

JavaScript

1. JavaScript 기본

JavaScript

- 웹 개발자가 알아야 할 3대 언어
 - HTML: 웹 문서의 내용들을 정의함.
 - CSS: 웹 문서의 구성, 배치, 방식 등을 정의함.
 - JavaScript: 웹 문서의 동작 방식 등을 정의함.

JavaScript 특징

- 클라이언트 프로그램
- 동적 바인딩
 - 바인딩이란 속성, 객체 또는 연산 기호를 연관시키는 것. 동적 바인딩이란 타입이 선언될 때 정해지지 않고, 값이 변수에 저장될 때 타입이 지정되는 것을 말한다.
- 객체 지향형
- HTML 문서에 삽입되는 스크립트언어로 웹 브라우저에서 웹 문서 실행 시 같이 수행된다.

JavaScript 특징

- Script 언어란?
 - 기계어로 변환하기 위한 컴파일 과정이 필요 없이, 스크립트 언어를 해석하고 수행하는 프로그램을 통해 실행될 수 있다.
 - 스크립트 언어를 해석할 수 있는 프로그램은 반드시 필요.
 - 기계어로 변환되지 않기 때문에 OS에 의존성이 없다.
 - Window에서 프로그램 한 JavaScript 코드도 Linux에서 구동될 수 있다.
- 스크립트 언어를 해석하고 수행하는 프로그램은 OS에 의존성이 있다.
 - JavaScript를 해석하는 프로그램은 웹 브라우저이며, 이들 웹 브라우저는 OS에 영향을 받는다.

JavaScript 특징

- 사용자와 컴퓨터간의 상호 교환 적(interactive)인 웹 페이지 작성이 가능하다.
- 서버의 기능을 클라이언트에서 수행하도록 하기 때문에 낮은 성능의 저가 서버를 사용할 수 있다.
- 네트워크 트래픽을 감소 시킬 수 있다.
- 플랫폼 독립적이다. (스크립트 언어의 특징)
- 역동적인 웹 페이지 작성이 가능하다.
- 소스가 포함된 문서가 사용자에게 전달되어 그대로 노출된다.

JavaScript WHERE TO

JavaScript 삽입 방법

- Javascript 코드는 <script> start tag와 </script> end tag 사이에 정의한다.
- HTML 문서 안에서 어느 곳에 위치해도 무관하다.
 - <head> 안에 있는 경우.
 - <body> 안에 있는 경우.
 - <body> 앞에 있는 경우.
 - <body> 맨 마지막에 있는 경우: Script loading이 마지막에 진행되므로 page를 보여주는 시간이 다른 경우보다 빠를 수 있다.

JavaScript 삽입 방법

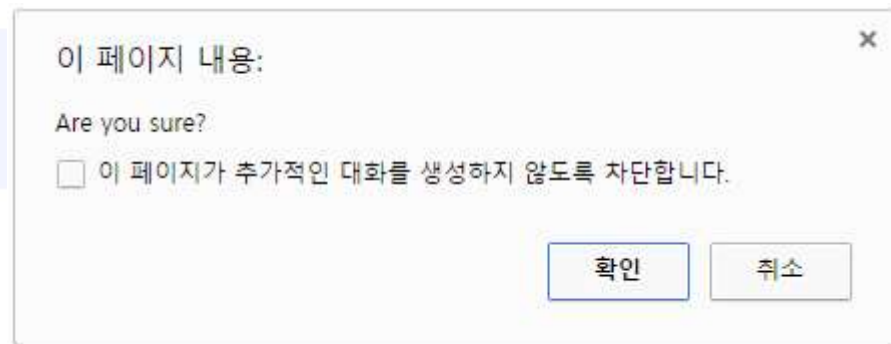
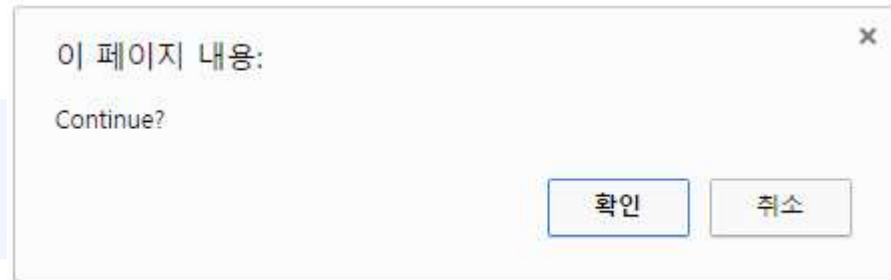
```
<html>
<head>
  <script>
    function help ()
    {
      confirm ("Continue?");
    }
  </script>
</head>
<body>
  <form>
    <script>
      help();
    </script>
    <input type="submit" onclick="check()" />
  </form>

  <script>
    function check ()
    {
      confirm ("Are you sure?");
    }
  </script>
</body>
</html>
```

