

- HTML 에 JavaScript 삽입

```
<script language="javascript">
```

JavaScript Statements

```
</script>
```

- HTML 에 JavaScript 코드 파일 삽입

```
<script language="javascript" src="src.js">
```

```
</script>
```

- 변수의 대소문자 구분

- 변수에 포함된 데이터의 형을 지정하지 않고, JavaScript 인터프리터가 변수에 포함된 데이터 형을 추적하고 변환.

- JavaScript 데이터 형

- 숫자형 : 정수와 부동 소수
- BOOLEAN : true, false
- STRING : 작은 따옴표나 큰 따옴표에 들어가 있는 값으로, 특수문자 포맷 사용시에는 'W'와 함께 사용.
- null : 아무런 값도 없는 것으로, 변수를 초기화 시키거나 어떤 값이나 이벤트를 지울 때 사용됨
- undefined : 변수만 만들고, 값을 할당하지 않은 상태

- 유형간의 변환

- 스트링 피연산자가 비스트링 연산자와 사용된 경우 다른 연산자를 모두 스트링으로 변환
- BOOLEAN 값은 1 과 0 으로 변환되어 수치 연산을 지원
- null 값은 스트링 연산에 대해서는 "null", 논리 연산에서는 false, 그리고 수치 연산에서는 0 으로 각각 변환
- 변환 함수
 - eval() : 스트링 표현식을 수치값으로 변환하고, 파라미터가 수치형태가 아닌 스트링 값이면 에러 발생
 - parseInt() : 스트링에 포함된 첫번째 정수를 리턴하고, 스트링이 정수로 시작되지 않으면 0 을 리턴
 - parseFloat() : 스트링에 포함된 첫번째 부동 소수를 리턴하거나 스트링이 적절한 부동 소수로 시작되지 않으면 0 을 리턴

- 배열 : 값의 시퀀스를 정렬할 수 있는 객체로, JavaScript 의 특수형으로 배열 사용 전엔 반드시 선언해야 한다.

- 배열 선언 예

```
arrayName = new Array(arrayLength)
```

```
arrayName = new Array()
```

```
arrayName = new Array(value0,value1, ..., valuen)
```

- JavaScript 만의 특수 연산자

- comma(,) 연산자 : 두 표현식을 계산하고, 두 번째 표현식의 값 리턴
- delete 연산자 : 객체의 프로퍼티나 배열 인덱스의 요소 삭제하고, 항상 undefined 값 리턴
- new 연산자 : 객체 형의 인스턴스를 만들기 위해 사용
- typeof 연산자 : 연산자의 형을 식별하는 스트링 값을 리턴

- void 연산자 : 값을 리턴하지 않는다
- 지역 변수와 전역 변수 : 함수 안에서만 사용되는 것을 지역 변수라 하고, 프로그램 내에서 사용되는 변수를 전역 변수라 하는 데, 지역 변수 사용시 반드시 var 키워드와 함께 선언해야 함

이벤트 처리!!

- 이벤트 정의와 사용
 - 이벤트 : 사용자가 웹페이지나 기타 다른 브라우저에 수행한 작업으로 인한 결과
 - 이벤트 처리 : 이벤트로 인해 수행되는 프로세스
 - 이벤트 핸들러 : 프로세스를 수행하는 코드
 - 사용 예 : 사용자가 링크 위로 마우스를 갖다 대면 다이얼로그 박스를 표시한다거나, 폼에 입력한 데이터를 검증한다거나, 버튼을 클릭할 때 애니메이션을 나타내거나, Java 애플릿과 브라우저 플러그인이 상호작용을 하도록 한다.

HTML 태그	JavaScript 이벤트	설명
다양	mouseMove	마우스 이동
<A>..<	Click	마우스로 링크를 클릭
	dbClick	마우스를 링크위에서 더블클릭
	mouseDown	마우스 버튼을 누름
	mouseUp	마우스 버튼을 놓음
	mouseOver	마우스를 링크위로 이동
	mouseOut	링크 위에 있던 마우스를 링크 밖으로 이동
	keyDown	사용자가 키를 누름
	keyUp	사용자가 키를 놓음
	keyPress	사용자가 키를 눌렀다가 놓음
	abort	사용자 액션으로 인해 이미지 로딩 작업을 중단함
	error	이미지 로딩하는 동안

		에러 발생
	load	이미지가 로드되고 화면에 나타남
	keyDown	사용자가 키를 누름
	keyUp	사용자가 키를 놓음
	keyPress	사용자가 키를 눌렀다가 놓음
<AREA>	mouseOver	마우스가 클라이언트측 이미지맵의 한 영역으로 이동함
	mouseOut	마우스가 이미지맵 영역 밖으로 이동
	dbClick	사용자가 이미지맵의 한 영역을 더블클릭함
<BODY>..</BODY>	Click	사용자가 문서의 본문을 클릭
	dbClick	문서의 본문을 더블 클릭함
	keyDown	키를 누름
	keyUp	키를 놓음
	keyPress	키를 눌렀다가 놓음
	mouseDown	마우스 버튼을 누름
	mouseUp	마우스 버튼을 놓음
<BODY>..</BODY> <FRAMESET>..</FRAMESET> <FRAME>..</FRAME>	blur	윈도우에서 현재 입력 포커스가 사라짐
	error	윈도우가 로드되는 동안 에러 발생
	focus	입력 포커스가 현재 윈도우로 이동
	load	윈도우 로딩이 완료됨
	unload	윈도우를 종료함
	move	윈도우가 이동됨
	resize	윈도우의 크기가 바뀜
	dragDrop	윈도우에 객체를 드롭
<FORM>..</FORM>	submit	사용자가 폼을 제출
	reset	사용자가 폼을 재설정
<INPUT TYPE="text">	blur	현재 입력 포커스가 텍스트 필드에서 사라짐

	focus	현재 입력 포커스가 텍스트 필드로 이동
	change	텍스트 필드가 변경되어 현재 입력 포커스가 사라짐
	select	텍스트 필드에 있는 텍스트가 선택됨
<INPUT TYPE="password">	blur	패스워드 필드에서 입력 포커스가 사라짐
	focus	패스워드 필드에 입력 포커스 생김
<TEXTAREA>..</TEXTAREA>	blur	텍스트 영역이 현재 입력 포커스가 사라짐
	focus	텍스트 영역에 입력 포커스 생김
	change	텍스트 영역이 변경되어 입력 포커스가 사라짐
	select	텍스트 영역에서 텍스트가 선택됨
	keyDown	키를 누름
	keyUp	키를 놓음
	keyPress	키를 눌렀다 놓음
<INPUT TYPE="button">	Click	버튼이 클릭됨
	blur	입력할 수 없도록 버튼이 흐려짐
	focus	입력할 수 있도록 포커스 생김
	mouseDown	버튼 위에서 왼쪽 마우스 버튼 누름
	mouseUp	버튼 위에서 왼쪽 마우스 버튼 놓음
<INPUT TYPE="submit">	Click	제출 버튼이 클릭됨
	blur	제출 버튼에서 입력 포커스가 사라짐
	focus	제출 버튼에 입력 포커스 생김
<INPUT TYPE="reset">	Click	리셋 버튼이 클릭됨
	blur	리셋 버튼에서 포커스가 사라짐

