



Chapter11

JavaScript 방문 카운터 만들기

HTML & JavaScript

1. 이벤트 핸들러

2. 브라우저 내장 객체

3. Window 객체

4. Document 객체

5. 방문 카운터 만들기

❖ 이벤트

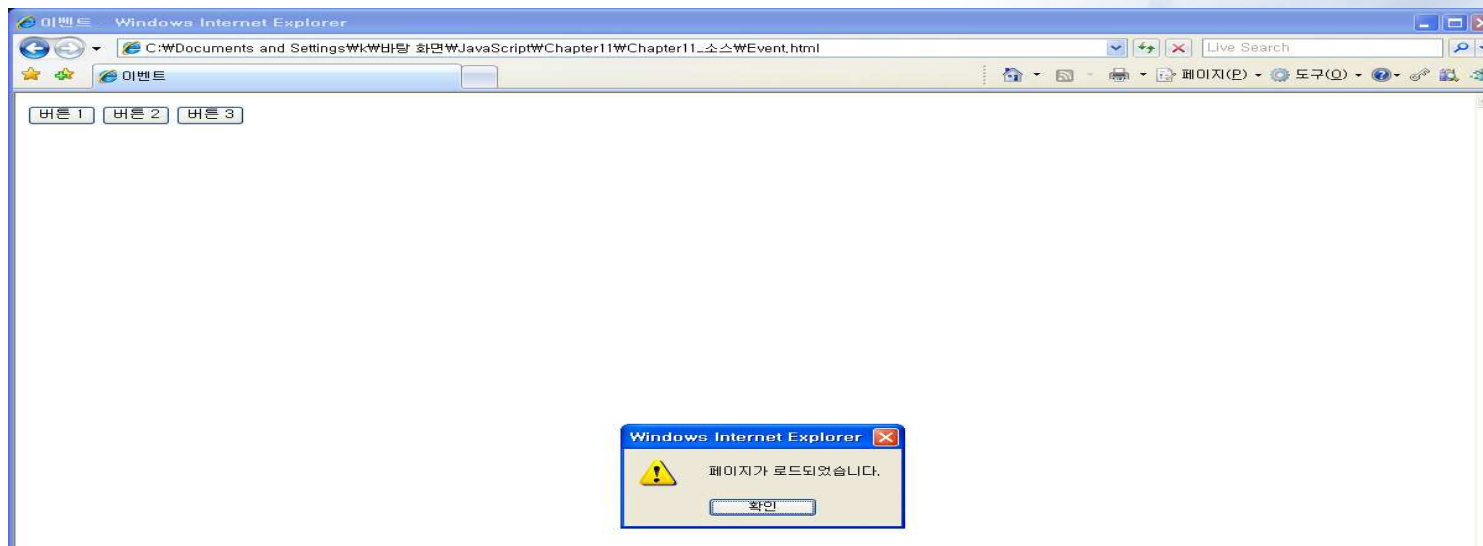
- 어떠한 일이 발생하는 것

❖ 이벤트 핸들러

- 이벤트가 발생하였을 때 호출하는 함수를 지정

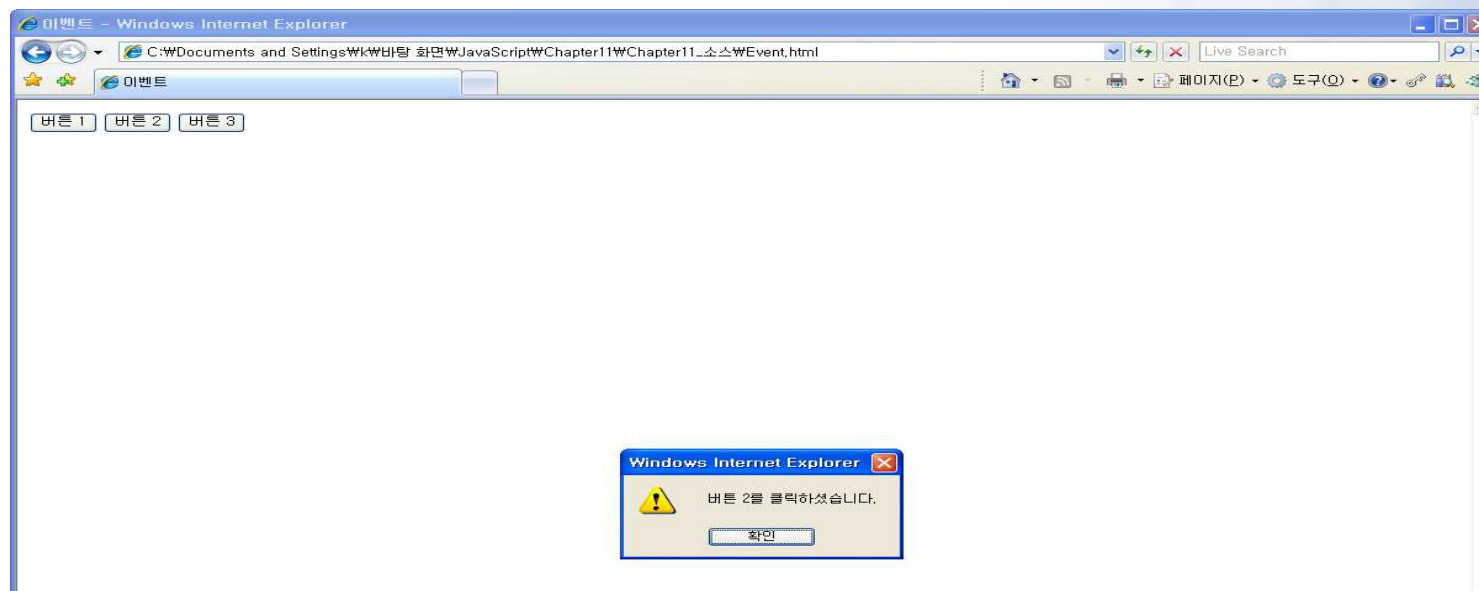
이벤트 핸들러	설명
onLoad	브라우저의 문서를 읽을 때 발생
onUnload	브라우저의 문서를 닫을 때 발생
onKeyDown	키보드의 키를 클릭했을 때 발생
onKeyPress	키보드의 키를 클릭했다가 놓았을 때 발생
onKeyUp	키보드의 키를 놓았을 때 발생
onMouseDown	마우스 버튼을 눌렀을 때 발생
onMouseMove	마우스를 이동하였을 때 발생
onMouseOut	마우스가 지정한 영역을 벗어났을 때 발생
onMouseOver	마우스가 지정한 영역 위에 위치할 때 발생
onMouseUp	마우스 버튼을 놓았을 때 발생
onDbClick	마우스를 더블 클릭했을 때 발생
onDragDrop	마우스를 드래그 했을 때 발생
onChange	입력양식의 값이 바뀌었을 때 발생
onClick	입력양식을 클릭했을 때 발생
onBlur	입력양식에서 포커스가 벗어났을 때 발생
onFocus	입력양식에 포커스가 위치했을 때 발생
onReset	입력양식을 초기화 시켰을 때 발생
onSelect	입력양식을 선택할 때 발생

```
<BODY onLoad="loadEvent()" onUnload="unloadEvent()">  
  
<FORM>  
  
<INPUT TYPE="button" VALUE="버튼 1"  
      onClick="clickEvent1()">
```

