



javascript 2강

양 명 속

[now4ever7@gmail.com]



목차

- location 객체
- history 객체
- screen 객체
- navigator 객체

- css 스타일 제어
- 이벤트 처리 - 이벤트, 이벤트 핸들러
- document 관련 객체
 - image 객체



location객체

location객체

현재 브라우저의 윈도우에 열린 **html 문서**의 주소를 포함한 URL에 관련된 각종 정보를 제공하는 객체로서 주소 관련 정보는 물론 특정 사이트에 접속하는 경우에도 사용됨

■ location객체

- 브라우저의 주소 표시줄과 관련된 객체
- 프로토콜의 종류, 호스트 이름, 문서 위치 등의 정보가 있음

```
<script>
// 출력합니다.
var output = "";
for (var key in location) {
    output += "▶"+key + ': ' + location[key] + '\ n';
}
alert(output);
</script>
```

브라우저마다 실행 결과가 조금씩 다른
속성만 출력하는 브라우저가 있고, 메서드까지 출력
하는 브라우저가 있음

```
▶replace: function () { [native code] }
▶assign: function () { [native code] }
▶ancestorOrigins: [object DOMStringList]
▶origin: file://
▶hash:
▶search:
▶pathname: /D:/d_java/luna_ws2/jsProject/js/
day2/9-9.html
▶port:
▶hostname:
▶host:
▶protocol: file:
▶href: file:///D:/d_java/luna_ws2/jsProject/js/
day2/9-9.html
▶reload: function reload() { [native code] }
```

확인



▶ hash:
▶ host:
▶ hostname:
▶ href: file:///D:/d_java/luna_ws2/jsProject/js/day2/9-9.html
▶ pathname: /D:/d_java/luna_ws2/jsProject/js/day2/9-9.html
▶ port:
▶ protocol: file:
▶ search:
▶ assign:
function assign() {
 [native code]
}

▶ reload:
function reload() {
 [native code]
}

▶ replace:
function replace() {
 [native code]
}

▶ toString:
function toString() {
 [native code]
}

확인

location객체

- 모든 브라우저의 location객체 속성과 메서드

속성 이름	설명	예
href	문서의 URL주소	
host	호스트 이름과 포트 번호	localhost:30763
hostname	호스트 이름	localhost
port	포트 번호	30763
pathname	디렉토리 경로	/Projects/Location.htm
hash	앵커 이름(#~)	#beta
search	요청 매개변수	?param=10
protocol	프로토콜 종류	http:

속성 이름	설명
assign(link)	현재 위치를 이동합니다.
reload()	새로고침합니다.
replace(link)	현재 위치를 이동합니다.



location객체

- location객체
 - 페이지를 이동할 때 많이 사용함
 - 다음 4가지 방법으로 페이지를 이동할 수 있음
 - assign() 메서드와 replace() 메서드 모두 페이지를 이동하지만 replace() 메서드는 뒤로 가기 버튼을 사용할 수 없음

```
location="http://www.naver.com";  
location.href="http://www.naver.com";  
location.assign("http://www.naver.com");  
location.replace("http://www.naver.com");
```

- 현재 페이지 새로고침하기

```
location=location;  
location.href=location.href;  
location.assign(location);  
location.replace(location);  
location.reload();
```

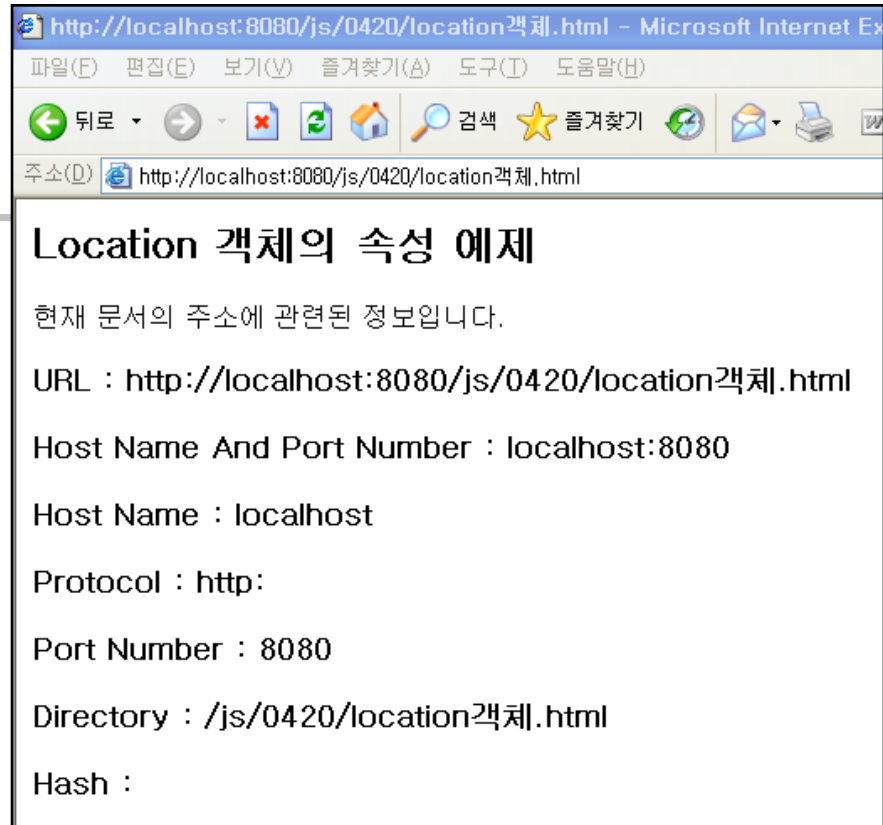
```
<script>
  var d = new Date();
  document.write(d.toLocaleString());

  function func1(){
    //location=location;
    //location.href=location.href;
    //location.assign(location);
    //location.replace(location);

    location.reload();
  }
</script>
</head>
<body>
<input type="button" value="새로고침" onclick="func1()"/>
</body>
```

2015. 4. 2. 오후 9:32:08 새로고침

예제 - location 객체



<BODY>

<H2> Location 객체의 속성 예제 </H2>

현재 문서의 주소에 관련된 정보입니다.<P>

<SCRIPT type="text/javascript">

document.write("<H3>")

document.write("URL : " + location.href + "<P>");

document.write("Host Name And Port Number : " + location.host + "<P>");

document.write("Host Name : " + location.hostname + "<P>");

document.write("Protocol : " + location.protocol + "<P>");

document.write("Port Number : " + location.port + "<P>");

document.write("Directory : " + location.pathname + "<P>");

document.write("Hash : " + location.hash);

document.write("</H3>")

</SCRIPT>

</BODY>

예제- location 객체

• location.href

- 문서의 URL주소를 알려주거나 특정 사이트에 접속하는 등의 기능을 가지고 있는 속성

```
<HEAD>
<SCRIPT type="text/javascript">
function m_site()
{
    window.location.href="http://www.google.co.kr";
}
function loadURL()
{
    location.href=document.frm1.site.options[document.frm1.site.selectedIndex].value;
}
</SCRIPT></HEAD>
<BODY>
    <H2> 사이트 이동 </H2>
    <FORM name="frm1">
        <INPUT TYPE = "button" VALUE="구글로 이동" onClick="m_site()"><p>

        <SELECT NAME="site">
            <OPTION VALUE="http://www.google.co.kr">구글 홈페이지
            <OPTION VALUE="http://www.naver.com" selected>네이버 홈페이지
            <OPTION VALUE="http://www.daum.net">다음 홈페이지
        </SELECT>
        <input type="button" VALUE="접속" onClick="loadURL()">
    </FORM>
</BODY>
```

location.href=document.frm1.site.value;



