Chapter08 JavaScript 시작하기

HTML & JavaScript

Contents

형식 : <BODY 속성=값 속성=값.....> 내용 </BODY>

속성 :
TOPMARGIN, LEFTMARGIN, BACKGROUND, BGCOLOR, TEXT, LINK, VLINK, ALINK

예제 :

```
<BODY BGCOLOR="yellow" TEXT="blue" LINK="red"
        VLINK="green" ALINK="gray">

BODY 태그의 속성

</BODY>
```

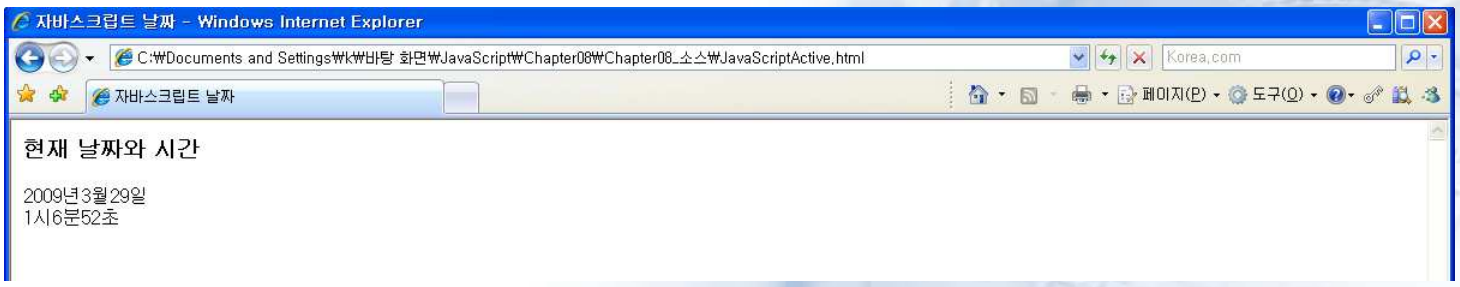
자바스크립트 정의

❖ 자바스크립트(JavaScript)

- 사용자와 페이지 간의 상호작용이 가능하게 하기 위해 사용
- Sun Microsystem사와 Netscape사가 공동으로 제작하여 보급한 언어
- 자바스크립트를 사용함으로써 사용자에게 동적인 웹 페이지를 제공해 줄 수 있음
- 기본적으로 HTML 내에 직접 삽입하여 사용
- 특별한 자료형이 필요 없이 var로 변수를 선언하면 자동으로 입력한 값에 따라 타입이 변형됨

프로그램 8-1

```
document.write("<H3> 현재 날짜와 시간 </H3>");
today = new Date();
document.write(today.getYear(), "년",
               today.getMonth()+1, "월",
               today.getDate(), "일 <BR>");
document.write(today.getHours(), "시",
               today.getMinutes()+1, "분",
               today.getSeconds(), "초");
```



[그림 8-1] JavaScriptActive의 실행결과

자바와 자바스크립트의 차이점

특성	자바	자바스크립트
작성 방법	별도의 파일 작성	HTML 내에 직접 삽입
실행 방식	서버에서 컴파일 된 후 클라이언트에서 수행됨	클라이언트에서 직접 해석되고 실행됨
변수 선언	변수의 자료형을 반드시 선언해야 함	변수의 자료형을 선언할 필요가 없음
객체지향	모든 객체를 정의하여 사용할 수 있는 객체지향 언어	클래스 선언이나 상속등의 개념이 없으므로 완전한 객체지향 언어가 아님
보안	소스를 볼 수 없으므로 보안을 유지할 수 있음	소스보기 메뉴를 통하여 소스를 볼 수 있으므로 보안성을 가지지 못함

자바스크립트 특징

❖ 자바스크립트 특징

- 스크립트언어
- 객체기반 언어
- HTML내에 삽입하여 사용
- 개발 및 결과 확인이 쉬움
- 빠른 개발 가능
- 배우기 쉬움
- 환경에 독립적

자바스크립트 단점

❖ 자바스크립트 단점

- 보안에 취약함
  - HTML안에 삽입하여 사용 하므로 웹 브라우저 메뉴의 소스 메뉴로 쉽게 코드 내용을 확인할 수 있음
- 제공하는 메소드의 부족
  - 브라우저의 기능과 밀접한 관계가 있기 때문에 자바스크립트 자체에서 제공하는 메소드는 브라우저에 제한적

