

JavaScript

HTML5 기술의 핵심



자바 vs 자바 스크립트

특징	자바 언어	자바스크립트
언어 종류	소스 파일을 컴파일하여 실행하는 컴파일 언어이다.	브라우저가 소스 코드를 직접 해석하여 실행하는 인터프리트 언어이다.
실행 방식	자바 가상 기계 위에서 실행한다.	브라우저 위에서 실행된다.
작성 위치	별도의 소스 파일에 작성	HTML 파일 안에 삽입 가능
변수 선언	변수의 타입을 반드시 선언해야 함	변수의 타입을 선언하지 않아도 사용 가능

자바 스크립트 역사

- 넷스케이프의 브렌던 아이크(Brendan Eich)가 개발
- Mocha 1995년 5월 처음 발표
- 1995년 9월 라이브스크립트(LiveScript)
- 1995년 12월 JavaScript
- 최신 버전은 자바스크립트 1.8.5
- ECMA(European Computer Manufacturer's Association)이 ECMAScript라는 이름으로 표준을 제정 -> ECMA-262



자바 스크립트 특징

- HTML에 포함, interpreter 방식, Browser에서 실행, Client-Side Script
- 동적 타이핑(dynamic typing): 실행시 값이 할당될 때 data type이 결정
- 구조적 프로그래밍 지원
- 객체 기반 (Object-based)언어 – 객체의 정의와 사용 지원
- 함수형 프로그래밍 지원
- 프로토타입-기반(prototype-based)

자바 스크립트의 용도

- 이벤트에 반응하는 동작을 구현
- AJAX: Background Asynchronous Data Communication Technology
- HTML 요소들의 크기나 색상을 동적으로 변경
- 게임이나 애니메이션
- 사용자가 입력한 값들을 검증

자바스크립트

- 자바스크립트 작성
 - 내부: <html>내부의 <script> </script>태그 안
 - 외부 *.js 파일 & <html>내부의 <script>태그
 - <script src="source.js"></script>
 - 선호되는 방법
 - inline style: 태그 내부
- HTML vs. JavaScript
 - HTML: 웹 문서의 텍스트(문자) 표현
 - JavaScript: 웹문서의 동적인 효과
 - 애니메이션, 대화상자표시, 입력값 검사 등

```
<html>
<body>
이제 html은 쉽습니다.<br/>
<script>
<!--
    document.writeln("JavaScript도 별것 아니군요.");
// -->
</script>
</body>
</html>
```

JavaScript 소스 작성 위치

- <script> 태그의 위치
 - <head> 내부
 - <body> 내부

자바스크립트의 객체

- 객체 개념 사용, but 객체지향이 아님
 - document, window, event 등
- document
 - 자바스크립트를 포함한 HTML문서 자체
- 객체에는 그 기능을 하는 메소드 존재
 - document.writeln()
- 내장객체, BOM객체(Browser관련), DOM객체(document관련), 사용자 정의 객체

자바스크립트: 기본 데이터 형

데이터형		설명
number	정수형	8진수, 10진수, 16진수의 3종류
	실수형	10.5, -3.1E5, 4E-2와 같이 소수부분이 있는 10진수와 지수로 표현된 숫자
boolean형		true와 false의 두 가지 값만을 가짐, 함수의 리턴 값이나 관계 연산자의 결과값으로 사용
String		따옴표에 묶어서 생성한 모든 데이터는 문자열의 형태임.
undefined		변수가 정의되지 않았거나 선언만 되고 정의안된 경우
null		값이 없음 또는 객체가 없음을 표시, 예외처리에 자주 사용, object형
object		객체형, json객체, array, function, date, RegExp 등

자바스크립트: 변수

- 값을 담아 두는 컨테이너, 박스, 그릇
- 변수의 선언: **var**
- 변수 선언의 위치
 - **전역변수**: 메서드 밖
 - **지역변수**: 메서드 내부
- 변수 선언시 값을 할당하여 사용해야 함
 - 값 할당하지 않으면 undefined

```
var a=15;
var b=10;
var total=a*b;
var soccer1="설기현";
var soccer2="박지성";
document.writeln('공격수는 '+ soccer1+' ' + '미드필더는 '+ soccer2+
                  '선수 입니다.');
```