사이트 이동

구글로 이동

구글 홈페이지 ▼ 접속

구글 홈페이지

네이버 홈페이지

다음 홈페이지

예제 - location 객체 메서드

```
<HEAD>
<SCRIPT type="text/javascript">
    function rel()
    {
        location.reload();
    }
    function rep()
    {
        location.replace("http://www.google.com");
    }
</SCRIPT>
</HEAD>
<BODY>
```

```
    <H2> Location 객체 메소드 예제 </H2>
```

```
    <FORM>
```

```
        <INPUT TYPE="button" VALUE="문서 다시 읽기" onClick="rel()"><p>
```

```
        <INPUT TYPE="button" VALUE="구글로 이동하기" onClick="rep()"><p>
```

```
        <a href="http://naver.com">네이버로 이동</a>
```

```
    </FORM>
```

```
    <div id="curTime"></div>
```

```
<SCRIPT type="text/javascript">
```

```
    today = new Date();
```

```
    document.getElementById("curTime").innerHTML = today.toLocaleString();
```

```
</SCRIPT>
```

```
</BODY>
```

Location 객체 메소드 예제

문서 다시 읽기

구글로 이동하기

[네이버로 이동](http://naver.com)

2014년 3월 12일 오전 11:30:53

- **replace()** 메서드 - 현재 실행중인 문서의 URL에 관한 히스토리를 제거하고 대체되기 때문에 **back** 버튼을 눌러도 이전 URL로 돌아갈 수 없음

실습 - 자동으로 웹 문서 새로 고치기

- 1분 간격으로 현재 문서를 새로 고침

현재 시간은 7시 32분 51초입니다.





history 객체

history 객체

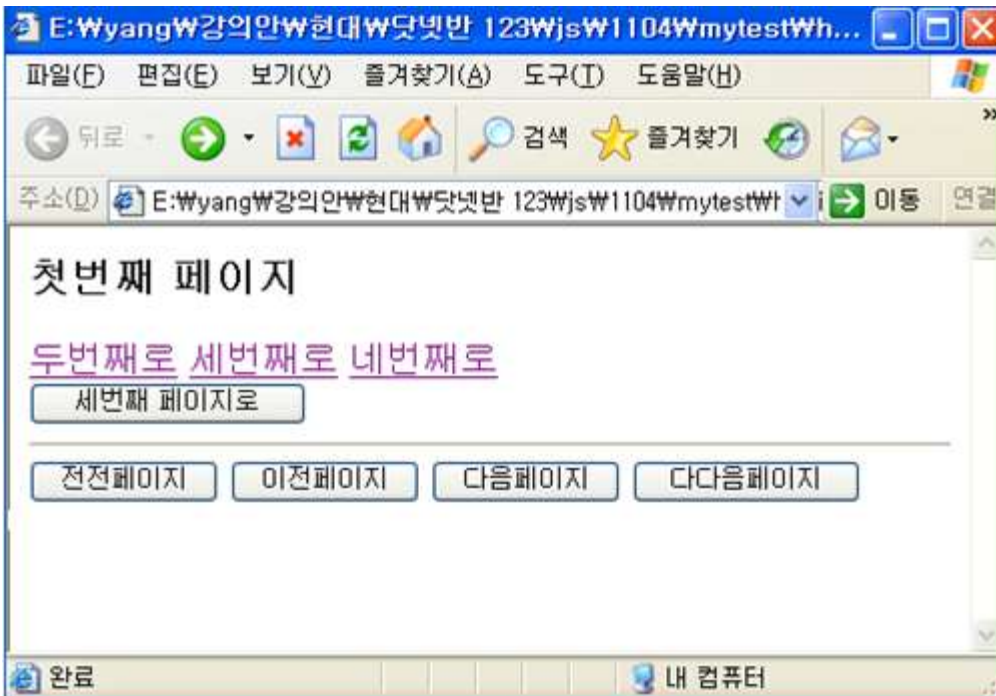
- 현재 윈도우에서 사용자가 가장 최근에 방문했던 사이트들의 URL 목록, 즉 히스토리를 관리하는 객체
 - 이 객체가 있어서 뒤로(back)버튼과 (forward)버튼이 제 기능을 할 수 있음

속성	기능
length	히스토리 리스트에 등록되어 있는 URL의 개수를 알려줌
current	현재 히스토리의 URL을 알려줌
next	다음 히스토리의 URL을 알려줌
previous	이전 히스토리의 URL을 알려줌

• 메서드

메서드	기능
back()	현재 페이지의 이전 페이지로 이동함, <code>history.go(-1)</code> 과 동일
forward()	현재 페이지의 다음 페이지로 이동함
go()	히스토리 목록 내의 임의의 페이지로 이동함, 양수값: 현재 문서의 다음 문서로 이동, 음수이면 이전 문서로 이동

예제 - history 객체



두번째 페이지

첫번째로 세번째로 네번째로

전전페이지

이전페이지

다음페이지

다다음페이지

세번째 페이지

첫번째로 두번째로 네번째로

전전페이지

이전페이지

다음페이지

다다음페이지

네번째 페이지

첫번째로 두번째로 세번째로

전전페이지

이전페이지

다음페이지

다다음페이지



예제 - history 객체

a.html

```
<body>
<h3>첫번째 페이지</h3>
<a href="b.html">두번째로</a>
<a href="c.html">세번째로</a>
<a href="d.html">네번째로</a><br>
<input type="button" value="세번째 페이지로" onclick="pageMove()">
<hr>

<input type="button" value="전전페이지" onclick="history.go(-2)">
<input type="button" value="이전페이지" onclick="history.back()">
<input type="button" value="다음페이지" onclick="history.forward()">
<input type="button" value="다다음페이지" onclick="history.go(2)">

<script>
  function pageMove(){
    location.href="c.html";
  }
</script>
</body>
```




예제 - history 객체

b.html

```
<BODY>
<h3>두번째 페이지</h3>
<a href="a.html">첫번째로</a>
<a href="c.html">세번째로</a>
<a href="d.html">네번째로</a><br>
<input type="button" value="전전페이지" onclick="history.go(-2)">
<input type="button" value="이전페이지" onclick="history.back()">
<input type="button" value="다음페이지" onclick="history.forward()">
<input type="button" value="다다음페이지" onclick="history.go(2)">
</BODY>
```

```
<BODY>
<h3>세번째 페이지</h3>
<a href="a.html">첫번째로</a>
<a href="b.html">두번째로</a>
<a href="d.html">네번째로</a><br>
<input type="button" value="전전페이지" onclick="history.go(-2)">
<input type="button" value="이전페이지" onclick="history.back()">
<input type="button" value="다음페이지" onclick="history.forward()">
<input type="button" value="다다음페이지" onclick="history.go(2)">
</BODY>
```

c.html



예제- history 객체

d.html

```
<BODY>
<h3>네번째 페이지</h3>
<a href="a.html">첫번째로</a>
<a href="b.html">두번째로</a>
<a href="c.html">세번째로</a><br>
<input type="button" value="전전페이지" onclick="history.go(-2)">
<input type="button" value="이전페이지" onclick="history.back()">
<input type="button" value="다음페이지" onclick="history.forward()">
<input type="button" value="다다음페이지" onclick="history.go(2)">
</BODY>
```



실습

뒤로

앞으로

새로고침

네이버로 이동

현재 날짜와 시간 : 2014년 3월 12일 수요일 오전 11:55:43

window 객체의 onload 이벤트 속성

■ onload 이벤트 속성

```
<script>
  window.onload = function () {

  };
</script>
```

- **window** 객체가 로드가 완료되고 자동으로 할당한 함수를 실행함
- **window** 객체가 로드가 완료되는 때
 - HTML 페이지에 존재하는 모든 태그가 화면에 올라가는 순간이 로드가 완료되는 순간임

- window 객체의 onload 속성을 사용
- 문서 객체의 속성 중 'on'으로 시작하는 속성을 이벤트 속성이라고 부르며 함수를 할당해야 함

