드의 이동을 통한 애니메이션 효과 구현

<h3> 피겨선수의 회전 연속동작 </h3>

<hr><div id="p1">

</div><hr>

<button onclick="setInterval(movingImages, 500);">Start</button>

<script>

function movingImages(){

var parent = document.getElementById('p1');

var imgTags = document.querySelectorAll('img');

var movingImage = imgTags[0];

movingImage.parentNode.removeChild(movingImage);

parent.appendChild(movingImage);

}

</script>



(2) DOM 제공 컬렉션 객체 이용

- HTML DOM은 많이 사용되는 요소들을 미리 컬렉션 객체들로 표현해서 제공함
- DOM 트리가 생성되면 자동으로 제공, 별도의 메서드를 호출하지 않고 사용할 수 있음

[표 8.4] DOM 제공 컬렉션 객체들

DOM 컬렉션 객체	기능
<code>anchors[]</code>	HTML 문서의 <a> 요소들을 순차적으로 접근함
<code>images[]</code>	HTML 문서의 요소들을 순차적으로 접근함
<code>forms[]</code>	HTML 문서의 <form> 요소들을 순차적으로 접근함
<code>elements[]</code>	<form> 요소 내의 자식요소들을 순차적으로 접근함

```
function movingImages(){
    var parent = document.getElementById('p1');
    var movingImage = document.images[0];
    movingImage.parentNode.removeChild(movingImage);
    parent.appendChild(movingImage);
}
```

[예 8.14]에서 `querySelectorAll()` 대신 `images[]` 컬렉션 객체를 사용할 수 있음

[많은 입력양식들을 포함하는 HTML 문서]

```
... 생략 ...
<form name='폼명0' ... > /* forms[0] */
<input name='입력요소명00' type=... value= ... /> /* elements[0] */
<input name='입력요소명01' type=... value= ... /> /* elements[1] */
...
</form>

...
<form name='폼명1' ... > /* forms[1] */
<input name='입력요소명10' type=... value= ... /> /* elements[0] */
<input name='입력요소명11' type=... value= ... /> /* elements[1] */
...
</form>

...
<form name='폼명2' ... > /* forms[2] */
<input name='입력요소명20' type=... value= ... /> /* elements[0] */
<input name='입력요소명21' type=... value= ... /> /* elements[1] */
...
</form>
... 생략 ...
```

- ① 폼명.입력요소명.속성 // <form>, <input>의 name 속성을 이용한 접근
예) document.write(document.폼명1.입력요소명11.value);
- ② forms[].입력요소명.속성 // 컬렉션객체 forms[]와 <input>의 name 속성을 이용한 접근
예) document.write(document.forms[1].입력요소명11.value);
- ③ forms[].elements[].속성 // 컬렉션객체 forms[], elements[]를 이용한 접근
예) document.write(document.forms[1].elements[1].value);

[특정 입력요소의 사용자 입력값 접근 방법]

[예 8.15] forms[] 컬렉션 객체의 사용

[illegible]

```
<script>
function computeScore() {
    var choice = new Array(); // 학생 응답
    var answer = [ 4, 2 ]; // 정답
    var sdept = document.stuInfo.dept.value;
    var sname = document.stuInfo.sName.value;
    var noti = document.getElementById('result');
    var score = 0;

    choice[0] = document.forms[1].p1.value;
    choice[1] = document.forms[2].p2.value;

    for (i=0; i<2; i++) {
        if (parseInt(choice[i]) == parseInt(answer[i]))
            score = score + 10;
    }
    noti.innerHTML = sname + " : " + score + "점" ;
}
</script>
```

[예 8.15] forms[] 컬렉션 객체의 사용(실행 예)

ch08-ex15.htm x

file:///C:/html5/ch08/ch08-ex15.htm ☆ ⋮

학 과 : 컴퓨터공학과 이 름 : 이나현

Q1. HTML5 프로그래밍에서 다루는 내용이 아닌 것은 ?

- ☐ HTML
- ☐ CSS3
- ☐ JavaScript
- ☒ PHP

Q2. 다음 중 그 성격이 다른 프로그램은 ?

- ☐ 사파리
- ☐ 파워포인트
- ☐ 파이어폭스
- ☒ 크롬

채점(20점 만점)

ch08-ex15.htm x

file:///C:/html5/ch08/ch08-ex15.htm ☆ ⋮

학 과 : 컴퓨터공학과 이 름 : 이나현

Q1. HTML5 프로그래밍에서 다루는 내용이 아닌 것은 ?

- ☐ HTML
- ☐ CSS3
- ☐ JavaScript
- ☒ PHP

Q2. 다음 중 그 성격이 다른 프로그램은 ?

- ☐ 사파리
- ☒ 파워포인트
- ☐ 파이어폭스
- ☐ 크롬

채점(20점 만점) 이나현 : 20점

[예 8.15] forms[] 컬렉션 객체의 사용

[illegible]

```
<form name="q1">Q1. HTML5 프로그래밍에서 다루는 내용이 아닌 것은 ? <br>
  <input type="radio" name="p1" value="1"> HTML </input> <br>
  <input type="radio" name="p1" value="2"> CSS3 </input> <br>
  <input type="radio" name="p1" value="3"> JavaScript </input> <br>
  <input type="radio" name="p1" value="4"> PHP </input> <br><br>
</form>
```

```
<form name="q2">Q2. 다음 중 그 성격이 다른 프로그램은 ? <br>
  <input type="radio" name="p2" value="1"> 사파리 </input> <br>
  <input type="radio" name="p2" value="2"> 파워포인트 </input> <br>
  <input type="radio" name="p2" value="3"> 파이어폭스 </input> <br>
  <input type="radio" name="p2" value="4"> 크롬 </input> <hr>
</form>
<button onclick="computeScore();" > 채점(20점 만점) </button>
<span id="result"> </span>
```

```
<script>
function computeScore() {
    var choice = new Array(); // 학생 응답
    var answer = [ 4, 2 ]; // 정답
    var sdept = document.stuInfo.dept.value;
    var sname = document.stuInfo.sName.value;
    var noti = document.getElementById('result');
    var score = 0;

    choice[0] = document.forms[1].p1.value;
    choice[1] = document.forms[2].p2.value;

    for (i=0; i<2; i++) {
        if (parseInt(choice[i]) == parseInt(answer[i]))
            score = score + 10;
    }
    noti.innerHTML = sname + " : " + score + "점" ;
}
</script>
```

(3) DOM 트리를 이용한 접근

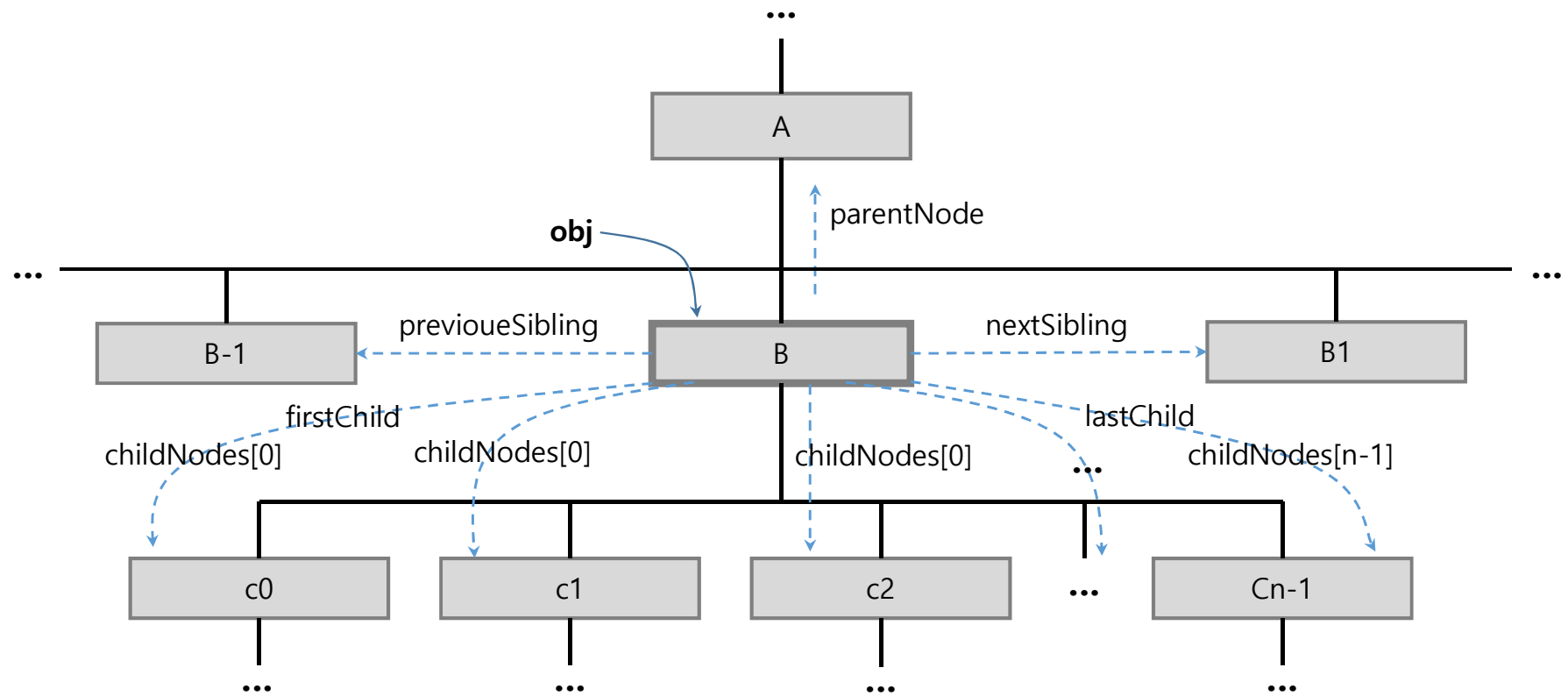
- 특정 요소를 기준으로 주변의 다른 요소들을 접근함
- DOM 트리에서 부모노드, 자식노드, 형제노드 탐색방법을 제공함

