



Chapter12

JavaScript 생일에 따른 별자리 이야기 페이지 만들기

HTML & JavaScript

1. History 객체

2. Location 객체

3. Frame 객체

4. Image 객체

5. Link 객체

6. Anchor 객체

7. Navigator 객체

8. 별자리 이야기 페이지 만들기

❖ History 객체

- Document 객체와 마찬가지로 Window 객체의 자식 객체
- 최근에 방문한 URL을 웹 브라우저가 저장해둔 히스토리 정보를 이용해서 앞, 뒤로 이동할 수 있음
- 한 가지 주의해야 할 점은 History 객체가 사용자가 방문했던 웹 사이트의 URL을 저장하고 있다고 해서 그 URL 자체를 가져올 수 없음
- Window 객체의 하위 객체이므로 History 객체를 사용할 때에는 앞에 창의 이름을 적는 것이 정석이지만 생략도 가능

속성	설명
length	브라우저의 history 목록에 저장된 URL 갯수

메소드	설명
back()	한 단계 전의 URL로 이동
forward()	한 단계 뒤 URL로 이동
go(숫자)	지정된 단계의 URL로 이동 (back()는 go(-1) 과 같음)

형식 :

history.속성

history.메소드()

속성 : length

속성 : back(), forward(), go()

예제 :

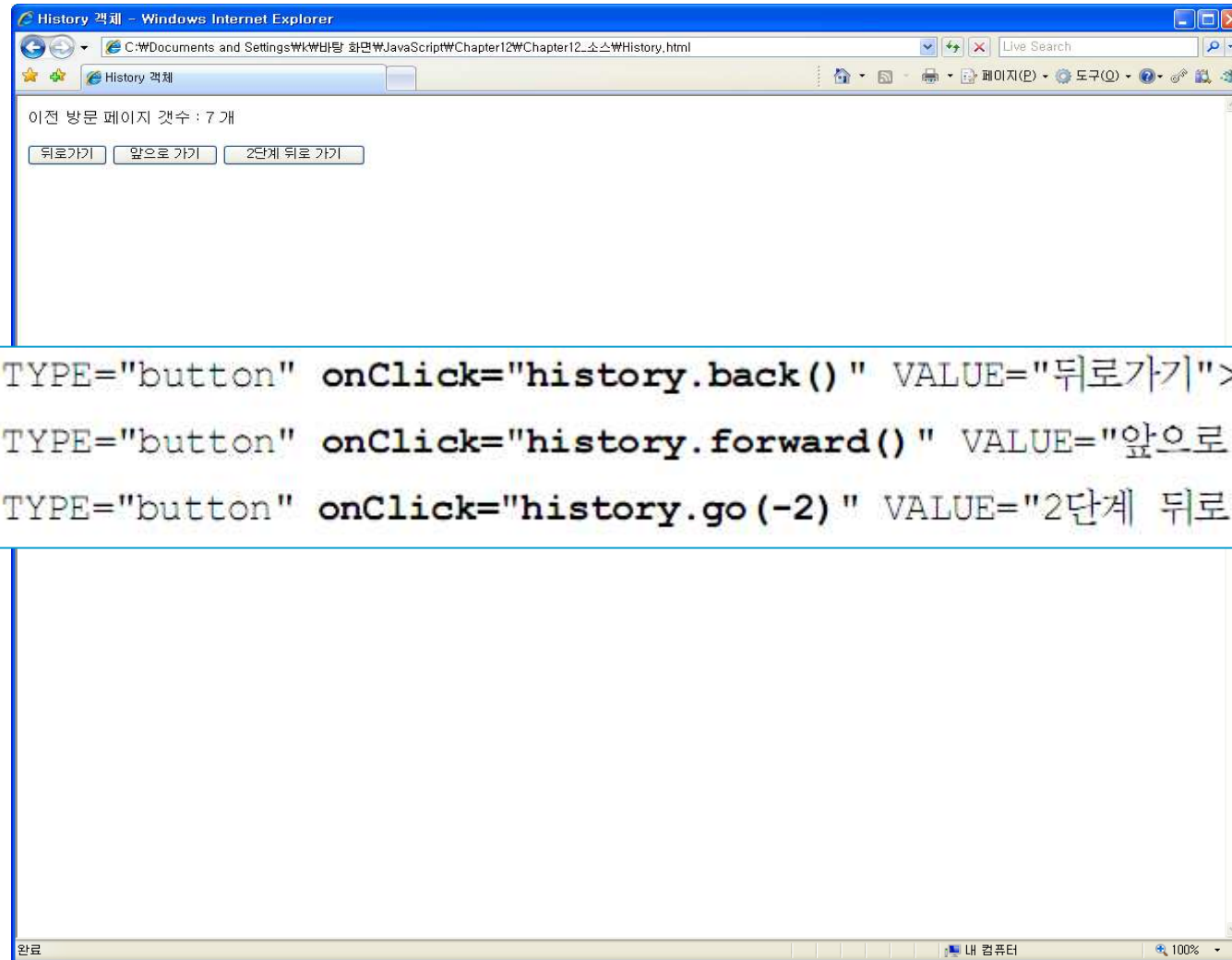
```
<SCRIPT LANGUAGE="JavaScript">
```

```
<!--
```

```
    document.write("이전 방문 페이지 갯수 : " + history.length + " 개");
```

```
//-->
```

```
</SCRIPT>
```