자바스크립트 기본 구조

❖ 자바스크립트 기본 구조

- <SCRIPT> 태그와 </SCRIPT> 태그 사이에 정의
- <SCRIPT> 태그의 LANGUAGE 속성의 값으로 'JavaScript'를 입력

형식 :

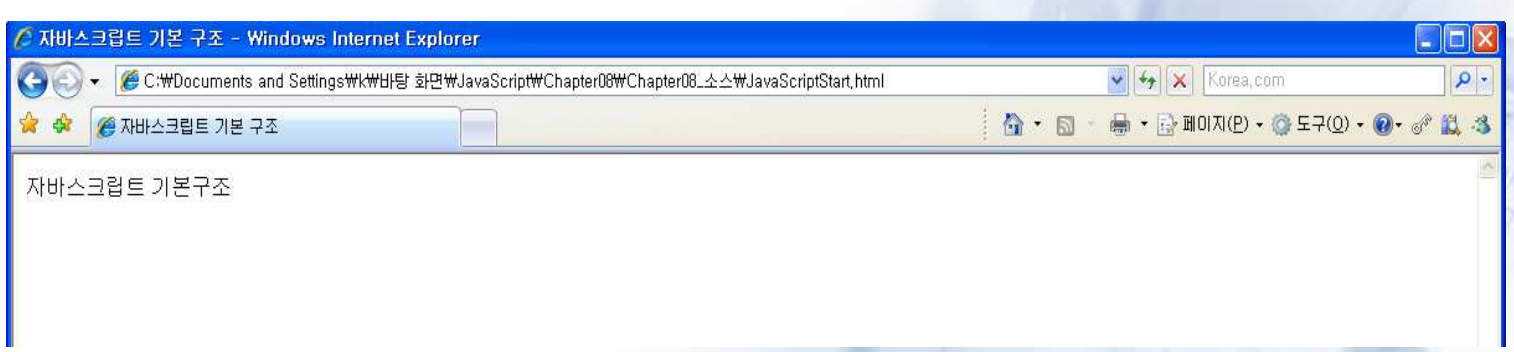
```
<SCRIPT LANGUAGE="JavaScript">
<!--
   자바스크립트 코드
//-->
</SCRIPT>
```

예제 :

```
<SCRIPT LANGUAGE="JavaScript">
<!--
   document.write("자바스크립트 기본구조");
//-->
</SCRIPT>
```

프로그램 8-2

```
<SCRIPT LANGUAGE="JavaScript">
<!--
    document.write("자바스크립트 기본구조");
//-->
</SCRIPT>
```



[그림 8-2] JavaScriptStart의 실행결과

자바스크립트 외부파일로 사용하기

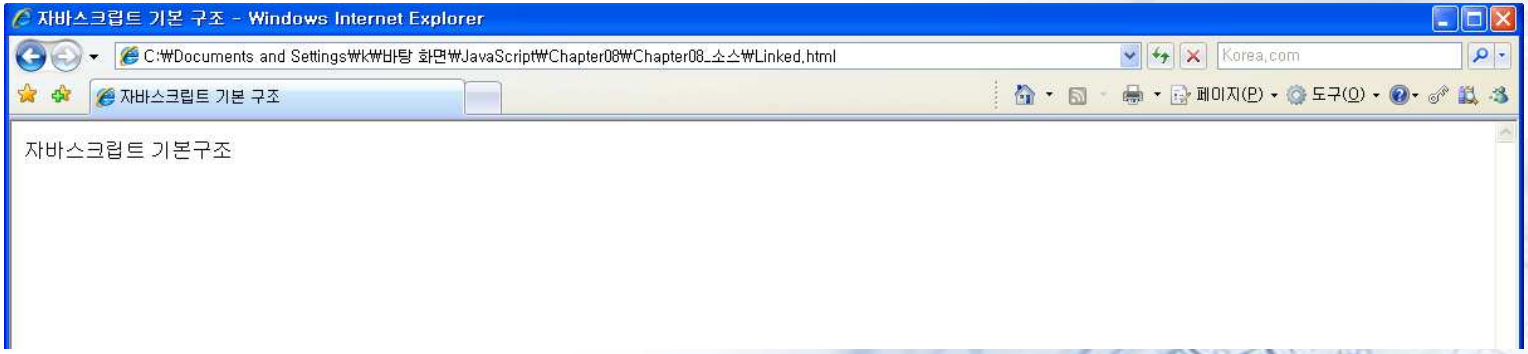
❖ 자바스크립트 외부파일로 사용하기

- 자바스크립트를 외부파일로 사용할 경우에는 확장자를 *.js로 지정해야 함
- 외부파일을 HTML 파일 안에서 호출할 경우에는 <SCRIPT> 태그의 SRC 속성의 값으로 파일의 이름을 입력

