

HTML5+ CSS3+ Javascript 웹 프로그래밍

A large field of blue flowers, likely cornflowers, stretches across the foreground and middle ground. A single, mature pine tree stands prominently in the center of the field. The background shows rolling hills under a clear blue sky. The number '09' is overlaid in large white font on the right side of the image.

09

이벤트 기초 및 활용

주교재 목차

2

HTML5

01장. 웹 프로그래밍과 HTML5 개요

02장. HTML 기본 문서 만들기

03장. HTML5 문서 구조화와 웹 폼

CSS3

04장. CSS3로 웹 페이지 꾸미기

05장. CSS3 고급 활용

Javascript

06장. 자바스크립트 언어

07장. 자바스크립트 코어 객체와 배열

08장. HTML DOM과 Document ★

09장. 이벤트 기초 및 활용 ★

10장. 윈도우와 브라우저 관련 객체(BOM)

11장. HTML5 캔버스 그래픽

12장. HTTP와 쿠키, 웹 스토리지

13장. 오디오, 비디오 제어 및 위치 정보 서비스, 웹 워커

14장. 웹 프로그래밍 응용 과제

3

강의 목표

1. 이벤트가 무엇이고 언제 발생하는지 안다.
2. 자바스크립트 코드로 이벤트 리스너를 작성할 수 있다.
3. 발생하는 이벤트가 DOM 트리를 따라 흘러가는 경로를 안다.
4. 문서와 이미지의 로딩 완료 시 호출되는 onload 리스너를 작성할 수 있다.
5. 폼에 발생하는 이벤트 리스너를 다룰 수 있다.
6. 마우스 관련 이벤트를 다룰 수 있다.
7. 키 관련 이벤트를 다룰 수 있다.

이벤트 개요

4

[참고]

- Procedural programming : C, BASIC
- Event-driven programming(OOP) : C++, JAVA

□ 이벤트(Event)

- 마우스 클릭, 키보드 입력, 이미지나 HTML 문서의 로딩, 타이머의 타임아웃 등 사용자의 입력 행위나 문서나 브라우저의 상태 변화를 자바스크립트 코드에게 알리는 통지(notification)

□ 이벤트 리스너(Event Listener)

이벤트가 발생하면 자동 호출되는 함수 형태로 구현

- 발생한 이벤트에 대처하기 위해 작성된 자바스크립트 코드

6장 자바스크립트 언어

onclick 이벤트
리스너 속성자바스크립트 코드
(이미지를 banana.png로 교체)

```

```

□ 이벤트 종류

- HTML5에서 이벤트 종류는 70여 가지 → 뒤에 정리
- 이벤트 리스너 이름은 이벤트 이름 앞에 on을 덧붙임

예) onmousedown : mousedown 이벤트의 리스너

onkeydown : keydown 이벤트의 리스너

on+이벤트 = 이벤트 리스너

브라우저에 발생하는 다양한 이벤트들

5

load 이벤트
(HTML 문서 전체 로딩 완료 시)

load 이벤트
(이미지의 로딩 완료 시)

click 이벤트
(마우스 클릭 시)

submit 이벤트
(submit 버튼 클릭 시)

reset 이벤트
(reset 버튼 클릭 시)

focus 이벤트
(입력하기 위해 선택 시)

keypress 이벤트
(키를 누를 때)

keyup 이벤트
(누른 키를 놓을 때)

dblclick 이벤트
(마우스 더블클릭 시)


change 이벤트
(라디오버튼 선택 시)

resize 이벤트
(윈도우 크기 변경 시)

이벤트가 뭐지

localhost/9/fig9-01.html

이벤트는 언제 발생하는 거야?



강아지를 분양합니다. 잘 키워주실 분 연락처 남겨 주세요.

도시 ☐ 서울 ☐ 부산 ☒ 광주

전화번호

HTML5 이벤트 리스너(event listener)(1/3)

6

New [attributes added in HTML5]	
onbeforeonload	<i>before onload event</i>
oncanplay	<i>media can start play</i>
oncanplaythrough	<i>media can be played to end</i>
oncontextmenu	<i>context menu is triggered</i>
ondrag	<i>element is dragged</i>
ondragend	<i>at the end of drag operation</i>
ondragenter	<i>element dragged on drop target</i>
ondragleave	<i>element leaves valid drop target</i>
ondragover	<i>element is dragged over drop target</i>
ondragstart	<i>at the start of drag operation</i>
ondrop	<i>dragged element is being dropped</i>
ondurationchange	<i>length of media is changed</i>
onemptied	<i>media resource element becomes empty</i>
onended	<i>media has reached end</i>
onerror	<i>error occurs</i>
onformchange	<i>form changes</i>
onforminput	<i>form gets user input</i>
onhaschange	<i>document has changed</i>
oninput	<i>message is triggered</i>
oninvalid	<i>element gets user input</i>

HTML5 이벤트 리스너(event listener)(2/3)

7

New [attributes added in HTML5]

onloadeddata	media data is loaded	onscroll	element's scrollbar is being scrolled
onloadedmetadata	duration of media element is loaded	onseeked	element's seeking attribute is not true
onloadstart	browser starts to load media data	onseeking	element's seeking attribute is true
onmessage	element is invalid	onstalled	there is error in fetching media data
onmousewheel	mouse wheel is being rotated	onstorage	document loads
onpause	media data is paused	onsuspend	browser has stopped fetching media data
onplay	media data is going to start playing	ontimeupdate	media changes its playing position
onplaying	media data has started playing	onvolumechange	media changes volume, also when mute
onprogress	browser is fetching media data	onwaiting	media has stopped playing
onratechange	media data's playing rate has changed		

HTML5 이벤트 리스너(event listener)(3/3)

Existing [attributes in HTML4 & 5]	
onabort	<i>abort event</i>
onbeforeunload	<i>before unload event</i>
onclick	<i>mouse clicked event</i>
ondblclick	<i>mouse double-clicked event</i>
ongesturechange	<i>gesture change event</i>
ongestureend	<i>end of gesture event</i>
ongesturestart	<i>start of gesture event</i>
onkeydown	<i>key pressed down event</i>
onkeypress	<i>key pressed event</i>
onkeyup	<i>key released event</i>
onmousedown	<i>mouse button pressed down event</i>
onmousemove	<i>mouse moved event</i>

onmouseout	<i>mouse cursor exited event</i>
onmouseover	<i>mouse cursor entered event</i>
onmouseup	<i>mouse button released event</i>
onmove	<i>move event</i>
onpaste	<i>paste event</i>
onresize	<i>resize event</i>
ontouchcancel	<i>touch canceled event</i>
ontouchend	<i>touch ended event</i>
ontouchmove	<i>touch move event</i>
ontouchstart	<i>touch started event</i>
onreset	<i>form is reset</i>

이벤트 리스너 작성

9

□ 이벤트 리스너 작성하는 3가지 방법

1. HTML 태그 내에 이벤트 리스너 작성 이미 학습
2. DOM 객체의 **이벤트 리스너 프로퍼티**에 작성
3. DOM 객체의 **addEventListener() 메소드** 이용
4. (2, 3을) 익명 함수로 작성

이벤트 리스너 작성

1. HTML 태그 내에 이벤트 리스너 작성
2. DOM 객체의 이벤트 리스너 프로퍼티에 작성
3. DOM 객체의 addEventListener() 메소드 이용
4. (2, 3을) 익명 함수로 작성

1. HTML 태그 내에 이벤트 리스너 작성

□ HTML 태그의 이벤트 리스너 속성에 리스너 코드 작성

예) <p>태그에 마우스 올리면 orchid, 내리면 white로 배경색 변경

```
<p onmouseover="this.style.backgroundColor='orchid'"
  onmouseout="this.style.backgroundColor='white'">
  마우스 올리면 orchid 색으로 변경
</p>
```

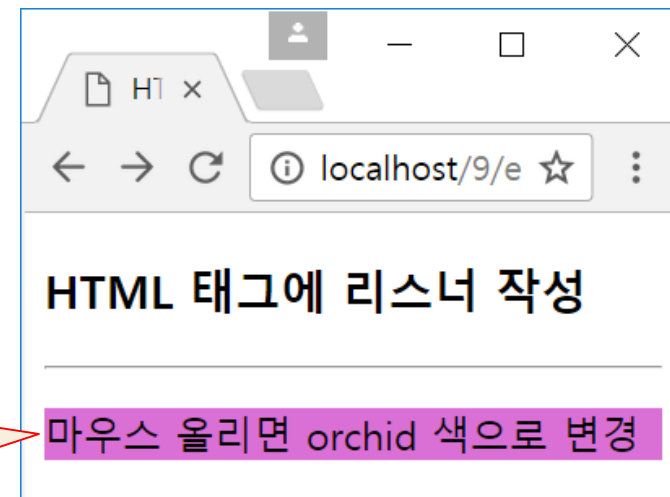
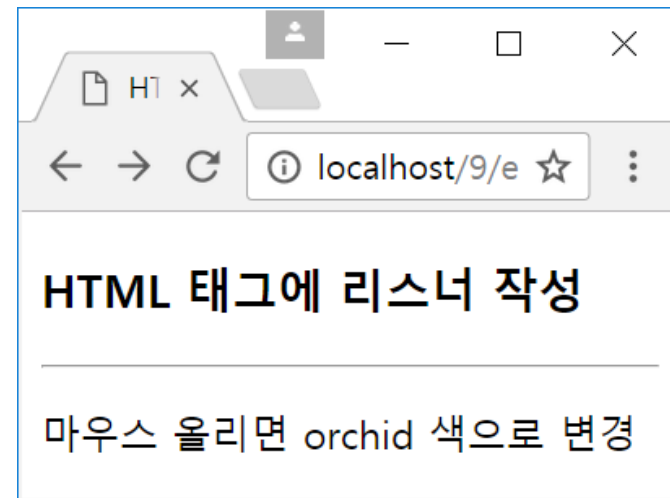
예제 9-1 HTML 태그 내에 이벤트 리스너 작성

10

```
<!DOCTYPE html>
<html>
<head>
<title>HTML 태그에 리스너 작성</title>
</head>
<body>
<p>HTML 태그에 리스너 작성</p>
<hr>
<p onmouseover="this.style.backgroundColor='orchid'"
  onmouseout="this.style.backgroundColor='white'">
  마우스 올리면 orchid 색으로 변경</p>
</body>
</html>
```

[실습-1] 마우스를 클릭하면(onclick) 글자색을 skyblue로 변경
[실습-2] 마우스를 더블클릭하면(ondblclick) 글자색을 green으로 변경

이곳에 마우스를 올리면
배경색 변함



2. DOM 객체의 이벤트 리스너 프로퍼티에 작성

11

□ DOM 객체의 이벤트 리스너 프로퍼티에 이벤트 리스너 코드 작성

적용할 DOM 객체를 찾은 후

- 이벤트 리스너 작성
- 1. HTML 태그 내에 이벤트 리스너 작성
 - 2. DOM 객체의 이벤트 리스너 프로퍼티에 작성
 - 3. DOM 객체의 addEventListener() 메소드 이용
 - 4. (2, 3을) 익명 함수로 작성

예) <p id="p">마우스 올리면 orchid 색으로 변경</p>

```
var p = document.getElementById("p");
p.onmouseover = over; // onmouseover 리스너 프로퍼티로 over() 함수 등록

function over() {
  ...
}
```

이벤트 리스너 프로퍼티

over(); // 잘못된 코드

뒤쪽 4. 익명 함수

함수 이름이 없음

```
var p = document.getElementById("p");
p.onmouseover = function () { // 마우스가 위에 올라가면 익명 함수 실행
  ...
};
```

cf) 예제 8-10 HTML 태그의 동적 추가 및 삭제

```
newDIV.onclick = function() { // 삽입한 태그를 클릭하면 삭제
  var p = this.parentElement; // 부모 HTML 태그 요소
  p.removeChild(this); // 자신을 부모로부터 제거
};
```

예제 9-2 DOM 객체의 이벤트 리스너 프로퍼티에 리스너 등록

12

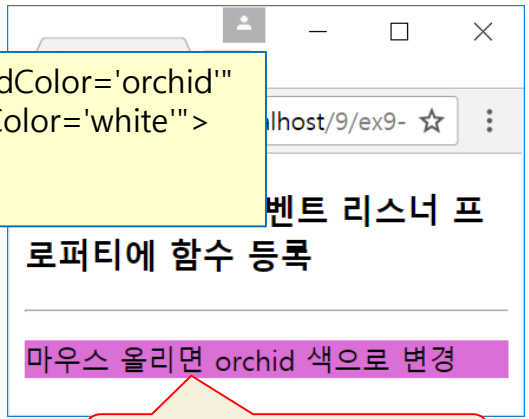
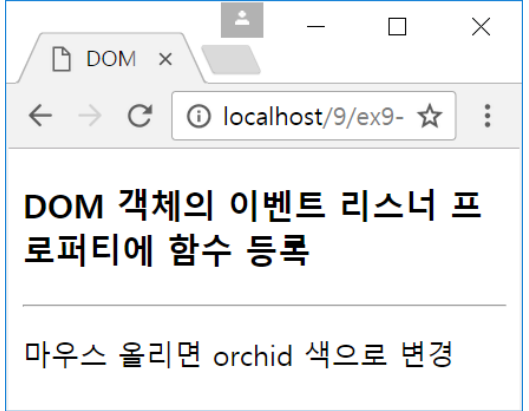
```
<!DOCTYPE html>
<html>
<head>
<title>DOM 객체의 이벤트 리스너 프로퍼티에 함수 등록</title>
<script>
var p;
function init() {
    p = document.getElementById("p");
    p.onmouseover = over;
    p.onmouseout = out;
}
function over() {
    p.style.backgroundColor="orchid";
}
function out() {
    p.style.backgroundColor="white";
}
</script>
</head>
<body onload="init()" >
<h3>DOM 객체의 이벤트 리스너 프로퍼티에 함수 등록</h3>
<hr>
<p id="p">마우스 올리면 orchid 색으로 변경</p>
</body> </html>
```

전역변수, 이유?