■ HTML 페이지 생성 순서

```
<head>
<title>Document Object Model</title>
  <script>document.write('Process - 0<br>')</script>
</head>
<body>
  <h1>Process - 1</h1>
  <script>document.write('Process - 2<br>')</script>
  <h2>Process - 2</h2>
  <script>document.write('Process - 3<br>')</script>
</body>
```



window 객체의 onload 이벤트 속성

- HTML 페이지 생성 순서

- 처음 HTML 페이지를 실행하면 문자열 'Process-0' 을 출력함
- 웹 브라우저가 코드를 아래로 읽으며 h1 태그 내부의 문자열 'Process-1'을 출력함
- 곧바로 아래의 script 태그를 실행하므로 문자열 'Process-2'를 출력함
- 다시 HTML 페이지를 읽으며 h2 태그 내부의 문자열 'Process-2' 를 출력함
- 그 아래의 코드를 읽으며 'Process-3'을 출력함



window 객체의 onload 이벤트 속성

- window 객체의 onload 이벤트 속성을 이용
 - 웹 브라우저의 모든 객체가 로드됐을 때 함수를 실행
 - script 태그가 h1 태그보다 위에 있지만 onload 이벤트 속성을 사용했기 때문에 나중에 출력됨

```
<script>
    window.onload = function () {
        document.getElementById("result").innerHTML = 'Procsss - 0';
    };
</script>
</head>
<body>
    <h1>Process - 1</h1>
    <h1>Process - 2</h1>
    <div id="result">div</div>
</body>
```



innerHTML 속성

- document.getElementById("id") 함수
 - id 속성의 값을 사용해서 원하는 HTML 태그를 사용할 수 있도록 해줌
- innerHTML 속성
 - HTML 요소의 innerHTML 속성에 HTML 코드 지정하기
 - innerHTML 속성에 원하는 HTML 코드만 값으로 주면 해당 내용이 변경됨

오늘 날짜는 <div id="info"></div> 입니다.

```
<script type="text/javascript">  
  var divinfo = document.getElementById('info');  
  divinfo.innerHTML = "<b>2013-10-21</b>";  
</script>
```

오늘 날짜는
2013-10-21
입니다.



자바스크립트 코드의 동작 원리

- 1. 브라우저가 html문서를 처음부터 한 줄씩 태그들을 해석해 나가면서 결과를 화면상에 출력함
- 2. 브라우저는 window, document, history, location, navigator의 5개 기본 객체를 자동으로 생성시킴
 - 이들 객체들은 하나의 html 페이지가 로드되면 그 내용에 관계없이 자동으로 생성되어 기본적으로 존재하게 되는 객체들임
- 3. html 문서 엔진이 <script>태그를 만나면 자바스크립트 엔진을 호출하여 코드의 실행을 지시하며, 자바스크립트 엔진은 <script>태그 내의 코드를 실행함
- 4. 브라우저는 문서의 끝까지 태그를 정상적으로 해석하여 출력하며, html 태그 중에 자바스크립트 객체 생성이 필요한 태그는 그에 상응하는 자바스크립트 객체를 생성함
 - 예를 들어 html 문서 엔진이 태그를 만나면 src 속성에 지정된 이미지를 화면에 출력하고, 내부적으로는 image 객체를 생성함
- 5. 사용자의 이벤트가 발생하면 그에 상응하는 이벤트 핸들러를 실행시킴



CSS 스타일 제어

CSS 스타일 제어

- 태그에 써 넣은 인라인 스타일을 자바스크립트를 통해 제어하는 방법

- (1) class 속성은 **className** 속성으로

- 자바스크립트에서는 html 태그의 class 속성에 접근하기 위해 className 속성을 이용함

- 예)

```
<img class="bigImage">
```

```
image객체.className = "bigImage";
```

- (2) style 속성을 의미하는 **style** 속성은 별도의 객체임

- html 태그의 style 속성 - 인라인 스타일 속성을 나열

- 자바스크립트에서 style 객체는 스타일 정보를 가진 별도의 객체임

- 예)

```
<img style="width:300px;height:100px">
```

```
image객체.style.width = "300px";
```

```
image객체.style.height = "100px";
```



CSS 스타일 제어-style 속성

- html객체의 style 속성을 이용하려면 스타일 속성 이름은 첫 단어와 두 번째 단어 사이의 하이픈이 없어지고, 두 번째 단어의 첫 글자를 대문자로 써야 함
 - 예)
 - <table style="background-color:black">
- table객체.style.backgroundColor = "black";



예제-스타일 className

```
<STYLE>
.on {color:steelblue; font-weight:bold}
</STYLE>
<script type="text/javascript">
    function applyClass() {
        var element = document.getElementById("h2");
        element.className="on";
    }

    function applyClass2(id, type) {
        var element = document.getElementById(id);
        element.className=type;
    }
</script>
</HEAD>
<BODY>
<h1 onmouseover="this.className='on';" onmouseout="this.className='';">
    자바스크립트</h1>
<h2 id="h2" onmouseover="applyClass()" onmouseout="this.className='';">
    className을 이용하여 스타일 제어</h2>

<h3 id="h3" onmouseover="applyClass2(this.id, 'on')" onmouseout="applyClass2(this.id, '')">
    스타일</h3>

</BODY>
```

자바스크립트

className을 이용하여 스타일 제어
스타일



CSS 스타일 제어 – className 속성

- className 속성
 - element의 클래스명을 바꾸거나 추가할 수 있음

element 클래스이름(className) 바꾸기

배경색상변경

테두리만들기

글씨굵게

초기화

```
document.getElementById("div1").className="s1";
```

예제

<BODY>

```
<div id="div1" class="s1">element 클래스이름(className) 바꾸기 </div>
```

```
<button onclick="changeClass('s1');">배경색상변경</button>
```

```
<button onclick="changeClass('s2');">테두리만들기</button>
```

```
<button onclick="changeClass('s3');">글씨굵게</button>
```

```
<button onclick="changeClass('s4');">초기화</button>
```

```
<script type="text/javascript">
```

```
function changeClass(target){
```

```
var preClass, newClass;
```

```
var element = document.getElementById("div1");
```

```
if(target=="s4"){
```

```
    element.className="";
```

```
}else{
```

```
    preClass = element.getAttribute("class"); //s2 s1
```

```
    newClass = target + " " + preClass;
```

```
    element.className=newClass;
```

```
}//if
```

```
}
```

```
</script>
```

```
<style type="text/css">
```

```
#div1{padding:20px 30px;}
```

```
.s1{background:#CCFFFF;}
```

```
.s2{border:2px solid red;}
```

```
.s3{font-weight:bold;}
```

```
</style>
```

예제 - style 속성 이용

```
<script type="text/javascript">
function applyStyle() {
    //var objLink = document.getElementById("a1");
    //var objImg = document.getElementById("img1");
    var objLink = document.links[0];
    var objImg = document.images[0];

    objLink.style.textDecoration = "none";
    objLink.style.fontWeight = "bold";
    objLink.style.backgroundColor = "#eee";
    objLink.style.color = "gray";

    objImg.style.borderStyle = "solid";
    objImg.style.borderColor = "green";
    objImg.style.borderWidth = "7px";

    //objImg.style.border= "7px solid red";
}
</script>
</head>
<body>
<a id="a1" href="http://www.naver.com">지식검색 네이버</a><br><br>
<br><br>
<input type="button" value="스타일 적용하기" onclick="applyStyle()">
```