함수형

내장형

행 입력 형



JavaScript 삽입 방법

```
<html>
  <head>
  </head>
  <body>
    <form>
      <script>
        help();
      </script>
      <input type="submit" onclick="check()">
    </form>

    <script>
      function help ()
      {
        confirm ("Continue?");
      }
      function check ()
      {
        confirm ("Are you sure?");
      }
    </script>
  </body>
</html>
```

function help()를 body의 아래 부분에 선언하면 어떤 현상이 발생하는가?

JavaScript 삽입 방법

- 외부 파일로부터 javascript 적용하기
 - HTML 문서를 간략화 할 수 있다.
 - JavaScript 소스의 노출을 방지하여 보안성을 향상 시킨다.
 - 하나 이상의 html 파일에서 동일한 JavaScript가 사용되는 경우 효율적이다.
 - <script> 태그에 src 속성을 이용하여 외부 파일을 불러올 수 있음.

```
<script src="자바스크립트 파일">
```

JavaScript 삽입 방법

```
<html>
  <head>
    <script src="week9.js"></script>
  </head>
  <body>
    <form>
      <script>
        help();
      </script>
      <input type="submit" onclick="check()">
    </form>
  </body>
</html>
```

```
function help ()
{
    confirm ("Continue?");
}
function check ()
{
    confirm ("Are you sure?");
}
```