문서 로딩이 완료되면
init() 함수 실행

<p onmouseover="this.style.backgroundColor='orchid'"
onmouseout="this.style.backgroundColor='white'">
마우스 올리면 orchid 색으로 변경
</p>

[실습] 마우스를 클릭하면(onclick) 글자색을 skyblue로
변경, 더블클릭하면(ondblclick) 글자색을 green으로 변
경하는 루틴 추가



이곳에 마우스를 올리거나/
벗어나면 배경색 변함

3. DOM 객체의 addEventListener() 메소드 활용

13

□ addEventListener() 메소드

이벤트 리스너 작성

1. HTML 태그 내에 이벤트 리스너 작성
2. DOM 객체의 이벤트 리스너 프로퍼티에 작성
3. DOM 객체의 **addEventListener()** 메소드 이용
4. (2, 3을) 익명 함수로 작성

addEventListener(eventName, listener[, useCapture])

- eventName : 이벤트 타입을 나타내는 문자열. click, load, keydown 등
- listener : 이벤트 리스너로 등록할 함수 이름
- useCapture : true이면 이벤트 흐름 중 캡처 단계에서 실행될 리스너(listener 함수) 등록.
false이면 버블 단계에서 실행될 리스너 등록. 생략 가능하며 디폴트는 false.
이벤트 흐름은 3절에서 자세히 설명

listener 함수를 eventName의 이벤트를 처리할 리스너로 등록한다.

예)

```
p.addEventListener("mouseover", over); // onmouseover 리스너로 over 함수 등록
```

onmouseover = **on** + 이벤트 이름

예제 9-3 addEventListener() 사용

14

```

<!DOCTYPE html>
<html>
<head>
<title>addEventListener()를 이용한 리스너 등록</title>
<script>
var p;
function init() {      // 문서가 완전히 로드되었을 때 호출
    p = document.getElementById("p");
    p.addEventListener("mouseover", over); // 이벤트 리스너 over 등록
    p.addEventListener("mouseout", out);   // 이벤트 리스너 out 등록
}
function over() {
    p.style.backgroundColor="orchid";
}
function out() {
    p.style.backgroundColor="white";
}
</script>
</head>
<body onload="init()">
<h3>addEventListener()를 이용한 리스너 등록</h3>
<hr>
<p id="p">마우스 올리면 orchid 색으로 변경</p>
</body></html>

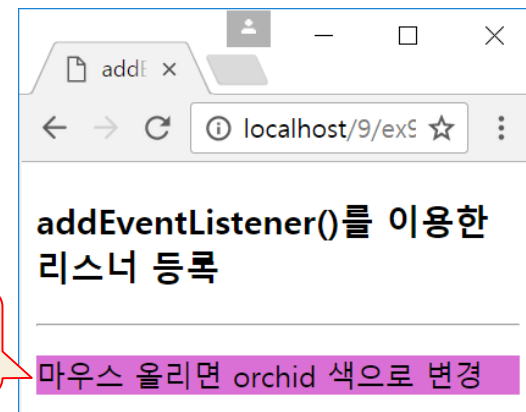
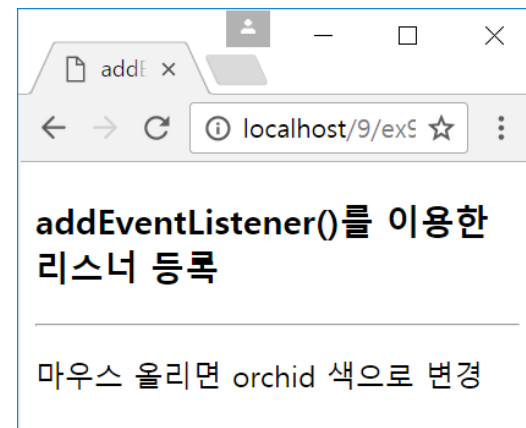
```

cf) 2번째 방법

p.onmouseover = over;
p.onmouseout = out;

이곳에 마우스를 올리면
배경색 변함

[실습] 마우스를 클릭하면(onclick) 글자색을 skyblue로 변경, 더블클릭하면(ondblclick) 글자색을 green으로 변경하는 루틴 추가



4. 익명 함수로 이벤트 리스너 작성

15

- 익명 함수(anonymous function)
 - ▣ 함수 이름 없이 필요한 곳에 함수의 코드를 바로 작성
 - ▣ 코드가 짧거나 오직 한 곳에서만 사용하는 경우, 익명 함수 편리

2. 이벤트 리스너
프로퍼티

```
p = document.getElementById("p");  
p.onmouseover = over; // over()를 onmouseover 리스너로 등록  
function over() {  
    p.style.backgroundColor="orchid";  
}
```

1) 그대로 대입, 2) 함수 이름 삭제,
3) DOM 객체(p)를 this로 변경



4. 익명 함수

```
p.onmouseover = function () {  
    this.style.backgroundColor = "orchid";  
};
```

함수이기 때문에 () 는 반드시

3. addEventListener()
메소드

```
p = document.getElementById("p");  
p.addEventListener("mouseover", over); // 이벤트 리스너 over 등록  
function over() {  
    p.style.backgroundColor="orchid";  
}
```

1) 그대로 대입, 2) 함수 이름 삭제,
3) DOM 객체(p)를 this로 변경



4. 익명 함수

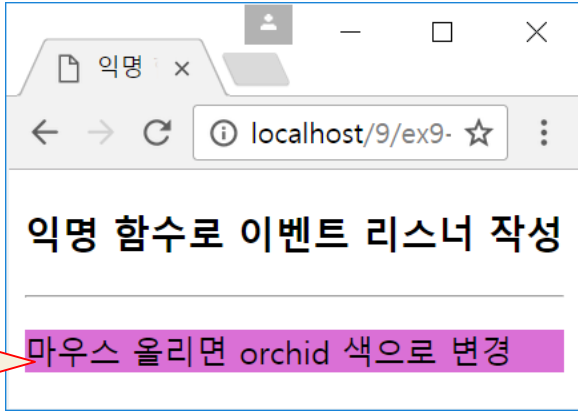
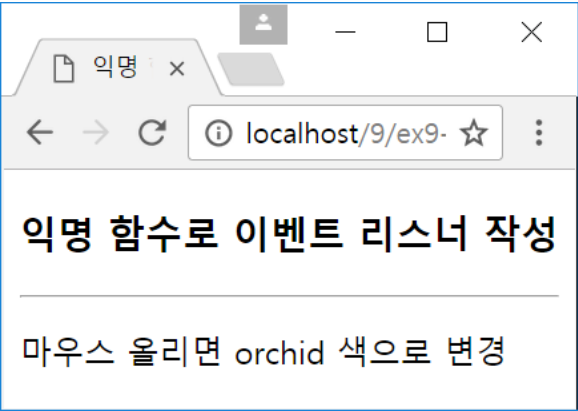
```
p.addEventListener("mouseover",  
    function () { this.style.backgroundColor = "orchid"; }; // 편의상 다음 줄에  
);
```

() 는 반드시

예제 9-4 익명 함수로 이벤트 리스너 작성

16

```
<!DOCTYPE html>
<html>
<head>
<title> 익명 함수로 이벤트 리스너 작성 </title>
<script>
var p; // 지역변수로 이동 가능
function init() { // 문서가 완전히 로드되었을 때 호출
p = document.getElementById("p");
p.onmouseover = function () { // 2→4. 익명 함수
this.style.backgroundColor = "orchid";
};
p.addEventListener("mouseout",
function () { this.style.backgroundColor="white"; } // 3→5. 익명 함수
);
}
</script>
</head>
<body onload="init()">
<h3> 익명 함수로 이벤트 리스너 작성 </h3>
<hr>
<p id="p">마우스 올리면 orchid 색으로 변경</p>
</body>
</html>
```



이곳에 마우스를 올리면 배경색 변함

[실습-1] 마우스를 클릭하면(onclick) 글자색을 skyblue로 변경, 더블클릭하면(ondbclick) 글자색을 green으로 변경 하는 루틴 추가
[실습-2] 예제 9-3의 over(), out() 함수를 익명 함수로 변경

- 이벤트 리스너 작성
- 1. HTML 태그 내에 이벤트 리스너 작성
 - 2. DOM 객체의 이벤트 리스너 프로퍼티에 작성
 - 3. DOM 객체의 addEventListener() 메소드 이용
 - 4. (2, 3을) 익명 함수로 작성

이벤트 리스너 작성 방법 4가지 비교

17

(1) HTML 태그 내

```
<p id="p" onmouseover="this.style.backgroundColor='orchid'"
  마우스 올리면 orchid 색으로 변경
</p>
```

(2) 이벤트 리스너
프로퍼티

```
p.onmouseover = over;
function over() {
  p.style.backgroundColor="orchid";
}
```

(3) addEventListener()
메소드 이용

```
p.addEventListener("mouseover", over);
function over() {
  p.style.backgroundColor="orchid";
}
```

(4) 익명 함수 이용

```
p.onmouseover = function () {
  this.style.backgroundColor="orchid";
};
```

(4) 익명 함수 이용

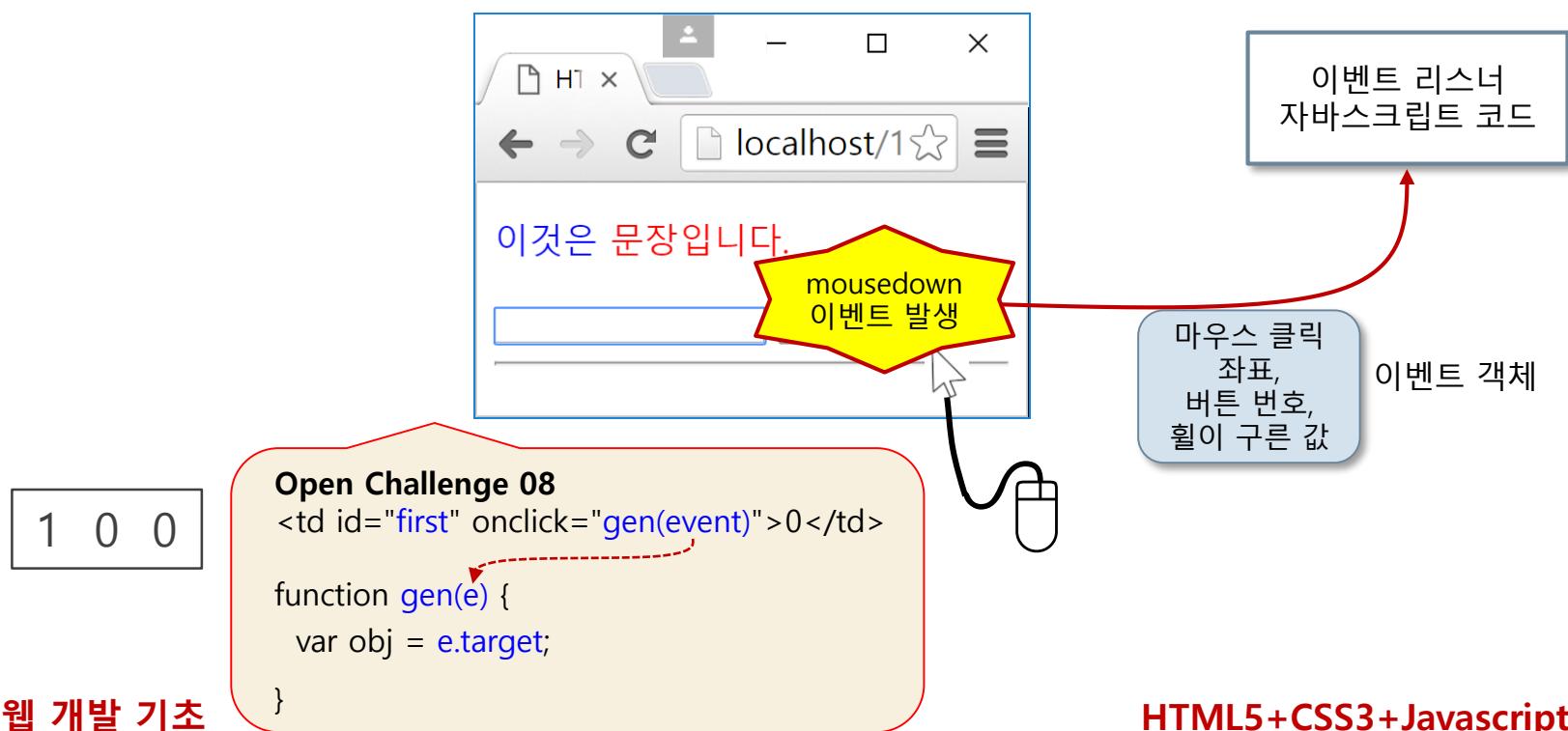
```
p.addEventListener("mouseover",
  function () { this.style.backgroundColor="orchid"; }
);
```

2. 이벤트 객체

이벤트 객체

18

- 이벤트 객체(event object)
 - 발생한 이벤트에 관련된 다양한 정보를 담은 객체
 - 예) mousedown 이벤트의 경우, 마우스 좌표와 버튼 번호 등
 - keydown 이벤트의 경우, 키 코드 값 등
 - 이벤트가 처리되고 나면 이벤트 객체 소멸



이벤트 객체 전달받기

19

- 이벤트 객체는 이벤트 리스너 함수의 **첫 번째 매개변수**로 전달

1. 이름을 가진 이벤트 리스너

```
function f(e) {           // 매개변수 e에 이벤트 객체 전달받음
    ...
}
obj.onclick = f;          // obj 객체의 onclick 리스너로 함수 f 등록
```

2. 익명 함수의 경우

```
obj.onclick = function(e) { // 매개변수 e에 이벤트 객체 전달받음
    ...
}
```