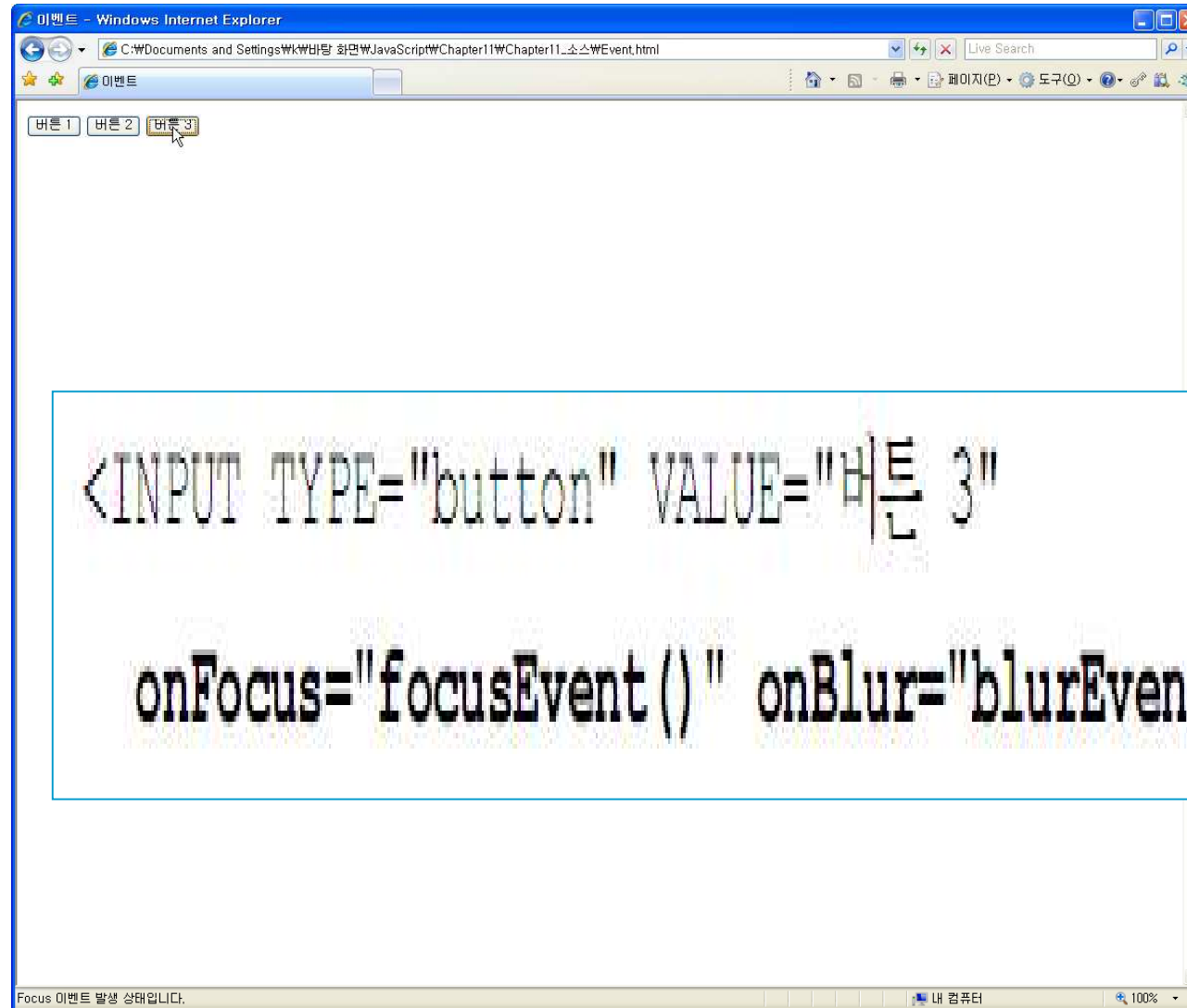


[그림 11-1] Event의 실행결과

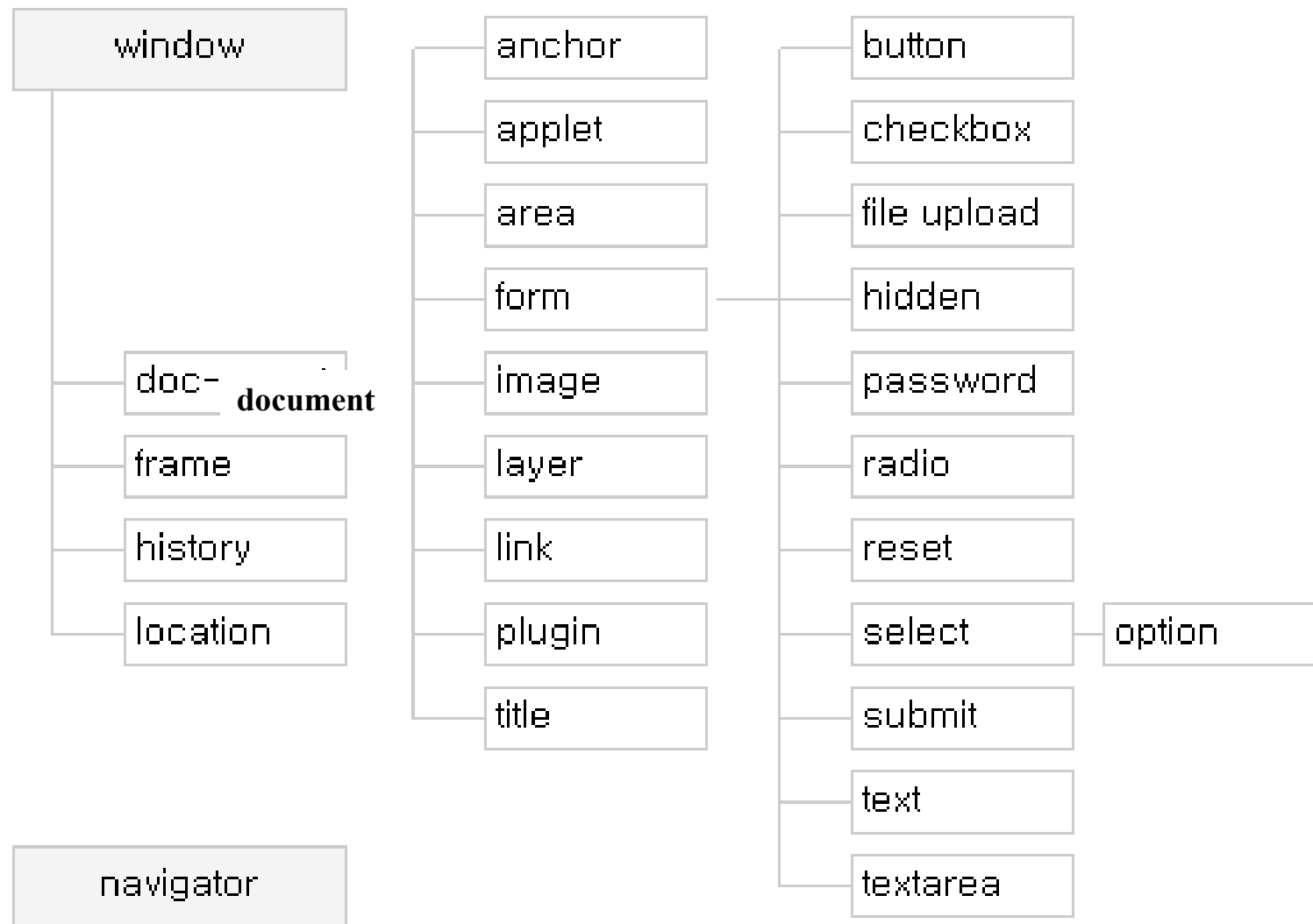
```
<INPUT TYPE="button" VALUE="버튼 2"  
onClick="clickEvent2 () ">
```



[그림 11-2] Event의 실행결과



[그림 11-3] Event의 실행결과



[그림 11-4] 브라우저 내장 객체 계층 구조