	focus	리셋 버튼에서 포커스가 놓임
<INPUT TYPE="radio">	Click	라디오 버튼이 클릭
	blur	라디오 버튼에서 입력 포커스가 사라짐
	focus	라디오 버튼에 입력 포커스 생김
<INPUT TYPE="checkbox">	Click	체크 박스가 클릭
	blur	체크 박스에서 입력 포커스가 사라짐
	focus	체크 박스에 입력 포커스 놓임
<INPUT TYPE="file">	blur	파일 업로드 폼 요소에서 입력 포커스 사라짐
	change	사용자가 업로드될 파일을 선택
	focus	파일 업로드 폼 요소에 입력 포커스 놓임
<SELECT>..</SELECT>	blur	선택 요소가 현재 입력 포커스 잃음
	change	선택 요소가 변경되어 입력 포커스가 사라짐
	focus	선택 요소에 현재 입력 포커스가 놓임

- 이벤트 처리 속성

이벤트핸들링속성	실행되는 상황
onAbort	이미지를 로딩하는 작업이 사용자의 한 행동으로 인해 취소되었음
onBlur	문서나 윈도우, 프레임세트, 폼 요소에서 현재 입력 포커스가 사라짐
onChange	텍스트 필드나 텍스트 영역, 파일 업로드 필드, 선택 항목이 변경되어 현재 입력 포커스가 사라짐
onClick	링크나 클라이언트측 이미지맵 영역, 폼 요소가 클릭됨
onDbClick	링크나 클라이언트측 이미지맵 영역, 문서가 더블 클릭됨
onDragDrop	드래그된 객체가 윈도우나 프레임에 드롭됨

onError	이미지나 윈도우, 프레임을 로딩하는 동안 에러가 발생함
onFocus	문서나 윈도우, 프레임 세트, 폼 요소에 입력 포커스 놓임
onKeyDown	키를 누름
onKeyPress	키를 눌렀다 놓음
onKeyUp	키를 놓음
onLoad	이미지나 문서, 프레임이 로드됨
onMouseDown	마우스 버튼 누름
onMouseMove	마우스를 이동함
onMouseOut	링크나 클라이언트측 이미지맵에서 마우스를 옮김
onMouseOver	마우스를 링크나 클라이언트측 이미지맵으로 옮김
onMouseUp	마우스 버튼을 놓음
onMove	사용자가 윈도우나 프레임을 이동함
onReset	폼의 리셋 버튼을 클릭하여 폼을 리셋시킴
onResize	사용자가 윈도우나 프레임의 크기를 조절
onSelect	텍스트는 텍스트 필드나 영역에서 선택됨
onSubmit	폼이 제출됨
onUnload	사용자가 문서나 프레임 세트를 종료함

객체 정의하기

- 객체 유형 정의
 - 프로퍼티 : 객체에 들어 있는 데이터 값에 액세스할 때 사용.
 - 메소드 : 객체에 어떤 작업을 할 때 사용하는 함수.
- 객체 유형 만들기

사용자가 직접 객체 유형을 정의하고 특정 객체 인스턴스를 만들 수 있는데 이렇게 만들려면 객체 유형의 특정 인스턴스를 만들 때 사용되는 함수를 정의하기만 하면 된다. 본래 이러한 생성자 함수는 다음과 같은 일을 한다.

 - 객체 유형의 프로퍼티에 값을 할당한다.
 - 객체 유형의 메소드로 사용할 수 있는 다른 함수를 지정한다.
- 객체 사용 예
 - table 객체의 정의 (table.js)


```
function table_getValue(row,col){
    return this.data[row* this.columns+col ];
}
function table_setValue(row,col,value){
    this.data[row* this.columns+col ]=value;
}
```

```

function table_set(contents){
    var n=contents.length;
    for(var j=0;j<this.data[j]=contents[j];
    }
function table_isHeader(row,col){
    return this.header[row* this.columns+col ];
}
function table_makeHeader(row,col){    this.header[row* this.columns+col ]=true;
}
function table_makeNormal(row,col){    this.header[row* this.columns+col ]=false;
}
function table_makeHeaderRow(row){    for(var j=0;j< this.columns+j)
    this.header[row* this.columns+col ]=true;
}
function table_makeHeaderColumn(col){    for(var i=0;i< this.rows;++i)
    this.header[i* this.columns+col ]=true;
}
function table_write(doc){
    doc.write("<TABLE BORDER="+ this.border+">");
    for(var i=0;i< this.rows;++i) {
        doc.write("<TR>");
        for(var j=0;j< this.columns;++j) {
            if( this.header[i* this.columns+j ]) {
                doc.write("<TH>");
                doc.write( this.data[i* this.columns+j ]);
                doc.write("</TH>");
            }
            else {
                doc.write("<TD>");
                doc.write( this.data[i* this.columns+j ]);
                doc.write("</TD>");
            }
        }
        doc.writeln("</TR>");
    }
    doc.writeln("</TABLE>");
}
function table(rows,columns) {
    this.rows=rows
    this.columns=columns
    this.border=0

```

```

this.data=new Array(rows*columns)
this.header=new Array(rows*columns)
this.getValue=table_getValue
this.setValue=table_setValue
this.set=table_set
this.isHeader=table_isHeader
this.makeHeader=table_makeHeader
this.makeNormal=table_makeNormal
this.makeHeaderRow=table_makeHeaderRow
this.makeHeaderColumn=table_makeHeaderColumn
this.write=table_write
}

```

- o table 객체 사용하기

```

<HTML>
<HEAD>
<TITLE>Defining Object Types</TITLE>
<SCRIPT LANGUAGE="JavaScript" SRC="table.js"><!--
//--></SCRIPT>
</HEAD>
<BODY>
<H1>Defining Object Types</H1>
<SCRIPT LANGUAGE="JavaScript"><!--
t=new table(3,4)
contents=new
Array("This","is","a","test","of","the","table","object.","Let's","see","it","work")
t.set(contents)
t.border=4
t.makeHeaderColumn(0)
t.write(document)
//--></SCRIPT>
</BODY>
</HTML>