3. HTML 태그에 이벤트 리스너 : **event** 라는 이름으로 전달

```
function f(e) {
    ...
}
...
<button onclick="f(event)">버튼</button>
<div onclick="alert(event.type)">버튼</div>
```

event 라는 이름으로 이벤트 객체 전달받음

예제 9-5 이벤트 리스너에서 이벤트 객체 전달 받기

20

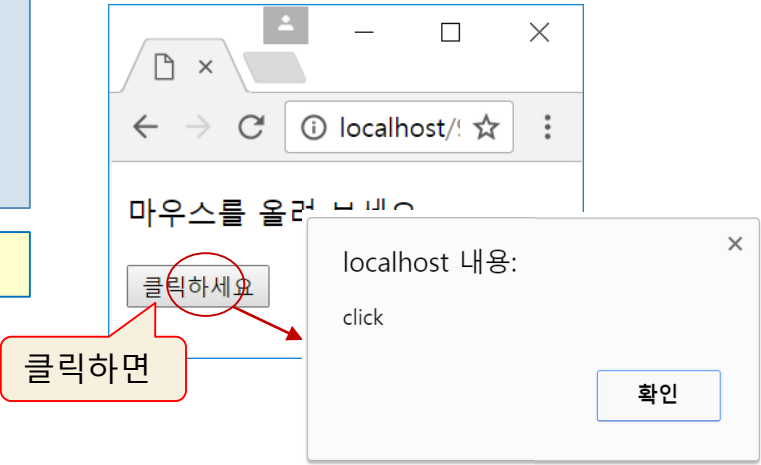
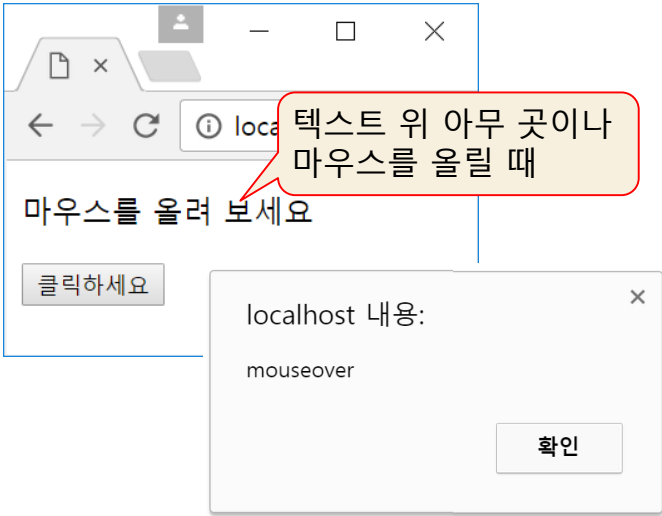
```
<!DOCTYPE html>
<html>
<head>
<title>이벤트 객체 전달받기</title>
</head>
<body>
<p id="p">마우스를 올려 보세요</p>
<button onclick="f(event)">클릭하세요</button>
<script>
function f(e) { // e는 현재 발생한 이벤트 객체
  alert(e.type); // 이벤트 종류 출력
}
document.getElementById("p").onmouseover = f;
</script>
</body>
</html>
```

chain rule

var p = document.getElementById("p");
p.onmouseover = f;

[실습] 함수 f()를 익명 함수로 변경 가능할까?

- 이벤트 리스너 작성
- 1. HTML 태그 내에 이벤트 리스너 작성
 - 2. DOM 객체의 이벤트 리스너 프로퍼티에 작성
 - 3. DOM 객체의 addEventListener() 메소드 이용
 - 4. (2, 3을) 익명 함수로 작성



이벤트 객체에 들어 있는 정보

- 현재 발생한 이벤트에 관한 다양한 정보
 - ▣ 이벤트 객체의 프로퍼티와 메소드로 알 수 있음
- 이벤트의 종류마다 조금씩 다름
 - ▣ 이벤트 객체의 공통 멤버

prevent :
금지/취소하다

멤버	종류	설명
type	프로퍼티	현재 발생한 이벤트의 종류를 나타내는 문자열(click, load 등)
target	프로퍼티	이벤트를 발생시킨 객체(DOM 객체 혹은 HTML 태그)
currentTarget	프로퍼티	현재 이벤트 리스너를 실행하고 있는 DOM 객체
defaultPrevented	프로퍼티	이벤트의 디폴트 행동이 취소되었는지를 나타내는 true/false 값
preventDefault()	메소드	이벤트의 디폴트 행동을 취소시키는 메소드

- target 프로퍼티
 - ▣ 이벤트 타겟 객체 가리킴
 - ▣ 이벤트 타겟 : 이벤트를 유발시킨 DOM 객체
- ex) <button> 태그의 버튼을 click하면, click 이벤트의 이벤트 타겟은 버튼

예제 9-6 이벤트 객체의 프로퍼티 출력

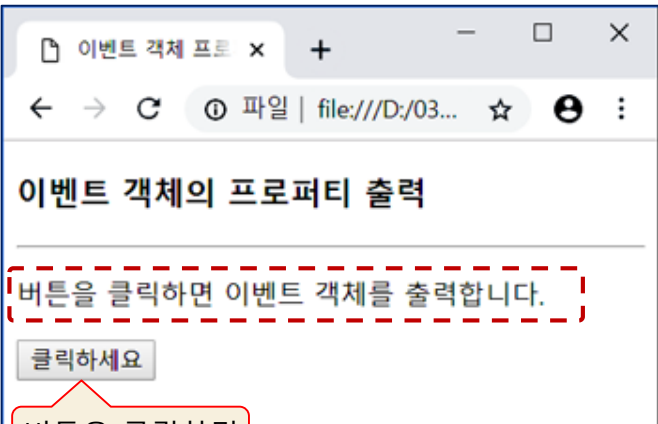
22

```
<!DOCTYPE html>
<html>
<head><title>이벤트 객체 프로퍼티</title>
</head>
<body>
<h3>이벤트 객체의 프로퍼티 출력</h3>
<hr>
<p id="p">버튼을 클릭하면 이벤트 객체를 출력합니다.</p>
<button onclick="f(event)">클릭하세요</button>
<script>
function f(e) {           // e는 현재 발생한 이벤트 객체
  var text = "type: " + e.type + "<br>"
               + "target: " + e.target + "<br>"
               + "currentTarget: " + e.currentTarget + "<br>"
               + "defaultPrevented: " + e.defaultPrevented;

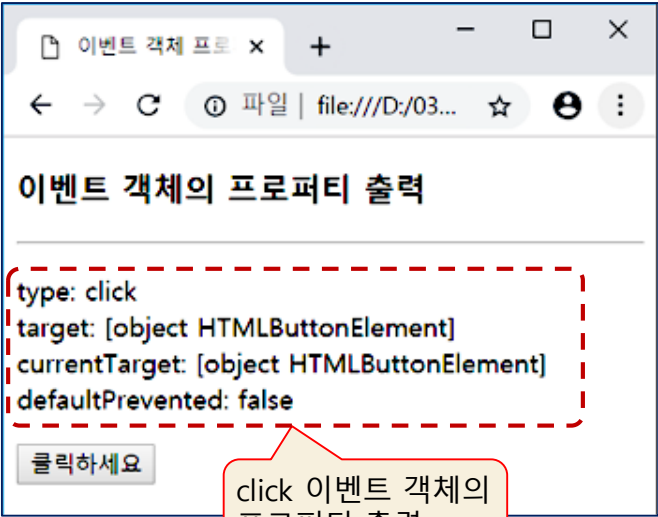
  var p = document.getElementById("p");
  p.innerHTML = text;      // 이벤트 객체의 프로퍼티 출력
}
</script>
</body>
</html>
```

HTML, cf) "\n"

\$("#p").html = text;



버튼을 클릭하면



click 이벤트 객체의 프로퍼티 출력

[실습] 버튼에 마우스를 올리는 경우(onmouseover)와 마우스가 벗어나는 경우(onmouseout) 를 추가

이벤트의 디폴트 행동 취소, preventDefault()

23

- 이벤트의 디폴트 행동이란?
 - ▣ 특정 이벤트에 대한 HTML 태그의 기본 행동
 - ▣ 사례
 - <a>의 click 이벤트의 디폴트 행동 : 웹 페이지 이동
 - submit 버튼의 click 이벤트의 디폴트 행동 : 폼 데이터 전송
 - <input type="checkbox">의 click 이벤트의 디폴트 행동 : 체크박스 선택

- 이벤트의 디폴트 행동을 막는 방법

- 1. 이벤트 리스너에서 false 리턴

```
<a href="http://www.naver.com" onclick="return false">  
    이동 안 되는 링크  
</a>
```

- 2. 이벤트 객체의 preventDefault() 메소드 호출

```
<a href="http://www.naver.com" onclick="event.preventDefault();">  
    이동 안 되는 링크  
</a>
```

- 이벤트 객체의 cancelable 프로퍼티가 true인 경우만 취소 가능

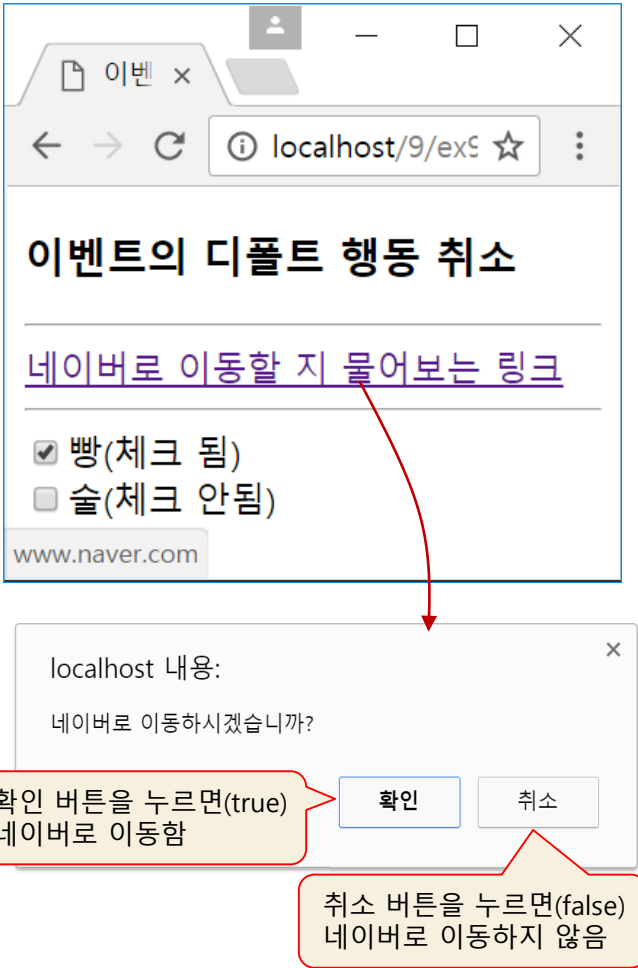
예제 9-7 이벤트의 디폴트 행동 취소

24

```
<!DOCTYPE html>
<html> <head> <title>이벤트의 디폴트 행동 취소</title>
<script>
function query() {
  var ret = confirm("네이버로 이동하시겠습니까?");
  return ret;           // confirm()의 리턴 값은 true 또는 false
}
function noAction(e) {
  e.preventDefault();  // 이벤트의 디폴트 행동 강제 취소
}
</script>
</head>
<body>
<h3>이벤트의 디폴트 행동 취소</h3>
<hr>
<a href="http://www.naver.com" onclick="return query()">네이버로
이동할 지 물어보는 링크</a>
<hr>
<form>
  <input type="checkbox">빵(체크 됨)<br>
  <input type="checkbox" onclick="noAction(event)">술(체크 안됨)
</form>
</body>
</html>
```

true or false

"event.preventDefault()"



[실습] 체크박스를 라디오 버튼으로 바꾸기

이벤트 흐름

25

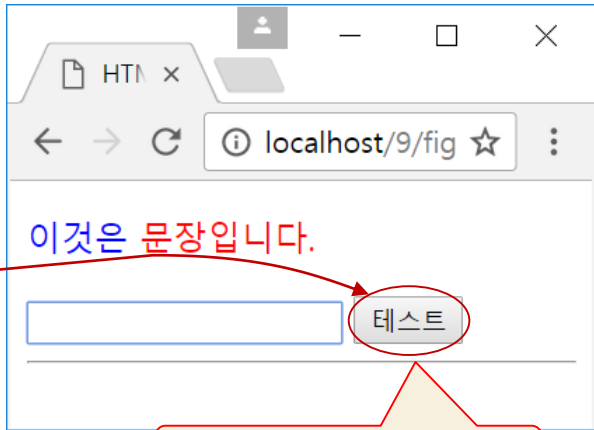
- 이벤트 흐름이란?
 - 이벤트가 발생하면 window 객체에 먼저 도달하고, DOM 트리를 따라 이벤트 타겟에 도착하고, 다시 반대 방향으로 흘러 window 객체에 도달한 다음 사라지는 과정
- 이벤트가 흘러가는 과정
 - 캡처 단계(capturing phase)
 - 이벤트가 window 객체에서 중간의 모든 DOM 객체를 거쳐 타겟 객체에 전달되는 과정
 - 이벤트가 거쳐가는 모든 DOM 객체(window 포함)의 이벤트 리스너 실행
 - 버블 단계(bubbling phase)
 - 이벤트가 타겟에서 중간의 모든 DOM 객체를 거쳐 window 객체에 전달되는 과정
 - 이벤트가 거쳐가는 모든 DOM 객체(window 포함)의 이벤트 리스너 실행
- DOM 객체에는 캡처 리스너와 버블 리스너 두 개 모두 작성할 수 있음