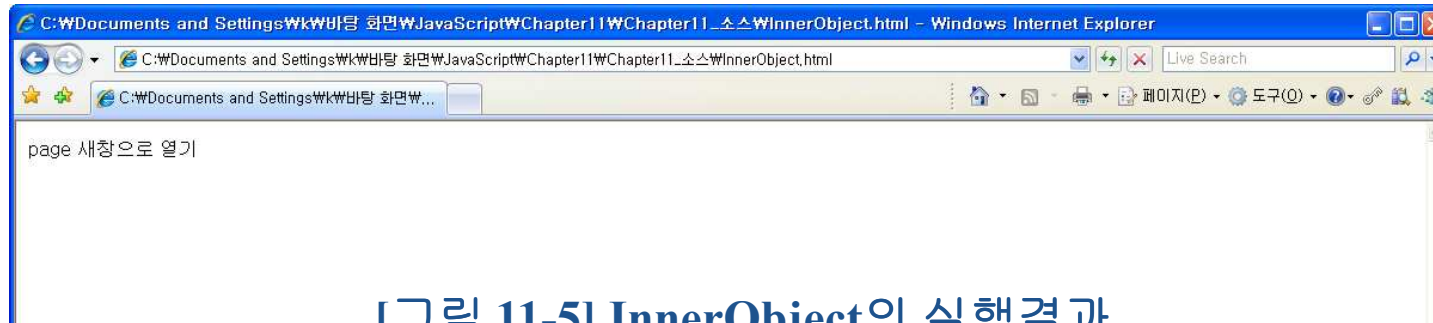
❖ **Window** 객체 : 최상위 객체로 브라우저 창에 관련된 속성과 메소드를 지원

- **Frame** 객체 : 여러 프레임을 처리
- **Location** 객체 : 브라우저 위치 등을 처리
- **History** 객체 : 브라우저로 본 히스토리 정보를 처리
- **Navigator** 객체 : 브라우저 식별
- **Document** 객체 : 링크, form 등 페이지 안의 태그를 처리하기 위해 사용

