



CHAPTER 10.

DOM과 이벤트 처리, 입력 검증



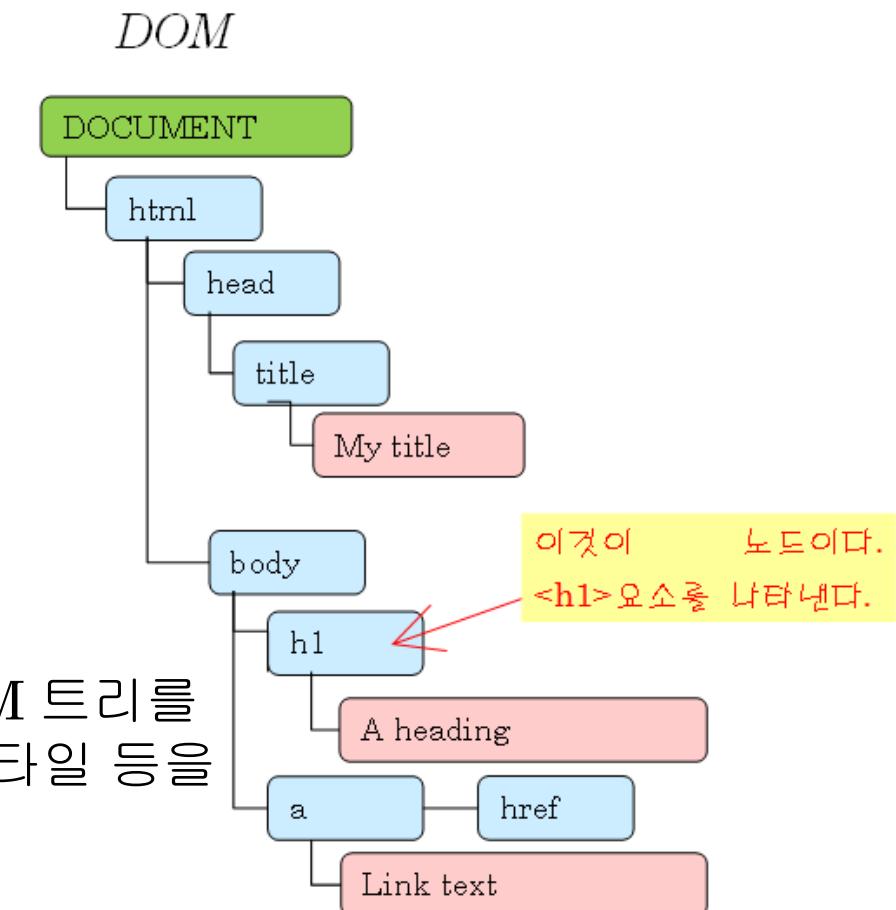


문서 객체 모델(DOM)

- Document Object Model

- DOM은 HTML 문서의 계층적인 구조를 트리(tree)로 표현
- 노드
 - 트리의 개별 구성요소
 - 문서 안의 요소나 텍스트

```
<html>
  <head>
    <title>My title</title>
  </head>
  <body>
    <h1>A heading</h1>
    <a href="#">Link text</a>
  </body>
</html>
```

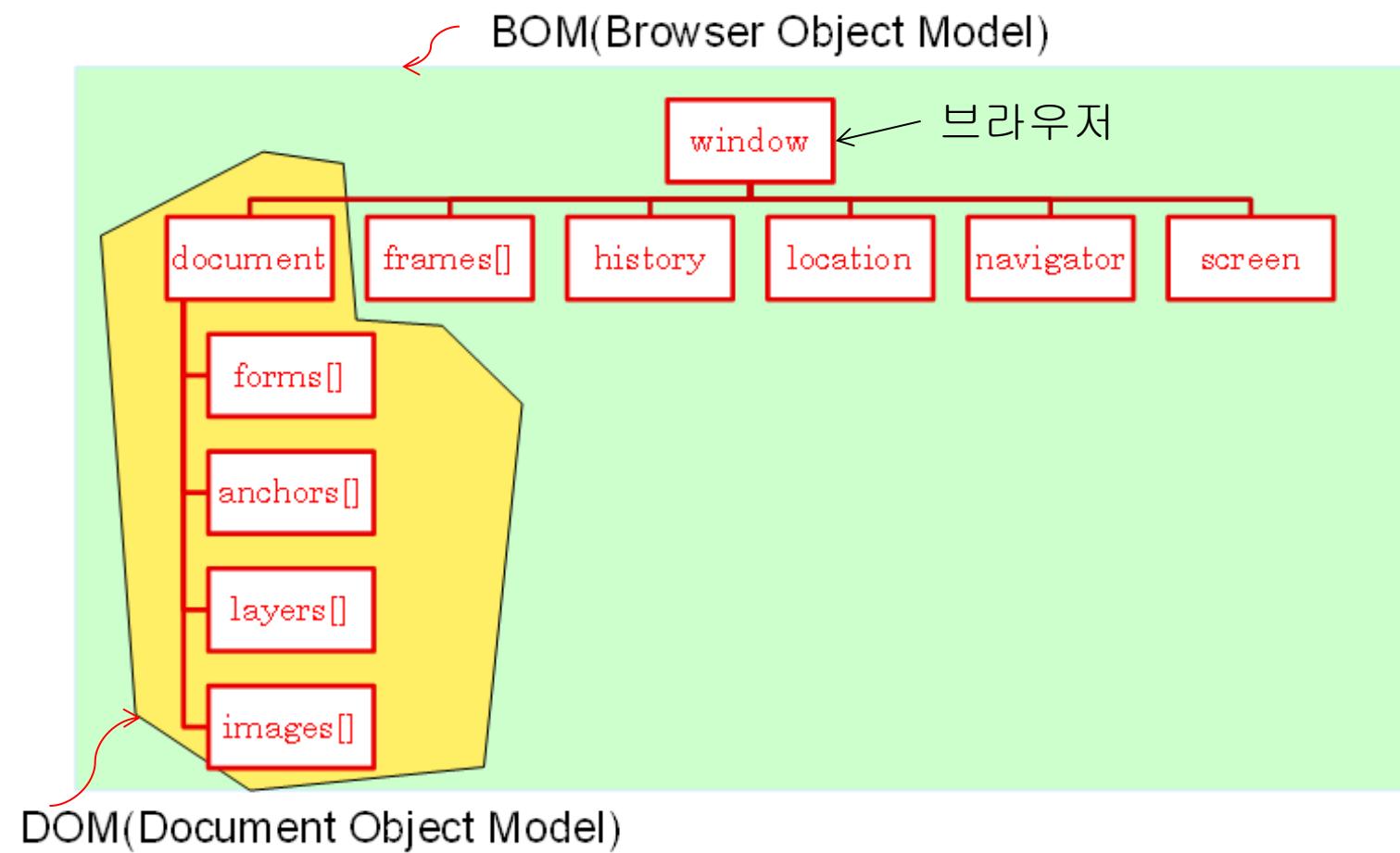


- 자바스크립트를 이용하여 DOM 트리를 순회하며 요소의 속성, CSS 스타일 등을 변경 가능



DOM과 BOM

- HTML 문서를 객체로 표현한 것을 DOM
- 웹브라우저를 객체로 표현한 것을 BOM(Browser Object Model)