[표 8.5] DOM 요소 노드의 속성 객체

DOM 인터페이스	기능
<code>childNodes[]</code>	노드 객체의 자식 노드들을 순차적으로 접근함
<code>firstChild</code>	노드 객체의 첫 번째 자식 노드를 접근함
<code>lastChild</code>	노드 객체의 마지막 자식 노드를 접근함
<code>parentNode</code>	노드 객체의 부모 노드를 접근함
<code>nextSibling</code>	노드 객체의 직후 형제 노드를 접근함
<code>previousSibling</code>	노드 객체의 직전 형제 노드를 접근함

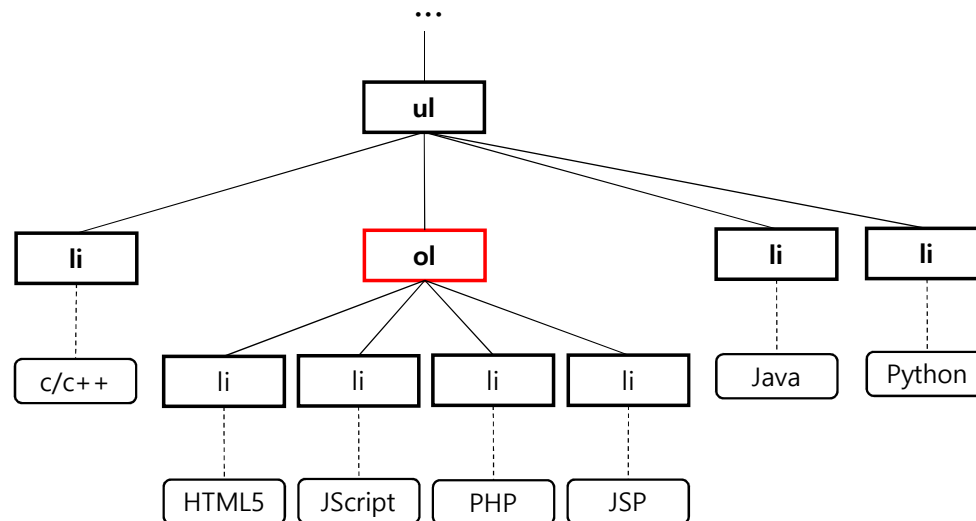
- HTML 태그들 사이의 공백문자들도 DOM 트리에서 각각 텍스트 노드들로 표현됨
 - ✓ 이것은 개발자가 예상하는 DOM 트리와는 차이가 있어 많은 혼란을 초래함(예 8.18)
 - ✓ 특히 형제 노드들이나 자식 노드들을 접근할 때 유의해야 함
- DOM 트리 구조를 정확하게 알고 있을 때만 DOM 트리 탐색 이용하는 것이 바람직함

그림 8.5 DOM 트리 예(노드 B 기준)



[예 8.16] DOM 트리 구조 확인

```
<body>
<h3>Programming Language</h3><hr>
<ul>
<li>C/C++</li><ol id="current"><li>HTML5</li><li>JScript</li><li>PHP</li><li>JSP</li></ol><li>Java</li><li>Python</li><li>FORTRAN</li></ul><hr>
<button onclick="f1();"> 현재위치 노드 </button> <span id="s1"></span><br>
<button onclick="f2();"> 직전 형제노드 </button> <span id="s2"></span><br>
<button onclick="f3();"> 직후 형제노드 </button> <span id="s3"></span><br>
<button onclick="f4();"> 첫번째 자식노드 </button> <span id="s4"></span><br>
<button onclick="f5();"> 두번째 자식노드 </button> <span id="s5"></span><br>
<button onclick="f6();"> 마지막 자식노드 </button> <span id="s6"></span><br>
```



ch08- x

file:///C:/html5...

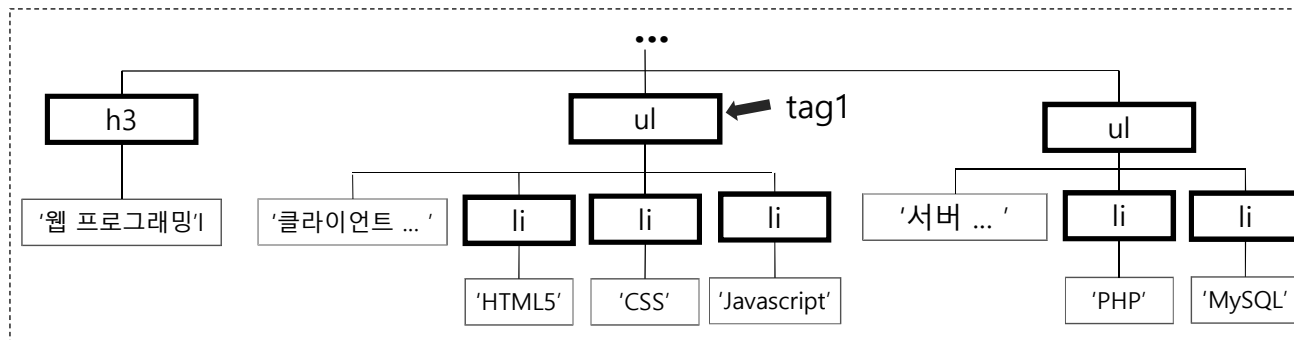
Programming Language

- C/C++
 - HTML5
 - JScript
 - PHP
 - JSP
- Java
- Python
- FORTRAN

현재위치 노드	빨강색 박스
직전 형제노드	C/C++
직후 형제노드	Java
첫번째 자식노드	HTML5
두번째 자식노드	JScript
마지막 자식노드	JSP

[예 8.17] HTML 문서 작성 예(1) : HTML 태그 사이에 공백문자 없음

```
<h3>웹프로그래밍</h3><ul id='t1'>클라이언트 프로그래밍<li>HTML5</li><li>CSS3</li><li>JavaScript</li></ul><ul>서버 프로그래밍<li>PHP</li><li>MySQL</li></ul>
```



DOM 예제

file:///C:/html5/ch08/...

웹프로그래밍

- 클라이언트 프로그래밍
 - HTML5
 - CSS3
 - JavaScript
- 서버 프로그래밍
 - PHP
 - MySQL

**** DOM 트리 탐색 ****

노드의 자식노드 개수: 4

0 번째 자식노드 : [object Text]

1 번째 자식노드 : [object HTMLLIElement]

2 번째 자식노드 : [object HTMLLIElement]

3 번째 자식노드 : [object HTMLLIElement]

[예 8.18] HTML 문서 작성 예(2) : HTML 태그 사이에 공백문자 있음

```
<h3>웹프로그래밍</h3>
<ul id='t1'>클라이언트 프로그래밍
  <li>HTML5</li>
  <li>CSS3</li>
  <li>JavaScript</li>
</ul>
<ul>서버 프로그래밍
  <li>PHP</li>
  <li>MySQL</li>
</ul>
```

```
<script>
  var tag1 = document.getElementById('t1');
  document.write('<hr> **** DOM 트리 탐색 **** <br>');
  document.write('노드의 자식노드 개수: ' + tag1.childNodes.length + '<br>');
  for ( i=0; i<tag1.childNodes.length ; i++) {
    document.write( i + ' 번째 자식노드 : ' + tag1.childNodes[i] + '<br>');
  }
</script>
```

DOM 예제

file:///C:/html5/ch08/...