지식검색 네이버



스타일 적용하기



지식검색 네이버



스타일 적용하기

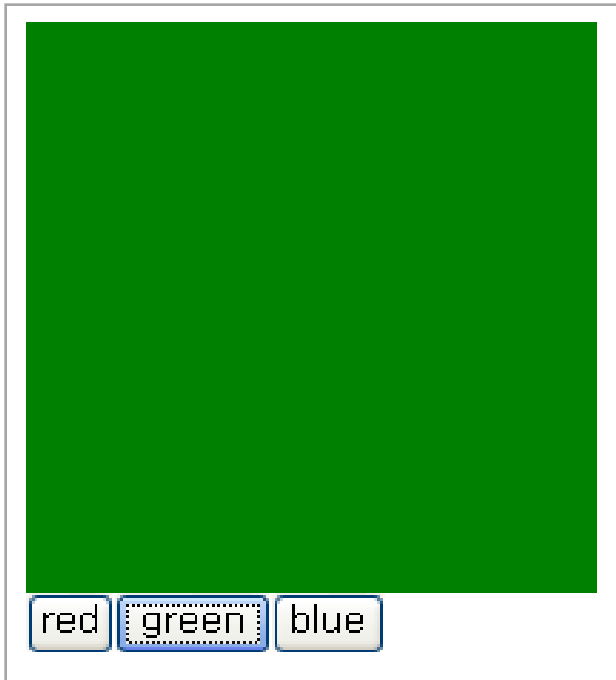
표 안에서 선택된 셀의 색상 바꾸기

```
<body>
<TABLE width = "450" >
<TR align="center">
    <TD WIDTH=50>번호</TD>
    <TD WIDTH=200>제목</TD>
    <TD WIDTH=100>작성자</TD>
    <TD WIDTH=100>날짜</TD>
</TR>
<TR align="center" onMouseOver="this.style.backgroundColor='lightcyan'"
onMouseOut="this.style.backgroundColor=''">
    <TD>1</TD>
    <TD>첫번째 글</TD>
    <TD>홍길동</TD>
    <TD>2009-10-20</TD>
</TR>
<TR align="center" onMouseOver="this.style.backgroundColor='lightcyan'"
onMouseOut="this.style.backgroundColor=''">
    <TD>2</TD>
    <TD>두번째 글</TD>
    <TD>김길동</TD>
    <TD>2009-10-19</TD>
</TR>
</TABLE>
</body>
```

번호	제목	작성자	날짜
1	첫번째 글	홍길동	2009-10-20
2	두번째 글	김길동	2009-10-19

실습

- className 오브젝트를 사용하여 각 버튼에 다른 효과를 부여해보자



```
<style type="text/css">
    div{
        width: 200px;
        height: 200px;border:solid gray 1px;
    }
    .red{background:red;}
    .green{background:green;}
    .blue{background:blue;}
</style>
</head>
<body>
<div id="targetDiv">
</div>
<input type="button" value="red">
<input type="button" value="green">
<input type="button" value="blue">
```



이벤트



이벤트 처리

■ 이벤트(Event)란

- 브라우저를 사용중인 사용자가 어떤 특정한 행동(action)을 했다는 신호를 의미하는 것
- 마우스 클릭이나 마우스 이동, 키보드 입력, 버튼 입력, 페이지 로드 등 사용자가 행한 다양한 모든 행동을 말함
- 사용자에게 의해 발생한 어떤 행동

■ 이벤트 핸들러(Event Handler)

- 자바스크립트는 이러한 사용자의 행동, 즉 이벤트가 발생하는지를 계속해서 체크하는데, 이벤트가 발생했음을 감지하게 되면 이벤트 핸들러를 이용하여 해당 이벤트에 대한 적절한 처리를 하게 됨
- 이벤트 핸들러 – 이벤트가 발생할 때마다 이벤트를 처리하기 위한 하나의 함수
- 예) 사용자가 버튼을 클릭하는 이벤트를 발생시킬 경우, 실제로 onClick 이라는 이벤트 핸들러에 의해 버튼 클릭 이벤트가 처리됨

이벤트와 이벤트 핸들러의 종류

이벤트	이벤트 핸들러	이벤트가 발생하는 경우	대상 객체
abort	onabort	이미지를 읽어 들이는 도중에 중지할 때	image
blur	onblur	윈도우, 폼, 구성 요소, 프레임이 포커스를 잃을 때(포커스가 다른 곳으로 이동하거나 벗어날 때)	button, checkbox, fileupload, password, radio, reset, select, text, textarea, window, layer
change	obchange	select, text, textarea 등의 입력 양식에 있던 내용(값)이 변경되고 포커스를 잃을 때 발생(콤보 상자에서 다른 데이터를 선택하는 경우도 해당)	fileupload, password, select, text, textarea
click	obclick	폼의 모든 구성 요소가 마우스로 클릭될 때	button, checkbox, link, radio, reset, submit
dblclick	ondblclick	폼의 모든 구성 요소가 마우스로 더블 클릭될 때	button, checkbox, link, radio, reset, submit
dragdrop	ondragdrop	마우스를 클릭한 상태에서 드래그했을 때	
error	onerror	문서나 이미지의 로딩시 오류가 발생할 때(넷스케이프에서 정상적으로 실행)	document, image
focus	onfocus	윈도우, 프레임, 프레임셋, 폼 요소가 포커스를 받을 때(입력 양식에 없던 커서가 생겨나는 순간)	button, checkbox, fileupload, password, radio, reset, select, text, textarea, window, layer

이벤트와 이벤트 핸들러의 종류

이벤트	이벤트 핸들러	이벤트가 발생하는 경우	대상 객체
keydown	onkeydown	사용자가 키를 누르는 순간 발생	document, image, link, textarea, text
keypress	onkeypress	사용자가 키를 눌렀다가 놓았을 때 발생	document, image, link, textarea, text
keyup	onkeyup	사용자가 누른 키를 놓은 순간 발생	document, image, link, textarea, text
load	onload	사용자가 브라우저를 이용해서 문서나 URL을 읽을 때(문서의 로딩이 완료되었을 때 발생)	image, window, layer
mousedown	onmousedown	사용자가 마우스를 누르는 순간	link, button, document
mousemove	onmousemove	사용자가 마우스를 움직이는 순간(넷스케이프에서 사용)	captureEvents 메서드를 가지고 있는 객체 (document, window)
mouseout	onmouseout	마우스 커서가 특정 영역을 벗어나는 순간	link, layer
mouseover	onmouseover	사용자가 마우스커서를 이미지맵이나 링크 위에 올려 놓을 때(마우스가 특정 영역 위로 지나가는 순간)	link, layer

키보드 관련 이벤트 : **keyDown -> keyPress -> keyUp** 순서로 발생

이벤트와 이벤트 핸들러의 종류

이벤트	이벤트 핸들러	이벤트가 발생하는 경우	대상 객체
mouseenter	onmouseenter	마우스가 요소의 경계 외부에서 내부로 이동할 때 발생함	link, layer
mouseleave	onmouseleave	마우스가 요소의 경계 내부에서 외부로 이동할 때 발생함	link, layer
mouseup	onmouseup	사용자가 마우스 단추에서 손을 떼는 순간 (버튼을 놓았을 때)	link, button, document
move	onmove	윈도우 창이나 프레임의 위치가 이동될 때(화면을 움직이는 순간), 넷스케이프에서만 실행	window, frame
reset	onreset	사용자가 reset 버튼을 누를 때(입력 양식의 내용을 초기화시킬 때)	form (reset단추에 설정되지 않고 form태그에 핸들러가 설정됨)
resize	onresize	윈도우나 프레임의 크기가 조절되는 순간(화면의 크기를 변경하는 순간)	window, frame
select	onselect	사용자가 입력 양식의 필드 내 텍스트 일부를 선택할 때(입력 양식의 내용 중 특정 내용을 선택하는 순간)	textarea, text
submit	onsubmit	사용자가 submit 버튼을 누를 때	form (submit단추에 설정되지 않고 form태그에 핸들러가 설정됨)
unload	onunload	사용자가 브라우저를 이용하여 읽어온 문서를 닫을 때(문서가 없어지는 순간)	document



이벤트 핸들러

- 이벤트 핸들러 – html과 함께 사용됨, on으로 시작
- 이벤트를 처리하는 방법
 - 이벤트 핸들러의 지정 위치에 따라 2가지 방법으로 구분
 - (1) html 태그를 이용하여 이벤트 핸들러를 지정하는 방법