HTML 요소 찾기

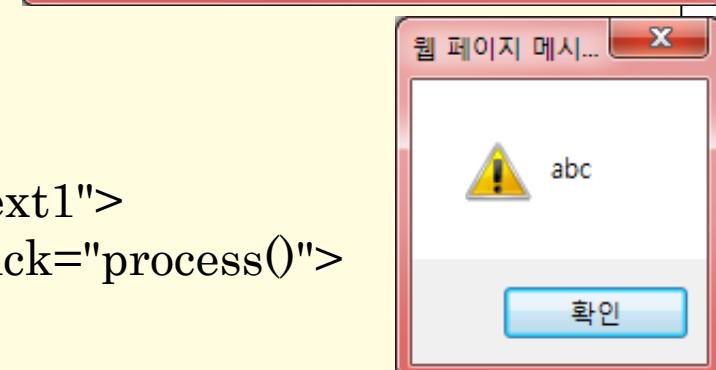
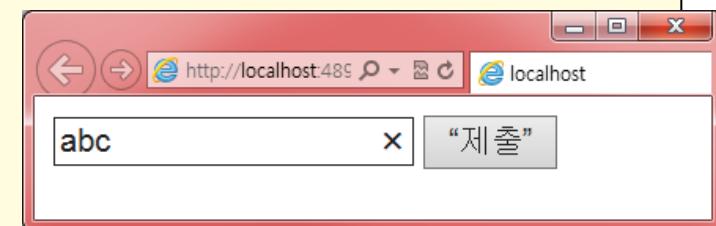
- 동적인 웹페이지를 작성하려면 원하는 요소를 찾아야 한다.
 - id로 찾기
 - 입력 양식 찾기
 - 태그 이름으로 찾기



id로 HTML 요소 찾기

```
<!DOCTYPE >
<html>

<head>
    <script>
        function process() {
            var obj = document.getElementById("target");
            alert(obj.value);
        }
    </script>
</head>
<body>
    <form name="myform">
        <input type="text" id="target" name="text1">
        <input type="submit" value="제출" onclick="process()">
    </form>
</body>
</html>
```

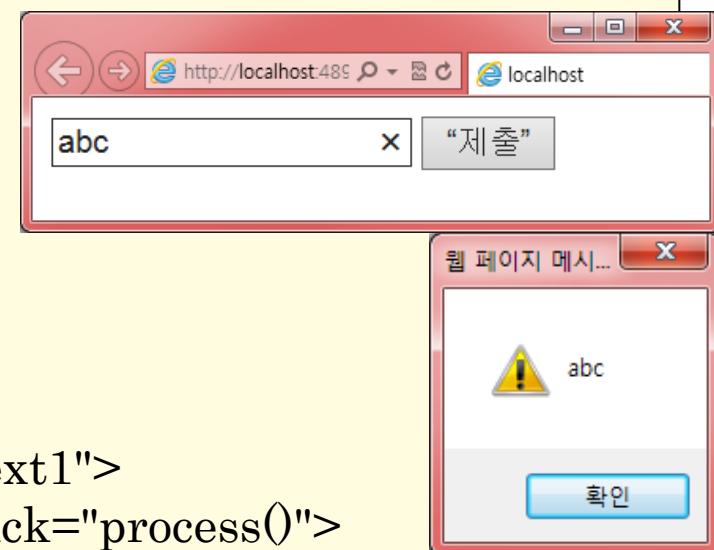




입력 양식 찾기

```
<!DOCTYPE >
<html>

<head>
    <script>
        function process() {
            var obj = document.myform.text1;
            alert(obj.value);
        }
    </script>
</head>
<body>
    <form name="myform">
        <input type="text" id="target" name="text1">
        <input type="submit" value="제출" onclick="process()">
    </form>
</body>
</html>
```

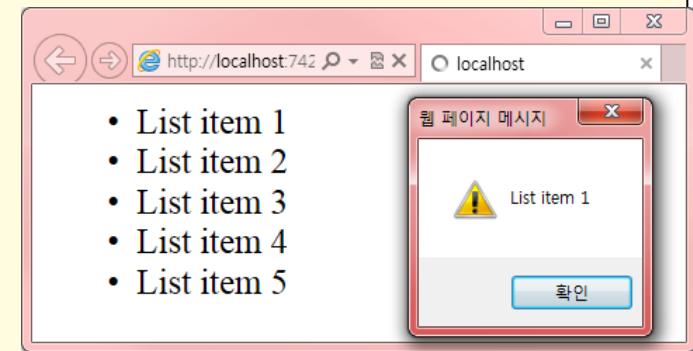




태그 이름으로 HTML 요소 찾기

```
<!DOCTYPE >
<html>
<body>
<ul>
    <li>List item 1</li>
    <li>List item 2</li>
    <li>List item 3</li>
    <li>List item 4</li>
    <li>List item 5</li>
</ul>
<script>
    var list = document.getElementsByTagName('ul')[0];
    var allItems = list.getElementsByTagName('li');

    for (var i = 0, length = allItems.length; i < length; i++) {
        alert(allItems[i].firstChild.data);
    }
</script>
</body>
</html>
```



[chap10_get_elem_tag.html](#)



DOM 트리 순회

- childNodes: 한 요소의 모든 자식 요소에 접근할 수 있다. 배열이 반환된다.
- firstChild: 'childNodes' 배열의 첫 번째 자식 노드가 반환된다. 'childNodes[0]'와 같다.
- lastChild: 'childNodes' 배열의 마지막 자식 노드가 반환된다. 'childNodes[childNodes.length-1]'와 같다.
- parentNode: 현재 노드의 부모 노드를 반환한다.
- nextSibling: 현재 노드의 다음 형제 노드를 반환한다.
- previousSibling: 현재 노드의 이전 형제 노드를 반환한다.



HTML 요소의 내용 변경

```
<!DOCTYPE html>
<html>
<head>
    <title></title>
    <script>
        function get() {
            var val = document.getElementById("ex").innerHTML;
            alert(val);
        }
        function set(v) {
            document.getElementById("ex").innerHTML = v;
        }
    </script>
</head>
<body>
    <div id="ex">여기가 div로 선언되었습니다.</div>
    <a href="#" onclick="get()">div 요소 내용 출력하기</a><br />
    <a href="#" onclick="set('변경되었습니다.')">div 요소 내용 변경하기</a>
</body>
</html>
```

innerHTML : 요소의 시작
태그와 종료 태그 사이에 놓여진
모든 HTML 코드와 텍스트



실행결과

The screenshot shows a web browser window with the URL `http://localhost:742`. The page content is as follows:

```
여기가 div로 선언되었습니다.  
div 요소 내용 출력하기  
div 요소 내용 변경하기
```

A red arrow points from the text "여기가 div로 선언되었습니다." to a confirmation dialog box titled "웹 페이지 메시지". The dialog box contains the message "여기가 div로 선언되었습니다." and a "확인" button.