웹프로그래밍

클라이언트 프로그래밍

- HTML5
- CSS3
- JavaScript

서버 프로그래밍

- PHP
- MySQL

**** DOM 트리 탐색 ****

노드의 자식노드 개수: 7

0 번째 자식노드 : [object Text]

1 번째 자식노드 : [object HTMLLIElement]

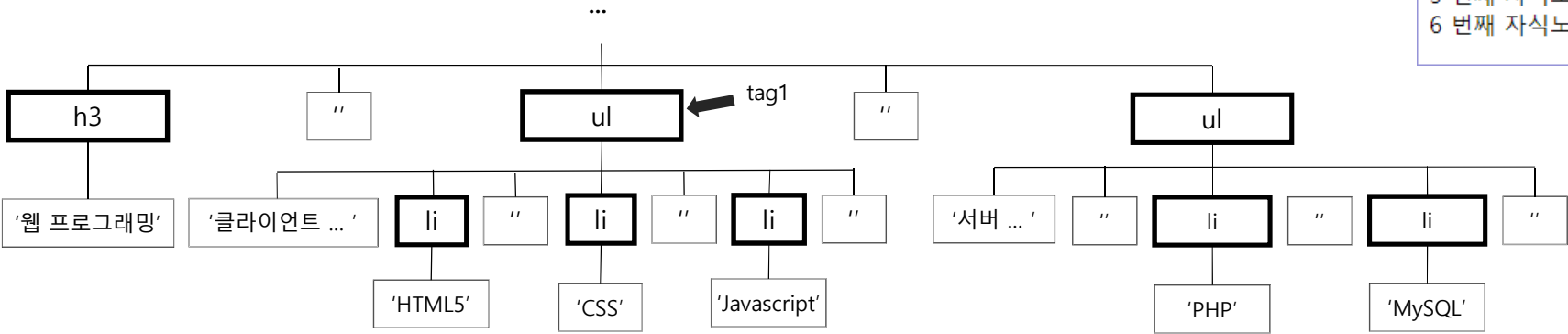
2 번째 자식노드 : [object Text]

3 번째 자식노드 : [object HTMLLIElement]

4 번째 자식노드 : [object Text]

5 번째 자식노드 : [object HTMLLIElement]

6 번째 자식노드 : [object Text]



8.3 HTML 입력양식의 데이터 처리

8.3.1 텍스트 필드의 입력 처리

- <input> 요소객체의 value 속성을 접근해서 처리함

[HTML 코드]

```
<input type="text" id="idName" name="varName" .... />  
....  
<div id="d1"></div> // 사용자 입력의 출력 공간
```

[자바스크립트 코드]

```
var obj=document.getElementById ("idName"); // ❶  
document.getElementById ("d1").innerHTML=obj.value); // 사용자입력의 출력
```

```

<form>
  중간시험 성적 : <input type="text" id="mid" /> <br>
  기말시험 성적 : <input type="text" id="last" /> <hr>
</form>
<button onclick="f1();" /> 처리결과 </button>
<div id="d1"> </div>

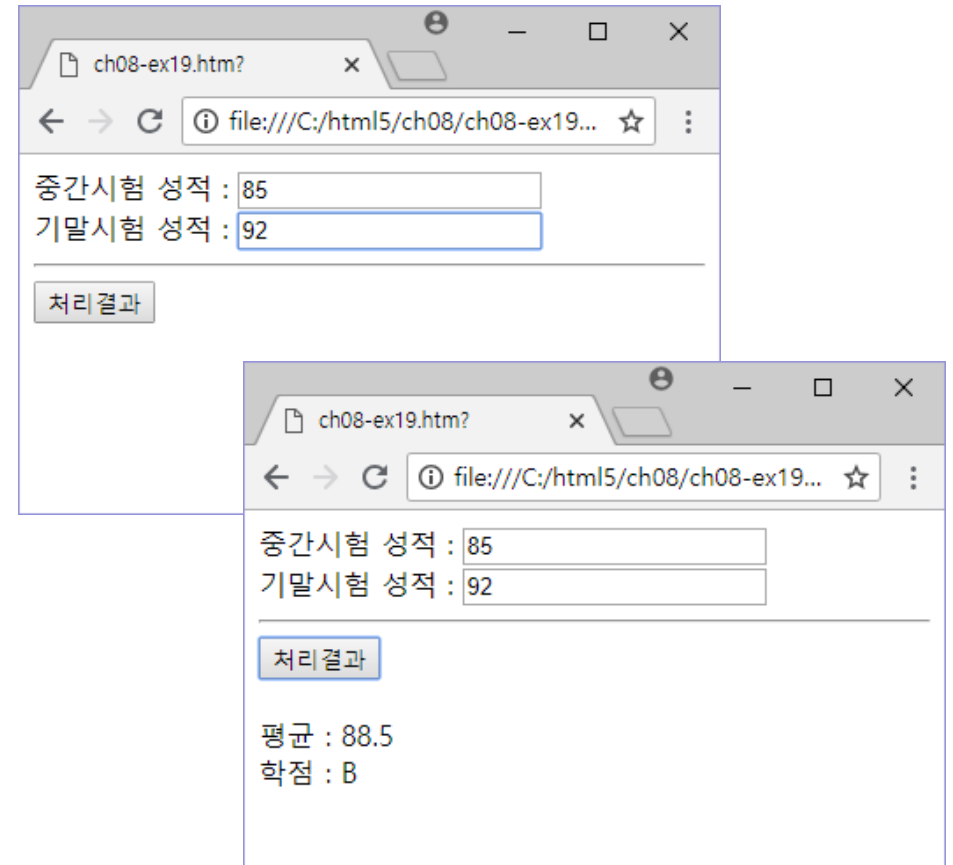
<script>
function f1() {
  var avg, grade, result;
  var s1 = document.getElementById('mid').value;
  var s2 = document.getElementById('last').value;
  var pTag = document.getElementById('d1');

  avg = (Number(s1)+Number(s2)) / 2;
  if (avg>=90) grade='A';
  else if (avg>=80) grade='B';
  else if (avg>=70) grade='C';
  else if (avg>=60) grade='D';
  else grade='F';

  result = "<br> 평균 : " + avg;
  result = result + "<br> 학점 : " + grade;
  pTag.innerHTML = result;
};
</script>

```

[예 8.19] 텍스트 입력필드의 사용자 입력 처리



8.3.2 라디오 버튼과 체크박스의 입력 처리

라디오 버튼의 입력 처리 예

[HTML 코드]

```
<input type="radio" name="rbotton" value="값1" checked />라디오버튼 라벨1  
<input type="radio" name="rbotton" value="값2" />라디오버튼 라벨2  
...  
<input type="radio" name="rbotton" value="값3" />라디오버튼 라벨n  
....  
<div id="d1"></div> // 사용자 입력의 출력 공간
```

[자바스크립트 코드]

```
var radios = document.getElementsByName("rbotton");  
for ( var i= 0; i<radios.length ; i++) {  
    if ( radios[i].checked == true ) {  
        radioValue = radios[i].value;  
        break;  
    }  
}  
document.getElementById ("d1").innerHTML=radioValue; // 입력값 출력
```

체크박스 입력 처리 예

[HTML 코드]

```
<input type="checkbox" name="cbox" value="값1" />체크박스 라벨1  
<input type="checkbox" name="cbox" value="값2" />체크박스 라벨2  
...  
<input type="checkbox" name="cbox" value="값3" />체크박스 라벨n  
....  
<div id="d1"></div> // 사용자 입력의 출력 공간
```

[자바스크립트 코드]

```
var checkobjs=document.getElementsByName("cbox");  
var cboxValue = ""  
for ( var i=-; i<checkobjs.length ; i++) {  
    if ( checkobjs[i].checked == true )  
        cboxValue = cboxValue + ',' + checkobjs[i].value;  
}  
document.getElementById("d1").innerHTML=cboxValue; // 입력값 출력
```

```

<script>
function dataReading() {
    var output1 = "", output2 = "";
    var years = document.getElementsByName("year");
    for ( var i= 0; i<years.length ; i++) {
        if ( years[i].checked == true ) {
            yearValue = years[i].value; break;
        }
    }
    output1 = output1 + '학년 : ' + yearValue;

    var subjects=document.getElementsByName("subject");
    var checkedValue = ""
    for ( i=0; i<subjects.length ; i++) {
        if (subjects[i].checked == true)
            checkedValue = checkedValue + ' ' + subjects[i].value;
    }
    output2 = output2 + '좋아하는 과목 : ' + checkedValue;

    var obj = document.getElementById("result");
    obj.innerHTML = output1 + '<br>' + output2 ;
}
</script>

```

[예 8.20] 라디오 버튼과 체크박스의 입력 처리

DOM 예제

file:///C:/html5/ch08/ch08-ex20.htm

Q1. 학생의 학년은 ?