```

- o 객체 유형에 프로퍼티와 메소드 추가

: prototype 프로퍼티를 통해서 인스턴스화할 수 있는 미리 정의된 객체 유형에 프로퍼티와 메소드 추가
사용 예

```

<HTML>
<HEAD>
<TITLE>Updating Object Types </TITLE>
<SCRIPT LANGUAGE="JavaScript" SRC="table.js"><!--
//--></SCRIPT>
</HEAD>

```



```

<BODY>
<H1>Updating Object Types</H1>
<SCRIPT LANGUAGE="JavaScript"><!--
function table_colorWrite(doc){
    .....
    함수 정의
    .....
    .....
}
t=new table(3,4)
table.prototype.bgColor="Cyan"
table.prototype.colorWrite=table_colorWrite
.....
.....
t.colorWrite(document)
//--></SCRIPT>
</BODY>
</HTML>

```

- 프로퍼티와 메소드 삭제

```

delete objectName.propertyName
delete objectName.methodName

```

브라우저 객체

객 체	용 도
window 객체	브라우저 윈도우나 윈도우 안에 있는 프레임에 액세스할 때 사용한다. 프로퍼티와 메소드를 참조할 때, window 객체가 존재하는 경우에는 "window." 접두사를 붙일 필요가 없다
document 객체	현재 윈도우에 로드되어 있는 문서에 액세스할 때 사용한다. document 객체는 콘텐츠를 제공하는 HTML 문서, 즉 HEAD 와 BODY 태그가 있는 문서를 의미한다.
location 객체	URL 을 나타낼 때 사용한다. 이 객체는 URL 객체를 만들거나 URL 의 일부분에 액세스하거나 기존의 URL 을 수정할 때 사용할 수 있다.
history 객체	한 윈도우 안에서 액세스된 URL 의

	히스토리를 유지할 때 사용
frame 객체, frames 배열	HTML 프레임에 액세스할 때 사용 frames 배열은 윈도우안에 있는 모든 프레임에 액세스할 때 사용
link 객체, links 배열	하이퍼텍스트 링크의 텍스트 기반이나 이미지 기반 소스 앵커(anchor)에 액세스할 때 사용 links 배열은 문서 안에 있는 모든 link 객체에 액세스할 때 사용한다. I.E.는 link 객체의 anchor 객체를 결합한다.
anchor 객체, anchors 배열	하이퍼텍스트 링크의 타겟에 액세스할 때 사용 anchors 배열은 문서 안에 있는 모든 anchor 객체에 액세스할 때 사용
image 객체, images 배열	HTML 문서에 삽입되어 있는 이미지에 액세스할 때 사용 images 배열은 문서 안에 있는 모든 image 객체에 액세스할 때 사용
area 객체	클라이언트측 이미지맵 안에 있는 영역에 액세스할 때 사용
applet 객체, applets 배열	Java 애플릿에 액세스할 때 사용 애플릿 배열은 문서 안에 있는 모든 애플릿에 액세스할 때 사용
event 객체, Event 객체	이벤트 발생에 대한 정보에 액세스할 때 사용 event 객체는 특정 이벤트에 대한 정보 제공 Event 객체는 이벤트를 식별하는 데 사용하는 상수 제공
form 객체, forms 배열	HTML 폼에 액세스시 사용 forms 배열은 문서 안에 있는 모든 폼에 액세스시 사용
element 객체	폼 안에 들어있는 모든 폼 요소에 액세스시 사용
text 객체	폼의 텍스트 필드에 액세스시 사용
textarea 객체	폼의 텍스트 영역 필드에 액세스시 사용
radio 객체	폼의 라디오 버튼 세트에 액세스하거나 세트 안에 있는 각각의 버튼에 액세스할 때 사용
checkbox 객체	폼의 체크 박스에 액세스할 때 사용
button 객체	Submit 나 Reset 버튼이 아닌 폼 버튼에 액세스시 사용
reset 객체	폼의 Reset 버튼에 액세스시 사용

selet 객체	폼의 선택 리스트에 액세스시 사용
option 객체	option 객체는 선택 리스트의 요소에 액세스시 사용
password 객체	폼의 패스워드 필드에 액세스시 사용
hidden 객체	폼의 숨겨진 필드에 액세스시 사용
FileUpload 객체	폼의 파일 업로드 요소에 액세스시 사용
navigator 객체	스크립트를 실행하고 있는 브라우저에 대한 정보에 액세스시 사용
screen 객체	사용자의 화면의 색상 깊이와 크기에 대한 정보에 액세스시 사용
embed 객체, embeds 배열	삽입된 객체에 액세스시 사용 embeds 배열은 문서 안에 삽입된 모든 객체에 대한 액세스 제공
contentType 객체, contentType 배열	브라우저가 제공하는 특정 MIME 유형에 대한 정보에 액세스시 사용 contentType 배열은 제공하는 모든 contentType 객체의 배열 I.E.는 빈 배열을 리턴하면서 contentType 에 대해서 임시적으로 지원
plugin 객체, plugins 배열	특정 브라우저 플러그인에 대한 정보에 액세스시 사용 plugins 배열은 브라우저가 지원하는 모든 플러그인의 배열 I.E.는 빈 배열을 리턴하면서 plugins 에 대해서 임시적으로 지원

window 객체

- 모든 브라우저 스크립트의 기본적인 것으로, 브라우저가 자동으로 정의하는 최상위 레벨의 객체이다. 현재 열려 있는 각 윈도우에 대해 별도의 window 객체가 정의된다.

window 객체의 프로퍼티

프로퍼티	설 명
closed	윈도우가 닫혀 있는지 식별
defaultStatus	브라우저 윈도우의 하단의 상태바에

	나타나는 디폴트 상대 메시지를 지정
document	윈도우에 표시되어 있는 현재 문서를 지정하는 객체
frames	윈도우 객체에 들어 있는 모든 프레임 객체로 구성된 배열
history	마지막으로 윈도우로 로드된 URL 의 리스트를 포함하는 윈도우의 히스토리 객체
length	window 에 들어 있는 프레임의 수 식별
location	window 객체와 관련된 URL 을 지정하는 객체
name	윈도우의 이름 지정
offscreenBuffering	윈도우 정보의 오프스크린 버퍼링이 사용될 것인지를 지정하는 부울값 오프스크린 버퍼링은 윈도우를 나타내기 전에 윈도우의 모든 요소를 로드할 때 사용
opener	윈도우를 만들거나 열 수 있도록 해주는 window 객체 지정
parent	특정 윈도우를 포함하는 윈도우를 지정하는 시너님
self	참조될 현재 윈도우를 지정하는 시너님
status	브라우저 윈도의 하단의 상태 표시줄에 나타날 임시 메시지를 지정
top	중첩된 일련의 윈도우에서 맨 위에 있는 브라우저 윈도우를 의미하는 시너님
window	참조될 현재 윈도우를 식별하는 시너님