In the second screenshot, the page content has been updated:

```
변경되었습니다.  
div 요소 내용 출력하기  
div 요소 내용 변경하기
```

A red arrow points from the text "변경되었습니다." to the same confirmation dialog box.

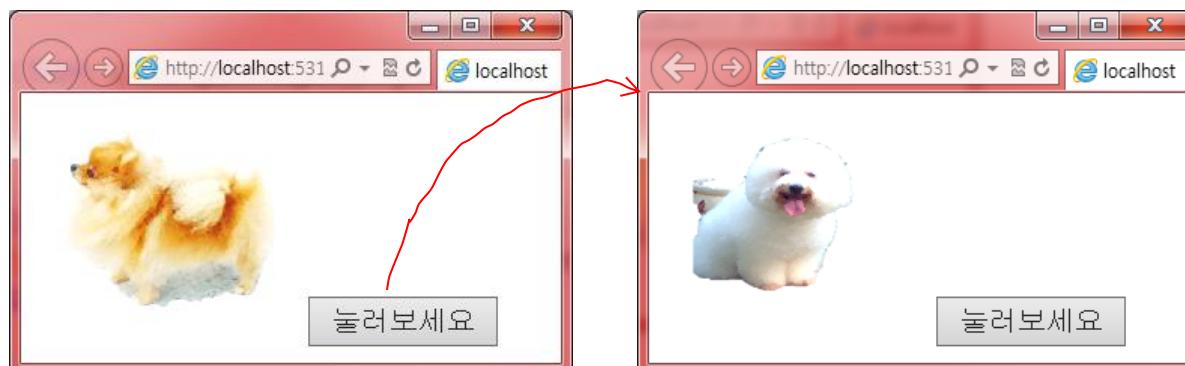
실행(클릭)



요소의 속성 변경하기

```
<!DOCTYPE html>
<html>
<body>
  
  <script>
    function changeImage() {
      document.getElementById("image").src = "poodle.png";
    }
  </script>
  <input type="button" onclick="changeImage()" value="눌러보세요" />
</body>
</html>
```

실행(클릭)

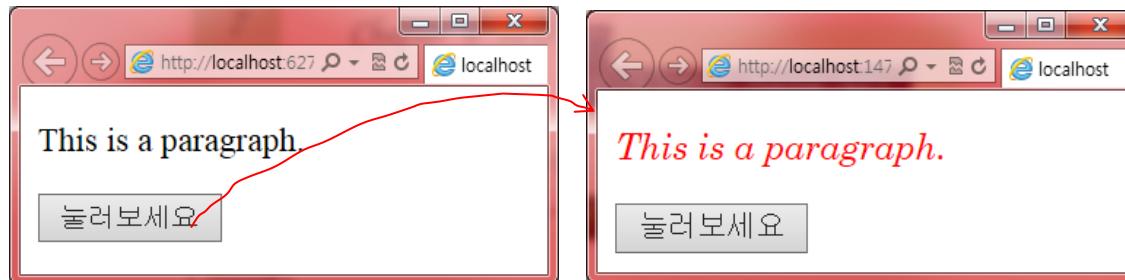




요소의 스타일 변경하기

```
<!DOCTYPE html>
<html>
<body>
<p id="p1">This is a paragraph.</p>
<script>
    function changeStyle() {
        document.getElementById("p1").style.color = "red";
        document.getElementById("p1").style.fontFamily = "Century
Schoolbook";
        document.getElementById("p1").style.fontStyle = "italic";
    }
</script>
<input type="button" onclick="changeStyle()" value="눌러보세요" />
</body>
</html>
```

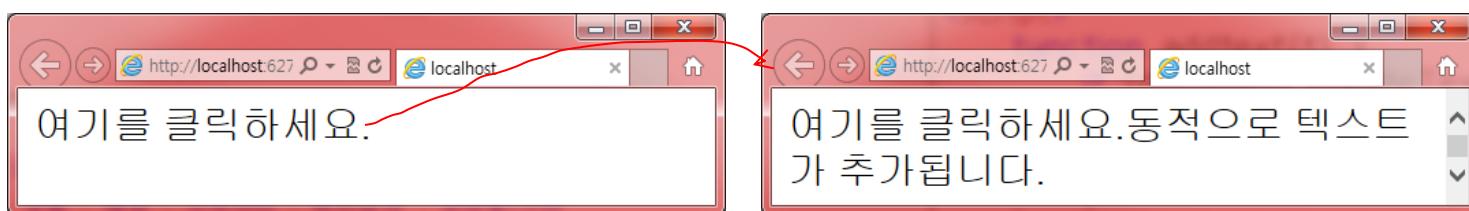
실행(클릭)





새로운 HTML 요소 생성

```
<script>
function addtext(t) {
    if (document.createTextNode) {
        var node = document.createTextNode(t);
        document.getElementById("target").appendChild(node);
    }
}
</script>
<div id="target" onclick="addtext('동적으로 텍스트가 추가됩니다.')"
style="font: 20px bold;">여기지를 클릭하세요.</div>
```

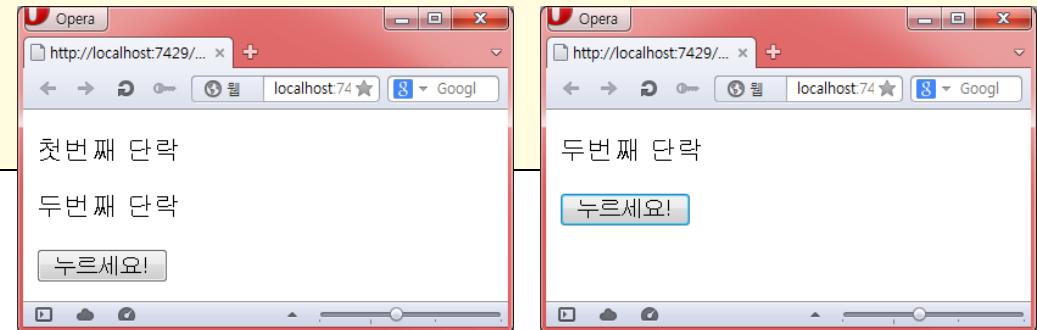


실행(클릭)



HTML 요소 삭제

```
<!DOCTYPE html>
<html>
  <head>
    <script>
      function removeNode() {
        var parent = document.getElementById("target");
        var child = document.getElementById("p1");
        parent.removeChild(child);
      }
    </script>
  </head>
  <body>
    <div id="target">
      <p id="p1">첫번째 단락</p>
      <p id="p2">두번째 단락</p>
    </div>
    <button onclick="removeNode()">누르세요!</button>
  </body>
</html>
```