프로그램 8-3, 프로그램 8-4

```
<SCRIPT SRC="JavaScript.js">

</SCRIPT>
```



[그림 8-3] Linked의 실행결과

자바스크립트 변수

❖ 변수

- 어떤 값을 저장하기 위한 공간

❖ 식별자

- 변수나 함수의 이름이 될 수 있는 프로그래머가 선언하는 단어를 의미
- 식별자 규칙
  - 영문자와 숫자, '_' 로 이루어져 있다.
  - 특수 문자나 메타 문자, 한글은 사용할 수 없다.
  - 첫 자로는 영문자만 사용할 수 있다.
  - 대소문자의 구별이 있다.
  - 예약어(자바스크립트에서 미리 정의해 놓은 용어)는 식별자로 사용할 수 없다.

자바스크립트 예약어

abstract	case	continue	extends	for
import	long	private	static	throw
var	boolean	catch	default	false
function	in	native	protected	super
throw	void	break	char	do
final	goto	instanceof	new	public
switch	transient	while	byte	class
double	finally	if	int	null
return	synchronized	true	with	case
const	else	float	implements	interface
package	short	this	try	

사용 가능한 식별자와 사용 불가능한 식별자

식별자		설명
사용 가능한 식별자	strName	문자만으로 구성 되어 있으므로 식별자
	strName123	첫 문자가 영문자로 시작되고 문자와 숫자만으로 구성되어 있으므로 식별자
	strName_123	첫 문자가 문자로 시작되고 문자, 숫자, _로 이루어져 있으므로 식별자
사용 불가능한 식별자	123strName	첫 문자가 숫자로 시작되므로 식별자로 사용할 수 없음
	strName&123	특수문자 &가 포함되어 있으므로 식별자로 사용할 수 없음
	while	예약어는 식별자로 사용할 수 없음
	str!Name	특수문자 !가 포함되어 있으므로 식별자로 사용할 수 없음
	변수	한글은 식별자로 쓸 수 없음

자바스크립트 형변환

❖ 자바스크립트 형변환

- 자료형을 특별히 지정하지 않고 실행 시에 자동으로 값에 따라 형 변환이 되어 사용

형식 :

```
var 식별자
   or
   var 식별자1, 식별자2.....
```

예제 :

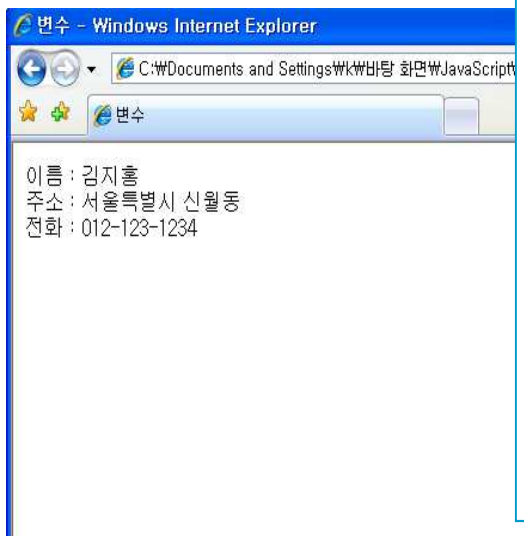
```
var strName;
var strName, intYear;
```

프로그램 8-5

```
<SCRIPT LANGUAGE="JavaScript">
<!--
    var strName, strAddress, strPhone;

    strName = "김지홍";
    strAddress = "서울특별시 신월동";
    strPhone = "012-123-1234";

    document.write("이름 : " + strName + "<BR>");
    document.write("주소 : " + strAddress + "<BR>");
    document.write("전화 : " + strPhone + "<BR>");
//-->
</SCRIPT>
```