JavaScript OBJECT

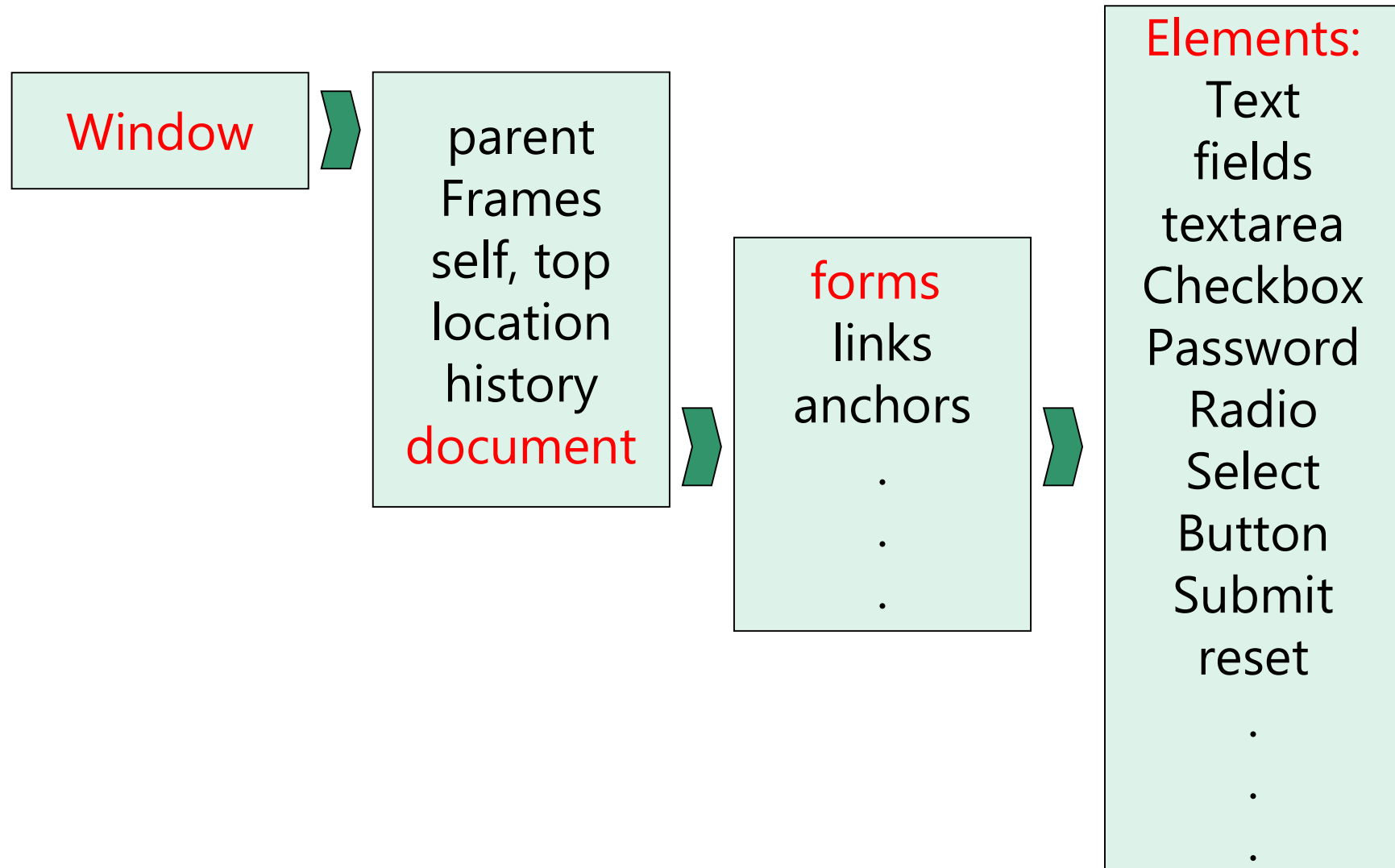
JavaScript 객체

- 자바 스크립트는 **객체 기반의 스크립트 언어**이다.
- 자바스크립트에서 **제공하는 객체**와 **사용자가 정의한 객체**가 있다.
- 자바스크립트에서 제공하는 객체 (내장형 객체)에는 Array 객체, String 객체, Math 객체, Date 객체, Number 객체 등이 있다.
- 문서를 보여 주는 창(window)을 최상위 객체로 하여 자바스크립트의 모든 작업을 수행한다.
 - 계층적 구조의 객체를 가지고 있다.

객체, 속성, 메소드의 예

	Object	Properties	Method
특징	프로그램 대상	객체의 속성, 성격, 특징	객체의 기능, 성능, 역할
예	자동차	이름, 무게, 전장, 색	앞으로 가기 정지하기 뒤로가기
예	사람	이름, 머리 색, 키, 몸무게	말하기 걷기 던지기

JavaScript 객체 계층구조



객체 표현법

- 객체명.속성 = 값
 - window.status = "Go!"
- 객체명.메소드
 - window.close()
 - window.open()
- 상위객체이름.하위객체이름.속성 = 값
 - window.document.myform.fname.value = "duksung"
 - window.document.forms["myform"].elements["fname"].value = 'duksung';
- window객체는 JavaScript에서 가장 최상위 객체이며 생략 가능하다.

window.document.myform.fname.value →
document.myform.fname.value

Document 객체

- 웹 문서의 색상 설정 관련 속성
 - fgColor, bgColor, alinkColor, linkColor, vlinkColor....
- 웹 문서와 관련된 정보를 다루는 속성
 - lastModified, location, referrer, title
- 웹 문서에 포함된 내용과 관련된 속성
 - anchors, cookie, forms, links
- document 객체의 메소드
 - open(), close(), clear(), write(), writeln()

HTML DOM (Document Object Model)

Window 객체

- window객체는 JavaScript에서 가장 최상위 객체이다.

```
<body>
  <button onclick="openWin()"> Open "myWindow"</button>
  <button onclick="closeWin()"> Close "myWindow"</button>

  <form name="myform">
    <input type="text" name="fname" value="Input your Univ." onclick="clrtext()">
  </form>
</body>

<script>
  var myWindow;

  function openWin() {
    myWindow = window.open("", "myWindow", "width=200, height=100");
    myWindow.document.write("<p>This is 'myWindow'</p>");
  }

  function closeWin() {
    myWindow.close();
  }

  function clrtext() {
    document.myform.fname.value='';
  }
</script>
```