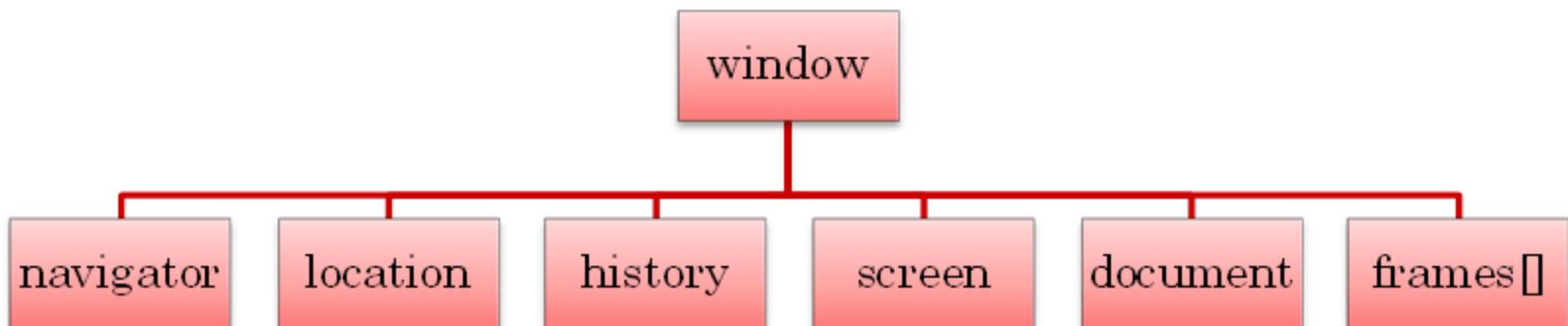


주의: 부모 노드를 알아야
자식 노드 삭제 가능



브라우저 객체 모델

- 브라우저 객체 모델(BOM: Browser Object Model): 웹 브라우저가 가지고 있는 모든 객체를 의미
- 최상위 객체는 window이고 그 아래로 navigator, location, history, screen, document, frames 객체가 있다.

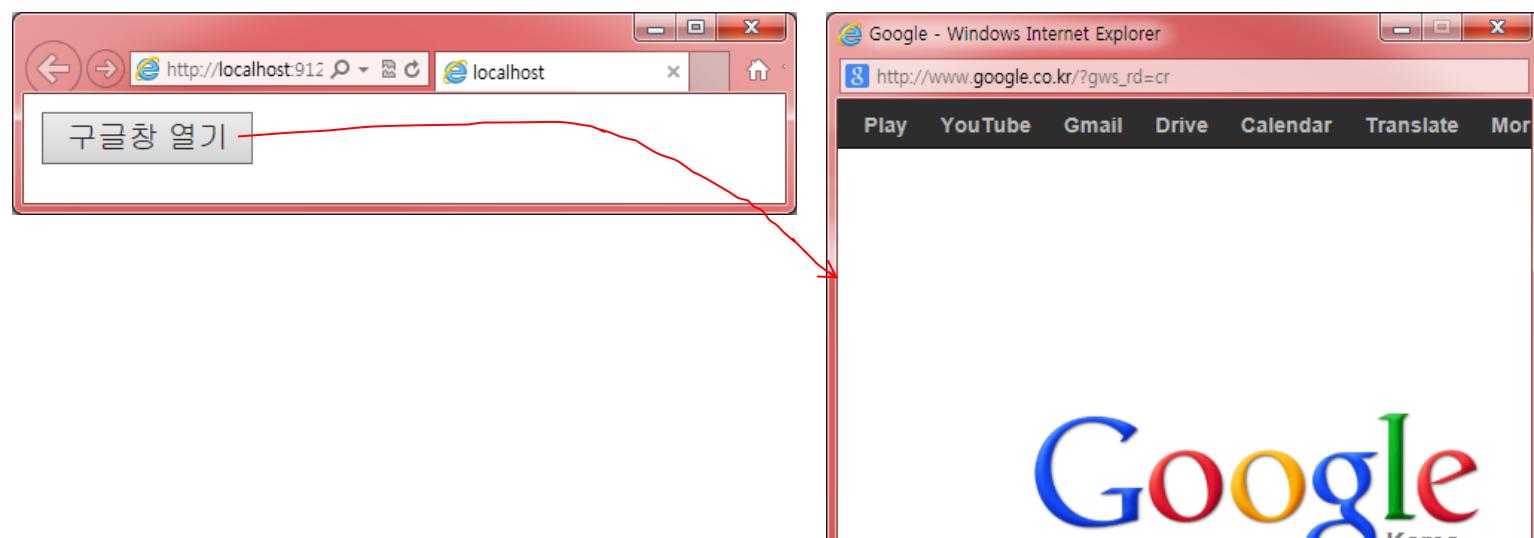


교재 p. 392 참조



새로운 윈도우 오픈 예제

```
<!DOCTYPE html>
<html>
<head></head>
<body>
<form>
    <input type="button" value="구글창 열기"
        onclick="window.open('http://www.google.com', '_blank', 'width=300,
height=300', true)">
</form>
</body>
</html>
```



Google
Korea



setTimeout()

setTimeout(code, millisec)

호출되는 함수의 이름, 호출되는 함수를 여기서 직접 정의할 수도 있다.

함수를 호출하기 전에 흘려야 하는 시간

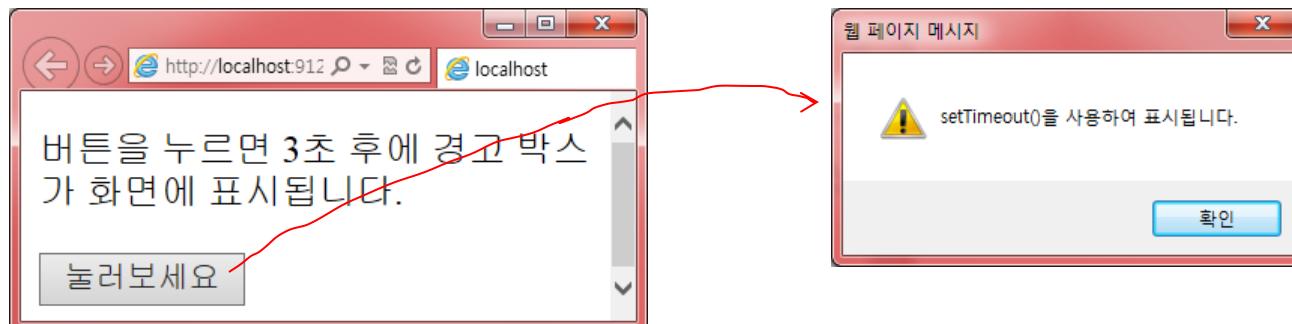
*setTimeout()*은 지정된 시간이 지난 후에 한번만 호출한다.