프로그램 11-2, 프로그램 11-3

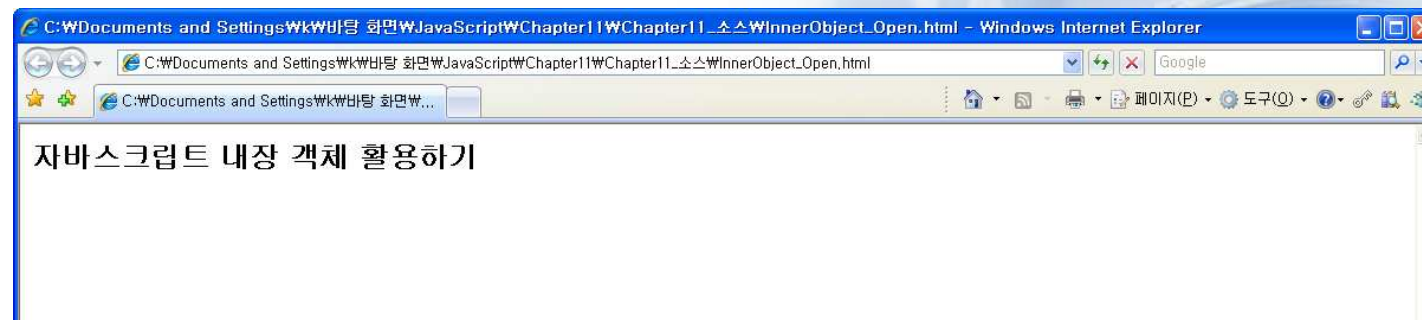
Chapter11

```
window.open("InnerObject_open.html");
```



[그림 11-5] InnerObject의 실행결과

```
document.write("<H1>" +  
"자바스크립트 내장 객체 활용하기" + "</H1>");
```



[그림 11-6] InnerObject open의 실행결과

❖ Window 객체

- 하나의 페이지 자체를 의미하며, 브라우저의 내장 객체 계층 구조 중에서 최상위에 있는 객체
- 하위 자식 객체를 다 포함하는 하나의 완결된 페이지를 다룸
- 생략하여 사용해도 됨

속성	설명
closed	창이 닫혀 있는 상태
defaultStatus	브라우저의 상태바의 초기 문자열
length	창 안의 프레임의 수
name	창 이름
opener	open() 메소드로 창을 연 페이지가 있는 윈도우
parent	프레임에서 현재 프레임의 상위 프레임 (부모)
self	현재 창 자신, window와 같음
status	브라우저의 상태바의 문자열
top	현재 프레임의 최상위 프레임

메소드	설명
alert(문자열)	경고창을 보여줌
blur()	포커스 제거
clearInterval(setInterval 식별자)	setInterval 메소드 정지
clearTimeout(setTimeout 식별자)	setTimeout 메소드를 정지
close()	창을 닫음
confirm(문자열)	확인, 취소를 선택할 수 있는 창을 보여줌
focus()	포커스를 줌
moveBy(x좌표, y좌표)	창을 상대좌표로 이동. 수평방향과 수직 방향의 이동량을 픽셀로 지정
moveTo(x좌표, y좌표)	창을 절대좌표로 이동. 창의 왼쪽 상단 모서리를 기준으로 픽셀을 지정
open(URL, 이름, [옵션])	새로운 창을 오픈
print()	화면의 내용을 프린트로 출력
prompt(문자열, 문자열)	입력이 있는 창을 보여줌
resizeBy(너비, 높이)	창의 크기를 상대 좌표로 재설정. 밑변의 모서리를 기준으로 수평, 수직 방향을 픽셀로 지정
resizeTo(너비, 높이)	창의 크기를 절대 좌표로 재설정. 창의 크기를 픽셀로 지정
scroll()	창을 스크롤 시킴
scrollBy(수평길이, 수직길이)	상대좌표를 창을 스크롤 함
scrollTo(수평길이, 수직길이)	절대좌표로 창을 스크롤 함. 창의 왼쪽 상단 모서리를 기준으로 픽셀로 지정
setInterval(함수, 시간(1/1000ms))	일정시간 간격으로 지정함수를 반복 호출.
setTimeout(함수, 시간(1/1000ms))	일정시간 후 지정함수를 호출.

형식 :

`window.객체`

`window.메소드`

`window.속성`

속성 :

`closed, defaultStatus, length, name, opener, parent, self, status, top`

메소드 :

`alert(), blur(), clearInterval(), clearTimeout(), close(), confirm(), focus(), moveBy(), moveTo(), open(), print(), prompt(), resizeBy(), resizeTo(), scroll(), scrollBy(), scrollTo(), setInterval(), setTimeout()`