새로운 윈도우 객체 생성

윈도우 생성 메소드

윈도우 제거 메소드

JavaScript EVENT & EVENTHANDER

Event 와 eventhandler

- 사용자로부터 취해진 동작을 Event라 한다. – 마우스를 움직이거나 키를 누르는 동작.
- 이 이벤트들 이름 앞에 'on'이란 단어를 붙여 eventhandler를 정의한다.
- 사용자의 행위 자체는 이벤트
 - 마우스를 누르는 행위
- 사용자의 행위를 처리하는 기능: 이벤트 핸들러
 - 마우스를 눌렀을 때

Event와 eventhandler

- 이벤트 핸들러의 종류
 - onLoad: 웹 브라우저에서 문서를 읽어 올 때
 - onUnload: 웹 브라우저에서 문서를 닫을 때
 - onMouseover: 마우스가 위에 있을 때
 - onMouseout: 마우스가 벗어 났을 때

```
<body onLoad="alert('Welcome!')" onUnload="alert('Good bye!')">
  <p id="evt" onMouseover="mover()" onMouseout="mout()">
    onLoad and onUnload Test page!
  </p>
  <script>
    function mover()
    {
      document.getElementById("evt").style.color = "red";
    }
    function mout()
    {
      document.getElementById("evt").style.color = "blue";
    }
  </script>
</body>
```

Event와 eventhandler

- 이벤트 핸들러의 종류

- onClick: 마우스 클릭할 때
- onFocus: 마우스의 커서가 위치했을 때
- onBlur: 마우스의 커서가 다른 곳으로 이동할 때

```
<body onLoad="alert('Welcome')" onUnload="alert('GoodBye')">
  <form>
    <input type="text" id="tst" name="test" onFocus="fcus()" onBlur="blr()">
    <input type="submit" onClick="clk()">
  </form>
</body>

<script>
  function clk()
  {
    alert("Are you sure?");
  }
  function fcus()
  {
    document.getElementById("tst").style.backgroundColor = "blue";
  }
  function blr()
  {
    document.getElementById("tst").style.backgroundColor = "yellow";
  }
</script>
```

JavaScript USAGE

JavaScript 구문

- 여러 명령문(statements)이 모여 program이 된다.
- 명령문(statements)은 값(values), 연산자(operators), 수식(expressions), 키워드(keywords), 주석(comments)으로 구성된다.

JavaScript Statements

- 각 명령문(statement)은 세미콜론(;)에 의해 구분된다.
- 따라서 여러 명령문(statement)이 같은 line에 있어도 문제가 되지 않는다.
- 마지막 명령문(statement)에는 세미콜론이 생략될 수 있다. 하지만, 반드시 붙이도록 한다.

JavaScript 여백

- JavaScript에서는 여백이 무시된다. 따라서, 스페이스를 이용하여 가독성을 높이는 것이 가능하다.

JavaScript VALUE, VARIABLE AND ARRAY

JavaScript Value

- 값에는 문자열, 숫자, 불리언, 배열, 객체가 있다.
- 리터럴(literal)과 변수(variable)
 - 변수를 선언하면서 값을 직접 지정하는 방식.

```
<p id=demo></p>
```

```
<script>
```

```
var cars = ["Saab", "Volvo", "BMW"];  
var owners = 'mason';  
var age = 40;
```

```
document.getElementById("demo").innerHTML = cars + " " + owners + " " + age;
```

```
</script>
```

- 값의 표현
 - 소수점 표현
 - 문자열
 - 이중 따옴표(“ ”) 그리고 단일 따옴표(‘ ’)으로 문자열을 표현한다.

JavaScript Value

- 변수 (variables) – 변할 수 있는 값
 - **var** 라는 키워드를 이용하여 변수를 선언한다.
 - 변수의 타입을 지정하여 선언하지 않고 변수에 할당하는 값에 따라 타입이 정해진다.
 - 변수에 값을 할당할 때는 (=) 등호 기호를 사용한다.
cf) 같음으로 표시할 때에는 (==) 기호를 사용한다.
 - 변수의 타입
 - 정수, 문자열, 소수점 등을 구분하여 변수를 사용할 필요가 없다.
 - var 키워드로 변수를 선언하고 = 기호로 값을 할당한다.