<html 태그 이벤트 핸들러 = “이벤트 핸들링을 위한 자바스크립트 코드”>

- <input type = “button” value=“확인” **onClick=“sample()”>**
- <body **onLoad = “alert(‘환영합니다’)”>**
- (1) 이벤트가 발생하면 이벤트 핸들러에 의해 지정한 함수 호출
- (2) 별도의 함수를 부르지 않고, 직접 자바스크립트 코드 작성
- (3) 이벤트 핸들러의 반환값 – true, false
 - **true가 반환되면 그 이후의 행동을 계속함, false가 반환되면 더 이상 작업을 진행하지 않음**



이벤트 핸들러

- (2) html객체에 이벤트 핸들러를 지정하는 방법

html객체.이벤트 핸들러 = 이벤트 처리 함수

- 해당 객체에서 해당 이벤트가 발생하면 정의된 함수를 호출함
- 예) `document.onmousedown = sample;`
 - 문서 안에서 마우스를 누르는 이벤트가 발생하면(onmousedown) 이벤트 처리 함수 sample을 호출

```
var btn = document.frm.btn;  
btn.onclick = sample2;
```


예제 1

```
<HEAD>
<script type="text/javascript">
    document.onmousedown = sample1;
    function sample1 ()
    {
        document.bgColor='pink';
    }
    function sample2 ()
    {
        document.body.style.backgroundColor="green";
    }
</script>
</HEAD>
<BODY>
<form name="frm">
    이름 : <input type="text" name = "str">
    <input type="button" value = "pink" onclick="sample1()" >
    <input type="button" value = "green" name ="btn" id="btn"><br>
</form>

<script type="text/javascript">
    //var btn = document.getElementById("btn");
    var btn = document.frm.btn;
    btn.onclick = sample2;
</script>
</BODY>
```



이메일 주소

@

직접입력



이메일 체크

```
<script type="text/javascript">
    function emailCheck(){
        var email2 = frm1.email2.value;
        if(email2=="etc"){
            frm1.email3.value="";
            frm1.email3.style.display = "inline";
            frm1.email3.focus();
        }else{
            frm1.email3.style.display = "none";
        }
    }
</script>
<form name="frm1" method="post" action="test.jsp">
이메일 주소 <input type="text" name="email1">@
    <select name="email2" onchange="emailCheck()">
        <option value="naver.com">naver.com</option>
        <option value="hanmail.net">hanmail.net</option>
        <option value="nate.com">nate.com</option>
        <option value="etc">직접입력</option>
    </select><input type="text" name="email3" style="display:none">
</form>
```



실습

- switch를 활용한 물고기 움직이기
 - 물고기 움직이기 예제 업그레이드
- 요구사항:
 - 01. 초기 시작 위치: $x(\text{left}) = 50$, $y(\text{top}) = 200$
 - 02. 왼쪽 화살표 버튼을 누르는 경우 왼쪽으로 -50만큼 이동
 - 03. 오른쪽 화살표 버튼을 누르는 경우 오른쪽으로 +50만큼 이동
 - 04. 위쪽 화살표 버튼을 누르는 경우 위쪽으로 -50만큼 이동
 - 05. 아래쪽 화살표 버튼을 누르는 경우 아래쪽으로 +50만큼 이동
 - 06. 단, 물고기가 움직이는 영역(`#panel`) 밖으로 넘어가면 안 됨
- 키보드 다운 이벤트 등록하기 (`keydown`)
- 키보드 입력 값 알아내기 (`event.keyCode`)
- 물고기 위치 움직이기
 - fish의 style
 - left: 물고기 X위치 값
 - Top: 물고기 Y위치 값

- 화살표 키에 따른 물고기 위치 움직이기
- `keyCode`
 - 37 : 왼쪽 화살표
 - 39 : 오른쪽 화살표
 - 38 : 위쪽 화살표
 - 40 : 아래쪽 화살표

과제 1

E:\WyangW강의안W현대W자바W자바...

우편번호 찾기

주소 입력 :

■ 회원 가입 버튼을 클릭하면 유효성 검사하기

회원 가입

* 표시 항목은 필수 입력 사항입니다.

* 사용자 이름	<input type="text"/>		
* 아이디	<input type="text"/>	<input type="button" value="중복확인"/>	
* 비밀번호	<input type="text"/>		
* 비밀번호 확인	<input type="text"/>		
주소	<input type="text"/> - <input type="text"/>	<input type="button" value="우편번호 찾기"/>	
	<input type="text"/>		
	<input type="text"/>		
휴대폰 번호	010 <input type="button" value="v"/>	- <input type="text"/>	- <input type="text"/>
이메일 주소	<input type="text"/>	@	naver.com <input type="button" value="v"/>
SMS 수신여부	<input type="radio"/> 수신 <input checked="" type="radio"/> 수신안함		
<input type="button" value="회원가입"/>			



과제 1

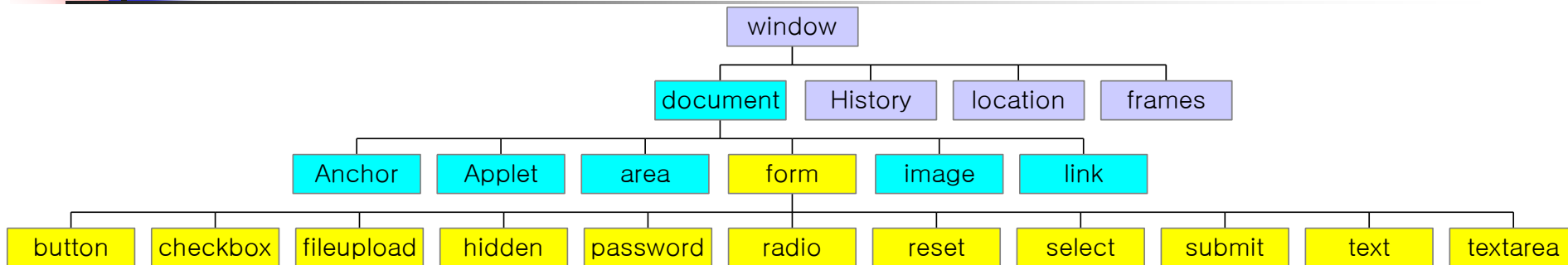
■ 유효성 검사

- 사용자이름, 아이디, 비밀번호는 반드시 입력하도록
- 아이디는 6자리~13자리 이어야 함
- 비밀번호와 비밀번호 확인은 일치해야 함
- 이메일 주소에서 직접입력을 선택하면 텍스트 상자가 보이도록

■ 새창 띄우기

- 우편번호 검색 버튼을 클릭하면 주소 검색 창이 뜨게 하기
- 주소 검색창에서 주소를 입력하고 버튼 클릭하면, 부모창에 주소가 입력되도록 하기

브라우저 객체 계층도



- 브라우저 객체 구조
 - 객체들은 계층구조로 이뤄져 있음
 - 최상위 객체로는 window 객체가 있으며 아래에는 frames, document, history, location 객체가 있으며 각각의 객체들은 아래로 또 다른 객체들로 파생
- window객체
 - 자바스크립트의 브라우저 객체의 최상위 객체, 전체 윈도우에 적용될 내용들을 가지고 있다
 - 윈도우마다 하나씩 존재하는 객체 (윈도우 창을 나타낼 때)
- document객체
 - 현재의 문서에 대한 내용을 가지고 있다.
 - 제목, 배경색, 폼에 대한 속성을 가지고 있다
 - html 페이지마다 하나씩 존재하는 객체 (웹 페이지를 나타낼 때)



image 객체

- image 객체

- ``태그를 이용해 만들어진 이미지에 관한 정보 및 속성을 추출하고, 제어하고, 설정할 수 있도록 하는 객체
- 화면에 보이고 있는 이미지의 이름, 크기, 테두리 굵기, 이미지 전송시 에러 발생 여부 등 이미지와 관련한 다양한 정보를 처리할 수 있음