window 객체의 메소드	
메 소 드	설 명
alert(text)	경고 다이얼로그 박스를 표시
blur()	포커스를 윈도우에서 옮긴다
setInterval(expression,milliseconds)	지정된 타임아웃 인터벌이 지난 이후에 표현식을 반복해서 평가하거나 함수를 불러온다.
clearInterval(interval)	이전에 설정된 인터벌 타이머를 클리어

setTimeout(expression,milliseconds)	타임아웃 기간이 지난 이후에 표현식을 평가하거나 함수를 호출한다.
clearTimeout(timer)	이전에 설정된 타임아웃을 클리어
close()	지정된 윈도우를 닫는다.
confirm(text)	확인 다이얼로그 박스를 나타낸다.
focus()	윈도우로 포커스를 가져간다.
open(url,name,[options])	새로운 윈도우를 열고 새로운 window 객체를 만듦
prompt(text,defaultInput)	프롬프트 다이얼로그 박스를 나타낸다.
scroll(x,y)	윈도우를 특정 위치까지 스크롤한다.

open() 메소드의 옵션		
옵 션	값	설 명
toolbar	yes no	윈도우에 툴바 제공
location	yes no	윈도우에 위치 필드 제공
directories	yes no	디렉토리 버튼 제공
status	yes no	상태 표시줄 제공
menubar	yes no	메뉴바 제공
scrollbars	yes no	스크롤 바 제공
resizable	yes no	윈도우 크기 조절 가능
width	정수	윈도우의 폭(픽셀)
height	정수	윈도우의 높이(픽셀)

frame 객체

- 프레임은 윈도우를 독립된 표시 영역들로 분할한 후, 이 영역들에 표시되는 정보들을 강력하게 컨트롤 할 수 있게 해준다.
- 프레임 객체의 프로퍼티와 메소드는 window 객체와 같지만, close() 메소드는 지원하지 않는다.

document 객체

- JavaScript 에서 아주 중요한 객체로, 이 객체를 사용하면 로드될 문서를 업데이트하고 로드된 문서 안에 있는 HTML 요소에 액세스할 수 있다.

프로퍼티	설 명
alinkColor	<BODY> 태그의 alink 속성의 값 지정
anchor	문서에 들어 있는 배열을 참조하는 객체
anchors	문서에 들어 있는 모든 앵커의 배열
applet	문서에 들어 있는 애플릿을 참조하는 객체
applets	문서에 들어 있는 모든 애플릿의 배열
area	문서 안의 이미지맵 영역을 참조하는 객체
bgColor	<BODY> 태그의 bgColor 속성의 값 식별
cookie	쿠키의 값 식별
domain	문서가 로드되는 서버의 도메인 이름 식별
embeds	문서안의 모든 플러그인의 배열
fgColor	<BODY> 태그의 text 속성값 지정
form	문서 안의 폼을 참조하는 객체
Forms[]	문서 안의 모든 폼의 배열
image	문서 안의 이미지를 참조하는 객체
Images[]	문서 안의 모든 이미지의 배열
lastModified	문서가 마지막으로 수정된 날짜
link	문서 안의 링크를 참조하는 객체
links	문서 안의 모든 링크의 배열
linkColor	<BODY> 태그의 link 속성의 값 식별
plugin	문서 안의 플러그인을 참조하는 객체
plugins[]	브라우저가 지원하는 플러그인을 나타내는 객체의 배열
referrer	문서에 대한 링크를 제공하는 문서의 URL
title	문서의 타이틀
URL	문서의 URL
vlinkColor	<BODY> 태그의 vlink 속성의 값 식별
메 소 드	설 명
close()	문서의 객체를 만드는 데 사용된 스트림
open([mimeType][,"replace"])	선택적인 MIME 유형으로 문서 객체를 만들 때 사용되는 흐름을 개시한다. "replace" 파라미터는 text/html MIME 유형과 함께

	사용되어 히스토리 리스트에 있는 현재 문서를 대체
<code>write(expr1[,expr2...,exprN])</code>	문서에 표현식의 값을 기록
<code>write(expr1[,expr2...,exprN])</code>	개행 문자가 다음에 따라오는 문서에 표현식의 값 기록

navigator 객체

- navigator 객체는 window 객체와 마찬가지로 브라우저 객체 모델에서 최상위 레벨의 객체이며, 스크립트를 실행할 때 사용되는 브라우저의 종류와 버전에 대한 정보 제공한다.

프로퍼티	브라우저 지원	설 명
<code>appCodeName</code>	N2, I.E3	브라우저 색상 이름
<code>AppMinorVersion</code>	I.E4	브라우저 버전 번호
<code>appName</code>	N2, I.E3	브라우저 이름
<code>appVersion</code>	N2, I.E3	브라우저의 버전
<code>browserLanguage</code>	I.E4	브라우저에 설정되어 있는 언어
<code>connectionSpeed</code>	I.E4	브라우저가 네트워크에 연결되는 속도
<code>cookieEnabled</code>	I.E4	브라우저가 쿠키를 허용하도록 설정되어 있는지의 여부
<code>cpuClass</code>	I.E4	브라우저 실행시 사용되는 마이크로프로세서의 유형
<code>onLine</code>	I.E4	브라우저가 현재 온라인 연결을 가지고 있는지 여부
<code>Language</code>	N4, I.E4	브라우저에 설정되어 있는 언어
<code>mimeTypes</code>	N3, I.E4	현재 브라우저가 지원하는 모든 MIME 유형의 배열
<code>platform</code>	N4, I.E4	브라우저가 실행되는 운영체제 플랫폼
<code>plugins</code>	N3, I.E4	현재 브라우저에 설치된 모든 플러그인의 배열
<code>systemLanguage</code>	I.E4	운영체제의 디폴트 언어
<code>userAgent</code>	N2, I.E3	브라우저에서 서버로 전송된 HTTP 프로토콜의 사용자 에이전트 헤더
<code>userLanguage</code>	I.E4	사용자가 사용하는 언어
<code>userProfile</code>	I.E4	사용자 프로파일 정보에 대한