[그림 8-4] Var의 실행결과

자바스크립트 자료형

❖ 자바스크립트 자료형

- 자료형의 선언이 필요 없이 var 자료형을 사용하고 저장되는 값에 따라 자동으로 형변환 됨

❖ 정수

- 제일 처음자리에 있는 숫자가 0일 경우 8진수로 인식하고 0x로 시작할 경우에는 16진수로 인식

❖ 부동소수점

- 12.34 또는 -12.34와 같이 소수가 포함되어 있는 10진수

자바스크립트 자료형

❖ Boolean

- true와 false 두 가지 값을 가지는 자료형

❖ Null

- 아무런 값도 없다는 것을 의미

❖ 문자열

- " "나 ' ' 사이에 들어가는 문자를 의미

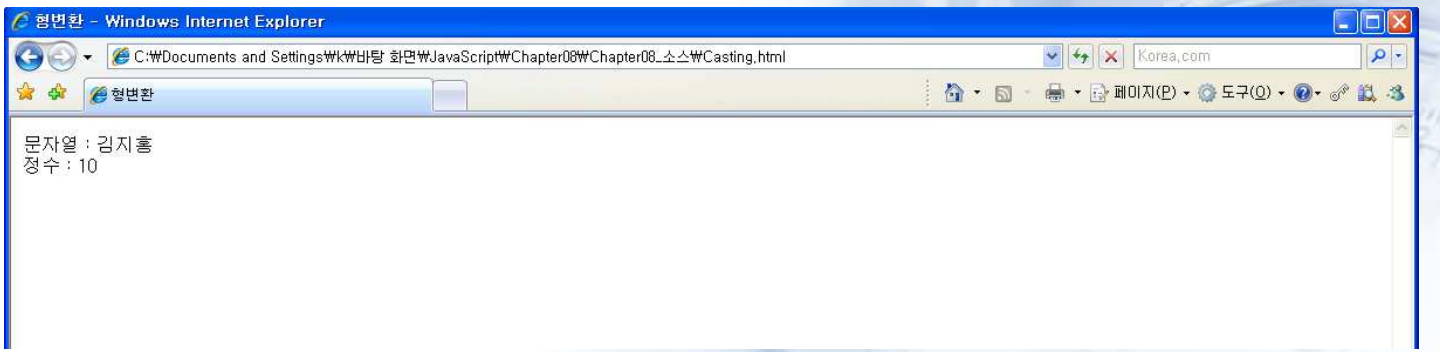
특수 문자의 종류

특수문자	설명
₩n	다음 줄로 이동
₩t	Tab 키 입력
₩b	BackSpace 키 입력
₩r	Return 키 입력
₩₩	₩ 기호 삽입
₩'	' 기호 삽입
₩"	" 기호 삽입

프로그램 8-6

```
<SCRIPT LANGUAGE="JavaScript">
<!--
    var myVar;
    myVar = "김지홍";
    document.write("문자열 : " + myVar + "<BR>");

    myVar = 10;
    document.write("정수 : " + myVar + "<BR>");
//-->
</SCRIPT>
```