- image객체는 이미지 정보를 배열로 저장하여 이용,

- 처음 나온 이미지의 정보는 `document.images[0]`에,
- 두 번째 나온 이미지 정보는 `document.images[1]`에 저장됨

```
document.이미지이름  
또는 document.getElementById("이미지아이디")  
또는 document.images[0]
```



image 객체 속성

속성	기능
border	이미지의 테두리 두께 및 테두리 지정 여부를 설정하거나 알려줌
width	이미지의 넓이를 설정하거나 알려줌
height	이미지의 높이를 설정하거나 알려줌
hspace	이미지의 가로 여백을 설정하거나 알려줌
vspace	이미지의 세로 여백을 설정하거나 알려줌
src	이미지의 URL주소를 나타냄
lowsrc	이미지의 URL주소를 나타내며, 해상도가 낮은 이미지를 전송하는 경우에 사용
name	이미지 객체의 이름을 알려줌
length	이미지의 수
complete	이미지의 로딩(전송) 완료 상태를 논리값(true/false)으로 알려줌

예제1 - image 객체

```
<script type="text/javascript">
function Information(idx){
    str = "문서내 이미지 개수:"+document.images.length+"\n";
    str += "이름 : " + document.images[idx].name + "\n";
    str += "파일 이름 : " + document.images[idx].src + "\n";
    str += "테두리 두께 : " + document.images[idx].border + "\n";
    str += "가로 길이 : " + document.images[idx].width + "\n";
    if(idx==0){
        str += "세로 길이 : " + document.dog1.height + "\n";
    }else if(idx==1){
        str += "세로 길이 : " + document.getElementById("dog2").height + "\n";
    }
    alert(str)
}
</script></head>
<body>
<H1> Image 객체 예제 </H1>
<IMG NAME="dog1" id="dog1" SRC="../images/1.png" BORDER=1 >
<IMG NAME="dog2" id="dog2" SRC="../images/2.png" BORDER=5 ><P>
다음 단추를 누르면 그림의 정보가 출력됩니다.<P>
<FORM>
    <INPUT TYPE="button" VALUE="왼쪽이미지 정보" onClick="Information(0)">
    <INPUT TYPE="button" VALUE="오른쪽이미지 정보" onClick="Information(1)">
</FORM>
</body>
```

Image 객체 예제



다음 단추를 누르면 그림의 정보가 출력됩니다.

왼쪽이미지 정보

오른쪽이미지 정보

```
문서내 이미지 개수:2
이름 : dog1
파일 이름 : file:///D:/d_java/luna_ws2/jsProject/js/images/
1.png
테두리 두께 : 1
가로 길이 : 180
세로 길이 : 135
```

예제 - 롤오버이미지

- off.jpg, on.jpg



JavaScript

그림에 마우스를 가져가면 다른 그림으로 바뀝니다.



JavaScript

그림에 마우스를 가져가면 다른 그림으로 바뀝니다.



예제

```
<SCRIPT type="text/javascript">
function changelImage(imgSrc, imgName){
    document.getElementById(imgName).src="../images/"+imgSrc+".jpg";
}
</SCRIPT>
</HEAD>

<BODY>
<A HREF="#" onmouseover="changelImage('on', 'roll')"
  onmouseout="changelImage('off', 'roll')" >
<IMG SRC="../images/off.jpg" NAME="roll" id="roll" style="border:0"></A><BR><BR>
<H3>그림에 마우스를 가져가면 다른 그림으로 바뀝니다.</H3>
```

실습-마우스 올리면 큰 이미지 보여주기

상품 이미지





실습

```
<h1>상품 이미지</h1>
```

```
<div align="center">
```

```
  <p>
```

```
    
```

```
  </p>
```

```
  <p>
```

```
    
```

```
    
```

```
    
```

```
    
```

```
  </p>
```

```
</div>
```

```
</body>
```



screen 객체

screen 객체

■ screen 객체

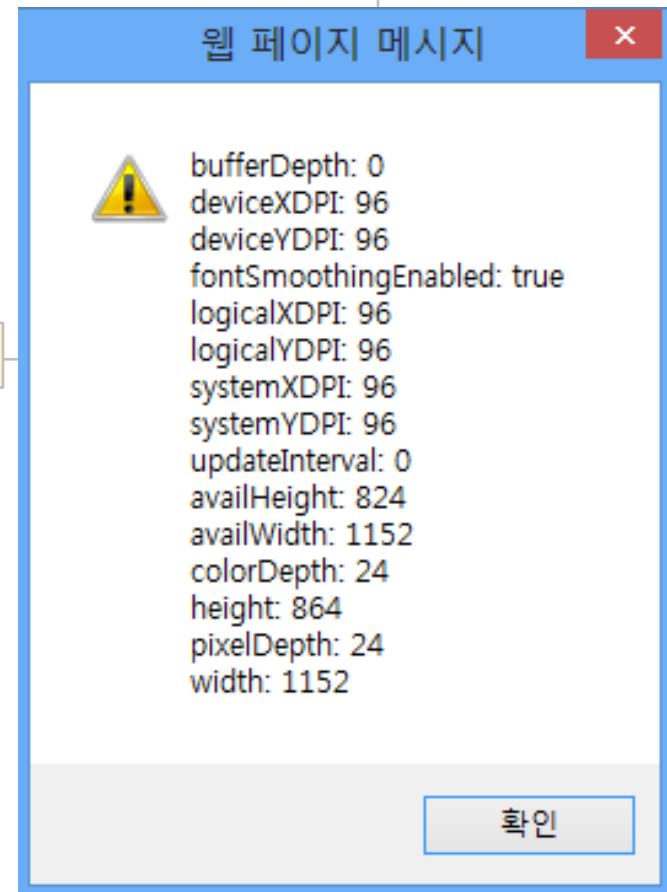
- 웹 브라우저의 화면이 아니라 운영체제 화면의 속성을 가지는 객체

```
<script>
  // 출력
  var output = "";
  for (var key in screen) {
    output += key + ': ' + screen[key] + '\n';
  }
  alert(output);
</script>
```

결과가 브라우저마다 다름

```
availWidth: 1152
availHeight: 824
availTop: 0
availLeft: 0
pixelDepth: 24
colorDepth: 24
width: 1152
height: 864
```

확인



```
<script>
```

```
// 변수를 선언합니다.
```

```
var child = window.open("", "width=300, height=200');
```

```
var width = screen.width;
```

```
var height = screen.height;
```

```
child.moveTo(0, 0); //절대적인 위치로 이동
```

```
child.resizeTo(width, height); //절대적인 크기-화면 크기와 동일하게 지정
```

```
//1초마다 10, 10 인 상대적인 위치로 이동
```

```
setInterval(function () {
```

```
    child.resizeBy(-20, -20);
```

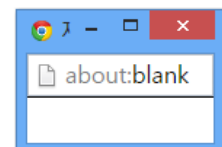
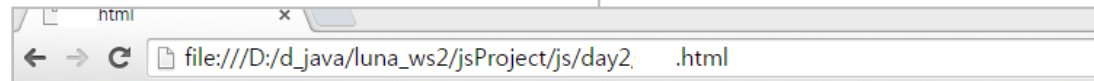
```
    child.moveBy(10, 10);
```

```
}, 1000);
```

```
</script>
```

- width, height 속성을 사용하면 손쉽게 window 객체를 화면에 띄워 만들 수 있음

- 팝업창을 생성하고 전체 화면으로 만듦
이어서 1초마다 크기를 변환하고 이동해 점
점 가운데로 수렴함



screen 객체

웹 브라우저의 화면이 아니라 운영체제 화면의 속성을 가지는 객체

- 사용중인 현재 화면의 해상도나 색상 수, 화면 크기 등의 정보를 알려주는 객체
 - 사용자의 화면 정보를 읽어서 그 상태에 따라 결과를 달리해야 하는 경우에 주로 사용됨
 - screen 객체도 별도의 객체 생성 과정 없이 직접 자신을 이용하여 객체를 사용함

속성	기능
availHeight	실제 화면의 높이(제목 표시줄이나 도구 모음 등은 제외)
availWidth	실제 화면의 넓이
colorDepth	사용 가능한 색상 수
height	전체 화면의 높이(제목 표시줄이나 도구 모음 등까지 포함)
pixelDepth	픽셀 당 비트 수
width	전체 화면의 넓이