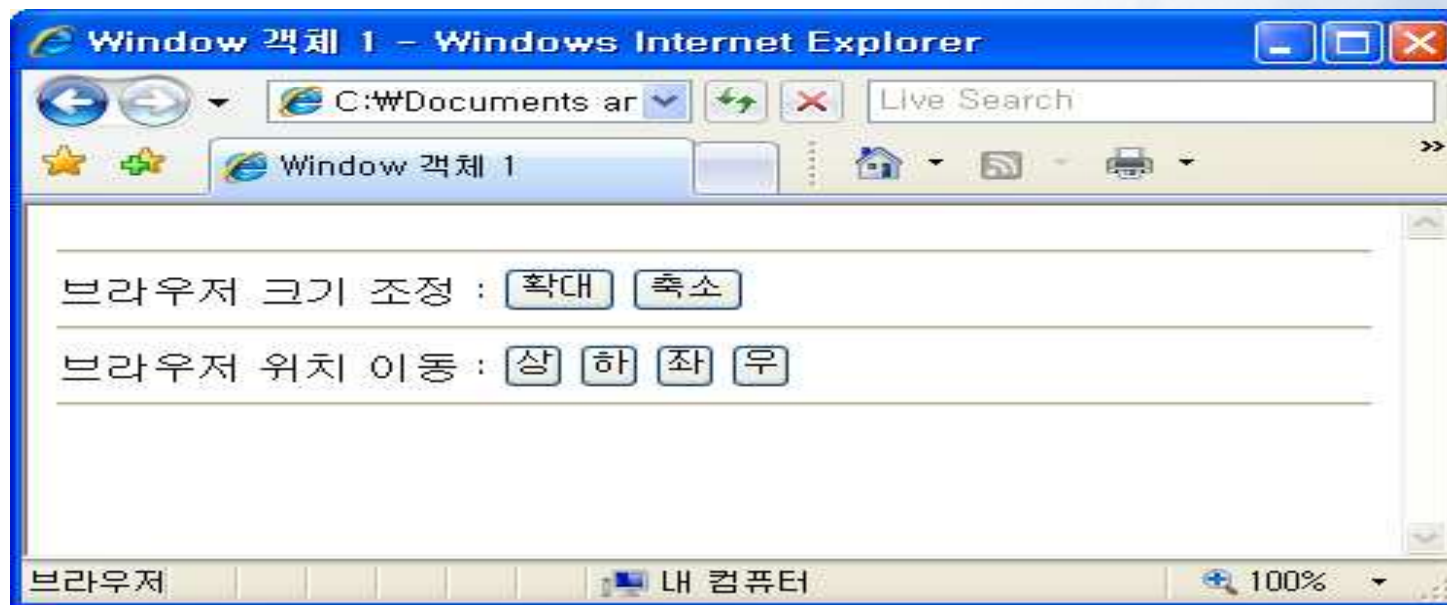
예제 :

```
<SCRIPT LANGUAGE="JavaScript">
<!--
    window.open("Innerobject_open.html");
//-->
</SCRIPT>
```



```
window.defaultStatus = "브라우저 크기/위치 조정하기"

function resize_by(x, y)
{
    window.resizeBy(x, y);
}
```



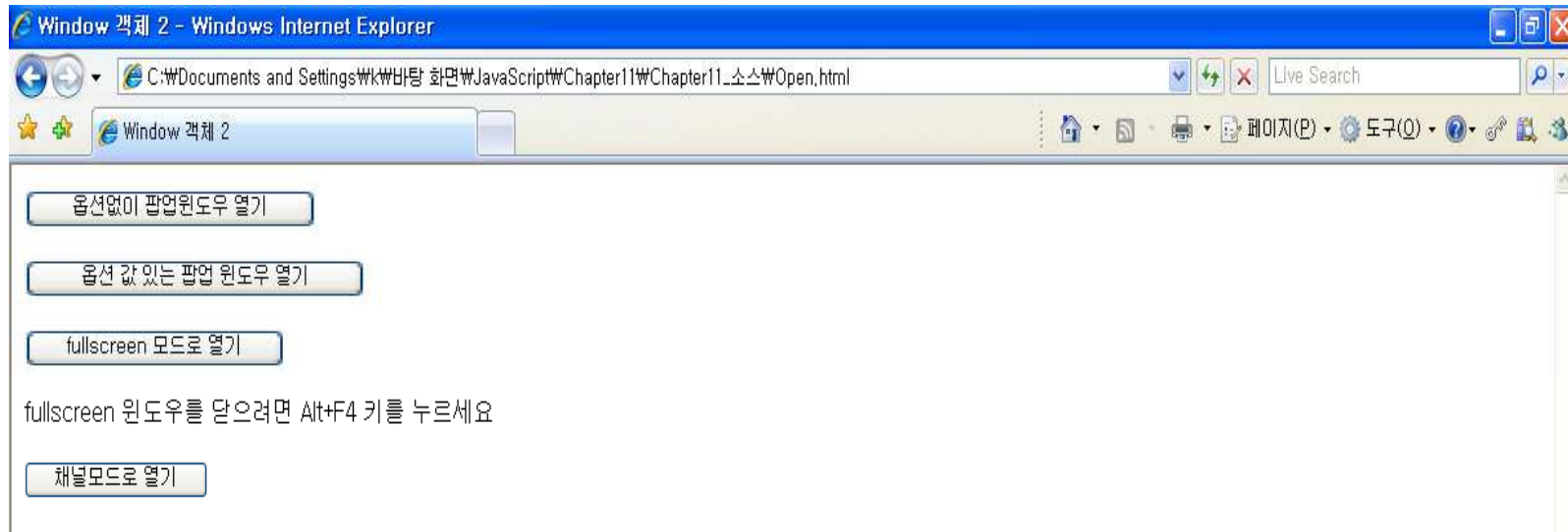
[그림 11-7] Moveby의 실행결과

❖ Window 객체의 open() 메소드

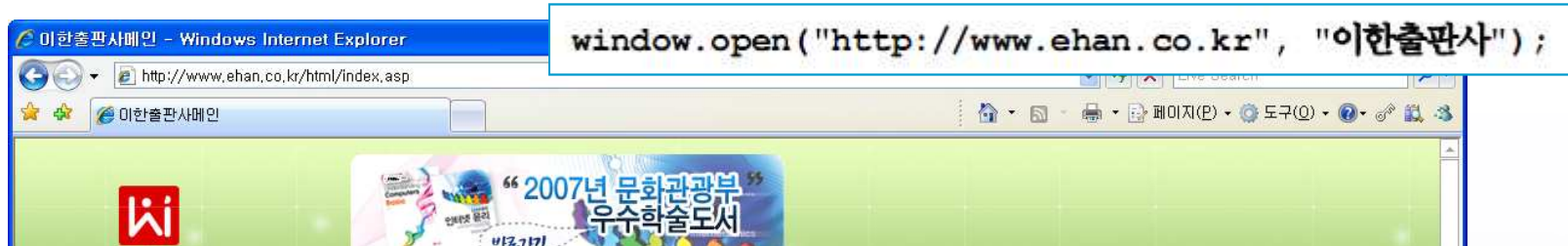
- 괄호 안에 아무 내용도 넣지 않으면 단순히 빈 창을 열게 되고, 열고자 하는 창의 URL을 넣으면 해당 URL의 창을 열음
- URL, 문서의 이름, 옵션 등 3개의 인수를 넣을 수 있음

```
window.open("http://www.ehan.co.kr", "이한", "width=500,height=300");
```


옵션	설명
menubar	메뉴바를 보여준다./숨긴다.
toolbar	도구막대를 보여준다./숨긴다.
directories	디렉토리바를 보여준다./숨긴다.
scrollbars	스크롤바를 보여준다./숨긴다.
status	상태 표시줄을 보여준다./숨긴다.
location	주소 표시줄을 보여준다./숨긴다.
width	팝업 윈도우의 가로를 지정 한다.
height	팝업 윈도우의 높이를 지정 한다.
left	팝업 윈도우의 x축 위치를 지정 한다.
top	팝업 윈도우의 y축 위치를 지정 한다.
resizable	팝업 윈도우의 크기를 사용자가 임의로 수정 할 수 있는지 여부를 지정 한다.
fullscreen	전체화면 모드로 연다.
channelmode	채널모드로 연다.