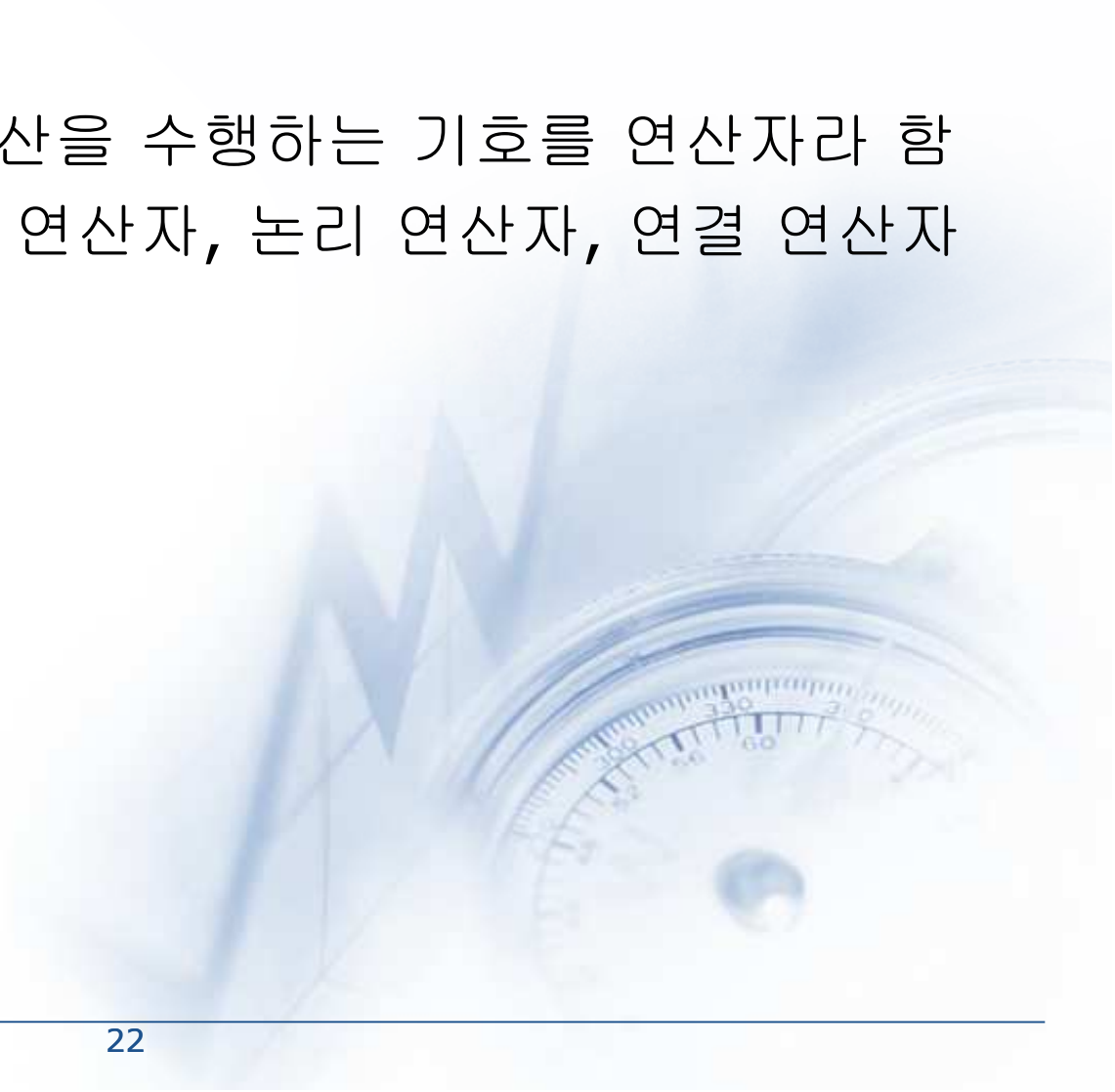


[그림 8-5] Casting의 실행결과

자바스크립트 연산자

❖ 자바스크립트 연산자

- 수학적, 논리적 연산을 수행하는 기호를 연산자라 함
- 산술 연산자, 비교 연산자, 논리 연산자, 연결 연산자 등이 있음



산술 연산자

❖산술 연산자

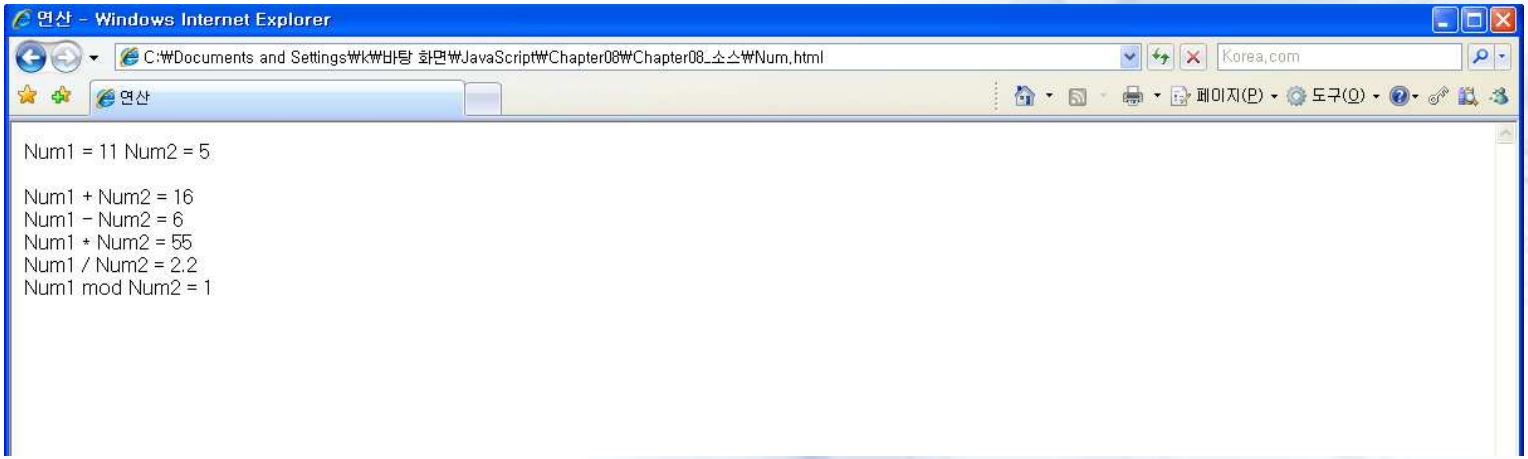
- 산술 계산을 위한 연산자
- 흔히 많이 사용하는 연산으로 덧셈, 뺄셈, 곱셈, 나눗셈, 나머지 등의 연산을 이야기 함