이벤트 흐름 사례

26

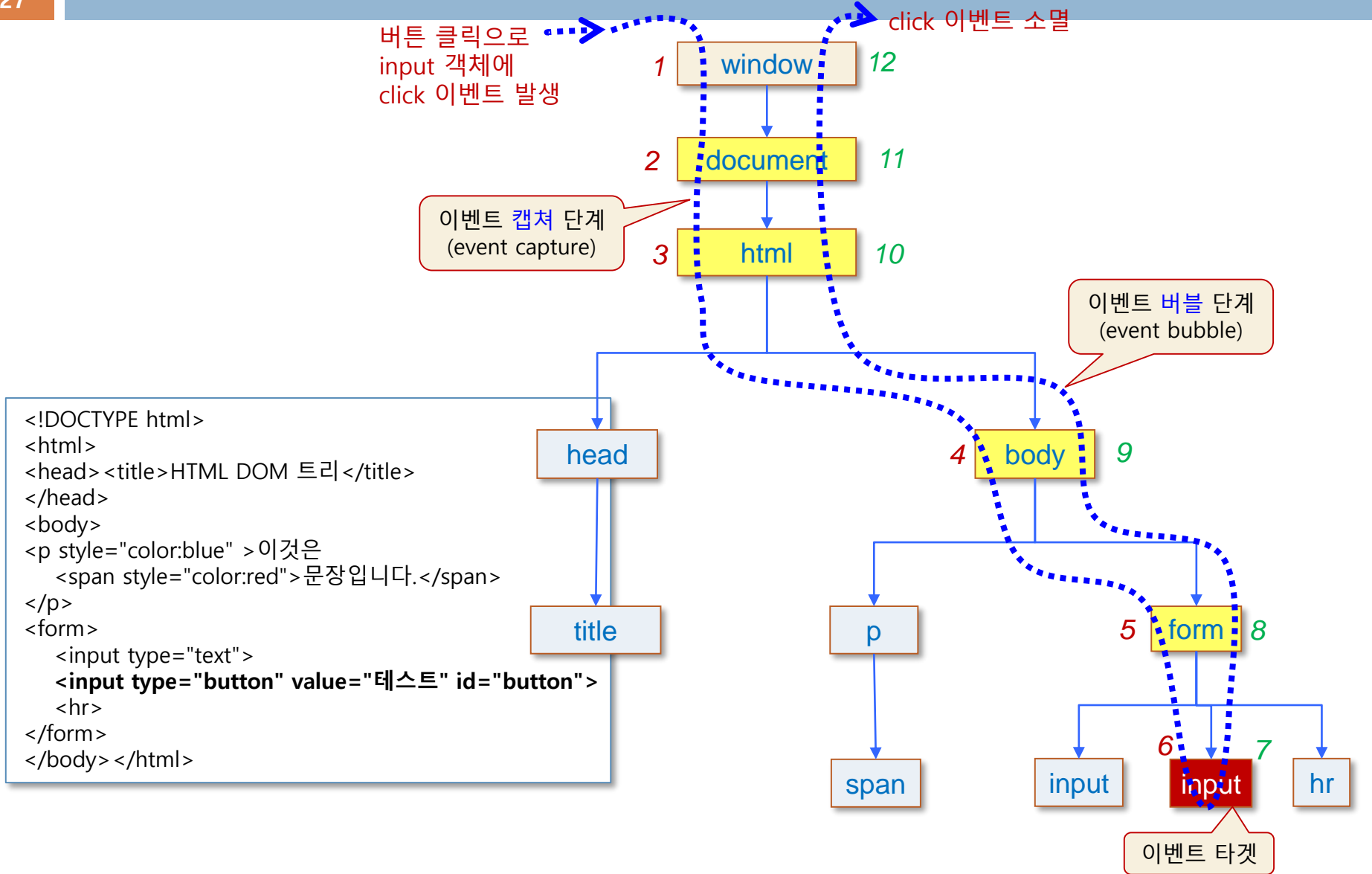
□ 샘플 웹 페이지

```
<!DOCTYPE html>
<html>
<head> <title>HTML DOM 트리</title>
</head>
<body>
<p style="color:blue" >이것은
  <span style="color:red">문장입니다.</span>
</p>
<form>
  <input type="text">
  <input type="button" value="테스트" id="button">
  <hr>
</form>
</body>
</html>
```



버튼 클릭, click 이벤트 발생

DOM 트리에서 이벤트 흐름



캡처 리스너와 버블 리스너

skip

28

- DOM 객체의 이벤트 리스너
 - ▣ 캡처 리스너와 버블 리스너를 모두 소유 가능
 - 이벤트 리스너 등록 시, 캡처 리스너인지 버블 리스너인지 구분
- 캡처 리스너와 버블 리스너 등록
 - ▣ addEventListener()의 3 번째 매개 변수 이용
 - true면 캡처 리스너, false면 버블 리스너

```
var b = document.getElementById("button");  
b.addEventListener("click", capFunc, true);    // 캡처 단계에서 capFunc() 실행  
b.addEventListener("click", bubbleFunc, false); // 버블 단계에서 bubbleFunc() 실행
```

- ▣ 다른 방법의 이벤트 리스너 등록의 경우
 - 버블 리스너로 자동 등록

예)

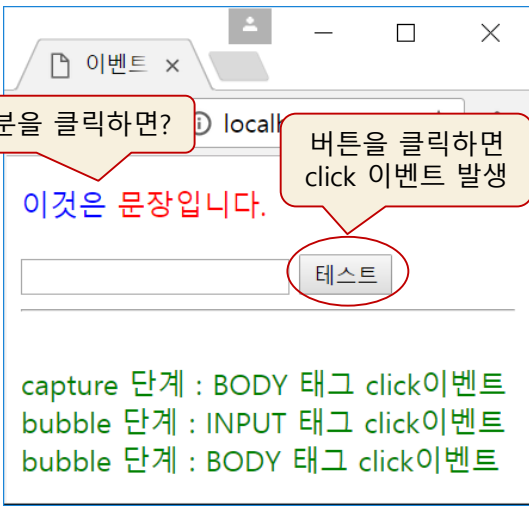
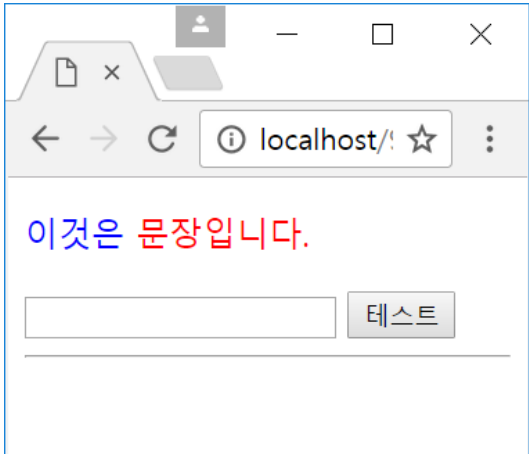
```
obj.onclick = function(e) {    // 버블 리스너도 작동  
    ...  
}
```


예제 9-8 이벤트 흐름

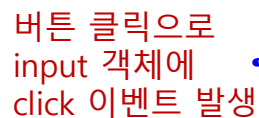
skip

29

```
<!DOCTYPE html>
<html> <head> <title>이벤트 흐름</title> </head>
<body>
<p style="color:blue">이것은 <span style="color:red" id="span">문장입니다.</span></p>
<form>
  <input type="text" name="s">
  <input type="button" value="테스트" id="button">
  <hr>
</form>
<div id="div" style="color:green"></div>
<script>
var div = document.getElementById("div"); // 이벤트 메시지 출력 공간
var button = document.getElementById("button");
// body 객체에 캡처 리스너 등록
document.body.addEventListener("click", capture, true); // 캡처 단계(1)
// 타겟 객체에 버블 리스너 등록
button.addEventListener("click", bubble, false); // 버블 단계(2)
// body 객체에 버블 리스너 등록
document.body.addEventListener("click", bubble, false); // 버블 단계(3)
function capture(e) { // e는 이벤트 객체
  var obj = e.currentTarget; // 현재 이벤트를 받은 DOM 객체
  var tagName = obj.tagName; // 태그 이름
  div.innerHTML += "<br>capture 단계 : " + tagName + " 태그 " + e.type + "이벤트";
}
function bubble(e) { // e는 이벤트 객체
  var obj = e.currentTarget; // 현재 이벤트를 받은 DOM 객체
  var tagName = obj.tagName; // 태그 이름
  div.innerHTML += "<br>bubble 단계 : " + tagName + " 태그 " + e.type + "이벤트";
}
</script>
</body></html>
```

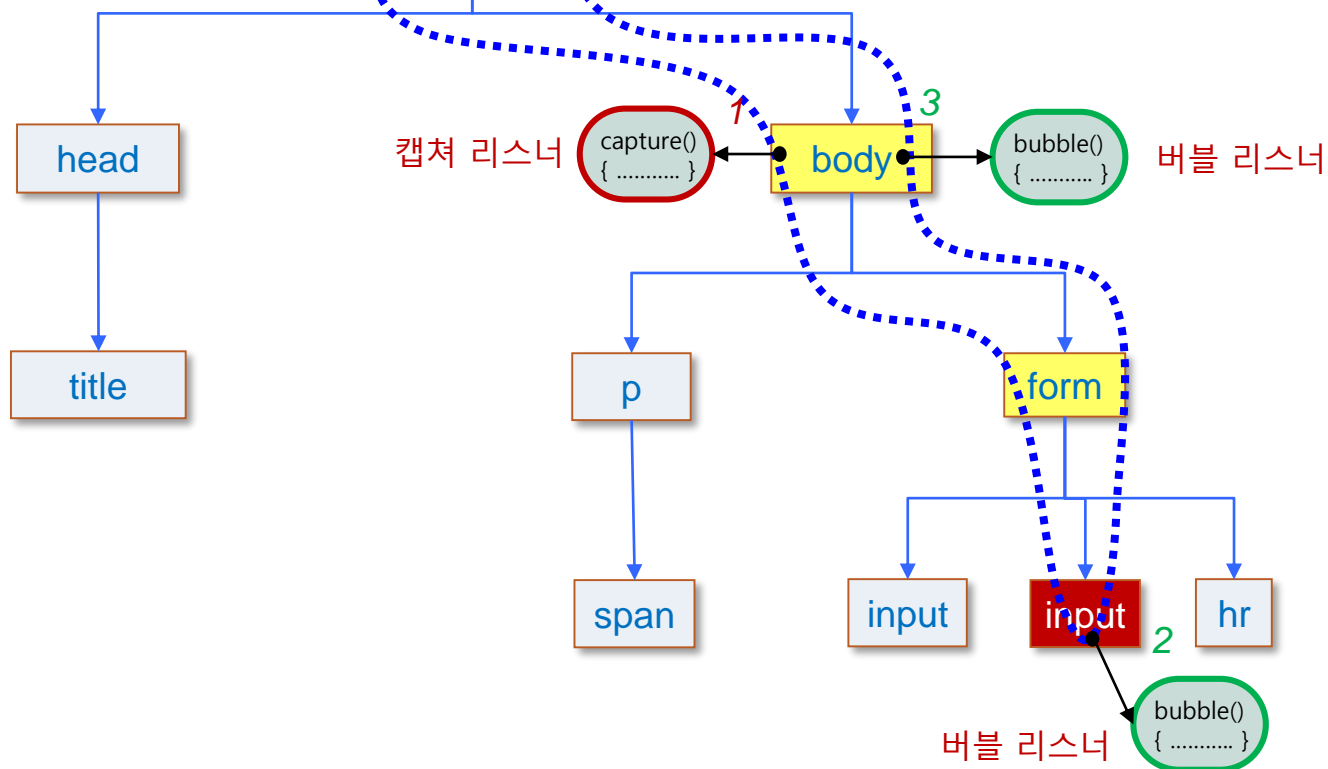


1. `<body>` 태그의 캡처 리스너 실행
2. `<input>` 태그의 버블 리스너 실행
3. `<body>` 태그의 버블 리스너 실행



click 이벤트 소멸

skip



이벤트 흐름 중단시키기

31

- 이벤트 객체의 stopPropagation() 호출
 - ▣ event.stopPropagation(); // event가 이벤트 객체일 때

마우스 핸들링

32

마우스 이벤트 객체의 프로퍼티

기준으로부터
상대 좌표
→ 37쪽 예시

프로퍼티	기준(0,0) +
x, y	(x, y)는 타겟 객체의 부모 객체 내에서의 마우스 좌표
screenX, screenY	(screenX, screenY)는 스크린을 기준으로 한 마우스 좌표
clientX, clientY	(clientX, clientY)는 브라우저 윈도우의 문서출력 영역 내에서의 마우스의 좌표
offsetX, offsetY	(offsetX, offsetY)는 타겟 객체 내에서의 마우스 좌표
button	눌려진 마우스 버튼 <ul style="list-style-type: none">• 0 : 아무 버튼도 눌러지지 않았음• 1 : 왼쪽 버튼이 눌러졌음• 2 : 오른쪽 버튼이 눌러졌음• 3 : 왼쪽, 오른쪽 버튼이 모두 눌러졌음• 4 : 중간 버튼이 눌러졌음
wheelDelta	마우스 휠이 구른 방향 <ul style="list-style-type: none">• 양수 : 위쪽으로 굴린 경우(실제 wheelDelta 값은 120)• 음수 : 아래쪽으로 굴린 경우(실제 wheelDelta 값은 -120)

- onclick
 - HTML 태그가 클릭될 때
- ondblclick
 - HTML 태그가 더블클릭될 때

★

예제 9-9 onclick 리스너로 계산기 만들기

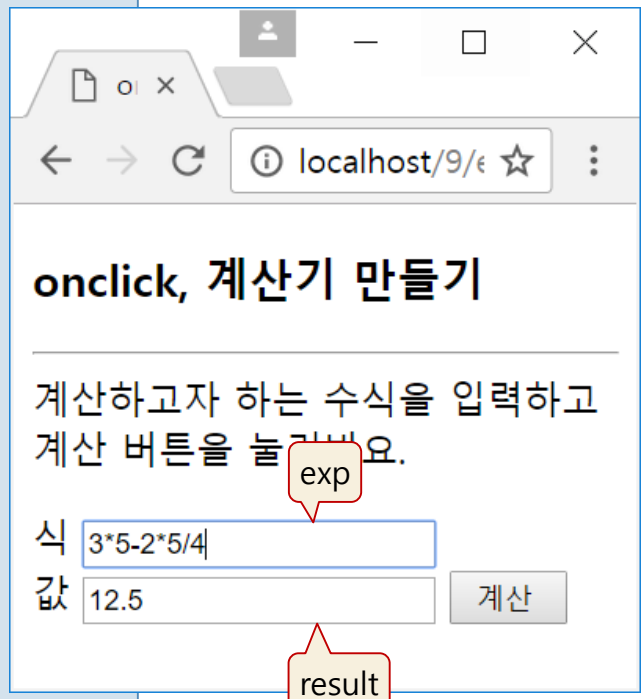
33

```
<!DOCTYPE html>
<html>
<head> <title>onclick</title>
<script>
function calculate() {
    var exp = document.getElementById("exp");
    var result = document.getElementById("result");
    result.value = eval(exp.value);
}
</script>
</head>
<body >
<h3>onclick, 계산기 만들기</h3>
<hr>
계산하고자 하는 수식을 입력하고 계산 버튼을 눌러봐요.
<br><br>
<form>
<label>식 <input type="text" id="exp" value=""> </label> <br>
<label>값 <input type="text" id="result"> </label>
<input type="button" value=" 계산 " onclick="calculate()" >
</form>
</body>
</html>
```