자바스크립트: 변수선언 예

```
<html>
<head> <title>변수 처리 예제</title> </head>
<body>
<h3>변수로 배경색을 지정한 예제입니다.</h3>
<script>
    var color = "yellow";

    document.bgColor = color;
    document.write("배경색은 '"+color+"' 입니다.<br />");
</script>
</body>
</html>
```

자바스크립트: 예약어

- 예약어는 변수명으로 사용 못함

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

자바스크립트: 산술,관계연산

산술연산자	설명
-	부호연산
+, -, *, /	사칙연산
%	나머지 연산(정수 나눗셈의 나머지값)
++, --	변수에 저장된 값을 각각 1증가, 1감소

※ +의 피연산자가
문자열이면
연결연산

관계연산자	설명
>	왼쪽이 오른쪽보다 크다
>=	왼쪽이 오른쪽보다 크거나 같다
<	왼쪽이 오른쪽보다 작다
<=	왼쪽이 오른쪽보다 작거나 같다
==	서로 같다(값)
===	서로 같다(값, 자료형)
!=	같지 않다
!==	같지 않다(값, 자료형)

자바스크립트: 논리, 대입연산

논리연산자	설명
!	논리 부정: 참이면 거짓, 거짓이면 참을 변경, NOT
&&	논리곱: 양쪽 다 참일 때만 참, AND
	논리합: 양쪽 중 한쪽이라도 참이면 참, OR
? :	조건 연산자(수식?실행1:실행2, 수식이 참이면 실행1이, 거짓이면 실행2가 실행됨)
^	XOR(배타적 논리합), 양쪽 값이 다르면 참, 같으면 거짓

대입연산자	설명
x=y	y값을 x에 대입
x+=y	x=x+y의 의미, x+=y가 실행속도 빠름
x-=y	x=x-y
x*=y	x=x*y
x/=y	x=x/y
x%=y	x=x%y, x를 y로 나눈 나머지 값을 x에 대입

자바스크립트: 연산자 예

```
<html>
<head> <title>연산자 처리 예제</title> </head>
<body>
<script>
var X = 100;
var Y = 30;

sum = X + Y;
minus = X - Y;

document.write("Num1 :" + X + ", Num2 :" + Y + "<br/>");
document.write("더하기 :" + sum + "<br/>");
document.write("빼기 :" + minus + "<br/>");
</script>
</body>
</html>
```

자바스크립트: 연산자 우선순위

- 소괄호 () 활용
- 대입연산자 - 제일 낮음

증감 연산자

- $++x$ 또는 $-x$
 - x 변수에 들어 있는 값을 먼저 1 더하거나 빼는 연산을 수행후 사용
- $x++$ 또는 $x--$
 - 산술식에 포함되어 있는 경우 이전 수식을 수행한 다음 1을 더하거나 1을 빼는 작업

연산자	기능	실행 의미	예
$++x$	선 증가	1을 증가	$x=3; ++x; \text{결과 } x=4$
$x++$	후 증가	1을 증가	$x=3; x++; \text{결과 } x=4$
$--x$	선 감소	1을 감소	$x=3; --x; \text{결과 } x=2$
$x--$	후 감소	1을 감소	$x=3; x--; \text{결과 } x=2$

- $y=x++$ 과 $y=++x$ 는 차이

수행 전

0
y

수행 후

6
y

수행 전

0
y

수행 후

5
y

수행 전

5
x

수행 후

6
x

선 증가

$y = ++x;$

후 증가

$y = x++;$

증감 연산자의 실습

```
<script>
  var x=5;
  var y=0;
  y = ++x; // x를 먼저 증가하고 y에 대입한다
  document.write("<선 증가><br/>");
  document.write("y = " + y + "<br/>");
  document.write("x = " + x + "<br/>");
  document.write("-----<br/>");
  var x=5;
  var y=0;
  y=x++; // x를 y에 대입하고 x를 증가한다
  document.write("<후 증가><br/>");
  document.write("y = " + y + "<br/>");
  document.write("x = " + x + "<br/>");
</script>
```

<선 증가>

y = 6

x = 6

<후 증가>

y = 5

x = 6

연결 연산자

- + 기호는 덧셈 이외에 문자열을 취급할 때 결합 기능을 수행
- 문자열과 숫자를 결합하면 모두 문자열로 나타나며, += 연산자는 결합 기능이 수행