[그림 11-8] Open의 실행결과



[그림 11-9] 옵션 없는 팝업

```

window.open("http://www.ehan.co.kr", "이한출판사",
            "width=500,height=300,left=0,top=0,resizable=no");
    
```



[그림 11-10] 옵션 있는 팝업



[그림 11-11] fullscreen 팝업



[그림 11-12] 채널 모드 팝업

❖ Document 객체

- HTML 문서를 의미
- Window 객체의 하위 객체로 웹 브라우저에서 실행되는 모든 HTML 문서에 대한 정보를 가지고 관리하는 객체
- <BODY> ~ </BODY> 태그 속의 내용과 관련된 처리
- 실제 이용은 이 객체에서 파생된 하위 객체 (Layer, Link, Image, Area, Anchor, Applet, Form) 등을 이용해 작업을 주로 수행

속성	설명
alinkColor	누르고 있는 동안의 링크문자열의 색을 설정
anchors[]	anchor 오브젝트 및 배열
applets[]	applet 객체 배열
bgColor	창의 배경색을 설정
cookie	쿠키파일의 정보
domain	현재 도메인 이름
embeds[]	plug in 배열정보
fgColor	문서 텍스트 색상
forms[]	form 오브젝트 및 배열
images[]	이미지 오브젝트 및 배열
lastModified	파일의 최종 변경 날짜를 알아냄
linkColor	링크색을 설정
links[]	링크 오브젝트 및 배열
plugins[]	embeds[] 배열의 동의어
referrer	이전 문서의 URL
title	문서의 제목을 설정하거나 반환
URL	현재의 URL
location	현재 문서의 URL
vlinkColor	이미 누른 적이 있는 링크 문자열의 색을 설정

메소드	설명
close()	문서에 내용입력을 마무리함
clear()	문서의 내용을 삭제
open([MIME 타입])	문서에 내용입력을 준비함
write(문자열)	문서에 내용을 출력
writeln(문자열)	줄바꾸기를 포함 문서에 내용을 출력
eval(수식)	문자열을 수치로 변경
toString()	오브젝트를 문자열로 변경
valueOf()	오브젝트 값을 표시
getSelection()	선택 범위 안에 포함되어 있는 문자열을 반환

형식 :

`document.객체`
`document.메소드`
`document.속성`

속성 :

`alinkColor`, `anchors[]`, `applets[]`, `bgColor`, `cookie`, `domain`, `embeds[]`, `fgColor`,
`form[]`, `images[]`, `lastModified`, `linkColor`, `linksp[`, `plugins[]`, `referrer`, `title`, `URL`,
`location`, `vlinkColor`

메소드 :

`close()`, `clear()`, `open()`, `write()`, `writeln()`, `eval()`, `toString()`, `valueOf()`,
`getSelection()`

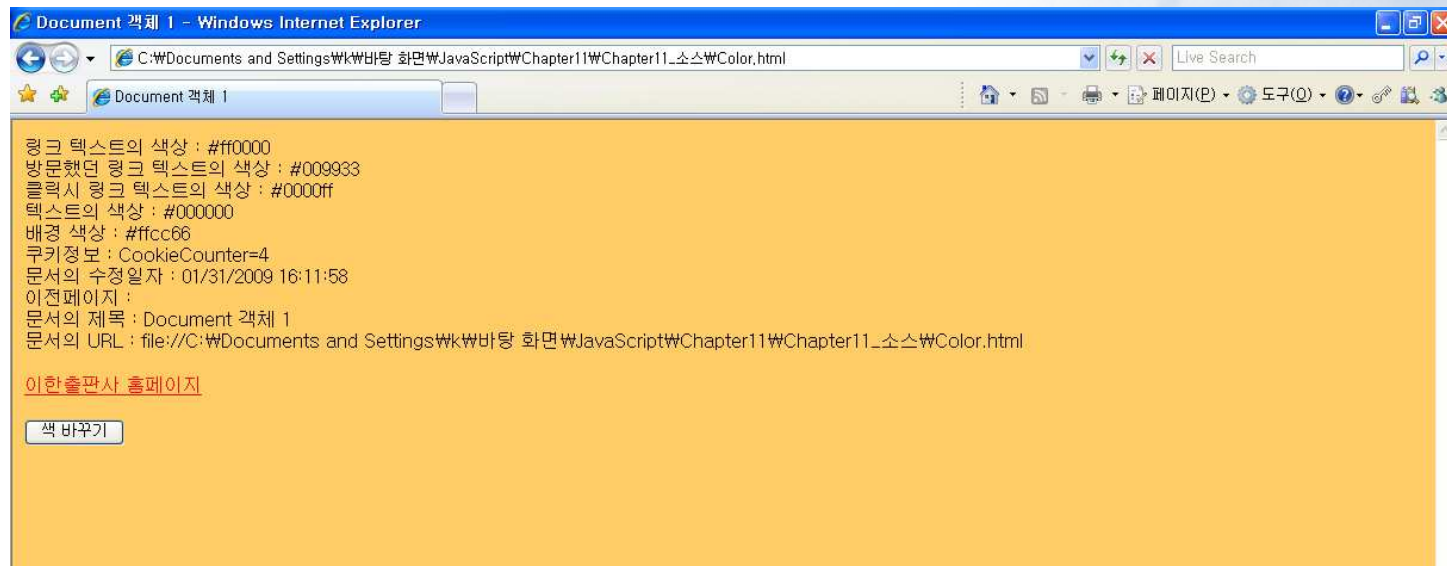
예제 :

```
<SCRIPT LANGUAGE="JavaScript">
<!--
    document.write("링크 텍스트의 색상 : "
        + document.linkColor + "<BR>");
    document.write("방문했던 링크 텍스트의 색상 : "
        + document.vlinkColor + "<BR>");
    document.write("클릭시 링크 텍스트의 색상 : "
        + document.alinkColor + "<BR>");
//-->
</SCRIPT>
```