값

exp

result



[실습] [계산] 버튼을 누르면 prompt()를 이용하여 식을 입력받는 방식으로 수정

여러 마우스 관련 이벤트 리스너

34

▣ 마우스 관련 이벤트 리스너 호출 경우

- onmousedown : HTML 태그에 마우스 버튼을 누르는 순간
- onmouseup : 눌려진 버튼이 놓여지는 순간
- onmouseover : 마우스가 HTML 태그 위로 올라오는 순간. 자식 영역 포함
- onmouseout : 마우스가 HTML 태그를 벗어나는 순간. 자식 영역 포함
- onmouseenter : 마우스가 HTML 태그 위로 올라오는 순간. 버블 단계 없음
- onmouseleave : 마우스가 HTML 태그를 벗어나는 순간. 버블 단계 없음
- onwheel : HTML 태그에 마우스 휠이 구르는 동안 계속 호출
 - 위쪽으로 굴린 경우 : wheelDelta 프로퍼티 값 양수(120)
 - 아래쪽으로 굴린 경우 : wheelDelta 프로퍼티 값 양수(-120)

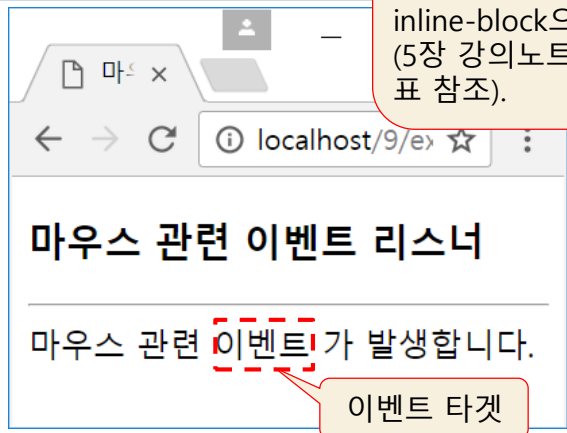
```
obj.onwheel = function (e) {  
    if(e.wheelDelta < 0) {           // 아래쪽으로 휠을 굴린 경우  
        ...  
    }  
    else {                           // 위쪽으로 휠을 굴린 경우  
        ...  
    }  
};
```

예제 9-10 마우스 관련 이벤트 리스너

```
<!DOCTYPE html>
<html>
<head> <title>마우스 관련 리스너</title>
<script>
var width=1;           // 테두리 두께
function down(obj) {
  obj.style.fontStyle = "italic";
}
function up(obj) {
  obj.style.fontStyle = "normal";
}
function over(obj) {
  obj.style.borderColor = "violet";
  // 테두리 폭이 0일 때 색은 보이지 않는다.
}
function out(obj) {
  obj.style.borderColor = "lightgray";
}
function wheel(e, obj) {    // e는 이벤트 객체
  if(e.wheelDelta < 0) {    // 휠을 아래로 굴릴 때
    width--;               // 폭 1 감소
    if(width < 0) width = 0; // 폭이 0보다 작아지지 않게
  } else                   // 휠을 위로 굴릴 때
    width++;               // 폭 1 증가
  obj.style.borderStyle = "ridge";
  obj.style.borderWidth = width+"px";
}
</script>
</head>
```

```
<body >
<h3>마우스 관련 이벤트 리스너</h3>
<hr>
<div>마우스 관련
  <span onmousedown="down(this)"
        onmouseup="up(this)"
        onmouseover="over(this)"
        onmouseout="out(this)"
        onwheel="wheel(event, this)"
        style="display: inline-block">이벤트
  </span>가 발생합니다.
</div>
</body>
</html>
```

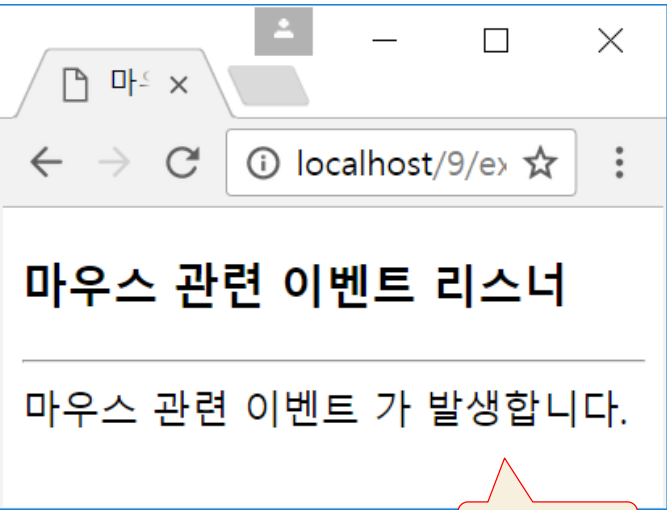

 태그는 디폴트로 inline 박스이므로 border 조절이 불가능하기 때문에 inline-block으로 변경 (5장 강의노트 6쪽 아래쪽 표 참조).



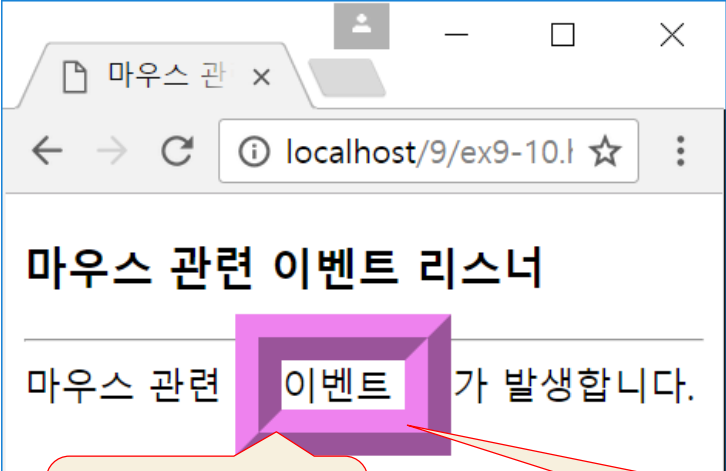
초기 화면

이벤트 타겟

- [실습-1] 마우스가 올라가면 글자색을 blue로 바꾸도록 수정
- [실습-2] '마우스'에도 동일한 이벤트 리스너 등록하기
- [실습-3] span 태그에 inline-block 속성을 부여하는 방식을 style 태그 방식으로 변경

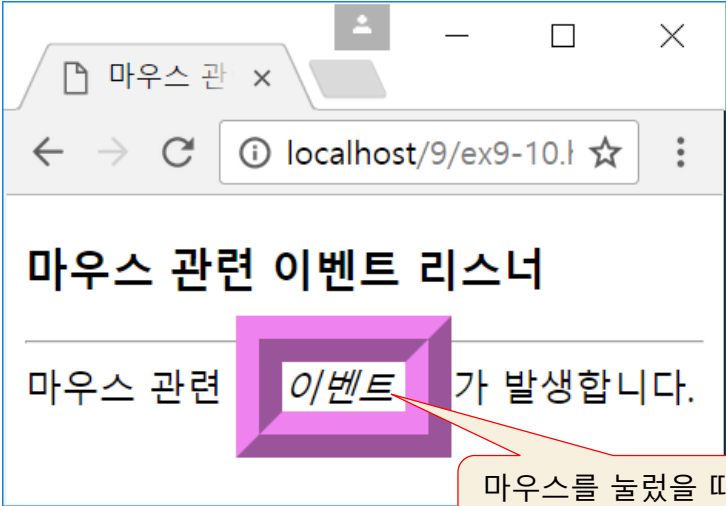


초기 화면

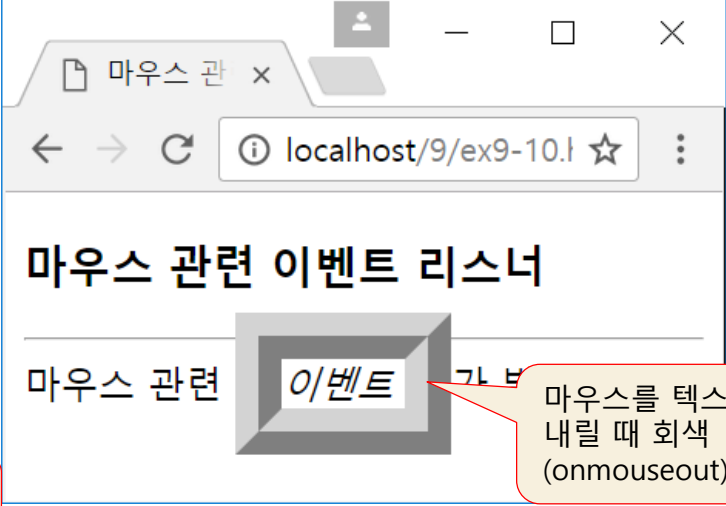


마우스를 텍스트 위로 올릴 때 violet 색 (onmouseover)

마우스를 위로 굴릴 때 두께가 1px씩 두꺼워짐(onwheel)



마우스를 눌렀을 때 Italic체 (onmousedown)



마우스를 텍스트에서 내릴 때 회색 (onmouseout)

예제 9-11 onmousemove와 마우스 위치 및 버튼

37

```
<!DOCTYPE html>
<html> <head> <title>마우스 이벤트 객체의 프로퍼티</title>
<style>
div { background : skyblue; width : 250px; }
</style></head>
<body>
<h3>마우스 이벤트 객체의 프로퍼티와 onmousemove</h3>
<hr>
이미지 위에 마우스를 움직일 때 onmousemove 리스너가 실행되고,
마우스의 위치를 보여줍니다.<br><br>

<br><br>
<div id="div"></div>
<script>
var div = document.getElementById("div");
function where(e) {
  var text = "버튼=" + e.button + "<br>";
  text += "(screenX, screenY)=" + e.screenX + "," + e.screenY + "<br>";
  text += "(clientX, clientY)=" + e.clientX + "," + e.clientY + "<br>";
  text += "(offsetX, offsetY)=" + e.offsetX + "," + e.offsetY + "<br>";
  text += "(x, y)=" + e.x + "," + e.y + "<br>";
  div.innerHTML = text;
}
</script>
</body> </html>
```

- [실습-1] onmousemove를 onmousedown으로 변경
- [실습-2] onmousemove를 img 태그에서 body 태그로 이동

모니터 (0,0)
(screenX, screenY)

408

478

브라우저 (0,0)
(clientX,clientY)

96

202

이미지 (0,0)

(88, 46)

마우스

버튼=0
(screenX, screenY)=408,478
(clientX, clientY)=96,202
(offsetX, offsetY)=88,46
(x, y)=96,202

두 좌표가 같은 이유는 객체의 부모가 <body>로써, 브라우저 윈도우이기 때문

브라우저 위치를 변경하고 실행
예) 모니터 좌측 상단에 모서리 맞추기

이미지 위에 마우스를 움직일 때 onmousemove 리스너가 실행되고, 마우스의 위치를 보여줍니다.

oncontextmenu

마우스 우클릭 금지되어
있는 웹페이지

38

- HTML 태그 위에서 **마우스 오른쪽 버튼** 클릭
 - ▣ 디폴트로 컨텍스트 메뉴(context menu) 출력
 - ‘소스 보기’나 ‘이미지 다운로드’ 등의 메뉴
 - ▣ oncontextmenu 리스너가 먼저 호출
 - false를 리턴하면 컨텍스트 메뉴를 출력하는 디폴트 행동 취소

```
document.oncontextmenu = function () {  
    ...  
    return false;           // 컨텍스트 메뉴 출력 금지  
}
```

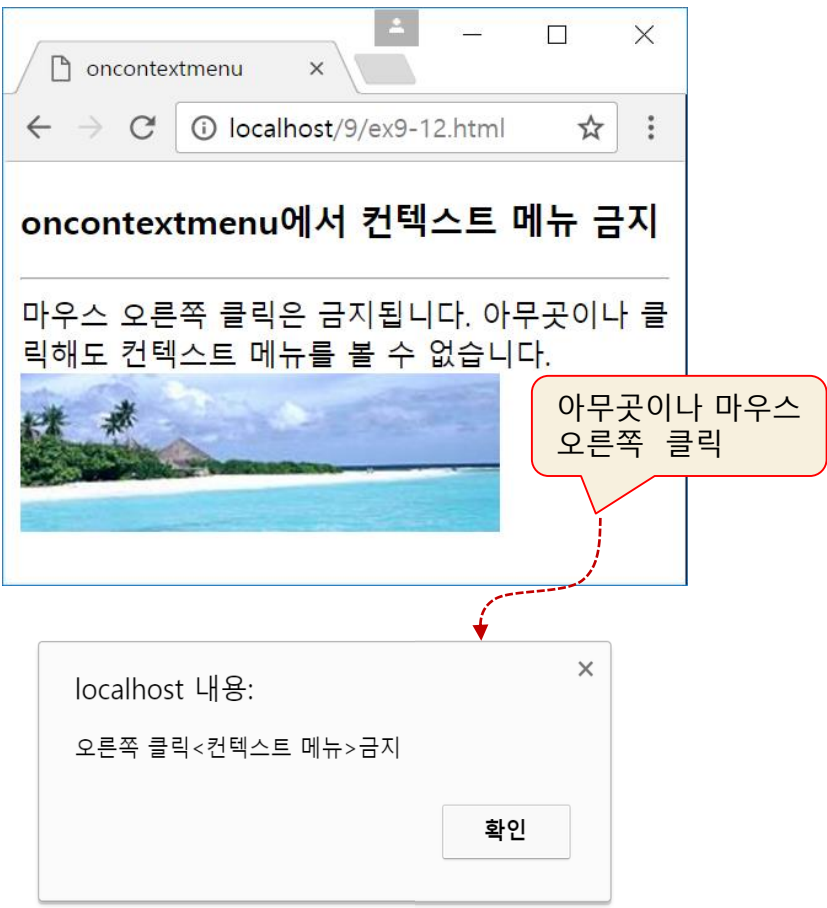
예제 9-12 oncontextmenu로 소스 보기나 이미지 다운로드 금지