"의적" + " " + "홍길동" // 의적 홍길동

"22" + 8 // 228

"의적" += " 홍길동" // 의적 홍길동

<h3>

<script>

var x = "의적";

var y = "홍길동";

document.write("x = " + x + "
");

document.write("y = " + y + "
");

document.write(x + " " + y, "
");

document.write("x += y --> " + (x += y) + "
", "
");

document.write("W" + "3" + "W" + "+3 = " + ("3"+3) + "
");

</script>

</h3>

x = 의적

y = 홍길동

의적 홍길동

x += y --> 의적홍길동

"3"+3 = 33

비교 연산자

- 연산의 결과는 true/false인 Boolean 값
- 문자열은 각 문자를 유니코드 값의 문자 대 문자로 비교
 - "A"와 "B"는 A가 B 보다 작게 비교

연산자/피연산자	의 미
x == y	x는 y와 같다.
x != y	x는 y와 같지 않다.
x > y	x는 y보다 크다.
x >= y	x는 y보다 크거나 같다.
x < y	x는 y보다 작다.
x === y	x는 y와 형과 값이 아주 같다.
x !== y	x는 y와 형과 값이 아주 같지 않다.

비교 연산자

- 예시

비교 수식	결과	비교 수식	결과
"KOREA" == "KOREA"	true	"KOREA" === "KOREA"	true
"korea" == "KOREA"	false	"korea" === "KOREA"	false
6 == 6.0	true	6 === 6.0	true
"55" == 55	true	"55" === 55	false
"5.5" == 5.5	true	"5.5" == 5.5	false
NaN == NaN	false	NaN === NaN	false
null == null	true	null === null	true
-0 == +0	true	-0 === +0	true
false == false	true	false === false	true
true == 1	true	true === 1	false
null == undefined	true	null === undefined	false

문자열의 비교 실습

```
<script>
```

```
var fruit1 = "pear";
```

```
var fruit2 = "peaR";
```

```
document.write( "<h2> fruit1  = "+ fruit1 + "<br/>");
```

```
document.write( "<h2> fruit2  = "+ fruit2 + "<br/>");
```

```
var result = fruit1 > fruit2;
```

```
document.write( "<h3> fruit1 > fruit2 = " + result + "<br/>");
```

```
result = fruit1 < fruit2;
```

```
document.write( " fruit1 < fruit2  = " + result + "<br/>");
```

```
result = fruit1 === fruit2;
```

```
document.write( "fruit1 === fruit2 = " + result + "<br/>");
```

```
</script>
```

fruit1 = pear.

fruit2 = peaR.

fruit1 > fruit2 = true.

fruit1 < fruit2 = false.

fruit1 === fruit2 = false.

논리 연산자

- 대부분 조건문, 반복문에서 사용
- 연산의 결과: Boolean 값을 반환
- AND(&&), OR(||), NOT(!)

```
if ( x > 5 && x < 10 )
```

```
// x는 5보다 크면서 10보다 작을 경우 true를 반환.
```

연산자/피연산자	의 미
num1 && num2	num1과 num2가 true 일 경우 true를 반환 둘 중 하나가 false이면 무조건 false를 반환
num1 num2	둘 중 하나가 true이면 true를 반환 둘다 false 일 경우에는 false를 반환
!num1	num1이 true이면 false, false이면 true를 반환

논리 연산자

- && 연산의 비교 결과

연산식	비교 결과
true && false	false
true && true	true
"honest" && true	true
true && ""	""
true && "honest"	honest
5 && 0	0
5 && -6	-6
5 && false	false
null && 0	null
null && ""	null
null && false	null
"hello" && true && 50	50
"this" && "that"	that