	엑세스를 제공하는 객체
--	--------------

navigator 객체의 메소드	
메 소 드	설 명
javaEnabled()	사용자가 브라우저의 Java 기능을 켜두었는지의 여부를 나타내는 부울값 리턴
taintEnabled()	사용자가 data tainting(보안 메커니즘)을 활성화했는지 여부를 나타내는 부울값 리턴
preference	서명이 되지 않은 스크립트가 보안 관련 프로퍼티를 얻고 설정할 때 사용

event 객체

event 객체의 프로퍼티

프로퍼티	브라우저	설 명
data	N	DragDrop 이벤트로 인해 드롭된 객체의 URL 이 들어있는 스트링 배열
height, width	N	윈도우나 프레임의 길이와 너비(픽셀표시)
pageX, pageY	N	픽셀로 나타낸 커서의 수평/수직 위치(페이지에서의 상대적위치)
screenX, screenY	N, I.E	픽셀로 나타낸 커서의 수평/수직 위치(화면에서의 상대적 위치)
layerX, layerY	N	픽셀로 나타낸 커서의 수평/수직 위치, 이벤트가 발생한 레이어에 대한 상대적 위치. Resize 이벤트와 함께 사용하면 layerX 와 layerY 가 이벤트가 타겟으로 하는 객체의 길이와 너비 지정
clientX 와 clientY	I.E	픽셀로 나타낸 커서의 수평/수직 위치, 이벤트가 발생한 웹페이지에서의 상대적 위치
offsetX, offsetY	I.E	픽셀로 나타낸 커서의 수평/수직 위치, 이벤트가 발생한 컨테이너에 대한 상대적 위치
X, Y	I.E	픽셀로 나타낸 커서의 수평/수직 위치, 이벤트가 발생한 문서에 대한 상대적 위치
target	N	이벤트가 전송된 원래 객체
srcElement	I.E	이벤트가 전송된 원래 객체
type	N, I.E	발생한 이벤트 유형
which	N	눌려진 마우스 버튼(왼:1, 가:2, 오:3)이나 눌려진 키의 ASCII 값
keyCode	I.E	키 누름과 연관된 Unicode 키 코드를 식별
button	I.E	이벤트가 발생했을 때 눌려진 마우스 버튼 식별(0:눌려진버튼없음, 1:왼, 2:오, 4:가)

modifiers	N	마우스나 키 이벤트와 연관된 수정자 키(ALT_MASK,CONTROL_MASK,SHIFT_MASK,META_MASK)를 식별
altkey,ctrlkey,shiftkey	I.E	true 나 false 로 설정하여 이벤트가 발생했을 때 Alt 키와 Control 키, Shift 키 중에 어떤 것이 눌러졌는지 알려준다.
cancelBubble	I.E	true 나 false 로 설정하여 이벤트 버블링을 취소하거나 활성화한다.
fromElement, toElement	I.E	이동 중인 HTML 요소 지정
reason	I.E	데이터 소스 객체에 대한 데이터 전송 상태를 나타내는데 사용
returnValue	I.E	true 나 false 로 설정하여 이벤트 핸들러의 리턴값을 나타낸다. 이벤트 핸들러에서 true 나 false 를 리턴하는 것과 같다.
srcFilter	I.E	onfilterchange 이벤트를 시작하는 filter 객체 지정

screen 객체

- screen 객체의 프로퍼티

- height : 사용자의 화면의 현재 높이(픽셀)
- width : 사용자의 화면의 현재 너비(픽셀)
- colorDepth : 사용자의 화면/비디오 카드에서 현재 지원하는 색상당 바이트 수

form 객체

- document 객체의 프로퍼티로 액세스되고, form 객체는 문서 안의 폼에 액세스할 수 있도록 해주고, 폼 관련 이벤트에 응답을 할 수 있도록 해주기 때문에 중요하다.

form 객체의 프로퍼티	
프로퍼티	설 명
action	<FORM> 태그의 HTML action 속성에 대한 액세스 제공
button	GUI 컨트롤 버튼을 나타내는 객체
checkbox	체크 박스 필드를 나타내는 객체
elements	폼 안에 포함되어 있는 모든 필드와 GUI 컨트롤을 포함하는 배열
encoding	<FORM> 태그의 HTML enctype 속성에 대한 액세스 제공

FileUpload	파일 업로드 폼 필드를 나타내는 객체
hidden	숨겨진 폼 필드를 나타내는 객체
length	elements 배열의 길이에 대한 액세스 제공
method	<FORM> 태그의 HTML method 속성에 대한 액세스 제공
name	폼의 이름 식별
password	패스워드 필드를 나타내는 객체
radio	라디오 버튼 필드를 나타내는 객체
reset	reset 버튼을 나타내는 객체
select	선택 항목 리스트를 나타내는 객체
submit	submit 버튼을 나타내는 객체
target	<FORM> 태그의 HTML target 속성에 대한 액세스 제공
text	텍스트 필드를 나타내는 객체
textarea	텍스트 영역 필드를 나타내는 객체

form 객체의 메소드	
메 소 드	설 명
handleEvent()	지정된 이벤트에 대한 폼의 이벤트 핸들러를 호출할 때 사용
submit()	폼을 제출시 사용
reset	폼의 엔트리를 디폴트 값으로 재설정시 사용

form 요소 객체의 프로퍼티		
객 체	프로퍼티	설 명
button	name	버튼의 name 속성에 대한 액세스 제공
	type	객체의 유형 지정
	value	객체의 값 지정
checkbox	checked	체크박스가 현재 체크되어 있는지를 식별
	defaultChecked	체크박스가 디폴트로 체크되어 있는지 식별
	name	체크박스의 HTML name 속성에 대한 액세스 제공
	type	객체의 유형 식별

	value	객체의 값 식별
FileUpload	name	객체의 name 속성에 대한 액세스 제공
	type	객체의 type 속성에 대한 액세스 제공
	value	객체의 값 식별
hidden	name	객체의 name 속성에 대한 액세스 제공
	type	객체의 유형 식별
	value	객체의 값 식별
password	defaultChecked	객체의 디폴트 값 식별
	name	객체의 name 속성에 대한 액세스 제공
	type	객체의 유형 식별
	value	객체의 값 식별
radio	checked	라디오 버튼이 체크되어 있는지 식별
	defaultChecked	라디오 버튼이 디폴트로 체크되어 있는지 식별
	name	객체의 name 속성에 대한 액세스 제공
	type	객체의 유형 식별
	value	객체의 값 식별
reset	name	객체의 name 속성에 대한 액세스 제공
	type	객체의 유형 식별
	value	객체의 값 식별
select	length	선택 리스트의 길이 식별
	name	객체의 name 속성에 대한 액세스 제공
	option	선택 리스트가 제공하는 옵션 식별하는 배열
	selectedIndex	선택 리스트 안에서 처음 선택된 옵션 식별
	type	객체의 유형 식별
submit	name	객체의 name 속서에 대한 액세스 제공
	type	객체의 유형 식별
	value	객체의 값 식별