예제

```
<!DOCTYPE html>
<html>
<head>
    <script>
        function showAlert() {
            setTimeout(function () { alert("setTimeout()을 사용하여
표시됩니다."), 3000);
        }
    </script>
</head>
<body>
    <p>버튼을 누르면 3초 후에 경고 박스가 화면에 표시됩니다. </p>
    <button onclick="showAlert()">눌러보세요</button>
</body>
</html>
```





setInterval()

setInterval()은
주기적으로
호출 한다.





예제

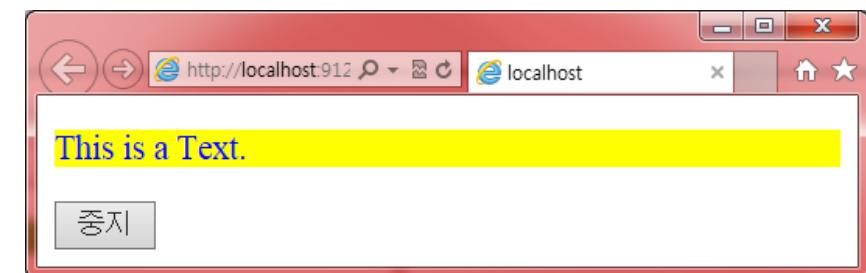
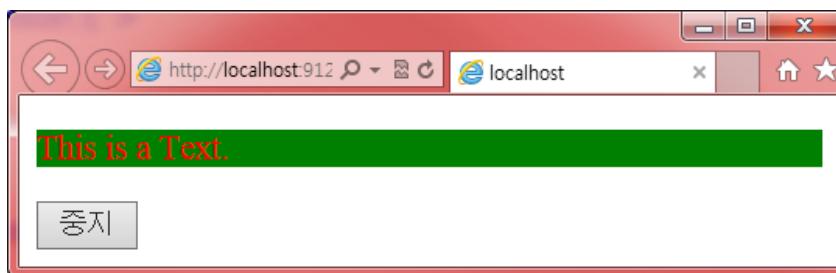
```
<!DOCTYPE html>
<html>
<head>

<script>
    var id;
    function changeColor() {
        id = setInterval(flashText, 500);
    }
    function flashText() {
        var elem = document.getElementById("target");
        elem.style.color = (elem.style.color == "red") ? "blue" : "red";
        elem.style.backgroundColor =
            (elem.style.backgroundColor == "green") ? "yellow" :
"green";
    }
    function stopTextColor() {
        clearInterval(id);
    }
</script>
</head>
```



예제

```
<body onload="changeColor();">
  <div id="target">
    <p>This is a Text.</p>
  </div>
  <button onclick="stopTextColor();">중지</button>
</body>
</html>
```



실행(클릭)



screen 객체

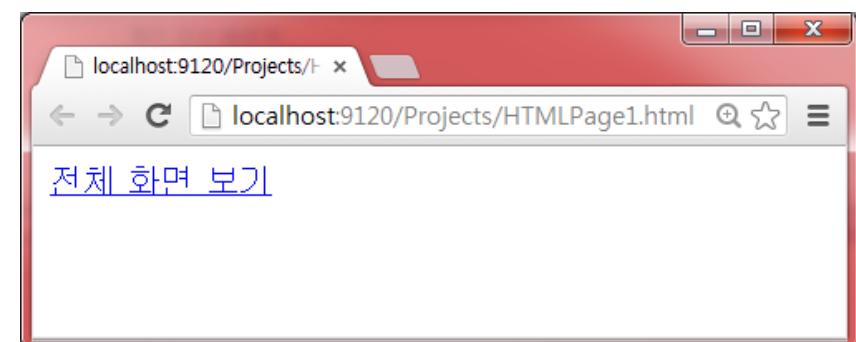
속성	설명
<u>availHeight</u>	화면의 높이를 반환(윈도우에서 태스크바를 제외한 영역)
<u>availWidth</u>	화면의 너비를 반환(윈도우에서 태스크바를 제외한 영역)
<u>colorDepth</u>	컬러 팔레트의 비트 깊이를 반환
<u>height</u>	화면의 전체 높이를 반환
<u>pixelDepth</u>	화면의 컬러 해상도(bits per pixel)를 반환
<u>width</u>	화면의 전체 너비를 반환



예제

```
<script>
    function maxopen(url, winattributes) {
        var maxwindow = window.open(url, "", winattributes)
        maxwindow.moveTo(0, 0);
        maxwindow.resizeTo(screen.availWidth, screen.availHeight)
    }

</script>
<a href="#" onClick="maxopen('http://www.google.com', 'resize=1,
scrollbars=1, status=1'); return false">전체 화면 보기</a>
```





location 객체

속성	설명
<u>hash</u>	URL 중에서 앵커 부분을 반환한다. (#section1과 같은 부분)
<u>host</u>	URL 중에서 hostname과 port를 반환
<u>hostname</u>	URL 중에서 hostname을 반환
<u>href</u>	전체 URL을 반환
<u>pathname</u>	URL 중에서 경로(path)를 반환
<u>port</u>	URL 중에서 port를 반환
<u>protocol</u>	URL 중에서 protocol 부분을 반환
<u>search</u>	URL 중에서 쿼리(query) 부분을 반환

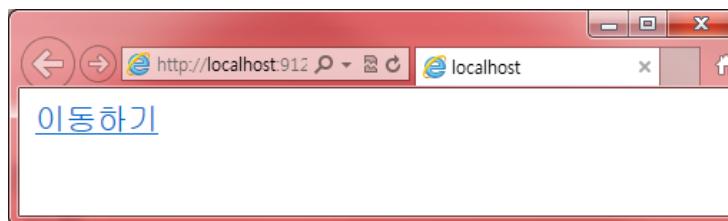
http://www.dragoneye.co.kr:80/column2/list.asp#top?table=table30&page=2

href	문서의 URL 주소	위 주소 전체
host	호스트 이름과 포트	http://www.dragoneye.co.kr:80
hostname	호스트 컴퓨터 이름	http://www.dragoneye.co.kr
hash	앵커 이름	#top
pathname	디렉토리 이하 경로	/column2/list.asp
port	포트 번호 부분	80
protocol	프로토콜 종류	http:
search	URL 조회부분	table=table30&page=2



예제

```
<script>
    function replace0 {
        location.replace("http://www.google.com")
    }
</script>
<a href="#" onClick="replace0">이동하기</a>
```