JavaScript Value

- 변수 (variables) – 변할 수 있는 값
 - 하나의 명령문(statement) 안에서 여러 개의 변수를 선언할 때에는 , 를 이용한다.
 - `var x = 1, y = 2, z = "hi, duksung"`
- 문자열도 연산자를 이용하여 서로 연결할 수 있다.
 - `var hi = "Good" + " " + "morning";`
- 이중 따옴표 안의 숫자와의 덧셈 연산은 문자열 간의 연결로 처리되고 결과 값은 문자열이 된다.
 - `"5" + 3 + 2; "5" + 3 * 2; 5 + "3" + 2`

JavaScript Value

변수의 타입은 고정되지 않고 할당되는 값에 따라 변한다.

x의 타입은 string에서 int로 변경되었으며,
y의 타입은 int에서 string으로 변경되었다.

```
<html>
  <head>
  </head>
  <body>
    <script>
      var x = "hello! world! ";
      var y = 2016;
      var z = 11;
      document.write (x + y + z + "<br>");
      document.write (y + z + x + "<br>");
      x = 2017;
      y = " is coming";
      document.write (x + z + y);
    </script>
  </body>
</html>
```

```
hello! world! 201611
2027hello! world!
2028 is coming
```

JavaScript Keywords

- 키워드 (Keywords)
 - 특별한 의미가 부여된 예약된 단어들.
 - 변수의 이름은 키워드와 달라야 한다.

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	

JavaScript 식별자

- 식별자 (Identifiers)
 - 변수, 키워드, 함수 등을 구분하기 위한 이름.
 - 이름 붙이기 규칙
 - 식별자의 첫 단어는 \$, _ 를 제외하고는 무조건 알파벳이어야 한다.
 - 첫 단어 이외의 단어는 문자와 숫자, 그리고 \$와 _ 가 올 수 있다.
 - 새로운 변수와 함수 이름 등은 키워드와 같은 이름을 사용해서는 안 된다.

JavaScript 식별자

- 대소문자 구분
 - JavaScript는 대소문자를 구분한다.
 - 식별자로 Var를 사용할 수 있다.
 - 키워드 var 와는 구분되기 때문이다.
 - lastName과 lastname은 서로 다른 변수이다.
- 여러 단어를 하나의 이름으로 사용하는 방법.
 - 어떠한 방법을 이용하던 무관하지만, 처음 단어는 소문자로 시작하고 이후부터는 단어의 시작은 대문자를 사용하는 것이 일반적이다.
 - firstName, secondName, paradiseCity

JavaScript 식별자

```
<html>
  <head>
  </head>
  <body>
    <script>
      var lastName = "This is lastName", lastname = "This is lastname";
      var _$name = "Duksung";
      var $_name_1$_ = "Univ";

      document.write (lastName + " and " + lastname + "<br>");
      document.write (_$name + " " + $_name_1$_);
    </script>
  </body>
</html>
```

JavaScript 주석

- 주석 (Comments)
 - // 문장
 - 명령문을 한 줄로 설명할 때 주로 사용하게 된다.
 - /* 문장 */
 - 함수의 기능이나 동작 구조 등의 내용이 긴 장문의 설명이 필요할 때 사용한다.
- 여러 단어를 하나의 이름으로 사용하는 방법.
 - 처음 단어는 소문자로 시작하고 이후부터는 단어의 시작은 대문자를 사용한다.
 - firstName, secondName, paradiseCity

JavaScript Data Types

- JavaScript에서의 자료형
 - 자료형의 선언이 필요 없이 var 자료형을 사용하고 저장되는 값에 따라 자동으로 형 변환 됨.
- 숫자형(numbers)
 - 정수와 실수 형이 있다.
- 논리형(boolean)
 - 참이나 거짓의 값을 가진다. true → 1, false → 0
- 문자열(string)
 - 하나 이상의 문자로 이루어진 문장을 표현한다.
 - " " 이나 ' '으로 표현 가능하다.
- 널(null)
 - 아무런 값이 없음을 나타낸다.