☐ 1학년 ☒ 2학년 ☐ 3학년 ☐ 4학년

Q2. 학생이 좋아하는 과목은 ?

☐ 알고리즘 ☒ 자료구조 ☐ 데이터베이스 ☐ 운영체제 ☐ 네트워크

학년 : 2
좋아하는 과목 : 자료구조

```

<form>
<h3>Q1. 학생의 학년은 ? </h3>
<input type="radio" name="year" value="1" />1학년
<input type="radio" name="year" value="2" checked />2학년
<input type="radio" name="year" value="3" />3학년
<input type="radio" name="year" value="4" />4학년
<br>
<h3>Q2. 학생이 좋아하는 과목은 ? </h3>
<input type="checkbox" name="subject" value="알고리즘" />알고리즘
<input type="checkbox" name="subject" value="자료구조" />자료구조
<input type="checkbox" name="subject" value="데이터베이스" />데이터베이스
<input type="checkbox" name="subject" value="운영체제" />운영체제
<input type="checkbox" name="subject" value="네트워크" />네트워크 <br>
<input type="button" value="입력 완료" onclick="dataReading();" />
</form>
<div id="result" > </div>

```


8.3.3 콤보박스의 입력 처리

관련 속성들	기능
length	select 객체의 속성, 옵션 항목 개수
selectedIndex	select 객체의 속성, 선택된 옵션 항목의 인덱스
size	select 객체의 속성, 콤보박스에 표시되는 옵션 항목 개수
options[]	select 객체의 자식 컬렉션 객체, 모든 옵션 항목들의 컬렉션
selected	option 객체의 속성, 선택 여부 표시 (true, false)
value	option 객체의 속성, 선택될 때의 데이터(문자열)

[HTML 코드]

```
<select id="s">
  <option value="값1" selected > </option>
  <option value="값1"> </option>
  ...
  <option value="값1"> </option>
</select>
<div id="d1"></div> // 사용자 입력의 출력 공간
```

[자바스크립트 코드]

```
var comboValue = ""; // 콤보박스의 사용자 입력 데이터
var obj=document.getElementById("s");

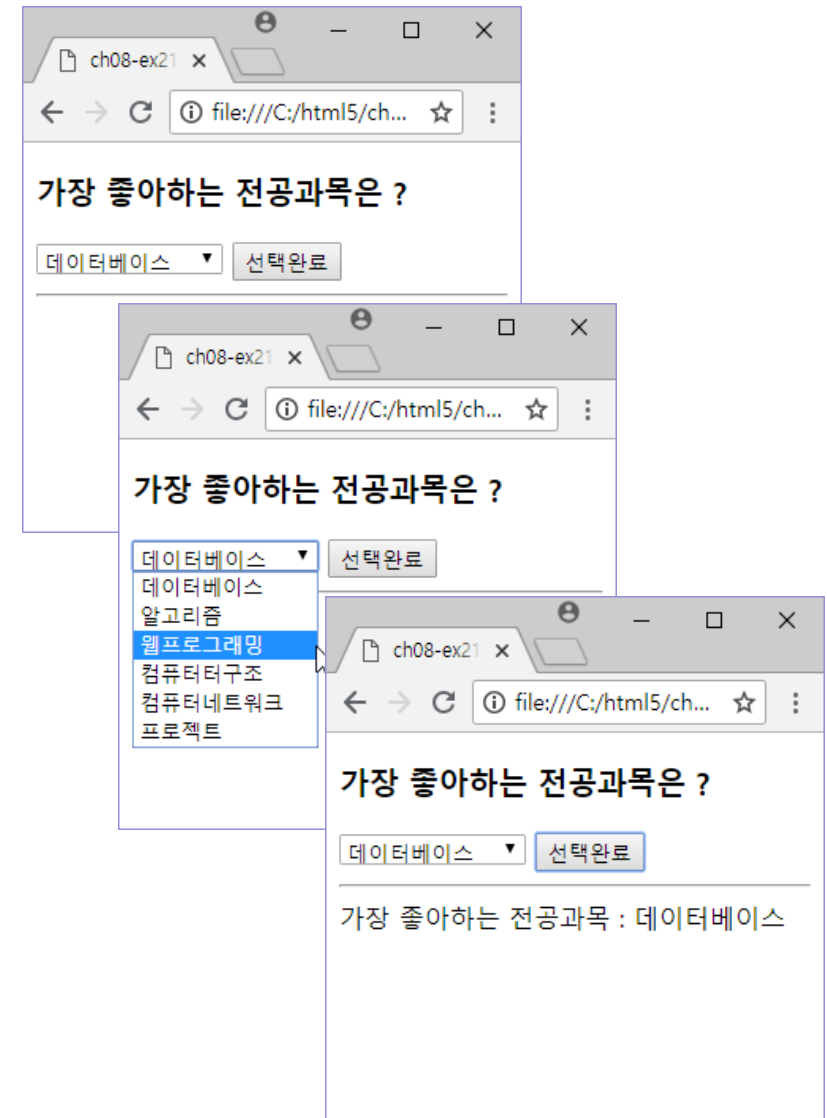
// ❶ 다중 선택 콤보박스의 데이터 확인
for (i=1; i<obj.length; i++)
  if (obj.options[i].selected==true)
    comboValue = comboValue + ', ' + obj.options[i].value;
}
document.getElementById("d1").innerHTML= comboValue;

// ❷ 단일 선택 콤보박스의 데이터 확인
comboValue=obj.options[obj.selectedIndex].value);
document.getElementById("d1").innerHTML= comboValue;
```

[예 8.21] 콤보박스 입력 처리

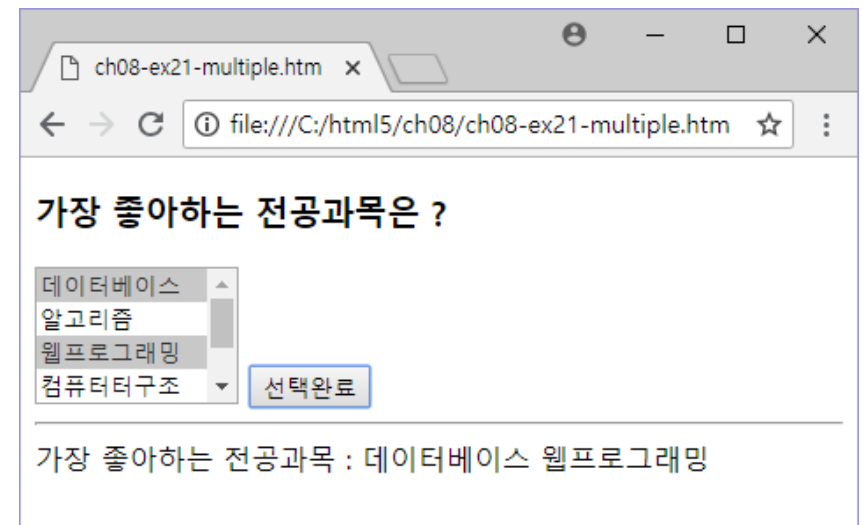
```
<script>
function dataReading() {
    var output = "가장 좋아하는 전공과목 : ";
    var obj = document.getElementById("s");
    output = output + obj.options[obj.selectedIndex].value;
    var obj = document.getElementById("result");
    obj.innerHTML = output ;
}
</script>

</head>
<body>
<h3>가장 좋아하는 전공과목은 ? </h3>
<select id="s" >
    <option value="데이터베이스">데이터베이스</option>
    <option value="알고리즘">알고리즘</option>
    <option value="웹프로그래밍" selected>웹프로그래밍</option>
    <option value="컴퓨터구조">컴퓨터구조</option>
    <option value="컴퓨터네트워크" >컴퓨터네트워크</option>
    <option value="프로젝트">프로젝트</option>
</select>
<input type="button" value="선택완료" onclick="dataReading();" />
<hr>
<div id="result" > </div>
```