논리 연산자

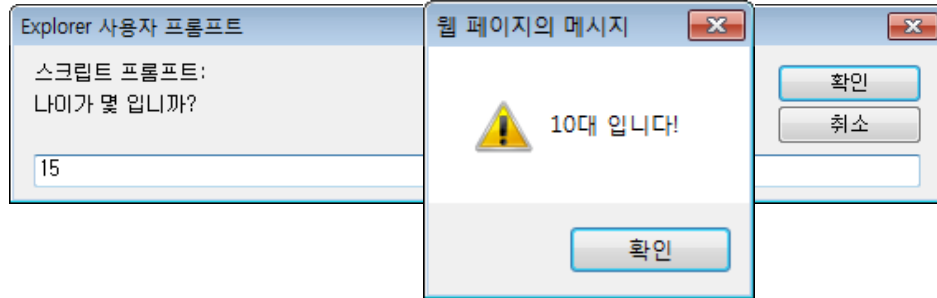
- || 연산의 비교 결과

연산식	비교 결과
true false	true
true true	true
"honest" true	honest
true ""	true
true "honest"	true
6 0	6
6 -6	6
6 false	6
null 0	0
null ""	""
null false	false
"hello" true 50	hello
"yes" "no"	yes

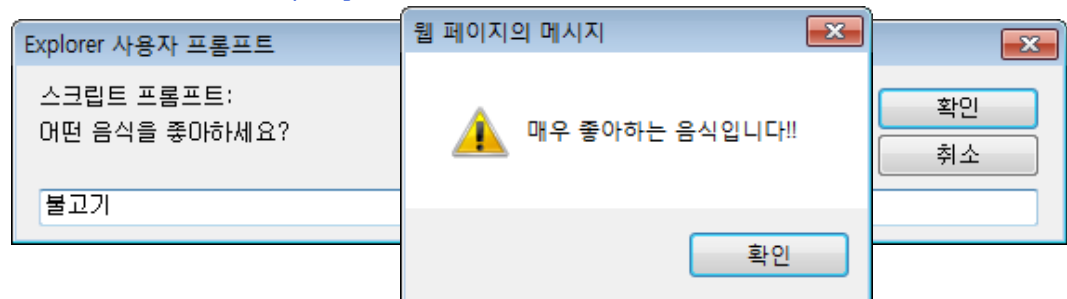
논리 연산자

- 실습

```
<script>
  var answer = prompt("나이가 몇 입니까? ", "");
  if (answer > 12 && answer < 20) {
    alert("10대 입니다!");
  } else {
    alert("10대가 아닙니다!");
  }
</script>
```



```
<script>
  var answer = prompt("어떤 음식을 좋아하세요? ", "");
  if ( answer == "불고기" ||
      answer == "짜장면" ||
      answer == "순대국") {
    alert("매우 좋아하는 음식입니다!!");
  } else { alert("좋아하는 음식이 없네요!!"); }
</script>
```



논리 연산자

- ! 연산의 결과

연산식	비교 결과
!"this"	false
!0	true
!2	false
!false	true
!null	true
!undefined	true

<script>

```
var answer = true;
```

```
document.write("true의 반대는 ?? " + ! answer + "<br/>");
```

```
document.write("!this 의 결과는 ?? " + !this + "<br/>");
```

```
document.write("!0 의 결과는 ?? " + !0 + "<br/>");
```

```
document.write("!undefined 의 결과는 ?? " + !undefined + "<br/>");
```

</script>

true의 반대는 ?? false

!this 의 결과는 ?? false

!0 의 결과는 ?? true

!undefined 의 결과는 ?? true

논리 연산자

```
<script>
```

```
var num1=50;  
var num2=100;  
var num3=0;  
document.write("num1 = " + num1+"<br>");  
document.write("num2 = " + num2+"<br>");  
document.write("num3 = " + num3+"<p>");
```

```
document.write("num1 && num2 = " +(num1 && num2)+"<br>");  
document.write("num1 || num2 = " +(num1 || num2)+"<p>");  
document.write("! num1 = " + !num1+"<br>");  
document.write("!(num1 && num2) = " +!(num1 && num2)+"<br>");  
document.write("!(num1 && num3) = " +!(num1 && num3)+"<br>");
```

```
</script>
```

```
num1 = 50.  
num2 = 100.  
num3 = 0.
```

```
num1 && num2 = 100.  
num1 || num2 = 50.
```

```
! num1 = false.  
!(num1 && num2) = false.  
!(num1 && num3) = true.
```