실행(클릭)



navigator 객체

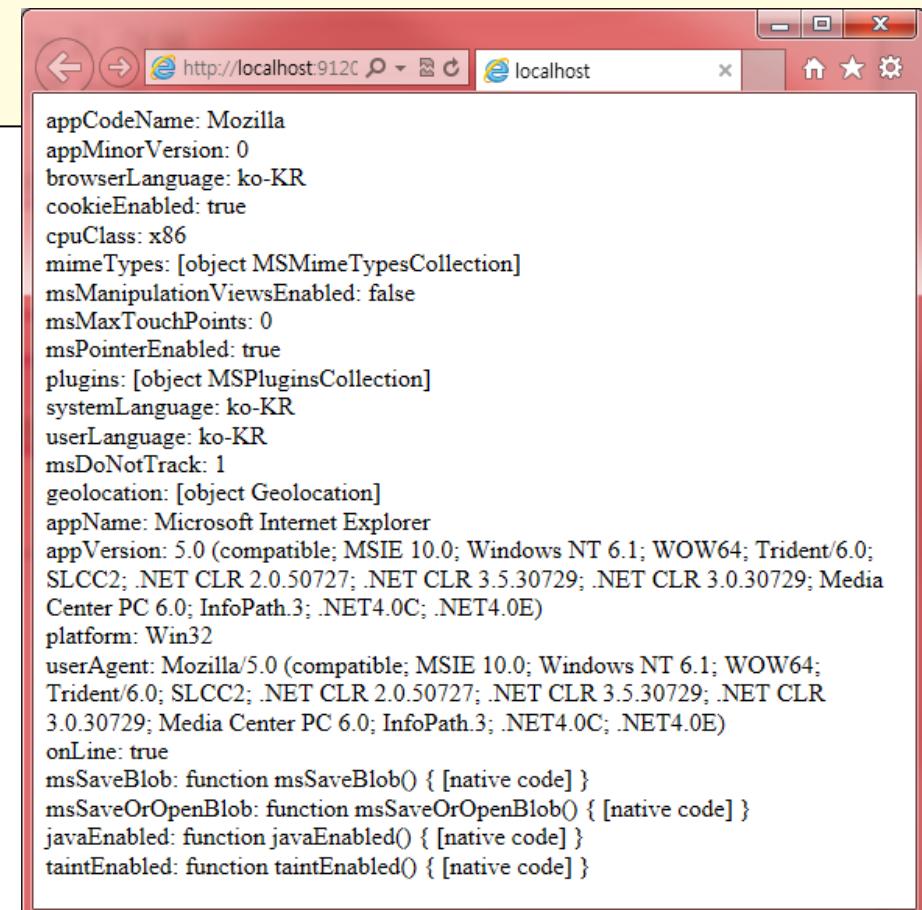
속성	설명
<u>appCodeName</u>	브라우저의 코드 네임
<u>appName</u>	브라우저의 이름
<u>appVersion</u>	브라우저의 버전 정보
<u>cookieEnabled</u>	브라우저에서 쿠키가 활성화되어 있는지 여부
<u>onLine</u>	브라우저가 인터넷에 연결되어 있으면 true
<u>platform</u>	브라우저가 <u>컴파일된</u> 플랫폼
<u>userAgent</u>	브라우저에서 서버로 가는 user-agent 헤더



예제

```
<script>
for (var key in navigator) {
    value = navigator[key];
    document.write(key + ": " + value + "<br>");
}
</script>
```

실행(클릭)



The screenshot shows a Microsoft Internet Explorer window with the URL `http://localhost:9120`. The page content displays various properties of the `navigator` object, such as `appCodeName`, `appMinorVersion`, `browserLanguage`, `cookieEnabled`, `cpuClass`, `mimeType`s, `msManipulationViewsEnabled`, `msMaxTouchPoints`, `msPointerEnabled`, `plugins`, `systemLanguage`, `userLanguage`, `msDoNotTrack`, `geolocation`, `appName`, `appVersion`, `platform`, `userAgent`, `onLine`, and several methods like `msSaveBlob`, `msSaveOrOpenBlob`, `javaEnabled`, and `taintEnabled`.

```
appCodeName: Mozilla
appMinorVersion: 0
browserLanguage: ko-KR
cookieEnabled: true
cpuClass: x86
mimeType: [object MSMimeTypesCollection]
msManipulationViewsEnabled: false
msMaxTouchPoints: 0
msPointerEnabled: true
plugins: [object MSPluginsCollection]
systemLanguage: ko-KR
userLanguage: ko-KR
msDoNotTrack: 1
geolocation: [object Geolocation]
appName: Microsoft Internet Explorer
appVersion: 5.0 (compatible; MSIE 10.0; Windows NT 6.1; WOW64; Trident/6.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; InfoPath.3; .NET4.0C; .NET4.0E)
platform: Win32
userAgent: Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.1; WOW64; Trident/6.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; InfoPath.3; .NET4.0C; .NET4.0E)
onLine: true
msSaveBlob: function msSaveBlob() { [native code] }
msSaveOrOpenBlob: function msSaveOrOpenBlob() { [native code] }
javaEnabled: function javaEnabled() { [native code] }
taintEnabled: function taintEnabled() { [native code] }
```



이벤트 처리



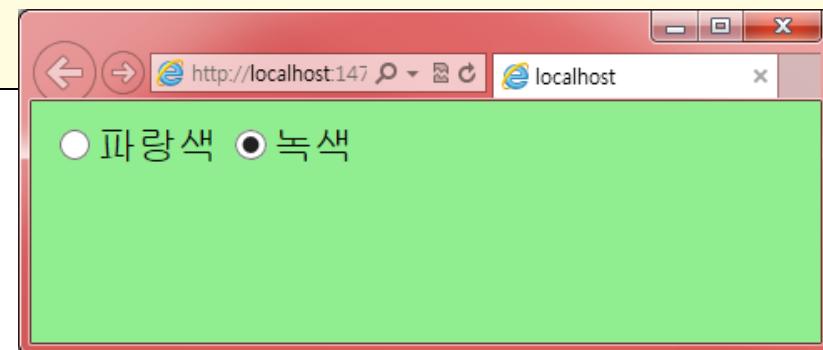
- 이벤트의 종류
 - 마우스 클릭
 - 웹 페이지 로딩
 - hovering
 - HTML 입력 양식에서 입력 박스를 선택하는 것
 - 키보드의 키를 누르는 것



onclick 이벤트

```
<!DOCTYPE html>
<html>
<head>
    <script>
        function changeColor(c) {
            document.getElementById("target").style.backgroundColor = c;
        }
    </script>
</head>
<body id="target">

    <form method="POST">
        <input type="radio" name="C1" value="v1"
            onclick="changeColor('lightblue')">파랑색
        <input type="radio" name="C1" value="v2"
            onclick="changeColor('lightgreen')">녹색
    </form>
</body>
</html>
```

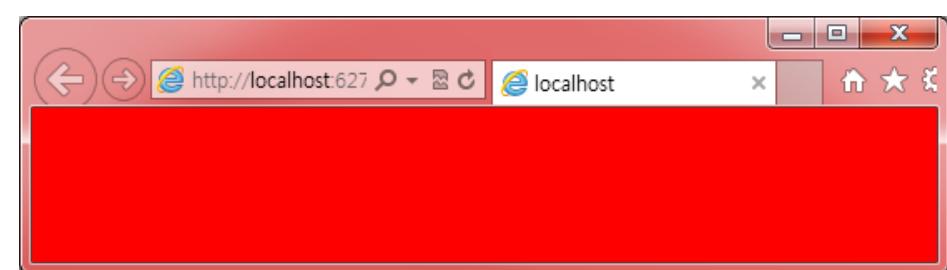
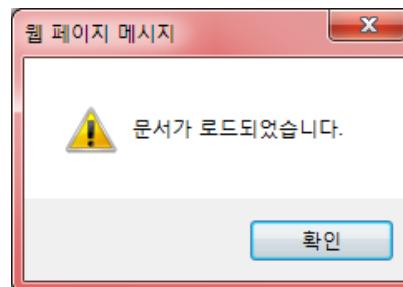