[그림 8.6] [예 8.21]의 다중 선택 콤보박스 입력 처리

```
<script>
function dataReading () {
    var output = "가장 좋아하는 전공과목 : ";
    var obj = document.getElementsByName ("course");
    for ( i=0; i < obj.length ; i++) {
        if (obj[i].selected) output = output + obj[i].value + " ";
    }
    var obj = document.getElementById ("result");
    obj.innerHTML = output ;
}
</script>
```



8.4 이벤트 처리

- 자바스크립트의 이벤트처리 지원
 - ✓ 이벤트속성 제공
 - ✓ 예) onclick, onmouseover 등
- 이벤트핸들러(event handler 또는 이벤트리스너(event listener))
 - ✓ 이벤트가 발생했을 때 실행되는 함수
- 이벤트 등록(event registration)
 - ✓ 이벤트와 이벤트핸들러를 연결하는 과정
- 이벤트 모델 : 이벤트 등록 방법
 - 1) 인라인 이벤트 모델(inline event model)
 - ✓ HTML 태그에서 이벤트속성에 자바스크립트 코드 또는 이벤트핸들러를 속성값으로 지정함
 - 2) 고전 이벤트 모델
 - 3) 표준 이벤트 모델

(1) 고전 이벤트 모델

- 자바스크립트에서 직접 DOM 요소객체의 이벤트 속성에 이벤트핸들러를 지정함

```
var obj = document.getElementById('id값'); // DOM 요소 객체;  
obj.이벤트 속성 = 이벤트핸들러;
```

- 자바스크립트에서 직접 DOM 요소객체의 이벤트속성을 지정할 때는 반드시 이벤트 속성값으로 이벤트핸들러 함수를 지정해야 함

[예 8.22]
고정 이벤트 모델을 이용한
[예 8.21]의 재작성

```
<script>  
window.onload = function() {  
  document.getElementById("b1").onclick=dataReading;  
}  
function dataReading() {...생략...}  
</script>  
  
<h3>가장좋아하는전공과목은?</h3>  
<select id="s " multiple>  
  ...생략...  
</select>  
<button id="b1">선택완료</button> <hr>  
<div id="result"> </div>
```

(2) 표준 이벤트 모델

- W3C에서 공식적으로 권장하는 표준 이벤트 처리 방법
- DOM 요소객체의 이벤트 속성 대신 메서드 `addEventListener()`를 이용해서 등록함
- 한 이벤트에 여러 이벤트핸들러들을 지정할 수도 있음

```
var obj = document.getElementById('id값'); // DOM 요소객체
obj.addEventListener('이벤트명1', 이벤트핸들러1, '이벤트명2', 이벤트핸들러2, ... );
```

[예 8.23] 표준 이벤트 모델을 이용한 [예 8.21]의 재작성

```
<script>
window.onload = function(){
  document.getElementById("b1").addEventListener("click",dataReading);
}
  function dataReading() { ... 생략 ... }
</script>
<h3>가장 좋아하는 전공과목은?</h3>
<select id="s" multiple> ...생략... </select>
<button id="b1">선택완료</button> <hr>
<div id="result"> </div>
```

8.4.2 이벤트 객체

- 이벤트에 대한 정보를 가지고 있는 객체
- 특정 이벤트가 발생하면 웹 브라우저는 이벤트 객체를 자동으로 생성함
- 이벤트가 발생하면, 웹 브라우저는 이벤트 객체 `e`를 생성해서 이벤트핸들러의 첫 번째 인수로 전달함
- 이벤트핸들러들은 첫 번째 인수로 `e`를 지정해서 이벤트 객체를 전달받아 처리할 수 있음
- 이벤트 객체를 사용하지 않는다면, 이벤트핸들러를 정의할 때 인수 `e`는 생략할 수 있음

```
function eHandler(e) { //코드생략 }
```

- ❶ 인라인-이벤트 모델 : `<HTML태그...onclick="eHandler(event);">...</HTML태그>`
- ❷ 고전 이벤트 모델 : `obj.onclick=eHandler;` 또는 `obj.onclick`
- ❸ 표준 이벤트 모델 : `obj.addEventListener("click", eHandler);`

[표 8.7] 이벤트 객체의 주요 속성과 메서드

속성 및 메서드	의미 또는 기능
<code>type</code>	이벤트 종류(문자열)
<code>target</code>	해당 이벤트를 발생시킨 객체
<code>defaultPrevented</code>	해당 이벤트의 디폴트 처리의 중지 여부 표시(true, false)
<code>preventDefault()</code>	해당 이벤트의 디폴트 처리를 중지함

[예 8.24] 이벤트 모델별 이벤트 객체 처리

```
<script>
  window.onload = function () {
    document.getElementById("old").onclick = eventHandler ;
    document.getElementById("old").onmouseover = eventHandler ;
    document.getElementById("standard").addEventListener("click", eventHandler);
    document.getElementById("standard").addEventListener("mouseover", eventHandler);
  }

  function eventHandler(e) {
    var output = e + " : " + e.type + " 이벤트가 발생함 ! ";
    var obj = document.getElementById("result");
    obj.innerHTML = output ;
  }
</script>
```

<h4> 이벤트 모델의 이벤트객체 처리 </h4> <hr>

인라인이벤트 모델 :

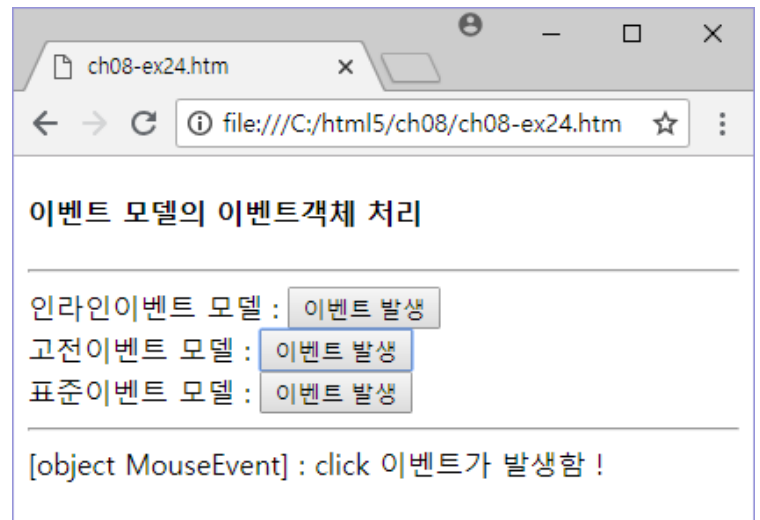
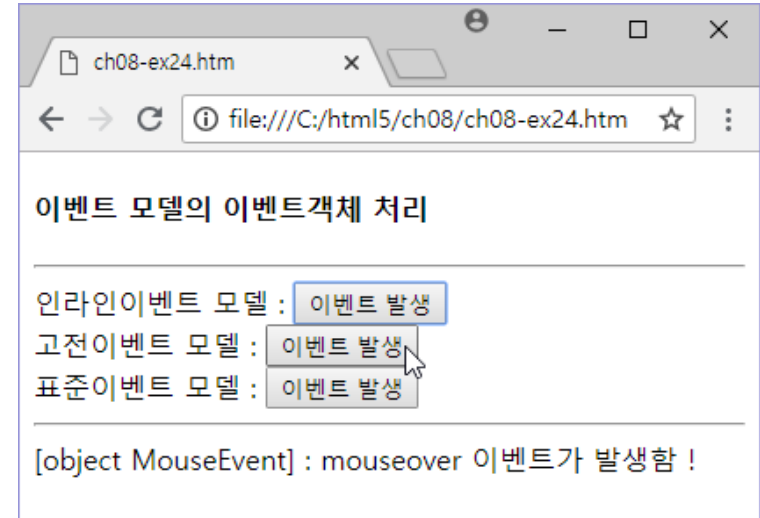
```
<button onclick="eventHandler(event);"
onmouseover="eventHandler(event);" >이벤트 발생</button> <br>
```

고전이벤트 모델 :

```
<button id="old" >이벤트 발생</button> <br>
```

표준이벤트 모델 :

```
<button id="standard" >이벤트 발생</button> <br> <hr>
<div id="result" > </div>
```