프로그램 11-6

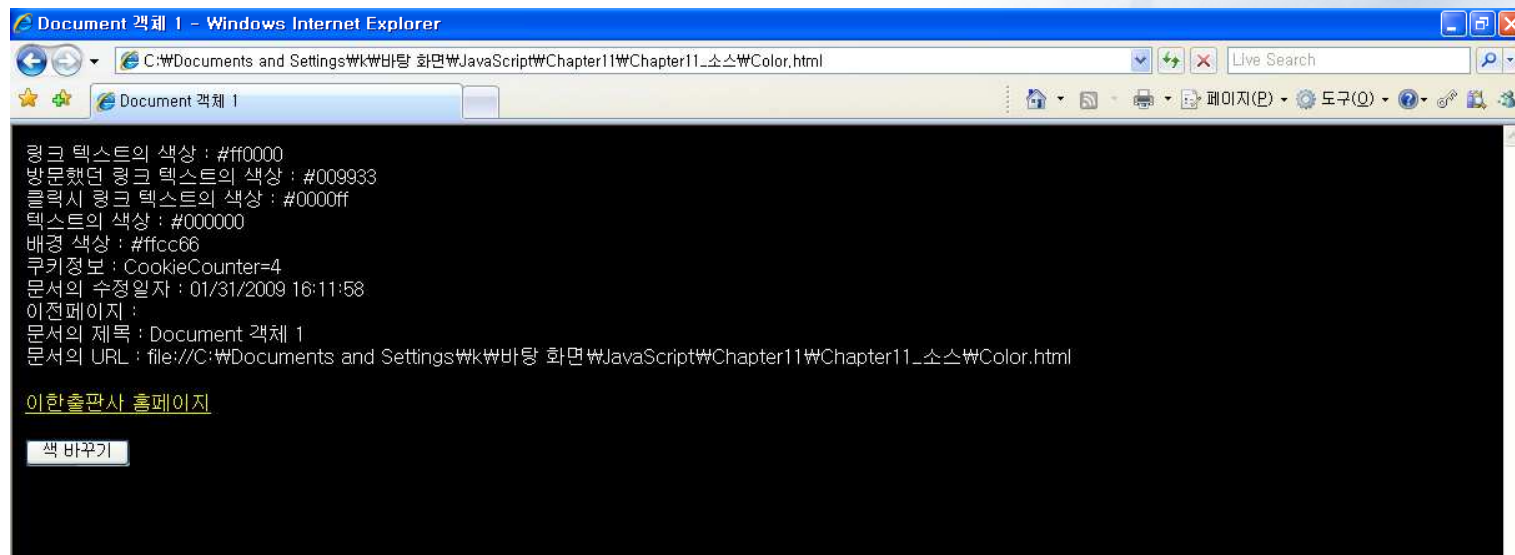
Chapter11

```
document.write("링크 텍스트의 색상 : "  
    + document.linkColor + "<BR>");  
document.write("방문했던 링크 텍스트의 색상 : "  
    + document.vlinkColor + "<BR>");  
document.write("클릭시 링크 텍스트의 색상 : "  
    + document.alinkColor + "<BR>");  
document.write("텍스트의 색상 : "  
    + document.fgColor + "<BR>");
```



[그림 11-13] Color의 실행결과

```
function myFunc() {
    document.fgColor = "white";
    document.linkColor = "yellow";
    document.vlinkColor = "silver";
    document.alinkColor = "red";
    document.bgColor = "black";
}
```



[그림 11-14] Color의 실행결과

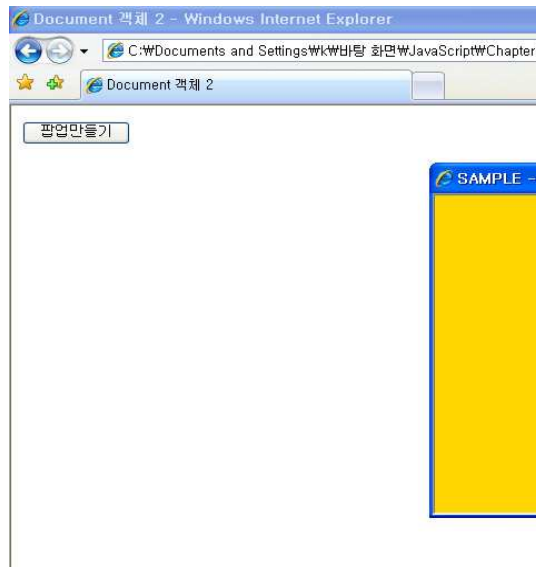
❖ open() 메소드, close() 메소드

- Window 객체의 open() 메소드와 close() 메소드는 문서 열고 닫을 때 사용하지만 Document 객체의 그것은 지정한 문서에 내용 입력을 준비하거나 마무리할 때 사용

❖ write() 메소드, writeln() 메소드

- write() 메소드는 지정한 문자열을 출력하지만 writeln() 메소드를 사용하면 지정한 문자열을 출력한 후 자동으로 줄을 바꾸어 준

```
myWin = window.open("", "popup_sample",
    "width=400,height=300");
myWin.document.open();
myWin.document.write
    ("<HTML><HEAD><TITLE>SAMPLE</TITLE></HEAD>");
myWin.document.write("<BODY bgcolor='gold'>");
myWin.document.write("<P ALIGN='center'>
    <B>안녕하세요? " + name + "님 </B></P>");
myWin.document.write("</BODY></HTML>");
myWin.document.close();
```



[그림 11-15] Popup의 실행결과

❖ 쿠키란?