39

```
<!DOCTYPE html>
<html>
<head> <title>oncontextmenu </title>
<script>
function hideMenu() {
    alert("오른쪽 클릭<컨텍스트 메뉴>금지");
    return false;
}
document.oncontextmenu=hideMenu;
</script>
</head>
<body>
<h3>oncontextmenu에서 컨텍스트 메뉴 금지
</h3>
<hr>
마우스 오른쪽 클릭은 금지됩니다. 아무곳이나
클릭해도 컨텍스트 메뉴를 볼 수 없습니다.

</body>
</html>
```

[실습-1] alert() 함수를 comment 처리
[실습-2] hideMenu()를 익명 함수로 수정



문서의 로딩 완료와 onload

40

□ onload 리스너

- 웹 문서 로딩 완료 시
- 이미지 로딩 완료 시

▣ window 객체에 발생

- 웹 페이지의 로딩 완료 시 자동으로 호출되는 이벤트 리스너
→ window 객체에 load 이벤트가 발생

▣ onload 리스너 작성 방법

1. `window.onload="alert('onload');"`

2. `<body onload="alert('onload');">`

=> 이 둘은 같은 표현임: `<body>` 태그에 onload 리스너를 만드는 것은 사실상 `window.onload`를 이용하는 것과 동일

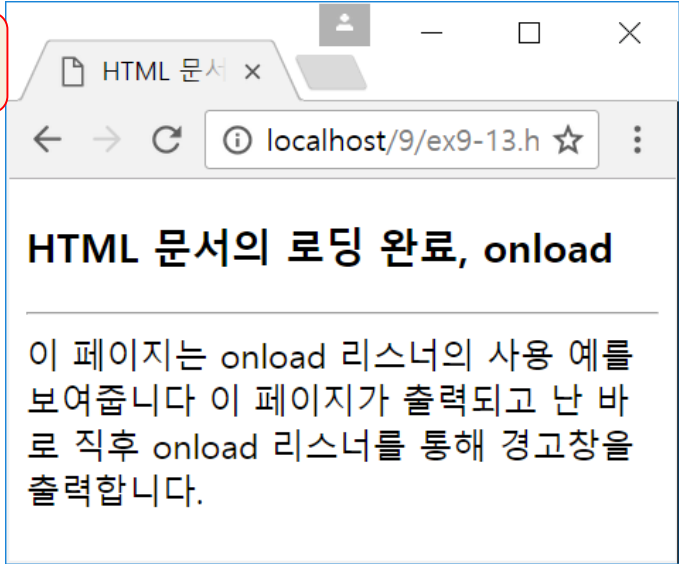
* `document.onload=function() { alert('onload') }` 는 최근에 와서 많은 브라우저에서 작동하지 않음

예제 9-13 onload에서 사이트 이전을 알리는 경고창 출력

41

```
<!DOCTYPE html>
<html>
<head> <title>HTML 문서의 onload</title>
</head>
<body onload="alert('이 사이트는 2017년 9월1일부터 www.js.co.kr로 옮겨지게 됩니다.');">
<h3>HTML 문서의 로딩 완료, onload</h3>
<hr>
이 페이지는 onload 리스너의 사용 예를 보여줍니다 이 페이지가 출력되고 난 바로 직후 onload 리스너를 통해 경고창을 출력합니다.
</body>
</html>
```

₩는 뒤에 <enter> 키를 무시하게 만듦



이 예제는 일부 브라우저에서 오동작하거나 오류가 발생할 수 있음



이미지 로딩 완료와 onload

42

- Image 객체
 - ▣ 태그에 의해 생성되는 DOM 객체
 - ▣ new Image(); 자바스크립트 코드에 의해 생성되는 객체
- onload
 - ▣ 이미지의 로딩이 완료되면 Image 객체에 발생하는 이벤트
- 새로운 이미지를 로딩하는 방법

이미지는 파일 크기가 크기때문에 로딩하는데 어느 정도 시간이 소요

```

```

```
var myImg = document.getElementById("myImg");  
myImg.src = "banana.png";
```

이미지 로딩 시 주의할 점

43

□ 잘못된 이미지 로딩 코드

- ▣ 이미지를 로딩하여 이미지 폭을 알아내는 코드

```
var myImg = document.getElementById("myImg");  
myImg.src = "banana.png";  
var width = myImg.width;           // banana.png 이미지의 폭
```

▣ 문제점

- myImg.src = "banana.png"; // 실행 직후 이미지 로딩 완료되지 않음
- var width = myImg.width; // 이미지 로딩 완료 전이면, myImg.width=0

□ 코드 수정

- ▣ onload 리스너에서 이미지 폭을 알아내는 코드 작성

```
var myImg = document.getElementById("myImg");  
myImg.src = "banana.png";           // 이미지 로딩 지시  
myImg.onload = function () {       // 이미지 로딩 완료 시 실행  
    var width = myImg.width;       // 정확한 이미지 폭 읽기  
}
```

★ 예제 9-14 onload로 이미지의 크기 알아내기

```
<!DOCTYPE html>
<html>
<head><title>onload로 이미지 크기 출력</title>
<script>
function changelImage() {
    var sel = document.getElementById("sel");           // 콤보 박스
    var img = document.getElementById("myImg");         // <img> 태그
    var index= sel.selectedIndex;                       // 선택된 옵션 인덱스: 0, 1, 2
    img.src = sel.options[index].value;                 // <option>의 value 속성
    img.onload = function () {                          // 이미지 크기 출력
        var mySpan = document.getElementById("mySpan");
        mySpan.innerHTML = img.width + "x" + img.height;
    }
}
</script>
</head>
<body onload="changelImage()">
<h3>onload로 이미지 크기 출력</h3>
<hr>
<form>
    <select id="sel" onchange="changelImage()">
        <option value="media/apple.png">사과 // value에 파일 이름 지정
        <option value="media/banana.png">바나나
        <option value="media/mango.png">망고
    </select>
    <span id="mySpan">이미지 크기</span>
</form>
<p></p>
</body></html>
```

이미지 로딩

이미지 로딩이 완료되면 발생

콤보 박스

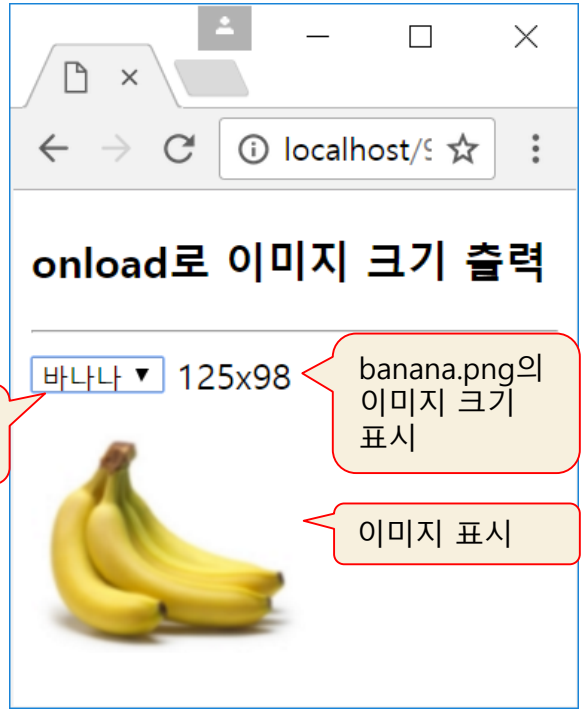
문서 로딩이 완료되면 발생

선택된 값이 바뀌면 발생

셋 중 하나에 selected를 추가하면?

초기 이미지

콤보 박스로 선택



banana.png의 이미지 크기 표시

이미지 표시

[실습-1] option 중 바나나에 selected 설정
[실습-2] 딸기(strawberry.png) 추가

new Image()로 이미지 로딩과 출력

45

- 동적으로 이미지 객체 생성
 - new Image() : 이미지 객체가 생기지만 화면에 출력되지는 않음
- new Image()의 이미지 객체에 이미지 로딩

```
var bananalmg = new Image(); // 이미지 객체 생성
bananalmg.src = "banana.png"; // 이미지 로딩
```

이미지를 메모리에 로딩만

- 로딩된 이미지 출력
 - 태그에 할당된 브라우저 공간에 이미지 출력

```

```

이미지 출력할 곳

```
var mylmg = document.getElementById("mylmg");
mylmg.src = bananalmg.src; // 이미지 출력
```

로딩된 이미지를 원하는 곳에 출력

★ 예제 9-15 new Image()로 이미지 로딩

46

```
<!DOCTYPE html>
<html> <head> <title>new Image()로 이미지 로딩</title>
<script>
// 배열에 저장된 이름의 이미지들을 미리 로딩
var files = ["media/penguins.png", "media/lighthouse.png",
            "media/Chrysanthemum.png", "media/Desert.png",
            "media/Hydrangeas.png", "media/Jellyfish.png",
            "media/Koala.png", "media/Tulips.png"];

var imgs = new Array();
for(var i=0; i<files.length; i++) {
    imgs[i] = new Image();
    imgs[i].src = files[i];
}

// 다음 이미지 출력
var next = 0;
function change(img) {
    next++;
    next %= imgs.length;
    img.src = imgs[next].src;
}

</script> </head>
<body>
<h3>new Image()로 이미지 로딩</h3>
<hr>
<p>이미지를 클릭하면 다음 이미지를 보여줍니다.</p>

</body>
</html>
```

이미지 로딩

화면에 이미지 출력

첫 번째 이미지

Array 생성

함수는 호출해야 실행, <script> 내부의 자동 실행

랜덤 값으로 수정

CSS로 수정

[실습-1] img 태그의 style을 CSS로 수정
[실습-2] 다음 이미지가 아니라 랜덤한 순서로 수정
[실습-3] [앞], [뒤] 버튼 추가, 버튼을 클릭하면 실행하도록 수정(뒤: 이전 이미지 표시)



onfocus와 onblur

47

□ 포커스(focus)

focus : 사진 찍을 때 피사체에 초점 맞추기
blur : 초점이 맞지 않아 흐릿한 경우

- 포커스는 현재 키 입력에 대한 독점권(입력 폼에서)
- 브라우저는 포커스를 가지고 있는 HTML 태그 요소에 키 공급

□ onfocus

- 포커스를 받을 때 발생하는 이벤트 리스너
예) 현재 HTML 요소를 클릭하면, 현재 HTML 요소가 포커스를 받는다.

□ onblur

- 포커스를 잃을 때 발생하는 이벤트 리스너
예) 다른 HTML 요소를 클릭하면, 현재 HTML 요소는 포커스를 잃는다.

예제 9-16 onfocus와 onblur: 입력 없이 다른 창으로 갈 수 없음

48

입력 폼에서 필수
입력 사항

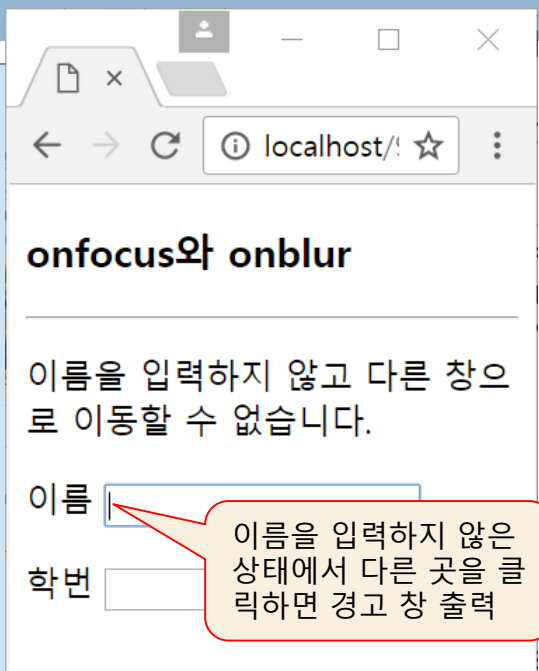
```
<!DOCTYPE html>
<html>
<head>
<title>onfocus와 onblur</title>
<script>
function checkFilled(obj) {
  if(obj.value == "") { // 입력 없음
    alert("enter name!");
    setTimeout(function() { obj.focus(); }, 1);
    // obj.focus();      // obj에 다시 포커스 주기
  }
}
</script>
</head>
<body onload="document.getElementById('name').focus();">
<h3>onfocus와 onblur</h3>
<hr>
<p>이름을 입력하지 않고 다른 창으로 이동할 수 없습니다.</p>
<form>
  <label>이름<input type="text" id="name" onblur="checkFilled(this)"> </label> <p>
  <label>학번<input type="text"> </label>
</form>
</body>
</html>
```

localhost 내용:

enter name!