● 디폴트 이벤트(default event) 처리

- 웹 브라우저가 HTML 요소에서 특정 이벤트가 발생하면 수행되는 작업을 미리 정의해 둔 것
 - ✓ 예) <a> 태그 또는 입력양식의 전송버튼이나 리셋버튼의 기능
- 디폴트 이벤트 처리의 수행을 막을 필요성
 - ✓ 예) 데이터의 서버 전송에 앞서 사용자 입력의 유효성 검사를 수행함

[디폴트 이벤트
처리의 방지]

```
function eventHandler1 () { return false; }  
function eventHandler2 (e) { e.preventDefault();}
```

(1) false를 반환하는 이벤트핸들러 지정

- ① <tagName id="t1" onclick=" return false;" ... > ... </tagName>
- ② obj.onclick=eventHandler1; // obj=document.getElementById("t1");
- ③ obj.addEventListener("click", eventHandler1) ; // obj=document.getElementById("t1");

(2) 이벤트 객체 e의 preventDefault() 사용

- ① <tagName id="t1" onclick="event.preventDefault();" ... > ... </tagName>
- ② obj.onclick="eventHandler2"; // obj=document.getElementById("t1");
- ③ obj.addEventListener("click", eventHandler2); // obj=document.getElementById("t1");

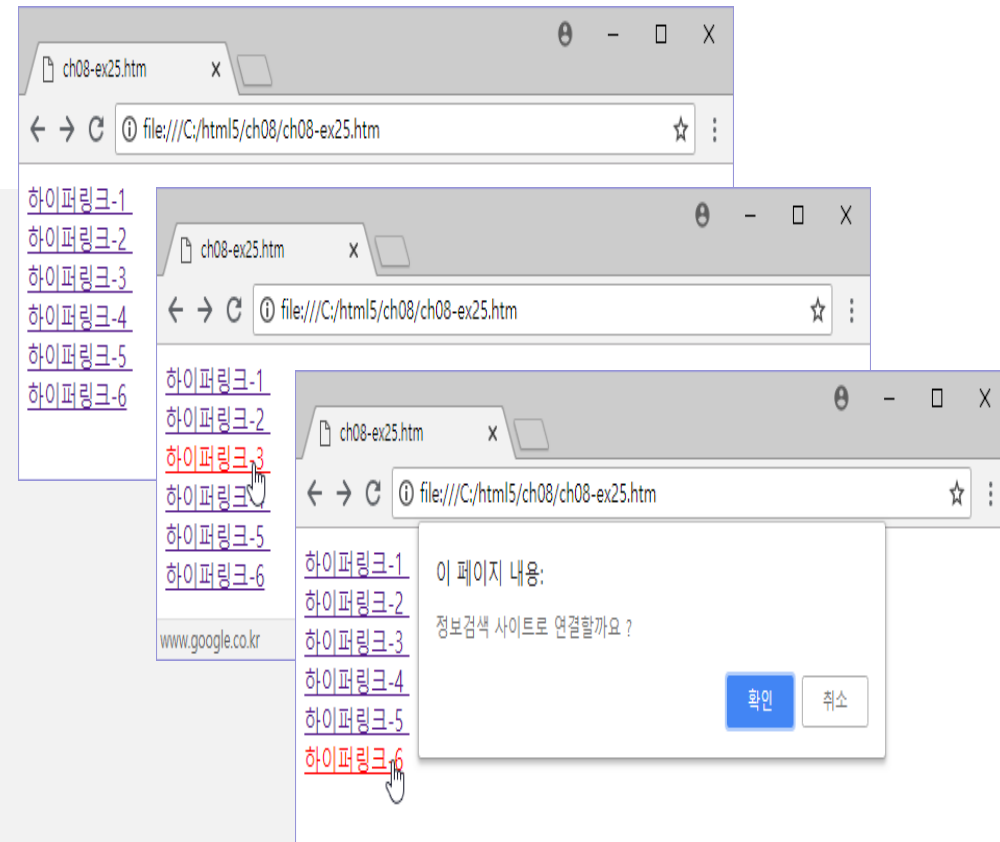
[예 8.25] <a> 요소의 디폴트 이벤트 처리

```
<script>
window.onload = function () {
    document.getElementById("a1").onclick = f;
    document.getElementById("a2").addEventListener("click", f);
    document.getElementById("a3").onclick = f2;
}

function f(e) {
    e.preventDefault();
}

function f2(e) {
    var result = confirm("정보검색 사이트로 연결할까요 ?");
    if ( result == false ) e.preventDefault();
}
</script>
```

```
<a href="http://www.google.co.kr" target="_blank"> 하이퍼링크-1 </a><br>
<a href="http://www.google.co.kr" target="_blank" onclick=" return false; "> 하이퍼링크-2 </a><br>
<a href="http://www.google.co.kr" target="_blank" onclick=" f(event); "> 하이퍼링크-3 </a><br>
<a href="http://www.google.co.kr" target="_blank" id="a1"> 하이퍼링크-4 </a><br>
<a href="http://www.google.co.kr" target="_blank" id="a2"> 하이퍼링크-5 </a><br>
<a href="http://www.google.co.kr" target="_blank" id="a3"> 하이퍼링크-6 </a>
```



8.4.3 많이 사용되는 이벤트 처리

분류	이벤트	이벤트 의미
마우스 관련	click, dbclick	클릭, 더블클릭
	mousedown, mousemove, mouseup	누름, 누른 상태의 이동, 누른 상태에서 놓음
	mouseenter, mouseover, mouseout	객체 영역 진입함, 위치함, 벗어남
	contextmenu	디폴트 이벤트, 단축 메뉴 제공
키보드 관련	keypress, keydown, keyup	각각 키 누름, 키 입력, 누른 키를 놓음
폼태그 관련	change	입력값이 변경됨
	focus, blur	포커스(데이터 입력 상태)를 가짐, 잃어버림
	select	선택함
윈도우 관련	load, unload	웹 문서를 웹 브라우저에 적재/제거함
	resize	웹 브라우저 창 크기를 변경함

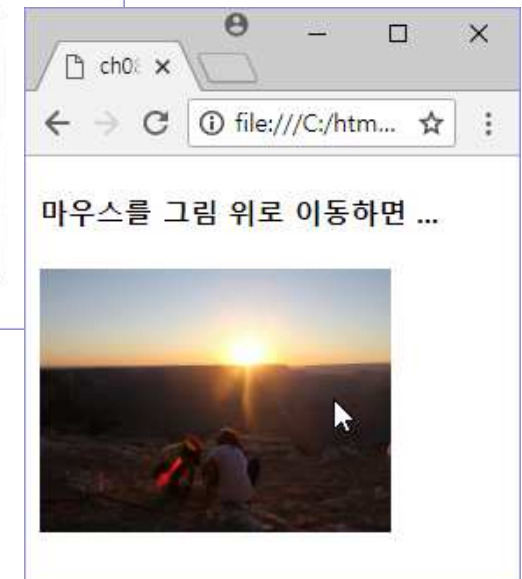
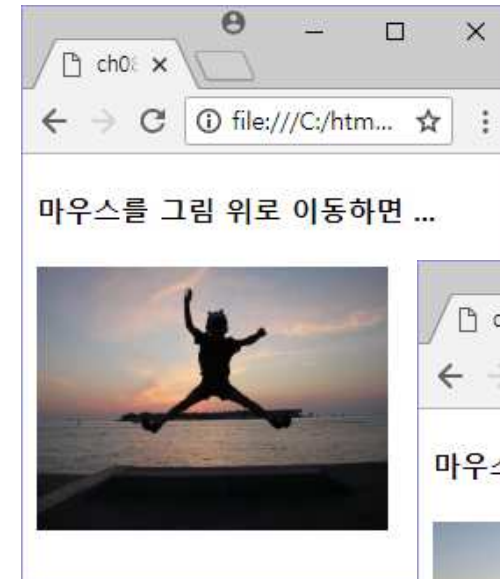
(1) mouseenter / mouseover / mouseleave / mouseout 이벤트

[예 8.26] mouseover/mouseout 이벤트 처리

```
<script>
  function f1(obj) { obj.src = "image2.jpg"; }

  function f2(obj) { obj.src = "image1.jpg"; }
</script>
<H4> 마우스를 그림 위로 이동하면 ... </h4>