자바스크립트: if문

- if(조건){
 statements
}
- if(조건){
 statements1
}else{
 statements2
}
- if(조건1){
 statements1
}else if(조건2){
 statements2
}else{
 statements3
}

```
<script>  
var currDate = new Date();  
var currTime = currDate.getHours();  
if(currTime < 12) {  
    document.write("오전이에요~~ 자~ 행복한  
                    하루를 시작해야죠??");  
} else if (currTime >=12){  
    document.write("오후네요. 날씨도 좋은데~  
                    농구 한게임 어떠세요?");  
}  
</script>
```

자바스크립트: switch문

- `switch(표현식){`
 `case 값1: statements 1; break;`
 `case 값2: statements 2; break;`
 ...
 `case 값n: statements n; break;`
 `default: statements d;`
}
- 표현식의 결과값(정수)과 case의 값과 일치하는 경우 해당 statements 실행, break문 실행
- break가 없으면 해당 case의 그 다음 case문의 statements도 실행
- 해당되는 case가 없으면 default 실행

자바스크립트: switch문 예

```
<script>
  var localDate = new Date();
  var localTime = localDate.getHours();
  switch (localTime)
  {
    case 4: document.write("새벽이군요.. 아침형 인간이시군요!"); break;
    case 9: document.write("업무 시작하는 시간이군요! 오늘도 열심히!");
            break;
    case 12: document.write("즐거운 점심시간~ 먹어야 산다~!!");
            break;
    case 19:
      document.write("퇴근해야 하는 시간!!! 오늘은 일찍 퇴근합시다!!");
      break;
    default:
      document.write("아무런 계획이 없는 시간이군요!! ");
  }
</script>
```

자바스크립트: for문

- `for(초기값;종료조건;증감식){`
 statements;
}
- `for([var]변수 in 증{객체 | 배열}){`
 statements;
}

```
<script>  
  for(x=0; x<3; x++){  
    document.write("당신이 최고예요!!! <br/>");  
  }  
</script>
```

```
<script>  
  for (var name in document){  
    console.log(name + ": " + document[name]);  
  }  
</script>
```

자바스크립트: 함수

- 빈번히 사용되는 기능의 소스코드 집합을 묶어 정의해 둔 것
- `function(변수리스트){ //익명 함수`
 statements;
}
- `function 함수이름(변수리스트){ //선언적 함수`
 statements;
}

자바스크립트: 내장 함수

내장 함수	기능
alert("출력내용")	경고창을 출력함
confirm("출력내용")	확인과 취소를 선택할 수 있는 창을 출력함
prompt("출력내용", "기본문구")	입력받을 수 있는 창을 출력함
eval()	문자열을 수식으로 변경하여 계산하여 출력함
isNaN()	입력값이 숫자인지 문자인지 구별하여 숫자가 아닌 경우에 true 값을 출력함
parseInt()	문자열을 정수로 변환하여 출력함
parseFloat()	문자열을 실수로 변환하여 출력함
escape()	ISO-Latin-1 문자셋을 아스키 값으로 변경하여 출력함
unescape()	아스키 값을 ISO-Latin-1 문자 셋으로 변경하여 출력함
String()	객체를 문자로 변환하여 출력함
Number()	객체를 숫자로 변환하여 출력함
isFinite()	입력 값이 유리수인지 판단하여 유리수의 경우에 true 값으로 변환하여 출력함

자바스크립트: 내장함수-`eval()`

- 문자열이나 표현식을 숫자로 변경(결과가 숫자형식이 되는)
- 문자 형태의 값을 연산하기 위해 사용
- 특정 스크립트를 실행시켜줌
 - `eval("50")-20;`
 - `value="50"; eval(value)-20;`

```
<html> <head> <title>eval 예제</title>
```

```
<script>
```

```
function evalSample() {
```

```
    var first, second, third, total;
```

```
    first  = prompt("첫번째 숫자를 입력해 주세요!", "20");
```

```
    second = prompt("연산자를 입력해 주세요!", "*");
```

```
    third  = prompt("두번째 숫자를 입력해 주세요!", "50");
```

```
    total = eval(parseInt(first) + second + parseInt(third));
```

```
    alert("first 과 third의 값을 second의 연산자로 처리한 수 = " + total);
```

```
}
```

```
</script> </head>
```

```
<body> <input type="button" value="눌러보아요!" onclick="evalSample();" />
```

```
</body></html>
```

자바스크립트: 내장함수-Array()