text	defaultValue	텍스트 필드에 나타나는 디폴트 텍스트를 식별
	name	객체의 name 속성에 대한 액세스 제공
	type	객체의 유형 식별
	value	객체의 값 식별
textarea	defaultValue	텍스트 영역 필드에 나타나게 되는 디폴트 텍스트 식별
	name	객체의 name 속성에 대한 액세스 제공
	type	객체의 유형 식별
	value	객체의 값 식별

form 요소 객체의 메소드		
객 체	메 소 드	설 명
button	Click()	클릭된 버튼 시뮬레이션
	blur()	포커스 잃음
	focus()	포커스 맞춤
checkbox	Click()	클릭된 체크 박스 시뮬레이션
	blur()	포커스 잃음
	focus()	포커스 맞춤
FileUpload	blur()	포커스 잃음
	focus()	포커스 맞춤
	select()	입력 영역을 선택
hidden	없음	
password	blur()	포커스 잃음
	focus()	포커스 맞춤
	select()	패스워드 필드에 나타나는 텍스트를 하이라이트
radio	Click()	라디오 버튼 클릭 시뮬레이션
	blur()	포커스 잃음
	focus()	포커스 맞춤
reset	Click()	리셋 버튼 클릭 시뮬레이션
	blur()	포커스 잃음
	focus()	포커스 맞춤
select	blur()	포커스 잃음

	focus()	포커스 맞춤
submit	Click()	제출 버튼 클릭 시뮬레이션
	blur()	포커스 잃음
	focus()	포커스 맞춤
text	blur()	포커스 잃음
	focus()	포커스 맞춤
	select()	텍스트 필드에 있는 텍스트 하이라이트
textarea	blur()	포커스 잃음
	focus()	포커스 맞춤
	select()	텍스트 영역에 있는 텍스트 하이라이트

location 객체

- 원도우에 로드되어 있는 현재 문서의 URL 에 액세스하거나 새로운 문서를 로드할 때 사용

location 객체의 프로퍼티	
프로퍼티	설 명
hash	URL 의 앵커 부분(존재하는 경우)
host	URL 의 hostname:port 부분
hostname	URL 의 host 부분
href	전체 URL
pathname	URL 의 경로명 부분
port	URL 의 포트 부분
protocol	URL 의 프로토콜 부분
search	URL 의 쿼리 스트링 부분

location 객체의 메소드

- reload() : 윈도우의 현재 문서를 브라우저의 Reload 버튼에서 사용하는 정책에 따라 다시 로드
 - Every time : 문서는 매번 서버에서 다시 로드
 - Once per session : 서버의 문서의 날짜가 캐시에 저장되어 있는 문서보다 더 최신 날짜라는 것을 나타내면 문서는 세션당 한 번씩 서버에서 다시 로드된다. 문서가 캐시에 없는 경우에는 서버에서 로드
 - Never : 문서는 가능하면 캐시에서 다시 로드, 그렇지 않으면 서버에서 로드
- replace() : URL 을 파라미터로 취하여, 현재 문서 히스토리 목록에 있는 현재 문서위로 그 URL 의 문서를 로드, 그러면 브라우저의 Back 버튼을 클릭하여 이전 문서로 돌아갈 수 없음

link 객체

- link 객체는 document 객체의 프로퍼티로, 문서에 들어있는 텍스트나 이미지 링크를 캡슐화
- links 배열은 문서에 들어있는 모든 링크의 배열
- link 객체의 메소드
 - handleEvent() : event 객체를 인자로 취하며 그 이벤트에 대해 적당한 이벤트 핸들러 호출

link 객체의 프로퍼티	
프로퍼티	설 명
hash	URL의 앵커 부분(존재하는 경우)
host	URL의 hostname:port 부분
hostname	URL의 host 부분
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

anchor 객체

- HTML 문서 안에서 이름이 지정된 오프셋으로 사용되는 앵커 의미
- anchors 배열에는 문서의 모든 앵커가 들어있음
- 프로퍼티나 메소드 또는 이벤트를 전혀 가지고 있지 않음
- HTML 문서와 관련하여 정의된 이름이 지정된 오프셋을 추적할 때 사용
- anchor 객체는 HREF 속성을 포함하는 경우에 link 객체가 된다.

history 객체

- history 객체의 프로퍼티
 - current : 윈도우에 나타나는 현재 문서의 URL
 - length : History 리스트의 길이
 - next : History 리스트에서의 다음 URL
 - previous : History 리스트에서의 이전 URL
- history 객체의 메소드
 - back() : History 리스트에 이전 문서를 로드, 브라우저의 Back 버튼을 클릭하는 것과 같은 효과
 - forward() : History 리스트에 다음 문서를 로드, 브라우저의 Forward 버튼을 클릭하는 것과 같은 효과
 - go() : History 리스트에 있는 특정 문서로 감

- `go(n)` : $n > 0$ 인 경우, 이 메소드는 History 리스트에서 n 개의 엔트리가 앞에 있는 문서를 로드, $n = 0$ 인 경우에는 현재 문서가 다시 로드되고, $n < 0$ 인 경우엔 History 리스트에서 n 개의 엔트리 뒤에 있는 문서를 로드
- `go(string)` : `go()`는 이 스트링을 서브스트링으로 갖고 있는 URL 의 History 리스트에서 가장 가까운 문서를 로드

image 객체

- document 객체의 프로퍼티
- 문서와 함께 로드된 이미지에 대한 액세스 제공
- images 배열은 문서 안에 지정되어 있는 각각의 태그에 대한 엔트리가 들어있음
- image 객체 유형을 사용하면 키워드와 생성자로 새로운 image 객체를 명시적으로 만들 수 있다. `Image()` 생성자는 웹페이지의 일부로서 처음에 나타나지 않는 이미지를 만들고 미리 로드할 때 사용한다. 이러한 image 객체는 브라우저의 캐시에 저장되면 이미 나타난 이미지를 대체할 때 사용
- * `Image()` 생성자를 사용하여 캐시된 이미지 만드는 예 `cachedImage=new Image()`
`cachedImage.src="myImage.gif"`
- = > 첫번째 문장은 새로운 image 객체를 만들고 그것을 `cachedImage` 변수에 대입하고, 두번째 문장은 image 객체의 `src` 프로퍼티를 `myImage.gif` 이미지 파일로 설정한다. 이 경우 `myImage.gif` 는 브라우저 캐시로 로드된다. 그러면 로드된 이미지는 `cachedImage` 변수를 사용하여 참조 가능하다.