[그림 12-1] History의 실행결과

❖ Location 객체

- 현재 브라우저에 열려진 문서의 URL에 대한 정보를 제공하는 객체
- 주소 표시줄에 나오는 프로토콜 종류, 호스트 이름, 문서 위치 등을 처리하는데 사용

Location 객체의 속성, 메소드

Chapter12

속성	설명
hash	앵커 이름(하이퍼링크 # 이하)
host	호스트 이름과 포트
hostname	호스트 컴퓨터 이름
href	문서의 URL 주소
pathname	디렉토리 이하 경로
port	포트번호
protocol	프로토콜 종류
search	URL 쿼리정보 (? 이하)

메소드	설명
reload()	문서를 새로 읽어 들인다. 새로고침
replace()	현재 URL을 새로운 URL로 바꿈

형식 :

`location.속성`

`location.메소드()`

속성 : `hash, host, hostname, href, pathname, port, protocol, search`

속성 : `reload(), replace()`

예제 :

```
<SCRIPT LANGUAGE="JavaScript">
```

```
<!--
```

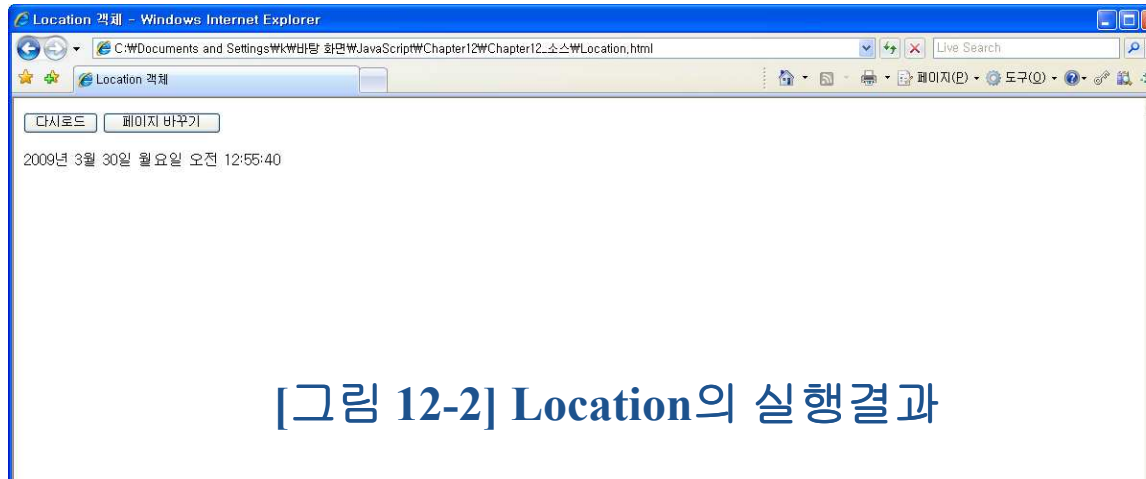
```
    document.write("HREF : " + location.href + "<BR>");
```

```
    document.write("PROTOCOL : " + location.protocol + "<BR>");
```

```
    document.write("HOST : " + location.host);
```

```
//-->
```

```
</SCRIPT>
```



[그림 12-2] Location의 실행결과



```
<INPUT TYPE="button" VALUE="다시로드"
onClick="location.reload()">
<INPUT TYPE="button" VALUE="페이지 바꾸기"
onClick="location.replace('http://www.ehan.co.kr')">
```

[그림 12-3] Location의 실행결과

❖ Frame 객체

- Window 객체의 자식 객체로 브라우저 내의 모든 프레임에 대한 정보를 가지고 있음
- 여러 개의 프레임이 있을 때 `frames[]` 배열 형식으로 각 프레임에 접근할 수 있음
- `length` 속성으로 창이 몇 개의 프레임으로 구성되어 있는지 알아낼 수 있음
- 프레임 자체가 창을 의미하므로 **Frame** 객체는 **Window** 객체의 속성, 메소드와 동일

속성	설명
closed	창이 닫혀 있는 상태
defaultStatus	브라우저의 상태바의 초기 문자열
length	창 안의 프레임의 수
name	창 이름
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

메소드	설명
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 :