- 배열객체의 변수 선언
 - myArray= new Array();
 - myArray=new Array(10)

속성, 메소드	설명
length	배열 요소의 개수
join()	배열의 모든 요소들을 하나의 문자열로 만들어 리턴
reverse()	배열 요소들의 순서를 역순으로 재배열
sort()	배열 요소들을 증가 순(오름차순)으로 정렬
concat(arr2)	현재 배열에 다른 arr2의 배열 요소들을 연결한 새로운 배열을 리턴
slice(a,b)	배열의 a번째요소부터 b번째요소를 포함하지 않은 배열값을 배열로 반환

자바스크립트: 내장함수-Array()

```
<script>
var myArray=new Array();
var titleArray=new Array("동호수를","이름을","평수를","방갯수를","화장실 갯수를");
i=0; k=0;flag=0;
while(confirm("아파트 정보를 입력할까요?")) {
    myArray[i]=new Array();
    for(j=0 ; j<5 ; j++) {
        myArray[i][j]=prompt(titleArray[j]+"입력해주세요","");
    }
    i++;
}
document.write("<table border='1' width='500'>");
document.write("<tr><td>동호수</td><td>이름</td><td>평수</td><td>방갯수</td>");
document.write("<td>욕실갯수</td></tr>");
for(i=0;i<myArray.length;i++) {
    document.write("<tr>");
    for(j=0;j<myArray[i].length;j++) {
        document.write("<td>"+myArray[i][j]+"</td>");
    }
    document.write("</tr>");
}
document.write("</table>");
</script>
```

자바스크립트: String관련

- 문자열 객체(String객체)의 변수에 대한 메소드 & 속성

속성		내용	
length		문자열이 가지고 있는 문자의 개수	
prototype		String 객체에 속성 추가	
메소드		내용	동일 기능 HTML 태그
문자 모양	big()	글자를 좀더 크게	<BIG> 글자 </BIG>
	small()	글자를 좀더 작게	<SMALL> 글자 </SMALL>
	bold()	볼드체	 글자
	fixed()	타자기체	<TT> 글자 </TT>
	italics()	이탤릭체	<I> 글자 </I>
	strike()	글자 가운데 줄긋기	<STRIKE> 글자 </STRIKE>
	sub()	아래 첨자	_{글자}
	sup()	윗 첨자	^{글자}
	fontcolor(색)	글자색	 글자
	fontsize(크기)	글자 크기	 글자

한글도 1로 처리

자바스크립트: String관련

- 문자열 객체(String객체)의 변수에 대한 메소드 & 속성-계속

메소드		내용	동일 기능 HTML 태그
하이퍼텍스트 연결	anchor("표식")	표식 지정	 글자
	link("위치")	하이퍼텍스트 연결	 글자
문자 처리	charAt(n)	n번째에 해당하는 문자 찾기	n 위치에 해당하는 문자 반환
	indexOf("문자열")	지정된 문자열의 위치값 계산(왼쪽부터)	문자열이 나오는 위치를 숫자로 반환
	lastIndexOf("문자열")	지정된 문자열의 위치값 계산(오른쪽부터)	
	substring(n, m)	n 번째부터 m-1 위치 사이에 있는 문자열 반환	
	concat("문자열")	문자열을 문자열 객체에 연결	
	slice(n, m)	substring과 동일, 음수값은 오른쪽부터 순번으로 계산	
	split("구분문자")	구분문자로 문자열 분리	
	toUpperCase()	대문자로 만들기	
	toLowerCase()	소문자로 만들기	

split(대상문자열, "기준문자열")
대상문자열을 기준문자로 잘라서 배열로 반환

```
<script>
function changeMoney(){
    var f = document.myForm;
    var tempString = "";
    var rsString = "";
    tempString = f.beforeValue.value;
    tempString = tempString.replace(/,/g, ''); //정규표현식: ,를 찾아 모두없앰
    for(x = 1; x <= tempString.length; x++){
        if(x > 1 && (x%3) == 1) {
            rsString = tempString.charAt(tempString.length - x) + "," + rsString;
        }else{
            rsString = tempString.charAt(tempString.length - x) + rsString;
        }
    }
    f.afterValue.value = rsString;
}
</script>
<form name="myForm">
    <input type="text" name="beforeValue"><br>
    <input type="button" value="바꾸기" onClick="changeMoney();"><br>
    <input type="text" name="afterValue">
</form>
```