```

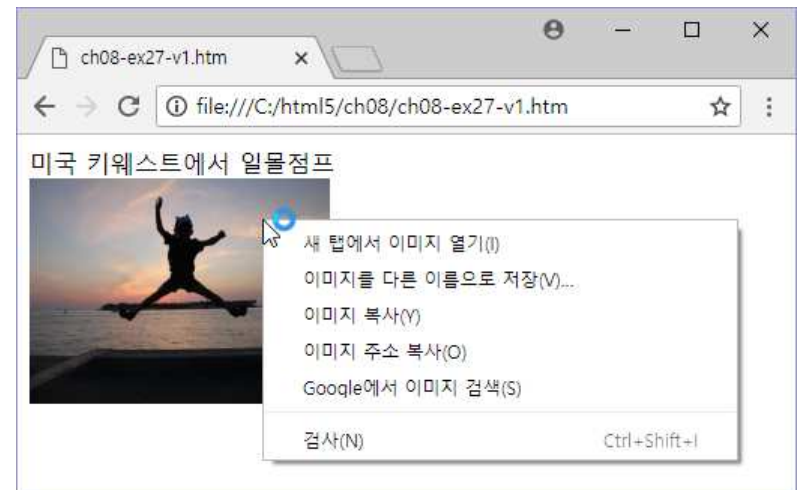
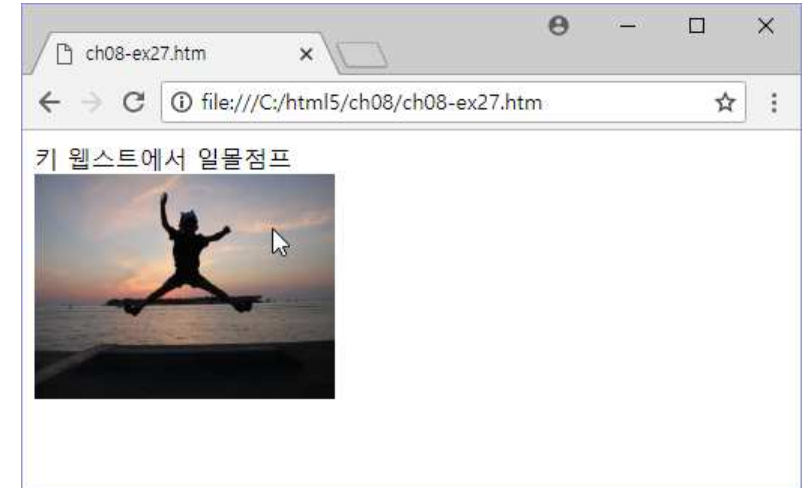


(2) contextmenu 이벤트

- 웹 브라우저 화면이나 특정 HTML 요소 위에서 마우스의 오른쪽 버튼을 누르면 발생하는 이벤트
- 소스보기, 이미지 다운로드 등과 같은 메뉴를 제공하는 컨텍스트 메뉴를 디폴트 이벤트 처리로 제공함

[예 8.27] contextmenu 이벤트 처리

```
키 웨스트에서 일몰점프<br>  
  
</div>
```

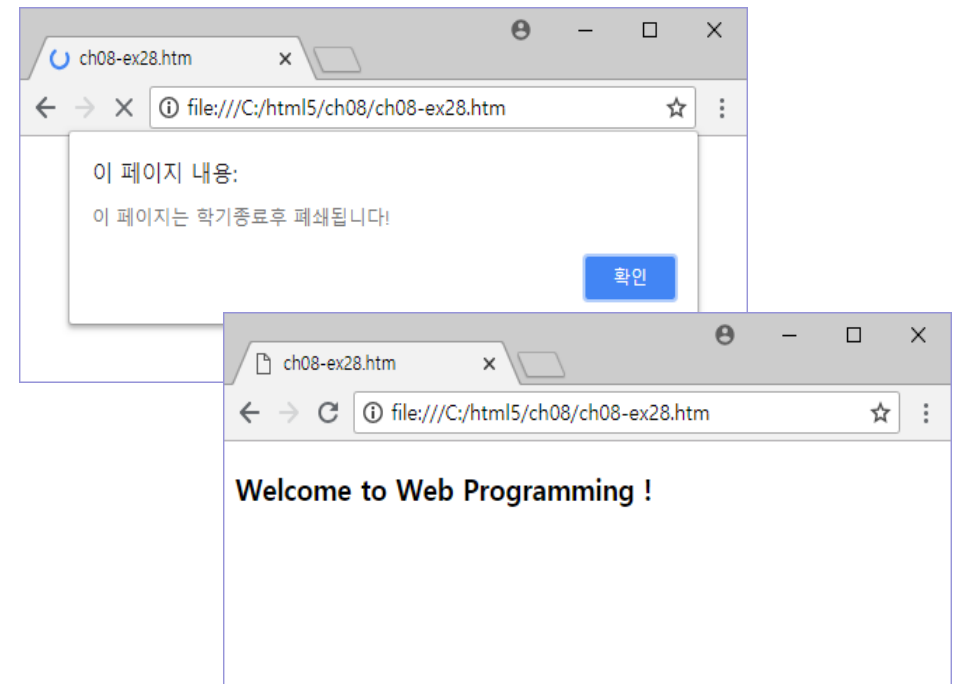


(3) load / unload 이벤트

- 웹 문서나 이미지 파일을 웹 브라우저로 로드하면 파일 용량에 따라 로딩되는 시간이 필요함
- load 이벤트는 **파일의 로드가 끝나면 발생함**

[예 8.28] load 이벤트 처리

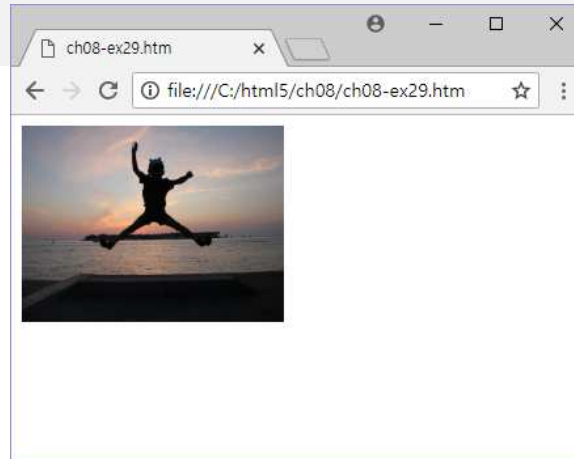
```
<body onload='f();' >
  <script>
    function f() {
      alert("이 페이지는 학기종료후 폐쇄됩니다!");
    }
  </script>
  <h3>Welcome to Web Programming !</h3>
</body>
```



[예 8.29] load 이벤트를 사용하지 않은 이미지 파일 크기 출력

```
 <br>
<p id="demo"></p>
<img id="fig2" src="" />

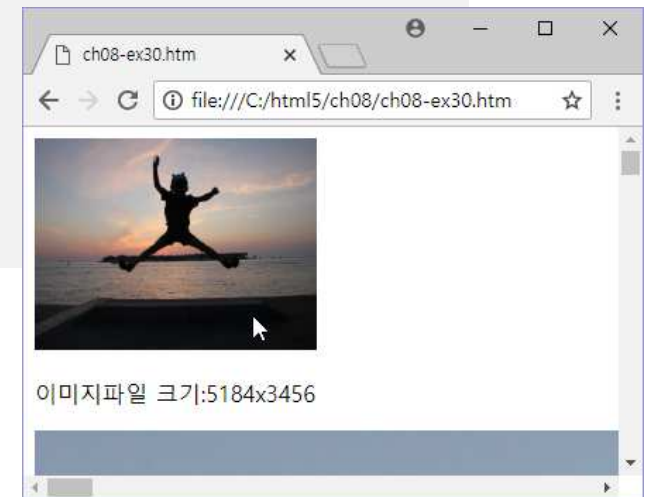
<script>
function f() {
    var obj=document.getElementById("fig2");
    obj.src="image2.jpg"; // 이미지 파일 로드 요청 및 백그라운드 로딩 작업시작
    document.getElementById("demo").innerHTML =
        '이미지파일 크기:' + obj.width + 'x' + obj.height;
}
</script>
```



[예 8.30] load 이벤트를 이용한 이미지 파일 크기 출력

```
 <br>
<p id="demo"> </p>
<img id="fig2" src="" />

<script>
function f() {
    var obj=document.getElementById("fig2");
    obj.src="image2.jpg"; // 이미지 파일 로드 요청 및 백그라운드 로딩 작업시작
    obj.onload = function () {
        document.getElementById("demo").innerHTML =
            '이미지파일 크기:' + obj.width + 'x' + obj.height;
    }
}
</script>
```