타겟(top,parent,self).frames[인덱스 번호].속성

타겟(top,parent,self).frames[인덱스 번호].메소드

타겟(top,parent,self).프레임이름.속성

타겟(top,parent,self).프레임이름.메소드

속성 : closed, defaultStatus, length, name, parent, self, status, top

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

예제 :

```
<SCRIPT LANGUAGE="JavaScript">
```

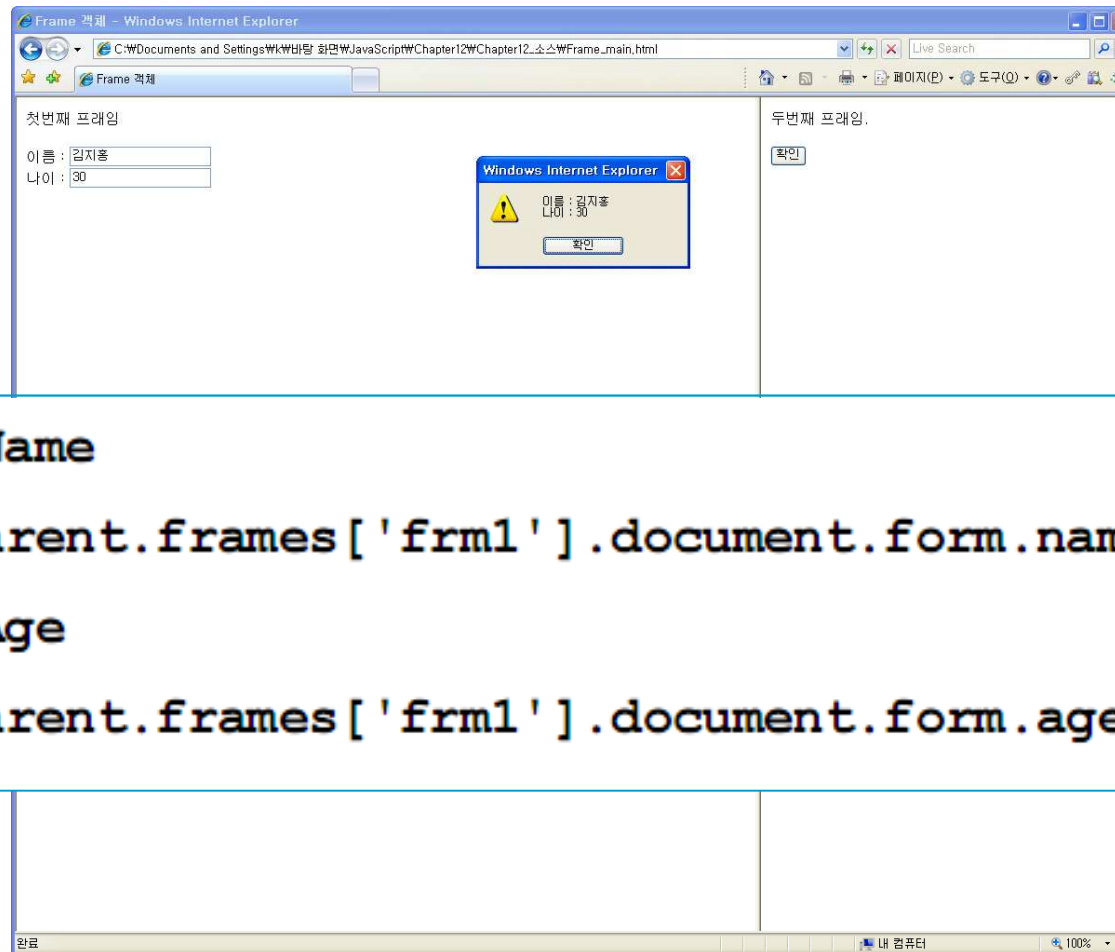
```
<!--
```

```
    parent.myframe.defaultStatus = "Frame 객체";
```

```
    document.write(self.frameRate);
```

```
//-->
```

```
</SCRIPT>
```

```
var strName
    = parent.frames[ 'frm1' ].document.form.name.value;
var strAge
    = parent.frames[ 'frm1' ].document.form.age.value;
```

[그림 12-4] Frame_main의 실행결과

❖ Image 객체

- Document 자식 객체로, 문서 내에 있는 이미지에 대한 정보를 담고 있음
- `` 과 같은 태그에 의해 생성되는 객체
- 기존 `` 태그의 모든 속성을 설정하거나 가져올 수 있음
- 실제 사용 시에는 문서 내의 이미지에 대한 설정을 변경하고자 할 때 주로 사용
- `` 태그의 `NAME` 속성의 값을 지정해야 자바 스크립트에서 접근을 할 수 있음
- `images[]`와 같은 배열 형식으로도 접근도 가능

속성	설명
border	외곽선 굵기를 설정/반환
complete	이미지 로드가 완료되었는지 여부
height	높이를 설정/반환
width	너비를 설정/반환
hspace	좌측 여백 값을 설정/반환
vspace	상단 여백 값을 설정/반환
length	높이를 설정/반환
align	정렬방식을 설정/반환
name	이미지 이름 설정/반환
src	이미지 파일 경로 설정/반환

형식 :

`document.이미지 이름.속성`

`document.images[인덱스 번호].속성`

속성 :

`border, complete, height, width, hspace, vspace, length, length, align, name, src`