자바스크립트: 내장함수-alert()

- 경고창 표시: alert(message);

```
<head><script>
  function InitMessage() {
    alert("내 생에 첫 Alert창~~ 입니다.");
  }
</script></head>
<body onLoad="InitMessage()">
  <h3> Alert 창을 구경해 보아요~~~ </h3>
  <h3> 내생에 첫 Alert 창~ </h3>
</body>
```

자바스크립트: 내장함수-isFinite()

- 파라미터값이 유한수인지 무한수인지 표시: 유한수-true, 무한수-false

```
<body>
```

유한수이면 True 리턴.~ , 무한수 이면 False 리턴.!

```
<br/><br/><br/>
```

```
<script>
```

```
document.write("MyLove  &nbsp;&nbsp;&nbsp;" + isFinite("MyLove") + "<br/>");
```

```
document.write("1976/12/01  &nbsp;&nbsp;&nbsp;" + isFinite("1976/12/01")+  
                "<br/>");
```

```
document.write("300  &nbsp;&nbsp;&nbsp;" + isFinite(365)+ "<br/>");
```

```
document.write("-1.23  &nbsp;&nbsp;&nbsp;" + isFinite(-1.23)+ "<br/>");
```

```
document.write("0  &nbsp;&nbsp;&nbsp;" + isFinite(0)+ "<br/>");
```

```
</script>
```

```
</body>
```


자바스크립트: 내장함수-escape()

- 문자열을 ASCII값으로 반환해주는 함수
 - 특수문자 처리를 위해 사용
 - get방식 전송시 보안을 위해
 - url 인코딩
 - 문자열이 16진수로 변경

```
<script>
```

```
function changeEscape(){
```

```
    var str = "";
```

```
    str = "황사가 점점 심해 지네요. 마스크가 필수품이 될것 같네요!! ";
```

```
    str = str + "?!=())#%&  ";
```

```
    str = str + "This is english!!";
```

```
    alert(escape(str));
```

```
}
```

```
</script>
```

```
<input type="button" value="escape()변환" onClick="changeEscape()">
```

자바스크립트: 내장함수-unescape()

- 16진수의 인코딩된 것을 문자열로 복원

```
<script>
function changeUnescape(){
    var str = "";
    str = "유혹적인 팜파탈, 구스타프 클림트";
    str = str + "- 1862.7.14~1918.2.6";
    str = str + "- Gustav Klimt is Austrian, a painter";

    str = escape(str);
    document.write("escape() => " + str + "<br/>");

    alert("unescape => " + unescape(str));
}
</script>
<input type="button" value="unescape() 변환" onClick="changeUnescape()">
```

자바스크립트: 내장 함수-isNaN()

- 인수값이 문자인지 숫자인지를 구별
 - 문자이면 true, 숫자이면 false

```
<script>
```

```
function isNaNCheck(){  
    document.write("MyLove " + isNaN("MyLove")+ "<br/>");  
    document.write("1976/12/01 " + isNaN("1976/12/01")+ "<br/>");  
    document.write("300 " + isNaN(123)+ "<br/>");  
    document.write("-1.23 " + isNaN(-1.23)+ "<br/>");  
    document.write("0 " + isNaN(0)+ "<br/>");  
}
```

```
</script>
```

```
<input type="button" value="isNaN 결과" onClick="isNaNCheck()">
```

자바스크립트: 내장 함수-parseInt()

- 문자열을 정수로 변환

- 8진수, 16진수 문자열을 10진수로 변환
- 숫자, 문자가 혼합: 숫자만 출력
- 실수인 경우 소수점 이하 버리고 정수만 반환
- `parseInt("28",16)`: 0x28을 10진수로 변환

<script>

```
function parse(){
```

```
var f = document.form1;          var x = new Array();
x[0] = "parseInt(128)= " + parseInt("128")+"\\n";
x[1] = "parseInt(3.14)= " + parseInt("3.14")+"\\n";
x[2] = "parseInt(0811)= " + parseInt("0811")+"\\n";
x[3] = "parseInt(18000원)= " + parseInt("18000원")+"\\n";
x[4] = "parseInt(13,8)= " + parseInt("13",8)+"\\n";
x[5] = "parseInt(28,16)= " + parseInt("28",16)+"\\n";
x[6] = "parseInt(0x11)= " + parseInt("0x11")+"\\n";
x[7] = "parseInt(10 20 30)= " + parseInt("10 20 30")+"\\n";
f.result.value=x;
```