모든 브라우저가 공통적으로 가지는 screen 객체의 속성



예제

```
<HTML>
  <HEAD>
  </HEAD>
  <BODY>
    <H1> screen 객체 예제 </H1>
    <SCRIPT type="text/javascript">
      document.write("픽셀 당 높이 : " + screen.height + "<P>");
      document.write("픽셀 당 넓이 : " + screen.width + "<P>");
      document.write("사용 가능한 화면의 높이 : " + screen.availHeight + "<P>");
      document.write("사용 가능한 화면의 넓이 : " + screen.availWidth + "<P>");
      document.write("픽셀 당 비트수 : " + screen.pixelDepth + "<P>");
      document.write("사용 가능한 색상수 : " + screen.colorDepth + "<P>");
    </SCRIPT>
  </BODY>
</HTML>
```

Screen 객체 예제

픽셀 당 높이 : 864

픽셀 당 넓이 : 1152

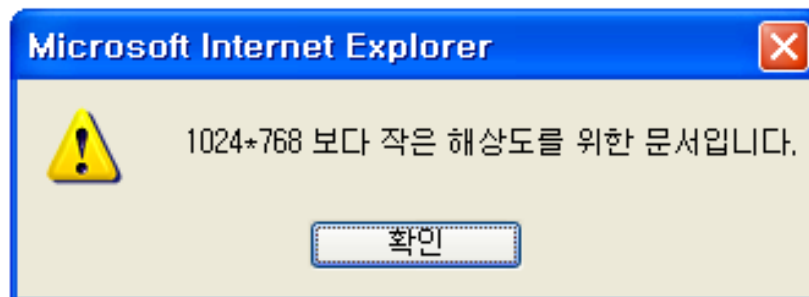
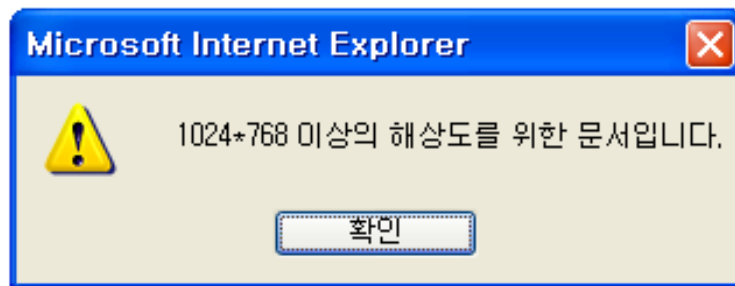
사용 가능한 화면의 높이 : 824

사용 가능한 화면의 넓이 : 1152

픽셀 당 비트수 : 24

사용 가능한 색상수 : 24

예제





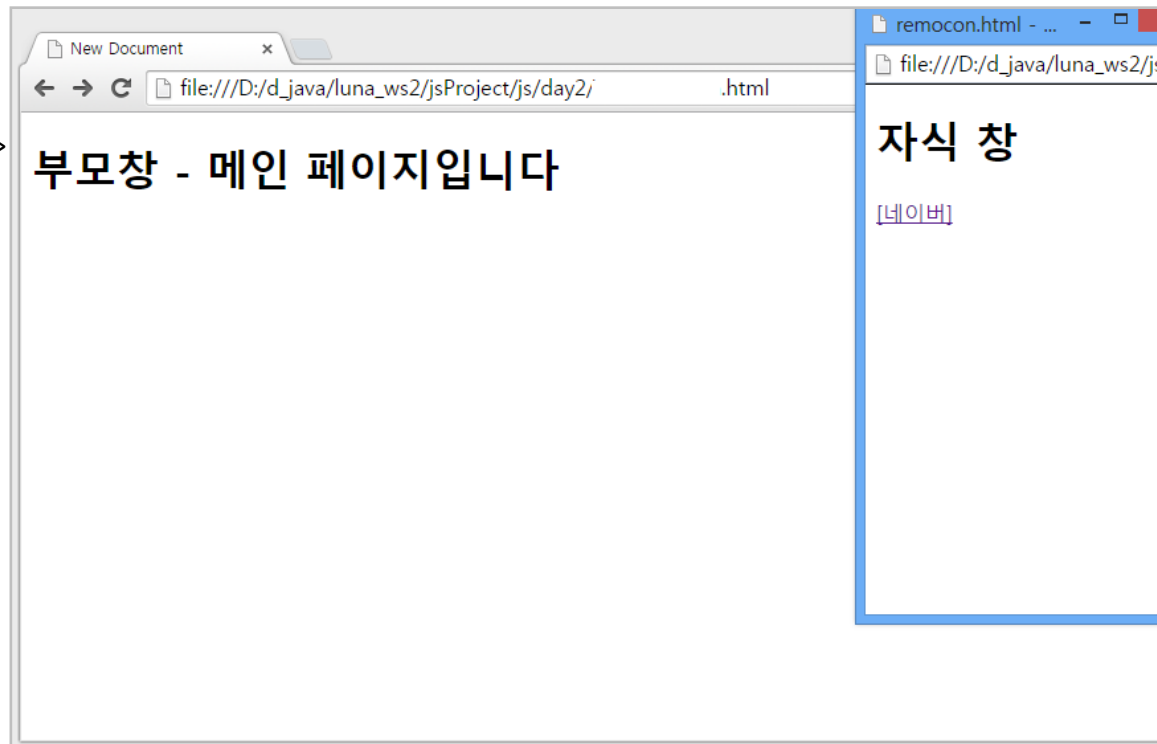
예제

```
<SCRIPT type="text/javascript">
var swidth=1024;
var sheight=768;
if (screen.width<swidth || screen.height<sheight)
{
    alert("1024*768 보다 작은 해상도를 위한 문서입니다.");
}
else
{
    alert("1024*768 이상의 해상도를 위한 문서입니다.");
}
</SCRIPT>
```

예제 - 리모콘 윈도우 만들기

```
<HTML>
<HEAD>
  <TITLE> New Document </TITLE>
  <SCRIPT type="text/javascript">
    opt="width=250,height=400,left=" + (screen.width-250) + ",top=0";
    window.open("remocon.html","",opt);
  </SCRIPT>
</HEAD>

<BODY>
  <h1>부모창 - 메인 페이지입니다</h1>
</BODY>
</HTML>
```





예제 - remococon.html

```
<HTML>
<HEAD>
<meta charset="EUC-KR">
<SCRIPT type="text/javascript">
function go(url){
    opener.location.href=url; //현재 윈도우를 연 윈도우에 들어갈 문서의 URL을 의미함
    self.close();
}
</SCRIPT>
</HEAD>

<BODY>
<h1>자식 창</h1>
<A HREF='#' onclick='go("http://www.naver.com")'>[네이버]</A><BR>
</BODY>
</HTML>
```



navigator 객체

navigator 객체

■ navigator 객체

- 웹 페이지를 실행하고 있는 브라우저에 대한 정보가 있음
- 많이 사용하는 속성

속성 이름	설명
appName	브라우저의 코드명
appName	브라우저의 이름
appVersion	브라우저의 버전
platform	사용중인 운영체제의 시스템 환경
userAgent	브라우저의 전체적인 정보

```
<script>
  // 출력합니다.
  var output = "";
  for (var key in navigator) {
    output += '●' + key + ': ' + navigator[key] + '\n';
  }
  alert(output);
</script>
```

navigator 객체의 속성을 모두 출력함

- serviceWorker: [object ServiceWorkerContainer]
- webkitPersistentStorage: [object DeprecatedStorageQuota]
- webkitTemporaryStorage: [object DeprecatedStorageQuota]
- geolocation: [object Geolocation]
- doNotTrack: null
- onLine: true
- languages: ko-KR,ko,en-US,en
- language: ko
- userAgent: Mozilla/5.0 (Windows NT 6.2; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/41.0.2272.101 Safari/537.36
- product: Gecko
- platform: Win32
- appVersion: 5.0 (Windows NT 6.2; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/41.0.2272.101 Safari/537.36
- appName: Netscape
- appCodeName: Mozilla
- hardwareConcurrency: 8
- maxTouchPoints: 0
- vendorSub:
- vendor: Google Inc.
- productSub: 20030107
- cookieEnabled: true
- mimeType: [object MimeTypeArray]
- plugins: [object PluginArray]
- javaEnabled: function javaEnabled() { [native code] }
- getStorageUpdates: function getStorageUpdates() { [native code] }
- getGamepads: function getGamepads() { [native code] }
- webkitGetUserMedia: function webkitGetUserMedia() { [native code] }
- vibrate: function vibrate() { [native code] }
- getBattery: function getBattery() { [native code] }