연산자	설명
+	더하기
–	빼기
*	곱하기
/	나누기
%	나머지

프로그램 8-7

```
    var intNum1, intNum2;
    intNum1 = 11;
    intNum2 = 5;

document.write("Num1 = " + intNum1 +
                " Num2 = " + intNum2  + "<BR><BR>");
document.write("Num1 + Num2 = " +
                (intNum1 + intNum2) + "<BR>");
document.write("Num1 - Num2 = " +
                (intNum1 - intNum2) + "<BR>");
```



[그림 8-6] Num의 실행결과

대입 연산자

❖ 대입 연산자

- '='를 말함
- 우변의 내용을 좌변의 변수에 대입할 경우 사용하는 연산자

❖ 연산 후 대입 연산자

- 좌변 변수의 원래 값에 일반적인 산술 연산 값을 저장할 경우 사용하는 연산자

연산자	설명
+=	덧셈 후 대입
-=	뺄셈 후 대입
*=	곱셈 후 대입
/=	나눗셈 후 대입
%=	나머지 연산 후 대입

증감 연산자

❖증감 연산자

- 증가, 감소 연산자는 변수의 앞이나 뒤에 붙여 사용되며 변수의 값을 1씩 증가시키거나 감소시킬 때 사용

연산자	설명
++	변수의 값 1씩 증가
--	변수의 값 1씩 감소

```
intY = ++intX; // 값 : intX = 11, intY = 11
intY = intX++; // 값 : intX = 11, intY = 10
```

비교 연산자

❖비교 연산자

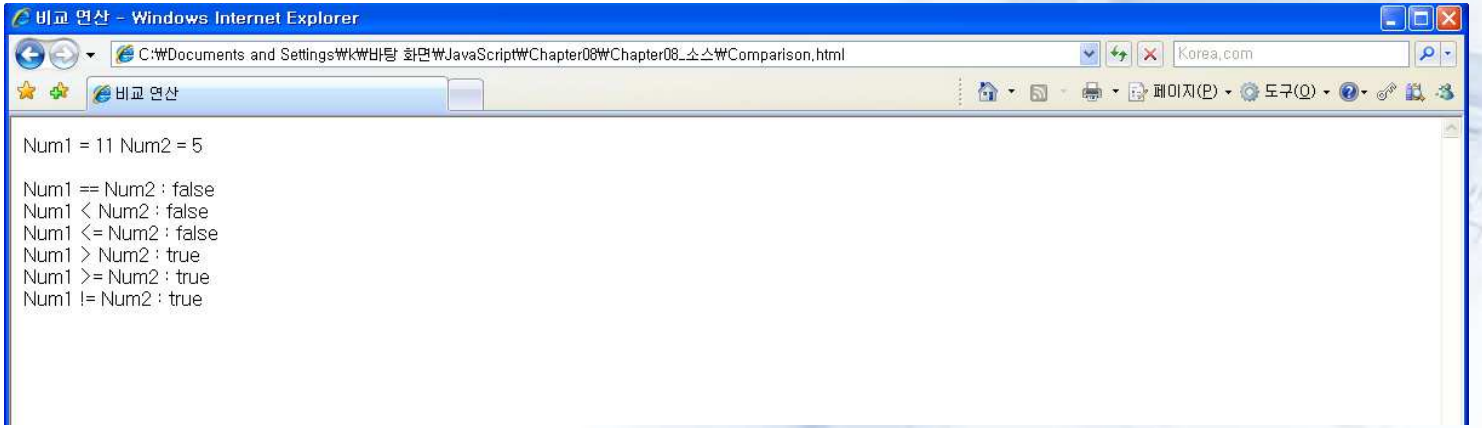
- 두 변수 사이의 관계를 비교하기 위하여 사용되는 연산자
- 연산 결과의 참과 거짓에 따라 Boolean 값인 true, false를 반환