확인

setTimeout(함수, 지연시간);
→ 지연시간(ms) 후에 함수 실행



이름을 입력하지 않은
상태에서 다른 곳을 클
릭하면 경고 창 출력

라디오버튼과 체크박스

라디오 버튼 객체 한 개만 선택

<input type="radio">로 만들어진 라디오 버튼 DOM 객체

```
<form>
  <input type="radio" name="city" value="seoul">서울
  <input type="radio" name="city" value="busan">부산
  <input type="radio" name="city" value="chunchen">춘천
</form>
```

그룹

☐ 서울 ☒ 부산 ☐ 춘천

라디오 버튼 객체들 알아내기

```
var kcity = document.getElementsByName("city"); // kcity[0], kcity[1], kcity[2]
```

체크박스 객체 여러 개 선택 가능

<input type="checkbox">로 만들어진 체크박스 DOM 객체

```
<form>
  <input type="checkbox" name="hap">모자 1만원
  <input type="checkbox" name="shoes">구두 3만원
  <input type="checkbox" name="bag">명품가방 8만원
</form>
```

그룹을 생성하지
않음

☐ 모자 1만원 ☒ 구두 3만원 ☒ 명품가방 8만원

★ 예제 9-17 선택된 라디오버튼 알아내기

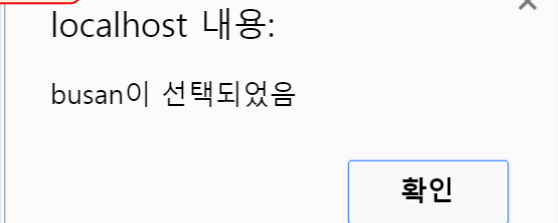
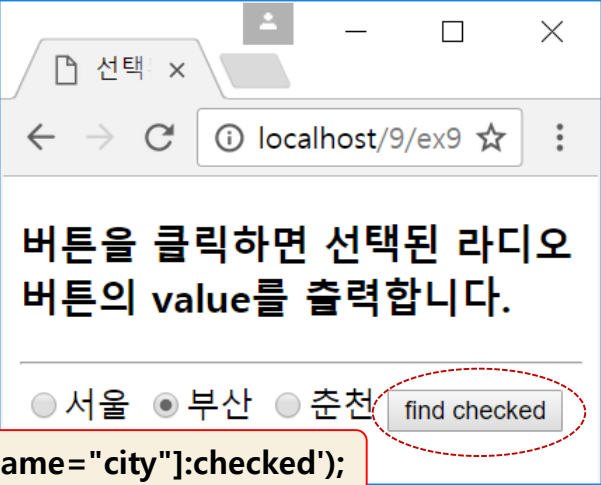
50

```
<!DOCTYPE html>
<html><head><title>선택된 라디오버튼 알아내기</title>
<script>
function findChecked() {
    var found = null; // 초기화
    var kcity = document.getElementsByName("city");
    for(var i=0; i<kcity.length; i++) { // 3
        if(kcity[i].checked == true)
            found = kcity[i];
    }
    if(found != null)
        alert(found.value + "이 선택되었음");
    else
        alert("선택된 것이 없음");
}
</script></head>
<body>
<h3>버튼을 클릭하면 선택된 라디오 버튼의 value를 출력합니다.</h3><hr>
<form>
    <input type="radio" name="city" value="seoul"> 서울
    <input type="radio" name="city" value="busan"> 부산
    <input type="radio" name="city" value="chunchen"> 춘천
    <input type="button" value="find checked" onclick="findChecked()">
</form>
</body></html>
```

found = document.querySelector('input[name="city"]:checked');

jQuery => \$("input[name='city']:checked").val()

버튼 클릭하면
결과 출력



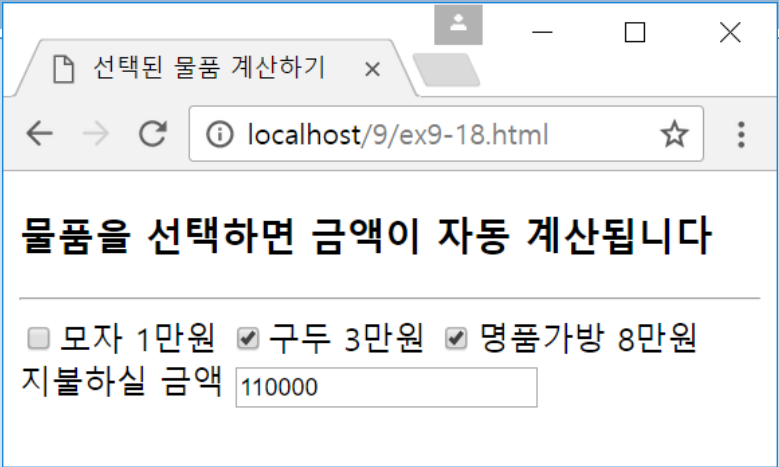
- [실습-1] '서울'이 초기값으로 선택되도록(checked) 수정
- [실습-2] 텍스트 상자를 추가 하고, 선택된 라디오 버튼의 value를 텍스트 상자에 출력
- [실습-3] <div> 태그에 출력

★ 예제 9-18 체크박스로 선택한 물품 계산

51

```
<!DOCTYPE html>
<html> <head> <title>선택된 물품 계산하기</title>
<script>
var sum=0;           // 초기화
function calc(cBox) {
  if(cBox.checked) sum += parseInt(cBox.value);
  else             sum -= parseInt(cBox.value);
  document.getElementById("sumtext").value = sum;
}
</script> </head>
<body>
<h3>물품을 선택하면 금액이 자동 계산됩니다</h3> <hr>
<form>
  <label> <input type="checkbox" name="hat" value="10000" onclick="calc(this)">모자 1만원</label>
  <label> <input type="checkbox" name="shoes" value="30000" onclick="calc(this)">구두 3만원</label>
  <label> <input type="checkbox" name="bag" value="80000" onclick="calc(this)">명품가방 8만원</label>
<br>
  <label>지불하실 금액 <input type="text" id="sumtext" value="0" ></label>
</form>
</body>
</html>
```

가격



클릭할 때마다 계산
결과 출력

전체를 순회하면서 check된
경우 처리하는 방법도 가능

[실습] 버튼을 클릭하면 합계를 계산하는 방식으로 수정하고, 초기화하는 버튼도 추가
(Hint) name을 같게 하고 name으로 객체를 선택한 후 반복문으로 계산

```
var fashion = document.getElementsByName("items");
for (var i=0; i<fashion.length; i++) {
  if (fashion[i].checked)
    ...
}
```

select 객체와 onchange

□ select 객체는 <select> 태그로 만들어진 콤보박스

■ option 객체는 <option>태그로 표현되는 옵션 아이템 배열

배열

```
<select id="fruits">
  <option value="1">딸기</option>
  <option value="2" selected>바나나</option>
  <option value="3">사과</option>
</select>
```

바나나 ▼

딸기

바나나

사과

■ 선택된 옵션 알아내기

```
var sel = document.getElementById("fruits");
var index = sel.selectedIndex; // index는 선택 상태의 옵션 인덱스, 0, 1, ...
```

■ 옵션 선택

```
var index = document.getElementById("fruits").selectedIndex;
```

options : 배열

```
sel.selectedIndex = 2; // 3번째 옵션 "사과" 선택
sel.options[2].selected = true; // 3번째 옵션 "사과" 선택
```

■ select와 onchange 리스너

■ 콤보박스에서 선택된 옵션이 변경되면 select 객체의 onchange 리스너 호출

```
<select id="fruits" onchange="drawImage()">...</select>
```


★ 예제 9-19 select 객체에서 선택한 과일 출력

53

```
<!DOCTYPE html>
<html>
<head> <title>select 객체에서 선택한 과일출력</title>
<script>
function drawImage() {
    var sel = document.getElementById("fruits");
    var img = document.getElementById("fruitimage");
    img.src = sel.options[sel.selectedIndex].value;
}
</script>
</head>
<body onload="drawImage()">
<h3>select 객체에서 선택한 과일 출력</h3> <hr>
과일을 선택하면 이미지가 출력됩니다.<p>
<form>
<select id="fruits" onchange="drawImage()">
    <option value="media/strawberry.png">딸기
    <option value="media/banana.png" selected>바나나
    <option value="media/apple.png">사과
</select>

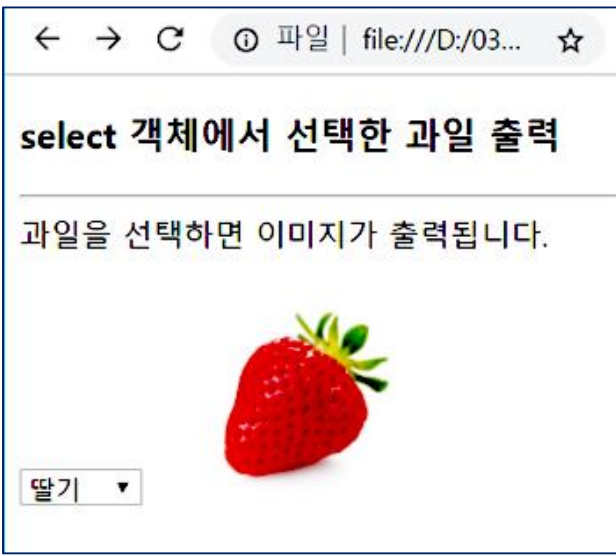
</form>
</body>
</html>
```

초기 이미지 출력



파일 이름 삭제 가능

[실습] 과일 이름을 선택하는 콤보박스와 이미지를 표 형태로 표시하도록 수정



키 이벤트

54

- onkeydown
 - ▣ 키가 눌러지는 순간 호출. 모든 키에 대해 작동
- onkeyup
 - ▣ 눌려진 키가 떼어지는 순간 호출
- onkeypress
 - ▣ 문자 키와 <Enter>, <Space>, <Esc> 키에 대해서만 눌러지는 순간에 추가 호출
 - ▣ 문자 키가 아닌 경우(<F1>, <Shift>, <PgDn>, , <Ins> 등) 호출되지 않음

★ 예제 9-20 키 이벤트 리스너와 이벤트 객체의 프로퍼티

55

```
<!DOCTYPE html>
<html> <head> <title>키 이벤트</title>
<script>
function whatKeyDown(e) {
  var str = "";
  var div = document.getElementById("div");
  div.innerHTML = ""; // div 객체 내용을 지운다.

  if(e.altKey) {
    if(e.altLeft) str += "왼쪽 Alt 키가 눌러졌습니다.";
    else str += "오른쪽 Alt 키가 눌러졌습니다.";
    str += "<br>";
  }

  if(e.shiftKey) {
    if(e.shiftLeft) str += "왼쪽 Shift 키가 눌러졌습니다.";
    else str += "오른쪽 Shift 키가 눌러졌습니다.";
    str += "<br>";
  }

  if(e.ctrlKey) {
    if(e.ctrlLeft) str += "왼쪽 Ctrl 키가 눌러졌습니다.";
    else str += "오른쪽 Ctrl 키가 눌러졌습니다.";
    str += "<br>";
  }

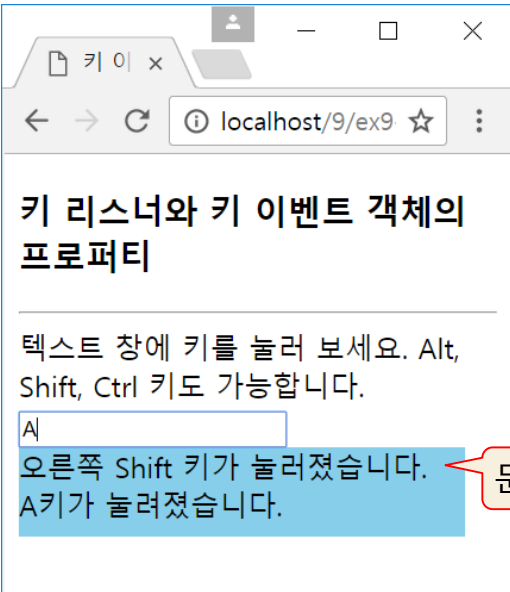
  str += String.fromCharCode(e.keyCode) + "키가 눌러졌습니다."
  div.innerHTML = str; // div 객체에 문자열을 출력한다.
}
</script>
</head>
```

keyCode에 해당하는 문자열을 반환하는 메소드

```
<body>
<h3>키 리스너와 키 이벤트 객체의 프로퍼티</h3>
<hr>
텍스트 창에 키를 눌러 보세요. Alt, Shift, Ctrl 키도 가능합니다.<br>
<input type="text" id="text" onkeydown="whatKeyDown(event)">
<div id="div" style="background-color:skyblue; width:250px; height:50px">
</div>
</body>
</html>
```

CSS로 수정

[실습] div 태그의 스타일 속성들을 style 태그로 구현



문자열 출력

onreset과 onsubmit

56

□ onreset

- ▣ reset 버튼(<input type="reset">) 클릭 시
- ▣ false를 리턴하면 폼이 초기화되지 않음

□ onsubmit

- ▣ submit(<input type="submit">) 버튼 클릭 시
- ▣ false를 리턴하면 폼 전송하지 않음

□ 리스너 작성

- ▣ onreset과 onsubmit 리스너는 <form> 태그에 달아야 한다.

```
<form onreset="..." onsubmit="...">
```

마우스 클릭 연습: 마우스 클릭을 연습하는 웹 페이지를 작성해 보자. 10x10의 셀을 가진 표를 만들고 랜덤하게 선택한 셀에 이미지를 출력하라. 이미지를 클릭하면 다른 셀을 랜덤하게 선택하여 이미지를 출력한다.