예제 :

```
<SCRIPT LANGUAGE="JavaScript">
```

```
<!--
```

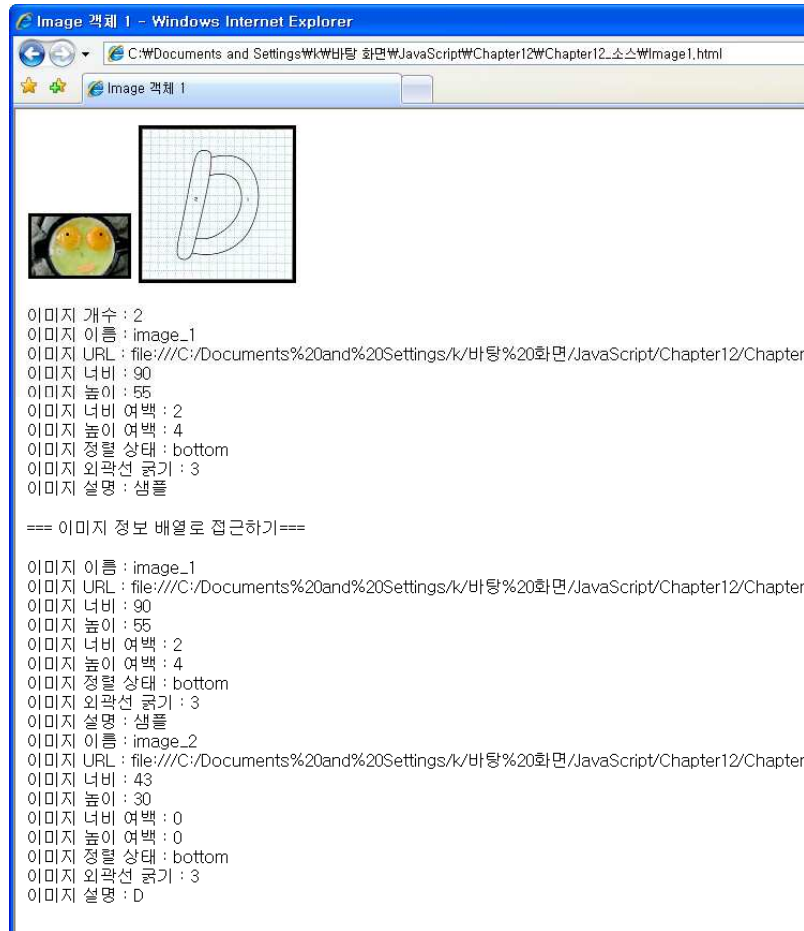
```
document.write("이미지 개수 : " + document.images.length + "<BR>");
```

```
document.write("이미지 이름 : " + document.images.name + "<BR>");
```

```
document.write("이미지 URL : " + document.images.src + "<BR>");
```

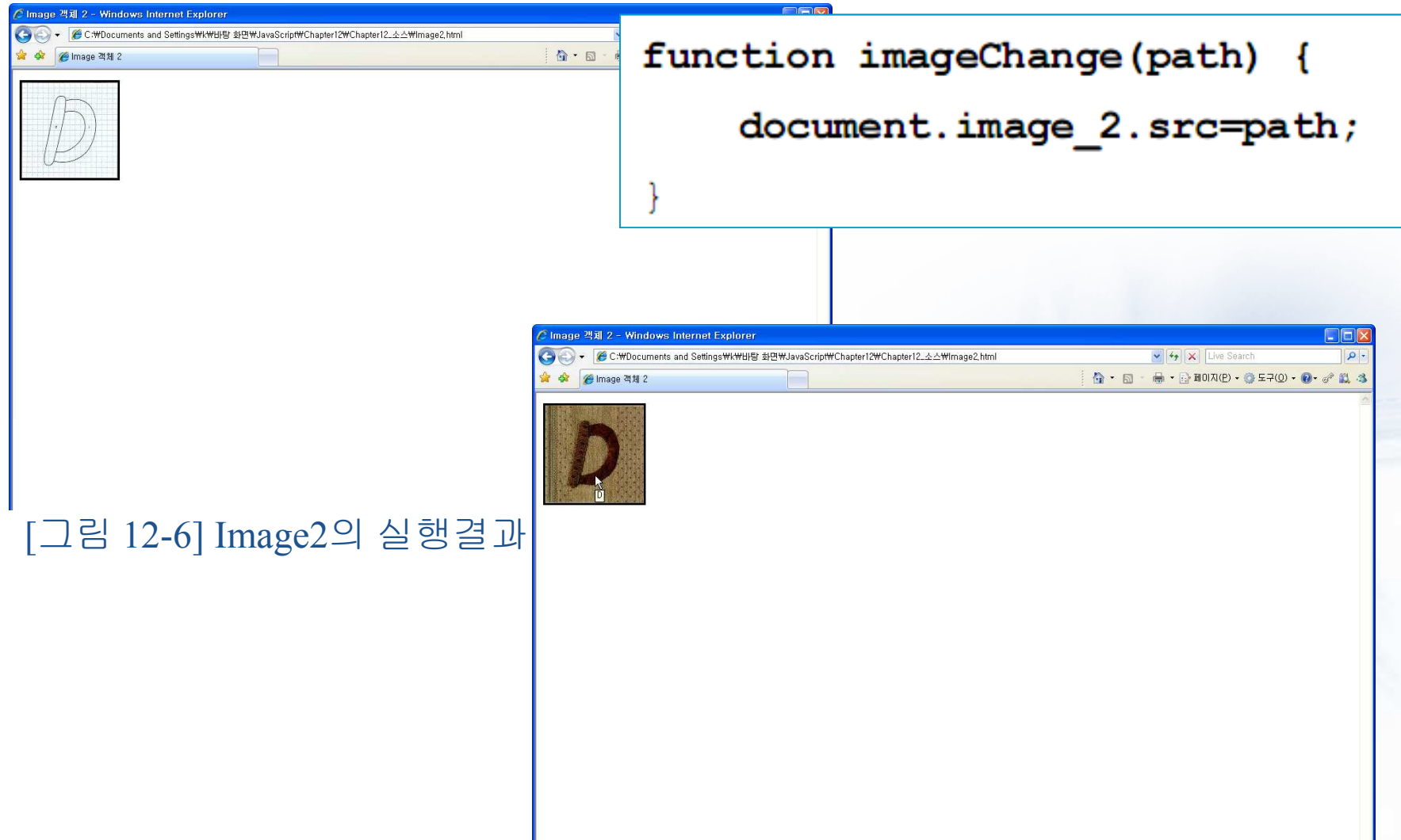
```
//-->
```

```
</SCRIPT>
```



```
document.write("이미지 개수 : "
    + document.images.length + "<BR>");
document.write("이미지 이름 : "
    + document.image_1.name + "<BR>");
document.write("이미지 URL : "
    + document.image_1.src + "<BR>");
document.write("이미지 너비 : "
    + document.image_1.width + "<BR>");
```