실행(클릭)



onload와 onunload 이벤트

```
<html>
<head>
    <script>
        function onLoadDoc() {
            alert("문서가 로드되었습니다.");
            document.body.style.backgroundColor = "red";
        }
    </script>
</head>
<body onload="onLoadDoc();">
</body>
</html>
```





onchange 이벤트

```
<!DOCTYPE html>
<html>
<head>
<script>
    function sub() {
        var x = document.getElementById("name");
        x.value = x.value.toLowerCase();
    }
</script>
</head>
<body>
영어단어: <input type="text" id="name" onchange="sub()">
<p>입력필드를 벗어나면 소문자로 변경됩니다.</p>
</body>
</html>
```

실행(클릭)

영어단어: ABOUT

입력필드를 벗어나면 소문자로 변경됩니다.

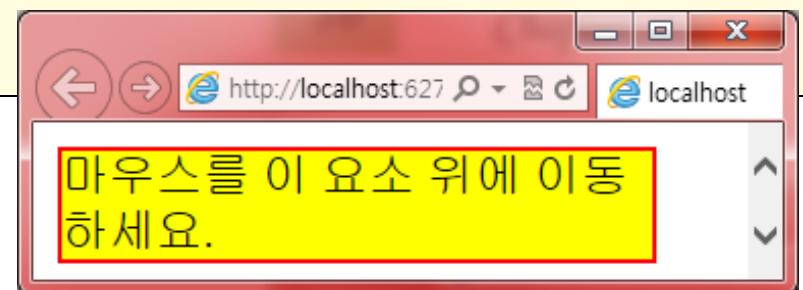
영어단어: about

입력필드를 벗어나면 소문자로 변경됩니다.



onmouseover 이벤트

```
<html>
<head>
<script>
    function OnMouseIn(elem)
    {
        elem.style.border = "2px solid red";
    }
    function OnMouseOut(elem)
    {
        elem.style.border = "";
    }
</script>
</head>
<body>
    <div style="background-color: yellow; width: 200px"
        onmouseover="OnMouseIn (this)" onmouseout="OnMouseOut (this)">
        마우스를 이 요소 위에 이동하세요.
    </div>
</body>
</html>
```



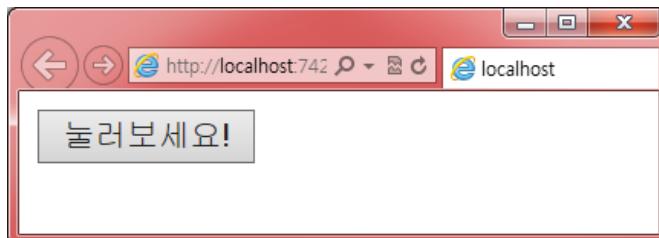
실행(클릭)



onmousedown 이벤트

```
<html>
<head>
<script>
    function OnButtonDown(button) {
        button.style.color = "#ff0000";
    }
    function OnButtonUp(button) {
        button.style.color = "#000000";
    }
</script>
</head>
<body>
    <button onmousedown="OnButtonDown (this)"
onmouseup="OnButtonUp (this)">눌러보세요!</button>
</body>
```

실행(클릭)





계산기 예제

```
<html>
<head>
<script>
    var expression="";
    function add(character) {
        expression = expression + character;
        document.getElementById("display").value = expression;
    }
    function compute() {
        document.getElementById("display").value = eval(expression);
    }
    function clearDisplay() {
        expression = "";
        document.getElementById("display").value = "0";
    }
</script>
<body>
<form>
```

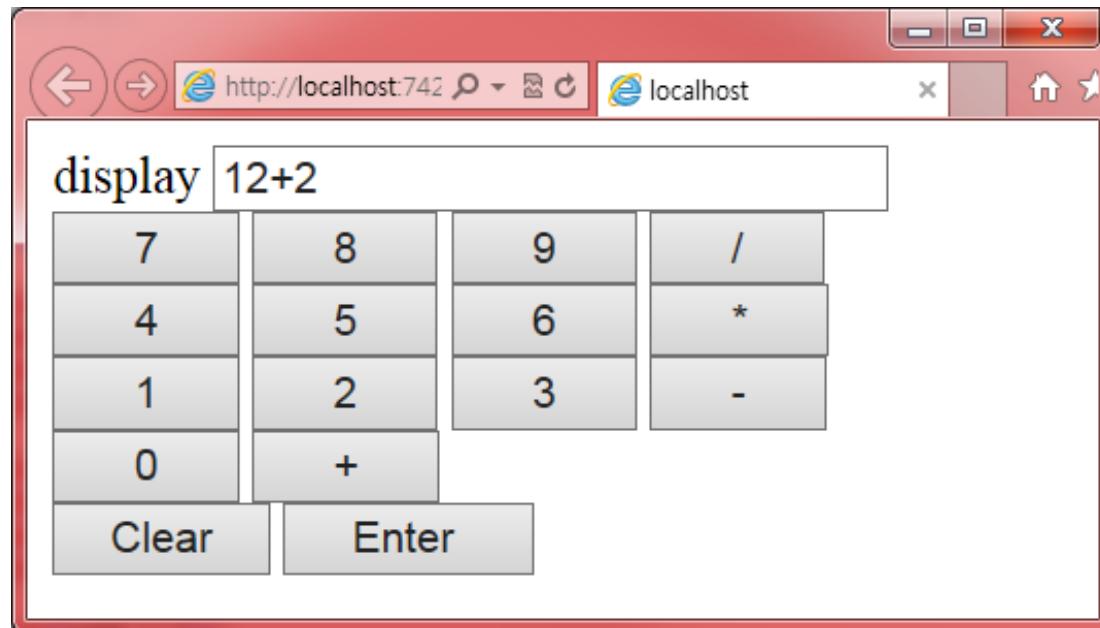


계산기 예제

```
display <input id="display" value="0" size="30"> <br>
<input type="button" value=" 7 " onclick="add('7')">
<input type="button" value=" 8 " onclick="add('8')">
<input type="button" value=" 9 " onclick="add('9')">
<input type="button" value=" / " onclick="add('/')"> <br>
<input type="button" value=" 4 " onclick="add('4')">
<input type="button" value=" 5 " onclick="add('5')">
<input type="button" value=" 6 " onclick="add('6')">
<input type="button" value=" * " onclick="add('*')"> <br>
<input type="button" value=" 1 " onclick="add('1')">
<input type="button" value=" 2 " onclick="add('2')">
<input type="button" value=" 3 " onclick="add('3')">
<input type="button" value=" - " onclick="add('-')"> <br>
<input type="button" value=" 0 " onclick="add('0')">
<input type="button" value=" + " onclick="add('+')"> <br>
<input type="button" value=" Clear " onclick="clearDisplay()">
<input type="button" value=" Enter " name="enter"
onclick="compute()">
</form>
</body>
</html>
```