(4) focus / blur 이벤트

■ focus 이벤트

- ✓ 입력양식에서 특정 입력필드가 포커스를 가질 때 발생하는 이벤트
- ✓ 해당 입력필드를 마우스로 선택하거나 autofocus 속성을 이용해서 자동포커스 이동을 지정했을 때 직전 입력필드에서 <Enter> 키를 누르면 발생됨

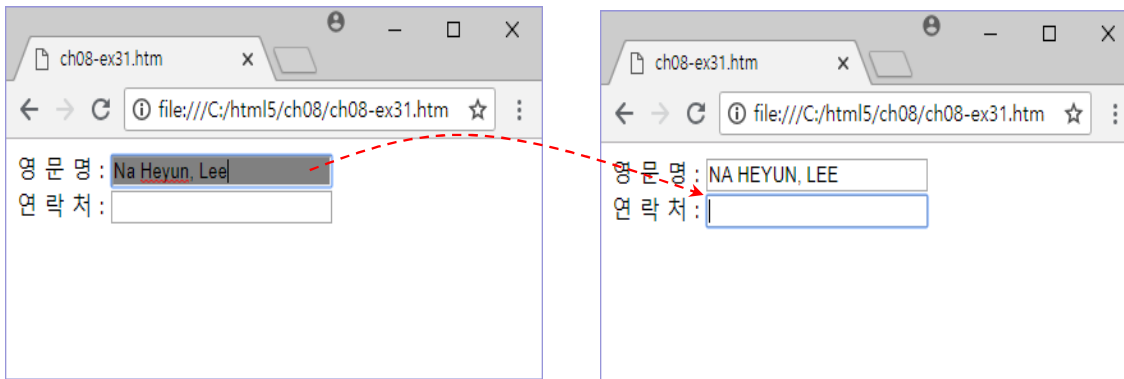
■ blur 이벤트

- ✓ 입력필드가 포커스를 잃게 될 때 발생하는 이벤트
- ✓ 현재 포커스를 가진 입력필드에서 엔터키를 누르거나 마우스로 다른 영역을 선택하면 발생함

[예 8.31] focus와 blur 이벤트 처리

```
<body>
영문명 : <input type="text" id="fname"
           onfocus="f1();"
           onblur="f2()" /> <br>
연락처 : <input type="text" />
<script>
function f1() {
    var x = document.getElementById("fname");
    x.style.background = "gray"; }

function f2() {
    var x = document.getElementById("fname");
    x.style.background = "white";
    x.value = x.value.toUpperCase();
}
</script>
```

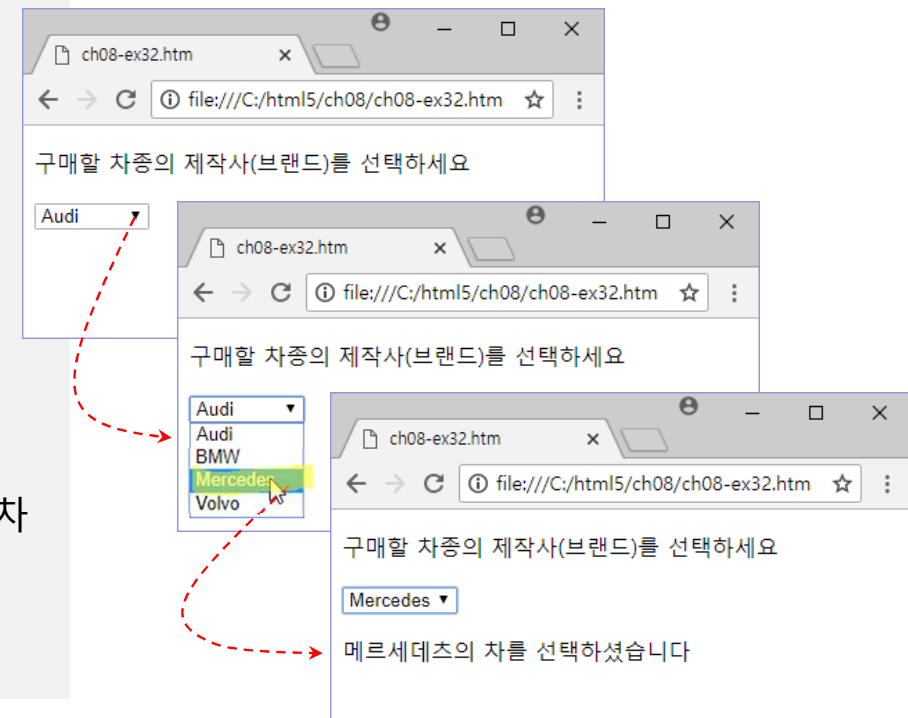


(5) change 이벤트

- 입력양식의 입력필드 값이 변경될 때마다 발생하는 이벤트
- `<input type="text">`, `<input type="password">`, `<input type="file">`, `<input type="radio">`, `<input type="checkbox">`, `<input type="range">`, `<input type="search">`, `<select>`, `<textarea>` 요소에서 change 이벤트 처리를 지정할 수 있음

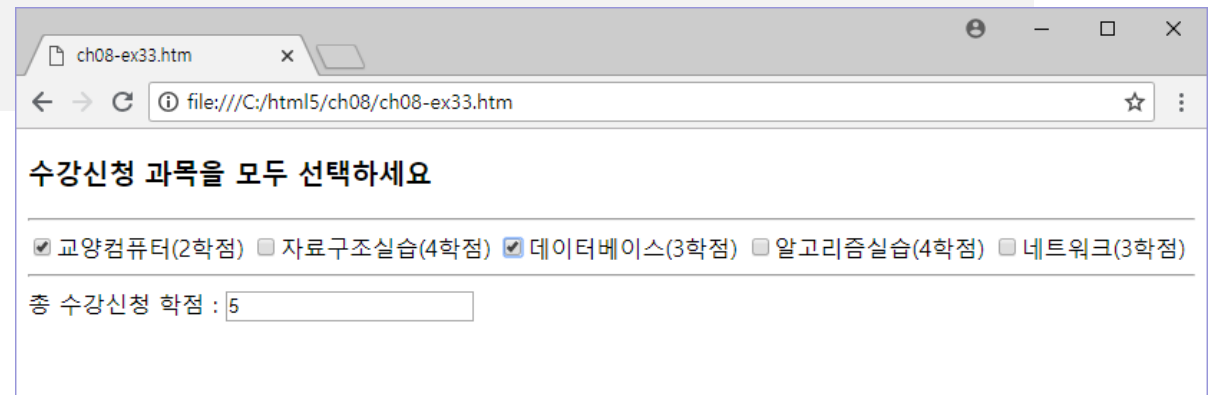
[예 8.32] 콤보박스 입력의 change 이벤트 처리

```
<p>구매할 차종의 제작사(브랜드)를 선택하세요 </p>
<select id="mySelect" onchange="f()">
  <option value="아우디">Audi
  <option value="BMW">BMW
  <option value="메르세데츠">Mercedes
  <option value="볼보">Volvo
</select>
<p id="demo"> </p>
<script>
  function f() {
    var x = document.getElementById("mySelect").value;
    document.getElementById("demo").innerHTML = x + "의 차
를 선택하셨습니다" ;
  }
</script>
```



[예 8.33] 체크박스 입력의 change 이벤트 처리

```
<h3> 수강신청 과목을 모두 선택하세요 </h3> <hr>
<input type="checkbox" name="s" value="2" onchange="apply(this)"/> 교양컴퓨터(2학점)
<input type="checkbox" name="s" value="4" onchange="apply(this)"/> 자료구조실습(4학점)
<input type="checkbox" name="s" value="3" onchange="apply(this)"/> 데이터베이스(3학점)
<input type="checkbox" name="s" value="4" onchange="apply(this)"/> 알고리즘실습(4학점)
<input type="checkbox" name="s" value="3" onchange="apply(this)"/> 네트워크(3학점) <hr>
총 수강신청 학점 : <input type="text" id="sum" value="" />
<script>
  var total=0;
  function apply(subject) {
    if (subject.checked==true) { total=total+parseInt(subject.value); }
    else { total=total- parseInt(subject.value); }
    document.getElementById("sum").value= total ;
  }
</script>
```



수강신청 과목을 모두 선택하세요

☒ 교양컴퓨터(2학점) ☐ 자료구조실습(4학점) ☒ 데이터베이스(3학점) ☐ 알고리즘실습(4학점) ☐ 네트워크(3학점)

총 수강신청 학점 : 5

Next 9장



jQuery