image 객체의 프로퍼티	
프로퍼티	설 명
<code>border</code>	 태그의 BORDER 속성의 값
<code>complete</code>	이미지가 완전히 로드되었는지 식별
<code>height</code>	 태그의 HEIGHT 속성의 값
<code>hspace</code>	 태그의 HSPACE 속성의 값
<code>lowsrc</code>	 태그의 LOWSRC 속성의 값
<code>name</code>	 태그의 NAME 속성의 값
<code>prototype</code>	image 객체에 사용자 지정 프로퍼티를 추가할 때 사용
<code>src</code>	 태그의 SRC 속성의 값
<code>vspace</code>	 태그의 VSPACE 속성의 값
<code>width</code>	 태그의 WIDTH 속성의 값

area 객체

- 이미지맵은 여러 가지 다른 영역으로 나누어져 있는 이미지로서 각각의 영역은 자체 URL 과 관련되어 있다. 그리고 이러한 영역과 관련된 사용자 처리 방식으로 area 객체를 제공

area 객체의 프로퍼티	
프로퍼티	설 명
hash	area 객체의 HREF 속성의 파일 오프셋 부분
host	area 객체의 HREF 속성의 호스트 이름 부분
hostname	area 객체의 HREF 속성의 host:port 부분
href	area 객체의 완전한 HREF 속성
pathname	area 객체의 HREF 속성의 경로명 부분
port	area 객체의 HREF 속성의 포트 부분
protocol	area 객체의 HREF 속성의 프로토콜 부분
search	area 객체의 HREF 속성의 쿼리 스트링 부분
target	area 객체의 TARGET 속성

Array 객체

- Array 객체를 사용하면 배열을 객체처럼 액세스 가능
- Array 객체의 프로퍼티
 - length : 배열의 길이 식별
 - prototype : 모든 객체 유형이 지원하는 프로퍼티로 객체 유형에 대해 추가적인 프로퍼티 및 메소드 정의 가능
- Array 객체의 메소드
 - toString() : 배열의 스트링 버전 리턴, 배열 요소는 콤마로 구분
 - join(separator) : 배열의 스트링 버전 리턴, 배열 요소는 seperator 스트링으로 구분, 분리자가 없으면 콤마로 구분
 - reverse() : 배열의 요소를 역순으로 바꿈
 - sort(comparisionFunction) : 비교 연산에 따라 배열의 요소 정렬. 비교 함수가 지정되면, 배열 요소는 사전순서로 정렬. 비교 연산이 지정되면 두개의 파라미터 p1,p2 를 취하고, p1 이 p2 보다 작은 경우에는 음의 정수를 리턴하고, 같으면 0 을 리턴하고, p1 이 p2 보다 크면 양의 정수 리턴

Boolean 객체

- Boolean 객체를 사용하면 부울값은 객체로서 액세스 가능
- Boolean 객체는 생성자에 대한 인자로서 값을 식별하여 만들어짐


```
myBoolean=new Boolean(false)
yourBoolean=new Boolean(true)
```


- Boolean 객체의 프로퍼티
 - prototype : 모든 객체 유형이 지원하는 프로퍼티로 객체 유형에 대해 추가적인 프로퍼티 및 메소드 정의 가능
- Boolean 객체의 메소드
 - toString() : 부울값에 해당하는 스트링 리턴
 - valueOf() : 객체의 값에 따라 true 나 false 로 리턴

Date 객체의 메소드	
메 소 드	설 명
getDate() getUTCDate() setDate() setUTCDate()	Date 객체의 날짜를 설정하거나 리턴
getDay() getUTCDay()	Date 객체의 한 주의 날짜를 설정하거나 리턴
getHours() getUTCHours() setHours() setUTCHours()	Date 객체의 시간을 설정하거나 리턴
getMilliseconds() getUTCMilliseconds() setMilliseconds() setUTCMilliseconds()	Date 객체의 밀리초 값을 설정하거나 리턴
getMinutes() getUTCMinutes() setMinutes() setUTCMinutes()	Date 객체의 분을 설정하거나 리턴
getMonth() getUTCMonth() setMonth() setUTCMonth()	Date 객체의 달을 설정하거나 리턴
getSeconds() getUTCSeconds() setSeconds() setUTCSeconds()	Date 객체의 초를 설정하거나 리턴
getTime() getTime()	Date 객체의 시간을 설정하거나 리턴
getTimeZoneOffset()	Date 객체의 시간대 오프셋(분)을 리턴
getYear() getFullYear() getUTCFullYear()	Date 객체의 연도를 리턴하거나 설정한다. 완전한 연도를 나타내는 방법으로 4 자리 연도 값을 사용한다.

setYear() setFullYear() setUTCFullYear()	
toGMTString()	날짜를 Internet GMT(Greenwich Mean Time) 포맷의 스트링으로 변환
toLocaleString()	날짜를 로케일(locale)포맷의 스트링으로 변환. 로케일 포맷이란 사용자가 위치해 있는 지형적 위치에서 일반적으로 사용하는 포맷 의미
toString()	Date 객체의 스트링 값을 리턴
valueOf()	1970 년 1 월 1 일 자정 이후의 밀리초 값 리턴
toUTCString()	UTC 에서 시간을 나타내는 스트링 값을 리턴

Date 생성자	
생 성 자	설 명
Date()	현재 날짜와 시간으로 Date 인스턴스 만듦
Date(dateString)	dateString 파라미터에 지정되어 있는 날짜로, Date 인스턴스를 만든다. dateString 의 포맷은 "월,일,연도,시:분:초"
Date(milliseconds)	1970 년 1 월 1 일 자정 이후 지정된 밀리초 값으로 Date 인스턴스를 만든다.
Date(year,month, day,hours,minutes, seconds,milliseconds)	년,월,일,시,분,초,밀리초 정수에 따라 지정된 날짜로 Date 인스턴스를 만든다. 연도와 월 파라미터는 제공되어야 하고 다른 나머지 파라미터가 포함되면 앞에 오는 모든 파라미터가 제공되어야

Function 객체

- 함수를 객체처럼 액세스 가능하고, 이 객체는 스크립트를 실행하는 동안에 함수를 동적으로 만들고 호출할 때 사용
- Function 객체는 함수의 파라미터와 본문을 Function() 생성자에 제공하면 된다.

variable=new Function("p1","p2", ... ,"pn","body")