[그림 12-5] Image1의 실행결과



[그림 12-6] Image2의 실행결과

[그림 12-7] Image2의 실행결과

❖ Link 객체

- Image 객체와 마찬가지로 Document 객체의 하위에 위치해 있는 객체로 문서에 들어있는 텍스트나 이미지의 링크 정보를 담고 있음
- <A HREF> 태그의 정보를 쉽게 알 수 있으며 변경할 수 있음
- Link 객체들을 배열로 모아 놓았을 경우에는 links[] 를 사용

속성	설명
hash	표식 이름을 반환한다. URL의 앵커 부분을 의미한다.
host	링크에 연결된 hostname:port 를 반환한다
hostname	URL의 host부분
href	전체 URL
pathname	URL의 경로명 부분
port	URL의 포트 부분
protocol	URL의 프로토콜 부분
search	URL의 쿼리 스트링 부분
target	링크의 HTML, target 속성 (URL에 연결된 후 보여줄 창)

형식 :

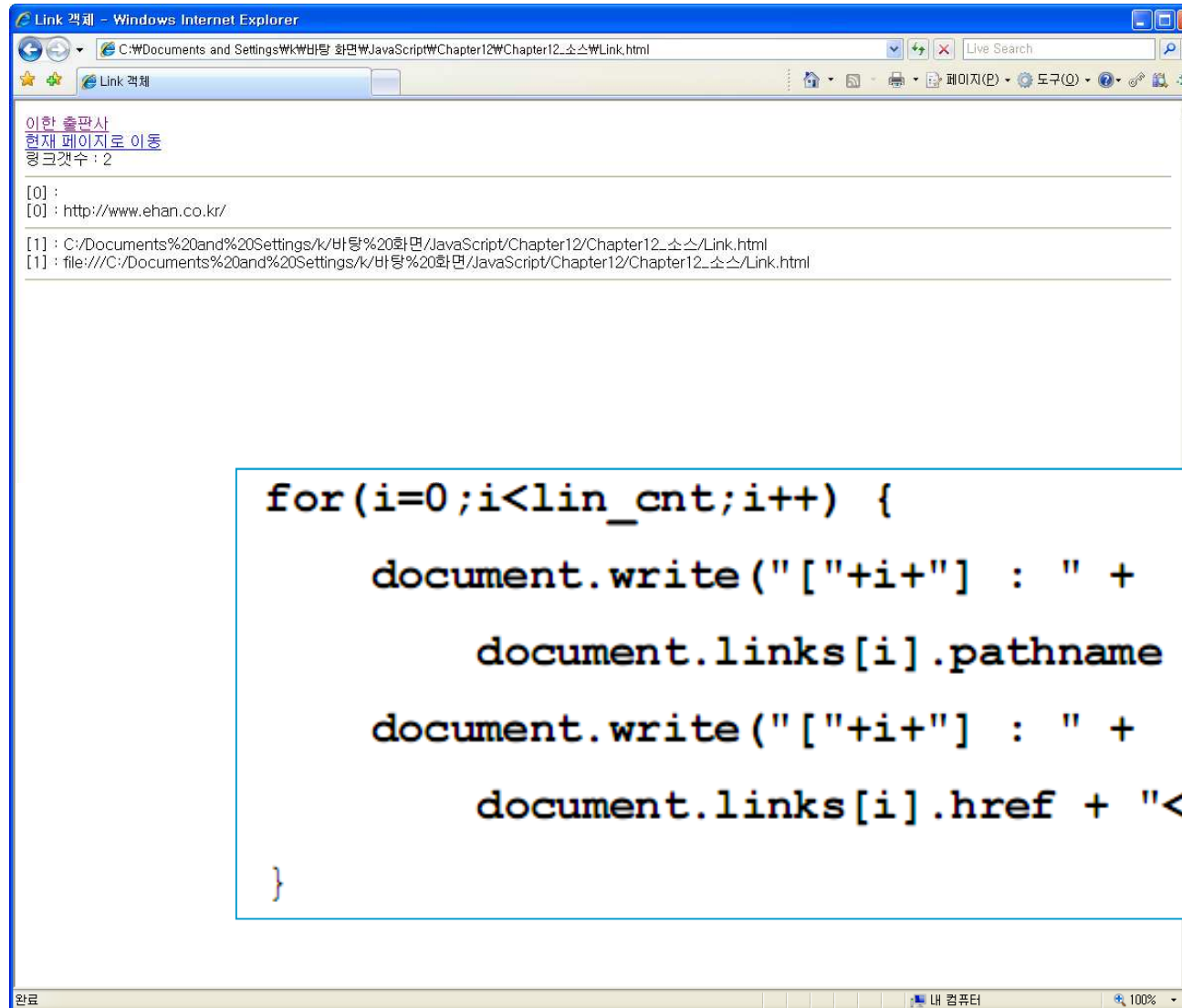
`document.links[인덱스 번호].속성`

속성 :

`hash, host, hostname, href, pathname, port, protocol, search, target`

예제 :

```
<SCRIPT LANGUAGE="JavaScript">
<!--
  document.write("Pathname : " + document.links[2].pathname + "<BR>");
  document.write("HREF : " + document.links[3].href);
//-->
</SCRIPT>
```



[그림 12-8] Link의 실행결과

❖ Anchor 객체

- Document 객체 하위에 위치해 있으며 문서에 있는 앵커(<A> 태그)의 정보를 담고 있음