연산자	설명
==	같다
<	보다 작다
<=	보다 작거나 같다
>	보다 크다
>=	보다 크거나 같다
!=	같지 않다

프로그램 8-8

```
var intNum1, intNum2;
intNum1 = 11;
intNum2 = 5;

document.write("Num1 = " + intNum1 +
                    " Num2 = " + intNum2  + "<BR><BR>");
document.write("Num1 == Num2 : " +
                    (intNum1 == intNum2) + "<BR>");
document.write("Num1 < Num2 : " +
                    (intNum1 < intNum2) + "<BR>");
```



[그림 8-7] Comparison의 실행결과

논리 연산자

❖ 논리 연산자

- 조건의 참과 거짓을 판단하기 위한 연산자
- 두 값을 비교하여 해당 연산자에 따라 참과 거짓을 판별

연산자	설명
&&	두 조건이 모두 참이면 참이 됨 (AND)
||	두 조건중 하나만 참이면 참이 됨 (OR)
!	조건 값을 반대로 만듬 (NOT)

비트 연산자

❖비트 연산자

- 각각의 비트 별로 연산을 하는 것으로 2진수 연산에 사용
- 참과 거짓에 따라 1, 0 값을 반환

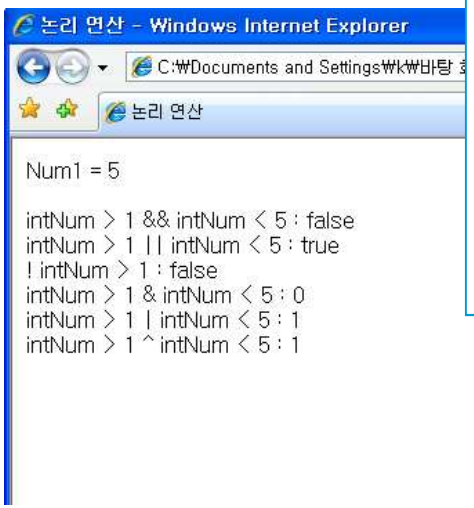
연산자	설명
&	두 조건이 모두 참이면 참이 됨 (AND)
|	두 조건중 하나만 참이면 참이 됨 (OR)
^	두 조건의 값이 같으면 참이 됨 (XOR)

프로그램 8-9

```
var intNum;

intNum = 5;


document.write("Num1 = " + intNum + "<BR><BR>");
document.write("intNum > 1 && intNum < 5 : " +
               (intNum > 1 && intNum < 5 ) + "<BR>");
document.write("intNum > 1 || intNum < 5 : " +
               (intNum > 1 || intNum < 5) + "<BR>");
document.write("! intNum > 1 : " +
               (! (intNum > 1)) + "<BR>");
```



[그림 8-8] Logic의 실행결과

이항 연산자

❖ 이항 연산자

- 어떠한 조건을 판단하여 두 개의 값 중 하나의 값을 선택하는 것을 말함
- 만약 조건이 참일 경우 앞의 값이 선택되고 조건이 거짓일 경우에는 뒤의 값이 선택

```
intNum = 10;
(intNum > 5) ? 1 : 0;
```

연결 연산자

❖ 연결 연산자

- 문자열을 합칠 때 필요한 연산자로 '+'를 사용

```
intX = 10;
strX = "연결";
strY = "연산자";

intX + strX; // 값 : 10연결

strX + strY; // 값 : 연결연산자
```

연산자 우선순위

구분	연산자
괄호	(), []
증감/부정	!, ++, --
산술	*, /, %
	+, -
비트쉬프트	<<, >>, >>>
비교	<, <=, >, =>
	==, !=
비트	&
	^
	|
논리	&&
	||
이항	?
대입	=