실행 결과



웹브라우저 실행



폼 데이터 접근

- getElementById()를 사용하거나 form[] 객체를 사용한다.
 - “form[]” 객체는 배열이며, “name” 속성을 이용하여 접근할 수 있다.

```
<input type="text" name="addr" onclick="display(this.form)"/>
```

```
function display(form) {
```

텍스트가 클릭되면

```
    alert(form["addr"].value);
```

```
}
```

↑
form 객체는 폼 안의 필드로 이루어진 배열로서
name을 식별자로 하여 필드값을 저장한다.





폼의 유효성 검증

- 입력 필드에서의 잘못을 검증하는 작업

```
<h3>회원가입</h3>
<form>
이름: <input type='text' id='name' /><br />
주소: <input type='text' id='addr' /><br />
생일: <input type='date' id='birthday' /><br />
아이디(6-8 문자):
<input type='text' id='username' /><br />
이메일: <input type='email' id='email' /><br />
휴대폰: <input type='tel' id='phone' /><br />
<input type='submit' value='확인' /><br />
</form>
```

회원가입

이름:

주소:

생일:

아이디(6-8 문자):

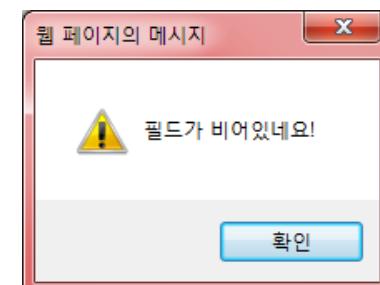
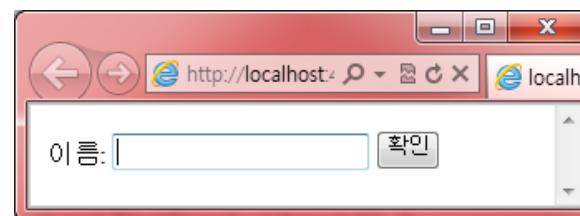
이메일:

휴대폰:



공백 검증

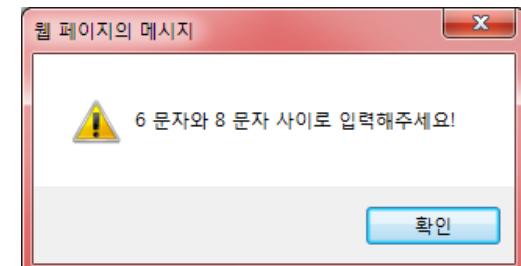
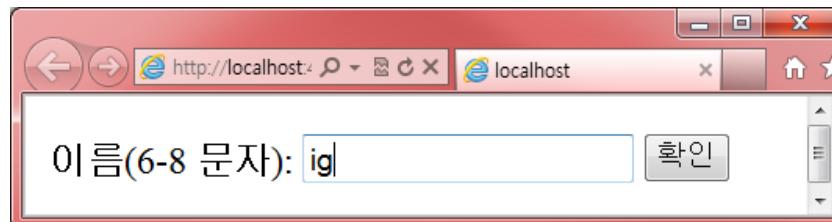
```
<script>  
function checkNotEmpty(field) {  
    if (field.value.length == 0) {  
        alert("필드가 비어있네요!");  
        field.focus();  
        return false;  
    }  
    return true;  
}  
</script>  
<form>  
이름: <input type='text' id='user' />  
<input type='button'  
    onclick="checkNotEmpty(document.getElementById('user'))"  
    value='확인' />  
</form>
```





데이터 길이 검증

```
<script>
    function checkLength(elem, min, max) {
        var s = elem.value;
        if (s.length >= min && s.length <= max) {
            return true;
        } else {
            alert(min + " 문자와 " + max + " 문자 사이로 입력해주세요!");
            elem.focus();
            return false;
        }
    }
</script>
<form>
    이름(6-8 문자): <input type='text' id='name' />
    <input type='button'
        onclick="checkLength(document.getElementById('name'), 6, 8)"
        value='확인' />
</form>
```





정규식

- 정규식(**regular expression**): 특정한 규칙을 가지고 있는 문자열들을 표현하는 수식이다
- (예) $^{\text{[0-9]}}+abc\$$

식	기능	설명
<code>^</code>	시작	문자열의 시작을 표시
<code>\$</code>	끝	문자열의 끝을 표시
<code>.</code>	문자	한 개의 문자와 일치
<code>\d</code>	숫자	한 개의 숫자와 일치
<code>\w</code>	문자와 숫자	한 개의 문자나 숫자와 일치
<code>\s</code>	공백문자	공백, 탭, 줄바꿈, 캐리지리턴 문자와 일치
<code>[]</code>	문자 종류, 문자 범위	[abc]는 a 또는 b 또는 c를 나타낸다. [a-z]는 a부터 z까지 중의 하나, [1-9]는 1부터 9까지 중의 하나를 나타낸다.



정규식

* : 0 번 또는 그 이상

+ : 1번 또는 그 이상

? : 0번 또는 1번

{n} : n번

() : 수학적인 표현식과 함께 하위 패턴들을 그룹 짓는데 사용.

{min,max} : 최소 min, 최대 max

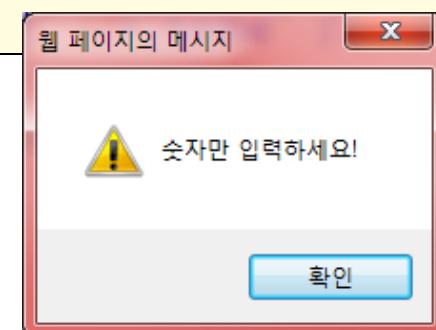
- `/./+`
- `/^\w*/`
- `/^([1-9][0-9])*$/`
- `/^\d{6}-\d{7}$/`
- `/^(Good)?Bye$/`
- `/^5[1-5][0-9]{14}$/`



숫자 검증 예제

```
<script>
    function checkNumeric(elem, msg) {
        var exp = /^[0-9]+$/;
        if (elem.value.match(exp)) {
            return true;
        } else {
            alert(msg);
            elem.focus();
            return false;
        }
    }
</script>
<form>
    전화번호(-없이 입력): <input type='text' id='phone' />
    <input type='button'
        onclick="checkNumeric(document.getElementById('phone'), '숫자만
        입력하세요!')"
        value='확인' />
</form>
```

전화번호(-없이 입력): 확인





Q & A