- Link 객체와 마찬가지로 Anchor 객체들을 배열로 모아 놓은 `anchors[]`가 존재 함
- 속성이나 메소드를 포함하고 있지 않으며 만약 <A> 태그에 **HREF** 속성이 포함되어 있을 경우에는 Link 객체가 됨

형식 :

`document.앵커 이름.속성`

`document.anchors[인덱스 번호].속성`

예제 :

```
<SCRIPT LANGUAGE="JavaScript">
```

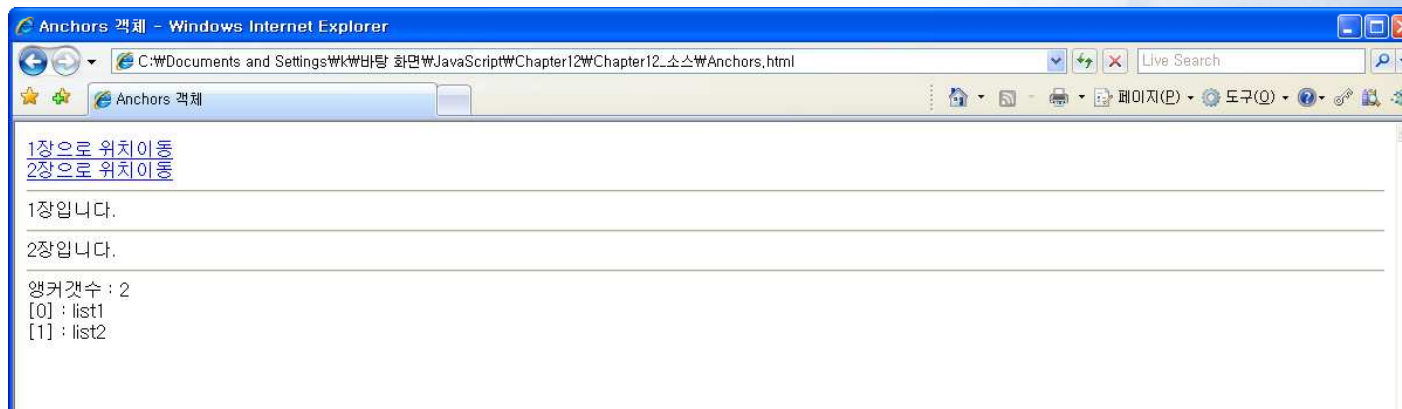
```
<!--
```

```
    document.write("Length : " + document.anchors[0].length + "<BR>");
```

```
//-->
```

```
</SCRIPT>
```

```
for(i=0;i<anch_cnt;i++) {  
    document.write ("["+i+"] : " +  
        document.anchors[i].name + "<BR>");  
}
```



[그림 12-9] Anchors의 실행결과

❖ Navigator 객체

- Window 객체와 마찬가지로 최상위 객체로서 웹 브라우저에 관련된 정보를 담고 있음
- 개발자가 고급 웹 사이트 제작 시 고려할 필요가 있을 뿐 실제로는 잘 사용하지 않음

속성	설명
appName	브라우저 코드 이름을 알려줌
appVersion	브라우저 이름을 알려줌
userAgent	브라우저의 버전 정보를 알려줌
platform	브라우저에서 서버로 전송된 HTTP 프로토콜의 사용자 에이전트 헤더
language	브라우저가 실행되는 운영체제 플랫폼
mimeType	브라우저가 사용하는 언어
plugins	브라우저에서 지원하는 MIME 타입
	브라우저에 설치된 플러그인

메소드	설명
javaEnabled()	사용자가 브라우저의 Java 기능을 켜두었는지의 여부

형식 :