- appCodeName: Mozilla
- appMinorVersion: 0
- browserLanguage: ko-KR
- cookieEnabled: true
- cpuClass: x86
- mimeType: [object MSMimeTypesCollection]
- msManipulationViewsEnabled: true
- 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 6.2; WOW64; Trident/6.0; .NET4.0E; .NET4.0C; .NET CLR 3.5.30729; .NET CLR 2.0.50727; .NET CLR 3.0.30729; InfoPath.2)
- platform: Win32
- userAgent: Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.2; WOW64; Trident/6.0; .NET4.0E; .NET4.0C; .NET CLR 3.5.30729; .NET CLR 2.0.50727; .NET CLR 3.0.30729; InfoPath.2)
- onLine: true
- msSaveBlob: function msSaveBlob() { [native code] }
- msSaveOrOpenBlob: function msSaveOrOpenBlob() { [native code] }
- javaEnabled: function javaEnabled() { [native code] }
- msLaunchUri: function msLaunchUri() { [native code] }
- taintEnabled: function taintEnabled() { [native code] }

브라우저마다 다른 결과가 나옴



navigator 객체

- navigator 객체
 - 유일하게 window 객체에서 파생되지 않고 독립적으로 존재하는 객체로,
 - 브라우저의 종류, 사용 언어, 시스템의 종류 등 현재 사용하고 있는 브라우저에 관한 대부분의 정보를 제공하는 객체
- 주로 어떤 브라우저를 이용하여 사용자가 홈페이지에 들어왔는지를 알아내는데 사용됨
 - 브라우저의 종류에 따라 자바스크립트를 인식하고 실행할 수 있거나 그렇지 않은 브라우저가 있기 때문



navigator 객체

속성	기능
appName	익스플로러, 넷스케이프 등의 브라우저 이름을 알려줌
appVersion	브라우저의 버전을 알려줌
appCodeName	브라우저의 코드명을 알려줌
userAgent	클라이언트에서 서버로 전송되는 HTTP 헤더 속의 User-Agent 정보를 알려줌
language	현재 브라우저가 사용하는 언어의 종류를 알려줌(익스플로러 지원 안함)
mimeType	현재 브라우저가 지원하고 있는 MIME 타입을 알려줌(익스플로러 지원 안함)
platform	현재 브라우저가 실행되고 있는 시스템의 종류를 알려줌
plugins	현재 브라우저에 설치된 플러그인들의 종류를 알려줌(익스플로러 지원 안함)

메서드	기능
javaEnabled()	현재 사용하는 브라우저가 자바를 사용할 수 있는지의 상태를 알아내는 것으로, 사용할 수 있으면 true, 그렇지 않으면 false 값을 반환함
plugins.refresh()	새로 설치된 플러그인을 사용 가능한 상태로 만듦
taintEnabled()	현재 브라우저가 문서 정보를 정상적으로 읽어 왔는지의 여부를 알아냄

예제 1 - navigator 객체

```
<BODY>
<SCRIPT type="text/javascript">
    document.write("[1] 애플리케이션 이름      : " + navigator.appName + "<P>");
    document.write("[2] 웹 브라우저 버전       : " + navigator.appVersion + "<P>");
    document.write("[3] 웹 브라우저 코드 이름 : " + navigator.appCodeName + "<P>");
    document.write("[4] User Agent 정보        : " + navigator.userAgent + "<P>");
    document.write("[5] 사용 언어              : " + navigator.language + "<P>");
    document.write("[6] 시스템의 종류         : " + navigator.platform + "<P>");
</SCRIPT>
</BODY>
```

[1] 애플리케이션 이름 : Microsoft Internet Explorer

[2] 웹 브라우저 버전 : 5.0 (compatible; MSIE 10.0; Windows NT 6.2; WOW64; Trident/6.0; .NET4.0E; .NET4.0C; .NET CLR 3.5.30729; .NET CLR 2.0.50727; .NET CLR 3.0.30729; InfoPath.2)

[3] 웹 브라우저 코드 이름 : Mozilla

[4] User Agent 정보 : Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.2; WOW64; Trident/6.0; .NET4.0E; .NET4.0C; .NET CLR 3.5.30729; .NET CLR 2.0.50727; .NET CLR 3.0.30729; InfoPath.2)

[5] 사용 언어 : undefined

[6] 시스템의 종류 : Win32



예제 1 – navigator 객체

- 파이어폭스

[1] 애플리케이션 이름 : Netscape
[2] 웹 브라우저 버전 : 5.0 (Windows; ko)
[3] 웹 브라우저 코드 이름 : Mozilla
[4] User Agent 정보 : Mozilla/5.0 (Windows; U; Windows NT 5.1; ko; rv:1.9.1.3) Gecko/20090824 Firefox/3.5.3 (.NET CLR 3.5.30729)
[5] 사용 언어 : ko
[6] 시스템의 종류 : Win32

- 크롬

[1] 애플리케이션 이름 : Netscape
[2] 웹 브라우저 버전 : 5.0 (Windows NT 6.2; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/41.0.2272.101 Safari/537.36
[3] 웹 브라우저 코드 이름 : Mozilla
[4] User Agent 정보 : Mozilla/5.0 (Windows NT 6.2; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/41.0.2272.101 Safari/537.36
[5] 사용 언어 : ko
[6] 시스템의 종류 : Win32

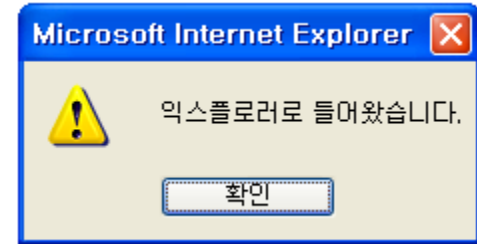
예제2 - navigator 객체

■ 브라우저 판별

```
<BODY>
<SCRIPT type="text/javascript">
var brow_IE = "Microsoft Internet Explorer"
var brow_Nets = "Netscape"

if (navigator.appName== brow_Nets)
    alert("넷스케이프로 들어왔습니다.");
else if (navigator.appName == brow_IE)
    alert("익스플로러로 들어왔습니다.");
else
    alert("IE나 Netscape 브라우저가 아니군요.");

/*
if(navigator.appName.indexOf("Microsoft") == -1)
    alert("익스플로러가 아닙니다!!");
else
    alert("익스플로러로 들어왔네요!!");
*/
</SCRIPT>
</BODY>
```





브라우저 판별

```
<body>
  <h1>브라우저 판별</h1>
  <div id="div1"></div>

  <script type="text/javascript">
    var userAgent = navigator.userAgent;
    var ua = userAgent.toLowerCase();
    console.log(ua);
    document.getElementById("div1").innerHTML=ua;

    if(ua.indexOf("trident")!=-1){
      alert("Microsoft Internet Explorer입니다");
    }else{
      alert("브라우저가 Explorer가 아닙니다");
    }
  </script>
</body>
```

브라우저 판별

mozilla/5.0 (windows nt 6.3; wow64; trident/7.0; .net4.0e; .net4.0c; .net clr 3.5.30729; .net clr 2.0.50727; .net clr 3.0.30729; gwx:downloaded; gwx:reserved; rv:11.0) like gecko