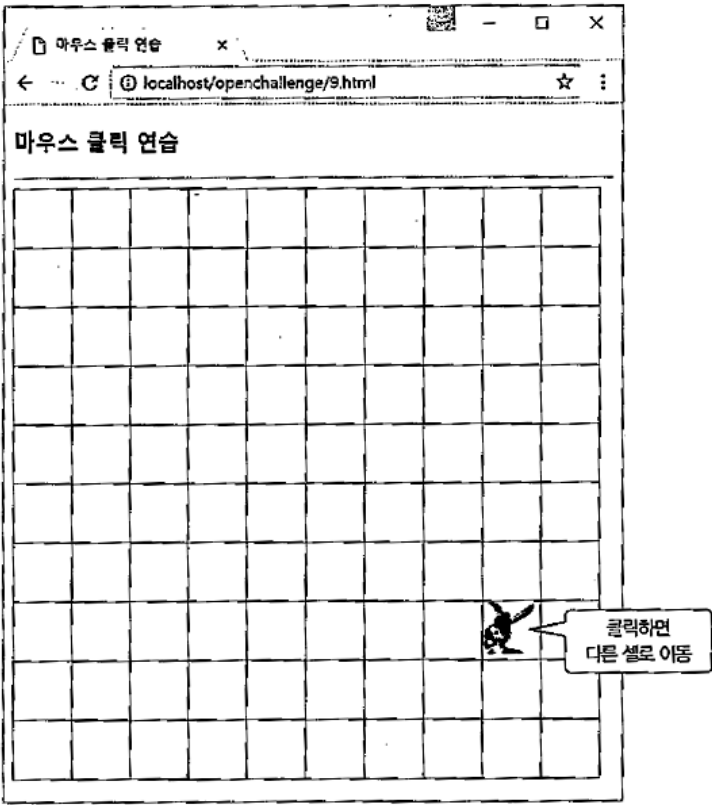
(힌트) 10x10셀을 만드는 방법은 여러 가지가 있다. 저자는 단순한 방법으로 작성하였다. 10x10셀을 가진 표를 만들고 각 셀에 를 삽입하여 이미지가 없게 하였다. 그리고 <body onload="">의 onload 리스너에 태그 중에서 한 개를 랜덤하게 선택하고, 이곳에 donkey.png 이미지를 출력하였다.

웹 페이지에 있는 모든 이미지는 다음 컬렉션으로 알 수 있다.

```
document.images[]
```

i번째 태그에 donkey.png 이미지를 출력하는 코드는 다음과 같다.

```
document.images[i].src = "donkey.png";
```



마우스 클릭 연습: 마우스 클릭을 연습하는 웹 페이지를 작성해 보자. 10x10의 셀을 가진 표를 만들고 랜덤하게 선택한 셀에 이미지를 출력하라. 이미지를 클릭하면 다른 셀을 랜덤하게 선택하여 이미지를 출력한다.

58

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>마우스 클릭 연습</title>
<style>
table {
  border : 1px solid blue;
  border-collapse : collapse;
}
td {
  border : 1px solid blue;
  width : 50px;
  height : 50px;
  padding : 0px;
}
</style>
<script>
var select=0;          // 선택된 셀 번호
var img = new Image();
img.src="media/donkey.png";  // 45x45 크기. 미리 로딩

function set() {
  // 100개의 이미지에 모두 click 리스너 등록
  // <img> 태그 엘리먼트가 공간을 차지하지만 보이지 않게 한다.

  movelImage();  // 셀을 선택하고 이미지 출력
}
```

```
// <img> 태그의 click 리스너
function movelImage() {          // 다른 셀로 이미지를 옮긴다.
  // 현재 <img>에 보여질 이미지를 제거하고, 보이지 않게 한다.

  // 0~99 사이의 랜덤한 셀을 선택

  // 새로 선택된 셀에 이미지 출력하고, 이미지를 보이게 한다.

}
</script>
</head>

<body onload="set()">
<h3>마우스 클릭 연습</h3>
<hr>
<table>          // 10x10 테이블
<tr><td><img src=""></td><td><img src=""></td><td><img
src=""></td><td><img src=""></td><td><img src=""></td><td><img
src=""></td><td><img src=""></td><td><img src=""></td><td><img
src=""></td><td><img src=""></td></tr>
...
<tr><td><img src=""></td><td><img src=""></td><td><img
src=""></td><td><img src=""></td><td><img src=""></td><td><img
src=""></td><td><img src=""></td><td><img src=""></td><td><img
src=""></td><td><img src=""></td></tr>
</table>
</body>
</html>
```

[실습] <table>을 직접 생성하지 않고 <table> element를 만들어서 document.body.appendChild()로 추가하기

59

1. 브라우저 내의 아무 곳이나 클릭하면 왼쪽 화면과 같고, 브라우저 바깥의 아무 곳에 마우스를 클릭하면 브라우저의 배경색이 lightgray로 바뀌도록 웹 페이지를 작성하라.

(힌트) 브라우저 바깥에 마우스를 클릭하면 브라우저의 window 객체는 포커스를 잃게 되어 blur 이벤트가 발생한다. window 객체에 onblur 리스너를 작성하면 된다.

포커스와 onblur,onfocus

브라우저 바깥에 마우스를 클릭하면 window 객체에 blur 이벤트가 발생하고 다시 마우스를 클릭하면 window 객체에 focus 이벤트가 발생한다.

(포커스를 가지고 있는 상태)

포커스와 onblur,onfocus

브라우저 바깥에 마우스를 클릭하면 window 객체에 blur 이벤트가 발생하고 다시 마우스를 클릭하면 window 객체에 focus 이벤트가 발생한다.

(포커스를 잃게 되는 경우)

```
<!DOCTYPE html>
<html>
<head><meta charset="UTF-8">
<title>onblur와 onfocus</title>
<script>
window.onblur = function(e) {
    document.body.style.backgroundColor = "lightgray";
}
window.onfocus = function(e) {
    document.body.style.backgroundColor = "white";
}
</script>
</head>
```

```
<body>
<h3>포커스와 onblur,onfocus</h3>
<hr>
<p>
브라우저 바깥에 마우스를 클릭하면 window 객체에 blur 이벤트가 발생하고 다시 마우스를 클릭하면 window 객체에 focus 이벤트가 발생한다.
</p>
</body>
</html>
```

[연습] <body> 태그에 onblur, onfocus 이벤트 리스너를 추가하는 방식으로 수정

3. 계산식을 입력받아 결과를 출력하는 웹 페이지를 작성하라. 식 입력 후 <Enter> 키를 치면 수식을 계산하고 결과를 출력한다.

60

(힌트) 예제 9-9를 참고하라. <Enter> 키의 keyCode 값은 13이므로 이벤트 객체 e에 대해, e.keyCode와 13을 비교하면 된다.

계산기 만들기

수식을 입력하고 <Enter>를 입력하세요.

식

값

```
<!DOCTYPE html>
<html> <head>
<meta charset="UTF-8">
<title>onkeydown</title>
<script>
function calculate(e) {
  if(e.keyCode == 13) {
    var exp = document.getElementById("exp");
    var result = document.getElementById("result");
    result.value = eval(exp.value);
  }
}
</script>
</head>
```

exp, result 변수를 사용하지 않고 축약

```
<body >
<h3>계산기 만들기</h3>
<hr>
수식을 입력하고 &lt;Enter&gt;를 입력하세요.
<p>
<form>
<label>식<input type="text" id="exp" value=""
onkeydown="calculate(event)"> </label> <br>
<label>값<input type="text" id="result"> </label>
</form>
</body>
</html>
```


5. 브라우저의 아무 곳이나 더블 클릭하면 바탕색이 랜덤하게 바뀌는 웹 페이지를 작성하라.

61

(힌트) 배경색을 바꾸는 자바스크립트 코드의 예를 들면 다음과 같다.

```
document.body.style.backgroundColor = "rgb(55, 46, 126)";
```

바탕 아무 곳이나 더블 클릭

바탕 아무 곳이나 **더블클릭**하면 배경색이 랜덤하게 변합니다.

바탕 아무 곳이나 더블 클릭

바탕 아무 곳이나 **더블클릭**하면 배경색이 랜덤하게 변합니다.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>자바스크립트 연습</title>
<script>
function randomBackground() {
  var r = Math.floor(Math.random()*256);
  var g = Math.floor(Math.random()*256);
  var b = Math.floor(Math.random()*256);
  document.body.style.backgroundColor = "rgb(" + r + "," + g + "," + b + ")";
}
</script>
</head>
```

```
<body ondblclick="randomBackground()">
<h3>바탕 아무 곳이나 더블 클릭</h3>
<hr>
<p>바탕 아무 곳이나 <strong>더블클릭</strong>하면
배경색이 랜덤하게 변합니다.
</p>
</body>
</html>
```

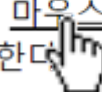
7. 웹 페이지에 있는 모든 `` 태그에 대해, 마우스가 올라오면 해당 태그의 텍스트에 밑줄을 긋고, 내려가면 밑줄을 지우는 자바스크립트 코드를 완성하라.

(힌트) 이벤트가 발생하면, 이벤트 객체를 검사하여 `e.target.tagName=="SPAN"`인 경우만 처리하면 된다. `tagName` 프로퍼티는 항상 대문자의 문자열을 가진다.

```
<!DOCTYPE html>
<html>
<head> <meta charset="UTF-8">
<title> 이벤트 객체의 target</title>
<script>
window.onmouseover = function(e) {
    // 이곳에 자바스크립트 코드 작성
}
window.onmouseout = function(e) {
    // 이곳에 자바스크립트 코드 작성
}
</script>
</head>
<body>
<h3>span 태그에만 onmouseover/onmouseout</h3>
<hr>
<p>span 태그에 대해서만 <span>마우스</span>가 올라올 때
밑줄이 그어지고 <span>마우스</span>가 내려갈 때 밑줄이 사라지도록
<span>자바스크립트 코드</span>를 작성한다.
</p>
</body>
</html>
```

span 태그에만 onmouseover/onmouseout

span 태그에 대해서만 마우스가 올라올 때 밑줄이 그어지고 마우스가 내려갈 때 밑줄이 사라지도록 자바스크립트 코드를 작성한다.



7. 웹 페이지에 있는 모든 태그에 대해, 마우스가 올라오면 해당 태그의 텍스트에 밑줄을 긋고, 내려가면 밑줄을 지우는 자바스크립트 코드를 완성하라.

63

```
<!DOCTYPE html>
<html>
<head><meta charset="UTF-8">
<title>이벤트 객체의 target</title>
<script>
window.onmouseover = function(e) {
    if(e.target.tagName.toLowerCase() != "span") return;
    e.target.style.textDecoration = "underline";
}
window.onmouseout = function(e) {
    if(e.target.tagName.toLowerCase() != "span") return;
    e.target.style.textDecoration = "none";
}
</script>
</head>
<body>
<h3>span 태그에만 onmouseover/onmouseout</h3>
<hr>
<p>span 태그에 대해서만 <span>마우스</span>가 올라올 때
밑줄이 그어지고 <span>마우스</span>가 내려갈 때 밑줄이 사라지도록
<span>자바스크립트 코드</span>를 작성한다.
</p>
</body>
</html>
```

9. 리스트를 만들고 아이템 영역 안에 마우스가 들어오면 다음과 같이 상세한 설명을 출력하는 웹 페이지를 작성하라.

(힌트) 태그에 onmouseover, onmouseout 리스너를 작성하고, onmouseover에서는 미리 position:absolute로 지정한 <div> 태그를 마우스가 올라온 위치로 배치하고 <div> 태그의 innerHTML에 적당한 텍스트를 출력한다. 또한 visibility 스타일에 "visible"을 주어 보이게 한다. onmouseout에서는 <div> 태그의 visibility 스타일에 "hidden"을 주어 보이지 않게 한다.

자전거로 대한민국 일주

- 빨간자전거를 타고 서울, 대전, 대구, 부산 찍고, 목포, 인천을 거쳐 달린다

책 100권 읽기

- 한국 고전 50권, 외국 고전 50권을 읽는 목표

철인 3종 경기 준비

- 아침에는 수영을 배우고, 점심먹고 자전거 훈련 30KM, 저녁에는 10KM 달리기 하기

자바스크립트 정복

- 문제가 많고 잘 정리된 자바스크립트 책을 구입 하여 본격적으로 연습문제 풀기

아이템에 마우스를 올리면 설명 출력

여름 방학 때 하고 싶은 것들

- 자전거로 대한민국 일주
- 책 100권 읽기
- 철인 3종 경기 준비
- 자바스크립트 정복

빨간자전거를 타고 서울, 대전, 대구, 부산 찍고, 목포, 인천을 거쳐 달린다

9. 리스트를 만들고 아이템 영역 안에 마우스가 들어오면 다음과 같이 상세한 설명을 출력하는 웹 페이지를 작성하라.

65

```
<!DOCTYPE html>
<html>
<head><meta charset="UTF-8">
<title>onmouseover과 onmouseout</title>
<script>
var text=["빨간자전거를 타고 서울, 대전, 대구, 부산 찍고, 목포, 인천을
거쳐 달린다",
        "한국 고전 50권, 외국 고전 50권을 읽는 목표",
        "아침에는 수영을 배우고, 점심 먹고 자전거 훈련 30KM, 저녁에
는 10KM 달리기 하기",
        "문제가 많고 잘 정리된 자바스크립트 책을 구입하여 본격적으
로 연습문제 풀기"];

var commentDiv;
function init() {
    commentDiv = document.getElementById("commentDiv");
    var liArray = document.getElementsByTagName("li");
    for(i=0; i<liArray.length; i++) {
        liArray[i].addEventListener("mouseover", over, false);
        liArray[i].addEventListener("mouseout", hideCommentDiv, false);
    }
}
function over(e) {
    var n=0;
    switch(e.target.id) {
        case "l0" : n = 0; break;
        case "l1" : n = 1; break;
        case "l2" : n = 2; break;
        case "l3" : n = 3; break;
    }
    setCommentDiv(text[n], e);
}
```

```
function setCommentDiv(comment, e) {
    commentDiv.innerHTML = comment;
    commentDiv.style.left = e.clientX + "px";
    commentDiv.style.top = e.clientY + "px";
    commentDiv.style.border = "1px solid yellowgreen";
    commentDiv.style.background = "aliceblue";
    commentDiv.style.visibility = "visible";
    commentDiv.style.width = "200px";
    commentDiv.style.height = "80px";
}

function hideCommentDiv() {
    commentDiv.style.visibility = "hidden";
}
</script>
</head>
<body onload="init()">
<h3>아이템에 마우스를 올리면 설명 출력</h3>
<hr>
<p>여름 방학 때 하고 싶은 것들</p>
<ul>
    <li id="l0">자전거로 대한민국 일주</li>
    <li id="l1">책 100권 읽기</li>
    <li id="l2">철인 3종 경기 준비</li>
    <li id="l3">자바스크립트 정복</li>
</ul>
<div id="commentDiv" style="position:absolute"></div>
</body>
</html>
```

<style> 태그로 구현

End of Chap 09