`navigator.속성`

`navigator.메소드()`

속성 :

`appName, appName, appVersion, userAgent, platform, language, mimeType, plugins`

속성 : `javaEnabled()`

예제 :

```
<SCRIPT LANGUAGE="JavaScript">
```

```
<!--
```

```
    document.write("appName : " + navigator.appName + "<BR>");
```

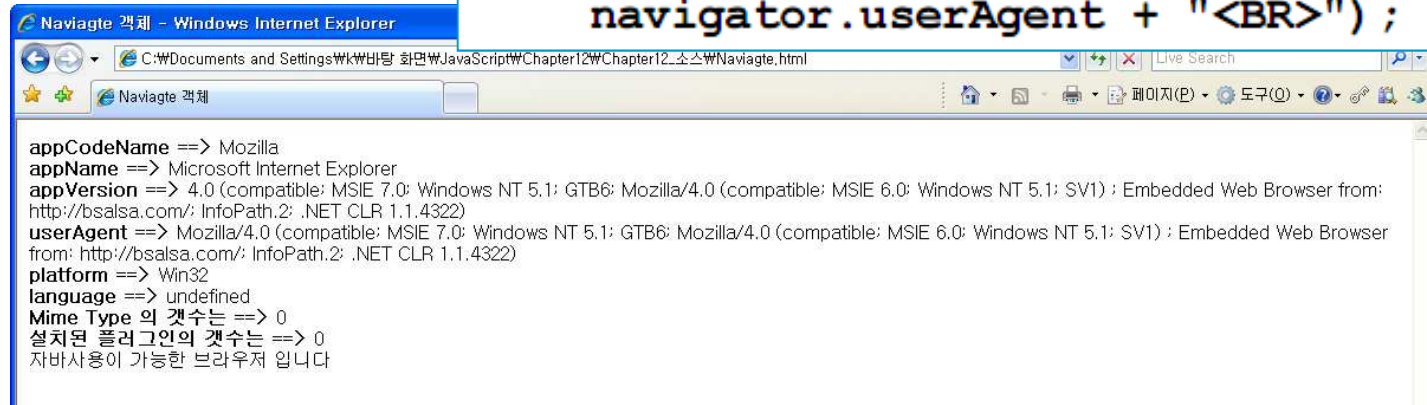
```
    document.write("appVersion : " + navigator.appVersion);
```

```
//-->
```

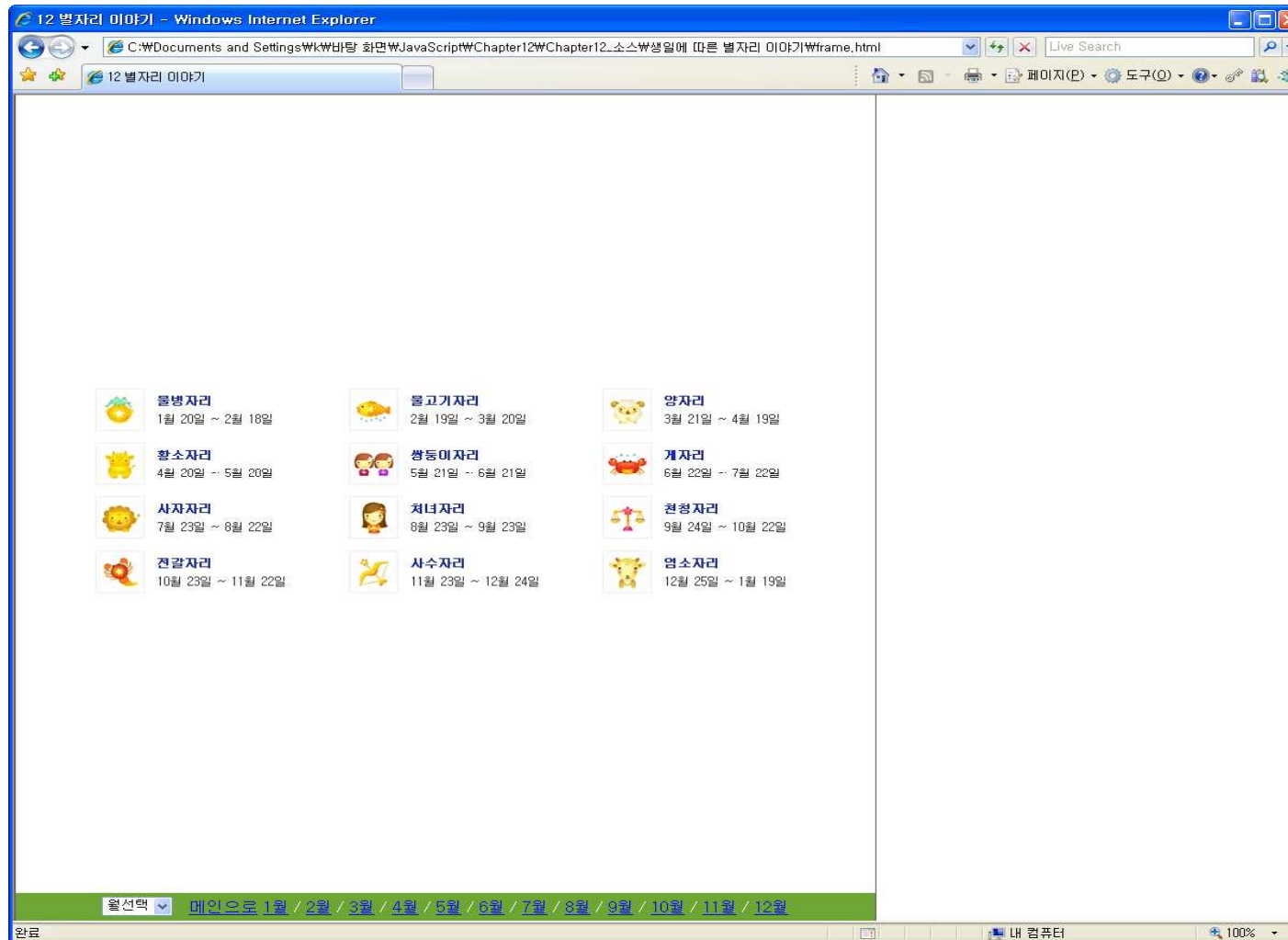
```
</SCRIPT>
```



```
document.write("<B> appCodeName ==> </B>" +
    navigator.appCodeName + "<BR>");
document.write("<B> appName ==> </B>" +
    navigator.appName + "<BR>");
document.write("<B> appVersion ==> </B>" +
    navigator.appVersion + "<BR>");
document.write("<B> userAgent ==> </B>" +
    navigator.userAgent + "<BR>");
```