- Function 객체 프로퍼티
 - length : 함수에 대해 정의된 파라미터의 숫자 식별
 - prototype : 모든 객체 유형이 지원하는 프로퍼티로 객체 유형에 대해 추가적인 프로퍼티 및 메소드 정의 가능
 - arguments : 호출시 함수에 전달되는 인자를 가리키는 배열
 - caller : 특정 함수를 호출한 함수를 가리킴
- Function 객체의 메소드
 - toString() : 함수의 스트링 형태 리턴
 - valueOf() : 함수 자체 리턴

Global 객체

- new Global() 을 통해서 명시적으로 만들거나, 참조할 수 없다. 대신 해당 프로퍼티와 메소드가 전역 변수와 함수로 직접 참조됨
- Global 객체의 프로퍼티
 - Nan : 숫자가 아니라는 의미
 - Infinity : 양수 무한대 값 의미
- Global 객체의 메소드
 - escape(string) : string 을 새로운 스트링으로 변환
 - eval(x) : 표현식 x 의 값을 계산하고 리턴
 - isFinite(number) : number 가 유한하면 true 를 리턴하고, 무한하면 false 를 리턴
 - isNaN(number) : number 가 숫자가 아니면 true 를 리턴하고, 숫자이면 false 를 리턴
 - parseFloat(string) : string 을 부동 소수 값으로 파싱
 - parseInt(string, radix) : string 을 밑이 radix 인 정수로 파싱
 - unescape(string) : escape()에 들어 있는 스트링을 원래의 값으로 되돌린다.

Math 객체

- 수학적 상수와 함수의 표준 라이브러리 제공
- Math 의 특수 인스턴스는 Math 가 내장 객체이고 객체 유형이 아니기 때문에 만들어지지 않는다.

Math 프로퍼티	
프로퍼티	설 명
E	Euler 의 상수. 이것은 계산을 하는 어디에서나 발견할 수 있으며 자연대수의 기초
LN2	2 의 자연대수. 이것은 자연대수와 밑이 2 인 대수 사이의 전환에 사용되는 간단한 상수
LN10	10 의 자연대수. LN2 와 마찬가지로 대수 변환에 사용

LOG2E	밑이 2 인 E 의 대수. 이것은 밑이 10 인 대수를 밑이 E 인 대수로 변환시 사용
PI	원의 원주 대 지름의 비율
SQRT1_2	1/2 의 제곱근은 많은 삼각법 계산에서 사용
SQRT2	2 의 제곱근은 대수식에서 흔히 사용

Math 메소드	
메 소 드	설 명
abs(x)	x 의 절대값 리턴
acos(x)	x 의 아크 코사인값 라디안으로 리턴
asin(x)	x 의 아크 사인값 라디안으로 리턴
atan(x)	x 의 아크 탄젠트 값을 라디안으로 리턴
atan2(x,y)	(x,y)에 해당하는 극좌표의 각도를 리턴
ceil(x)	x 보다 크거나 작은 최소 정수값 리턴
cos(x)	x 의 코사인 값 리턴
exp(x)	e^x 를 리턴
floor(x)	x 보다 작거나 같은 최대 정수값 리턴
log(x)	x 의 자연대수 리턴
max(x,y)	x, y 중 큰 값 리턴
min(x,y)	x, y 중 작은 값 리턴
pow(x,y)	x^y 를 리턴
random()	0 과 1 사이의 임의의 숫자 리턴
round(x)	x 의 가장 가까운 정수로 반올림되는 값 리턴
sin(x)	x 의 사인값 리턴
sqrt(x)	x 의 제곱근 리턴
tan(x)	x 의 탄젠트 값 리턴

Number 객체

- Number 객체 유형을 사용하면 숫자를 객체로 다룰 수 있다.
- Number 객체의 프로퍼티
 - MAX_VALUE : 숫자는 가능한 최대 수치값
 - MIN_VALUE : 숫자는 가능한 최소 수치값
 - NaN : 숫자가 아니다
 - NEGATIVE_INFINITY : 숫자가 음수 무한대 값

- POSITIVE_INFINITY : 숫자가 양수 무한대 값
 - prototype : 모든 객체 유형이 지원
- Number 객체의 메소드
 - toString(radix) : 밑이 radix 인 숫자를 나타내는 스트링 리턴
 - valueOf() : Number 객체의 수치값 리턴

Object 객체

- Object 객체는 다른 모든 객체들이 파생되어 나가는 기반 객체로 이것의 프로퍼티와 메소드는 다른 객체 유형들에서 사용 가능
- Object 객체의 프로퍼티
 - prototype : 모든 객체 유형이 지원
 - constructor : 객체 생성자의 이름 식별
- Object 객체의 메소드
 - toString() : 객체를 스트링 표현으로 바꾸는 역할
 - valueOf() : 객체와 관련된 경우의 원시값(숫자,스트링,부울값)을 리턴하고, 그렇지 않은 경우에는 객체 자체를 리턴

String 객체

- 스트링을 객체로 액세스 가능
- String 객체의 프로퍼티
 - length : 문자에서 스트링의 길이 알아내는 역할
 - prototype : 모든 객체 유형이 지원

String 메소드	
메 소 드	설 명
charAt(index)	메소드가 적용되는 스트링의 지정된 인덱스에 있는 문자로 구성된 스트링을 리턴
charCodeAt(index)	지정된 인덱스의 문자의 Unicode 인코딩 리턴
fromCharCode(codes)	문자 코드의 콤마로 구분된 시퀀스에서 스트링 만듦
indexOf(pattern)	스트링안에 들어있는 pattern 파라미터가 지정한 첫 번째 스트링의 인덱스 리턴, 패턴이 스트링 안에 들어있지 않으면 -1 리턴
indexOf(pattern,startIndex)	startIndex 가 지정한 위치에서 검색을 시작하는

	것을 제외하면 indexOf(pattern) 메소드와 같다.
lastIndexOf(pattern)	스트링에 들어 있는 pattern 파라미터가 지정한 마지막 스트링의 인덱스 리턴, 패턴이 스트링 안에 들어있지 않으면 -1 리턴
lastIndexOf(pattern,startIndex)	startIndex 가 지정한 위치에서 검색을 시작하는 것을 제외하면 lastIndexOf(pattern)과 같다.
splitSeparator()	하나의 스트링을 분리자를 기반으로 하여 서브스트링의 배열로 분리
substring(startIndex)	startIndex 에서 시작하는 스트링의 서브스트링을 리턴
substring(startIndex,endIndex)	startIndex 에서 시작하고, endIndex 에서 끝나는 스트링의 서브스트링을 리턴
toLowerCase()	소문자로 변환된 스트링의 복사본 리턴
toString()	객체의 스트링 값을 리턴
toUpperCase()	대문자로 변환된 스트링의 복사본 리턴