- 이름이 지정된 아주 적은 양의 데이터로서 사용자의 웹 브라우저에 저장되며 특정한 웹 사이트나 웹 페이지와 결합되어 사용
- 쿠키의 특징
 - 하나의 웹 브라우저에 쿠키를 **300개** 이상 보관할 수 없음
 - 동일 웹서버에서 쿠키를 **20개** 이상 보관할 수 없음
 - 각 쿠키의 크기는 **4KByte**를 넘지 못함
 - **domain** 속성을 특별히 지정하지 않으면 쿠키를 설정한 문서보다 상위 단계에서는 호출 될 수 없음

❖ setCookie() 메소드, getCookie() 메소드

- 쿠키를 설정하고 해당 값을 가져 올 때 사용

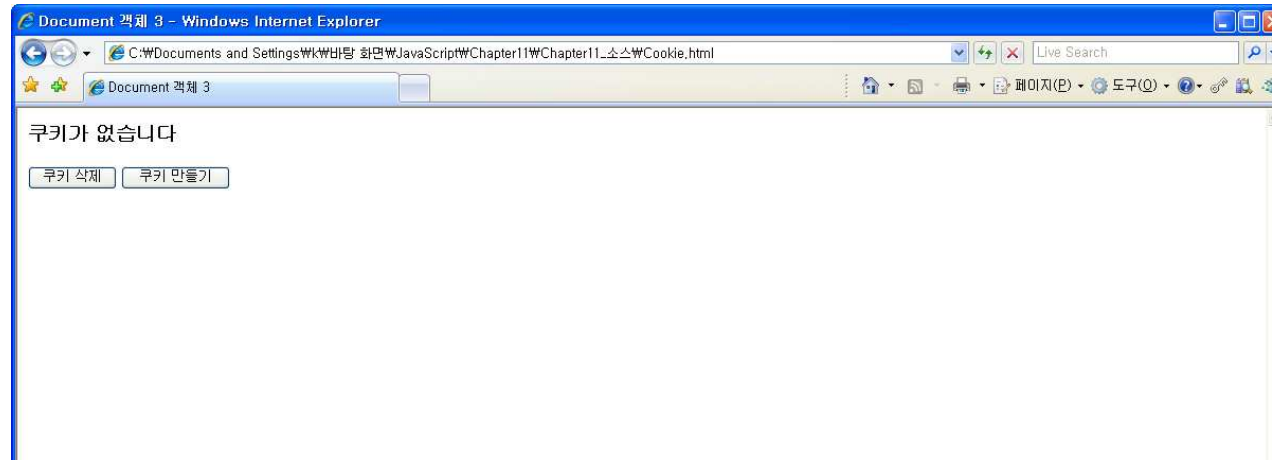
형식 :

```
setCookie(name, value, [expires], [path], [domain], [secure])
```

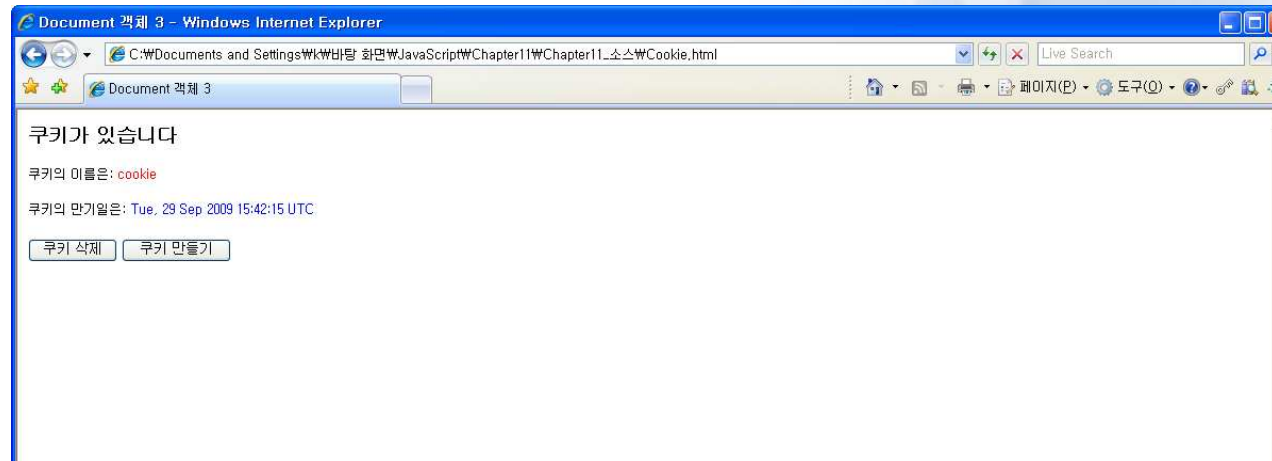
예제 :

```
<SCRIPT LANGUAGE="JavaScript">  
<!--  
    setCookie('myCookie', 111);  
//-->
```

설정	설명
name	쿠키에 저장하고자 하는 이름
value	name에 대한 값
expires	쿠키를 사용할 기간
path	쿠키를 사용할 웹 페이지
domain	쿠키를 사용할 도메인
secure	브라우저와 서버가 https 또는 다른 보안 프로토콜을 통해 연결될 때만 전송됨



[그림 11-16] Cookie의 실행결과



[그림 11-17] Cookie의 실행결과

프로그램 11-9, 프로그램 11-10

Chapter11



[그림 11-18] MemberInfo의 실행결과