[그림 12-10] Naviagte의 실행결과



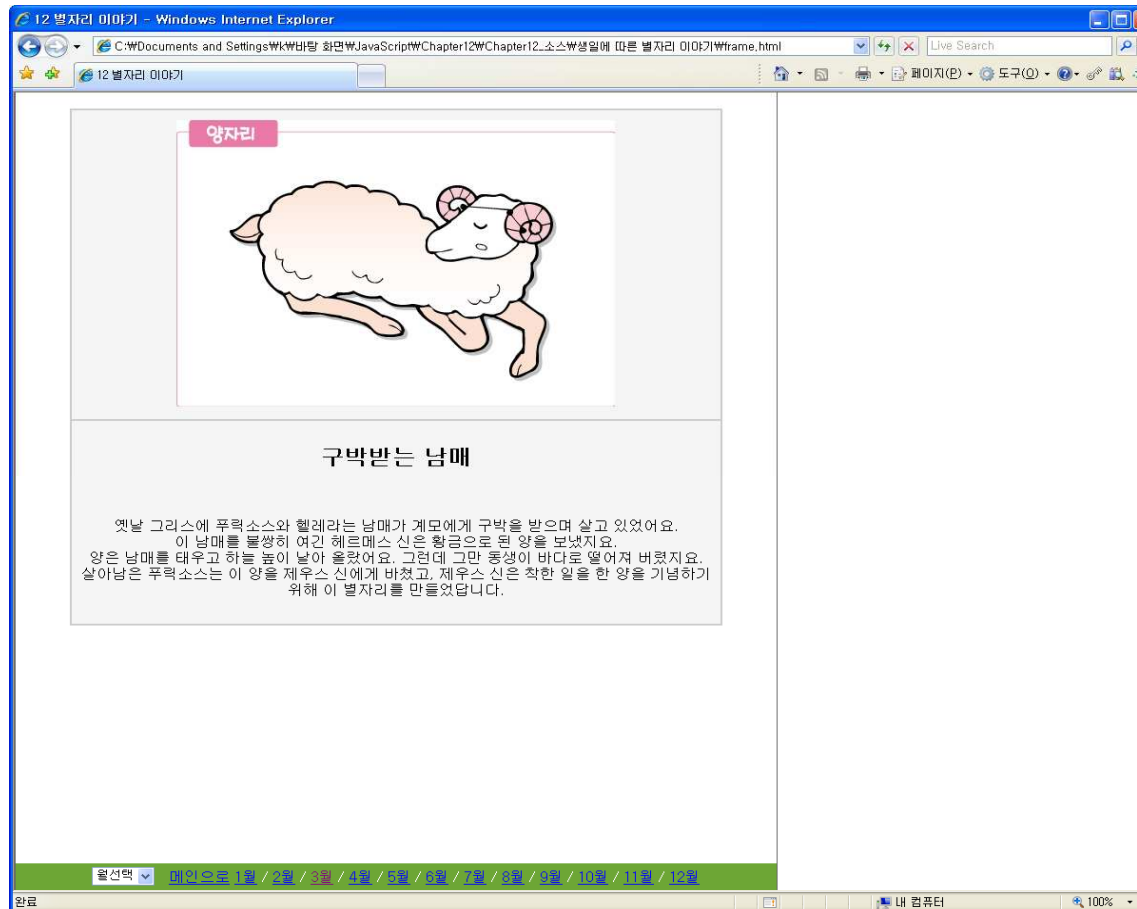
[그림 12-11] frame의 실행결과



[그림 12-12] popup의 실행결과



[그림 12-13] remote의 실행결과



[그림 12-14] frame의 실행결과