JavaScript Data Types

- 숫자형
 - 정수와 실수로 표현할 수 있다.
 - 8진수: 처음에 0으로 시작
 - 16 진수: 0x 로 시작.
 - exponential 을 사용하여 표현할 수 있다.

```
<html>
  <head>
  </head>
  <body>
    <script>
      var x1 = 34.11, x2 = 0x10, x3= 010;
      var y = 123e5, z = 123e-5;

      document.write(x1 + x2 + x3 + "<br>");
      document.write(y + "<br>" + z);
    </script>
  </body>
</html>
```

JavaScript Array

- 배열의 정의
 - 같은 타입, 같은 크기의 데이터를 2개 이상 붙여서 하나의 변수로 처리하는 것.

- 형식

```
var 배열 변수 명 = new Array();  
배열 변수 명[0] = value;  
배열 변수 명[1] = value;  
배열 변수 명[2] = value;
```

```
var cars = new Array();  
cars[0] = "Saab";  
cars[1] = "Volvo";  
cars[2] = "BMW";
```

```
literal 선언  
var array-name = [item1, item2...
```

```
var cars = ["Saab", "Volvo", "BMW"];
```

```
var cars = new Array("Saab", "Volvo", "BMW");
```

JavaScript OPERATORS

JavaScript 연산자

- 산술연산자
 - 더하기(+), 빼기(-), 곱하기(*), 나누기(/), 나머지(%)
- 연결연산자
 - (+) 문자와 문자, 문자와 변수를 연결할 때 사용.

JavaScript 연산자

- 관계(비교)연산자
 - 대소관계를 비교하여 참, 거짓을 결과 값으로 함.

기호	설명
$A == B$	서로 같음
$A != B$	서로 같지 않음
$A \geq B$	A가 B와 같거나 큼
$A \leq B$	A가 B보다 같거나 작음
$A > B$	A가 B보다 큼
$A < B$	A가 B보다 작음

JavaScript 연산자

- 조건연산자
 - 조건식의 결과가 참인지 거짓인지를 판단.

```
<script>
  function comp (a, b)
  {
    (a < b)?
      document.write (b + " is bigger than " + a):
      document.write (b + " isn't bigger than " + a);

    document.write ("<br> AND <br>");

    if (a < b)
      document.write (b + " is bigger than " + a);
    else
      document.write (b + " isn't bigger than " + a);
  }
  comp (4, 1);
</script>
```

JavaScript 연산자

- 논리연산자
 - 여러 값의 참 거짓임을 이용하여 전체의 값이 참인지 거짓인지를 판단하는 연산자.

기호	설명
A && B	A와 B가 모두 참이면 참.
A B	A와 B 중 하나가 참이면 참
!A	A가 참이면 거짓, 거짓이면 참

JavaScript 연산자

- 증감연산자
 - 변수의 값을 1 증가 시키거나 1 감소시키는 연산자.

기호	설명
<code>++A</code>	A를 1 증가 후 이후 작업
<code>A++</code>	작업 후 A를 1 증가
<code>--A</code>	A를 1 감소 후 이후 작업
<code>A--</code>	작업 후 A를 1 감소

JavaScript 연산자

- 대입연산자
 - 변수에 숫자나 문자열과 같은 값을 저장하는 연산자

기호	설명
$A = B$	A에 B 값을 저장
$A += B$	A 와 B를 더한 값을 A에 저장
$A -= B$	A에서 B를 뺀 값을 A에 저장
$A *= B$	A와 B를 곱한 값을 A에 저장
$A /= B$	A를 B로 나눈 값을 A에 저장
$A \% = B$	A를 B로 나눈 나머지 값을 A에 저장

JavaScript 연산자

```
<html>
  <head>
  </head>
  <body>
    <script>
      var z =11, y=11;
      document.write (z++ + "<br>");
      document.write (++y + "<br>");
    </script>
  </body>
</html>
```

++ 의 위치에 따라 출력 값이 달라진다.
증가를 시키고 함수를 수행 할 것인지,
함수를 수행 시키고 증가를 시킬 것인지
결정된다.

END