} </script>

```
<form name="form1"> <center><h1> ParseInt 테스트 입니다</h1><br/>  
    <input type="button" value="확인" onclick="parse()">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
    <input type="button" value="지우기" onclick="reset()"><br/>  
    <textarea name="result" style="height:15em; width:30em"></textarea></center>  
</form>
```

자바스크립트: 내장 함수-parsefloat()

- 문자열을 부동소수점으로 변환

<script>

```
function parse(){
    var f = document.form1;
    var x = new Array();
    x[0] = "parseFloat(128)= " + parseFloat("128") +"Wn";
    x[1] = "parseFloat(3.14)= " + parseFloat("3.14")+"Wn";
    x[2] = "parseFloat(3.14.87)= " + parseFloat("3.14.87")+"Wn";
    x[3] = "parseFloat(18000원)= " + parseFloat("18000원")+"Wn";
    x[4] = "parseFloat(18000.05원)= " + parseFloat("18000.05원")+"Wn";
    x[5] = "parseFloat(10.20.30)= " + parseFloat("10.20.30")+"Wn";
    f.result.value=x;
}
```

</script>

[illegible]

Form접근

- document객체 소속
- form객체 접근법
 - <form>태그 name속성에 이름 정의후 사용
 - document.form1.tel
 - document객체의 내부 form 객체배열로 접근
 - document.forms[0].tel

```
<form name="form1" action="go.jsp" method="post" target="_self" enctype="">
<input type="text" name="tel">
</form>
<form name="form2" action="go.jsp" method="post" enctype="">
<input type="text" name="tel">
</form>
```

Form 접근

- JavaScript의 폼엘리먼트객체 프로퍼티와 메소드

| 프로퍼티 | 내용 |
|---------------|---|
| elements | 폼 태그에 담긴 모든 폼 컨트롤 배열(form1.elements[0]는 tel임) |
| length | 폼 태그의 컨트롤 수 |
| name | 폼 태그의 이름 (form1, form2) |
| acceptcharset | 폼 태그가 지원하는 문자 집합 목록 |
| action | 폼 태그의 action값을 읽어오거나 값을 대입 (go.jsp) |
| enctype | 폼 태그의 contents 형을 읽어오거나 값을 대입(첨부파일 유무에 따라, multipart/form-data) |
| method | 폼 태그에 지정된 HTTP메소드를 읽어오거나 값을 대입 (post,get) |
| target | action의 대상을 읽어오거나 값을 대입(_blank,_self,_top,_parent, 프레임 이름) |

| 메소드 | 내용 |
|----------|---------------------|
| submit() | 폼 태그의 내용을 제출 |
| reset() | 폼 태그의 설정값을 초기상태로 변경 |

form 태그의 node 찾기

| | |
|-----|---|
| 형식 | document.폼객체명(form태그의name속성값).속성
document.폼객체명(form태그의name속성값).메소드 |
| 속성 | action: 서비스 요청 url
elements: form태그의 자식 노드들의 배열
name: form태그의 name속성값
target: form태그의 target속성값
method: form태그의 method속성값 |
| 메소드 | submit(): 입력 내용을 서버로 전송
reset(): 입력 내용의 초기화 |

<form name='userinfo' ...> ... </form>인 경우
document.userinfo.elements[1]
document.userinfo.name
document.userinfo.submit()

form 태그의 node 찾기

- <input type=''>: text or password, <textarea>

| | |
|-----|--|
| 형식 | document.폼객체명.text객체명.속성
document.폼객체명.text객체명.메소드 |
| 속성 | value: 입력창에 입력한 문자열
defaultValue: 기본적으로 설정되어 있는 문자열 |
| 메소드 | blur(): 입력창에 focus를 제거
focus(): 입력창에 focus를 주어 입력창에 커서를 표시
eval(): 입력창의 수식을 평가(계산, 실행)함
select(): 입력창에 입력한 문자열을 모두 선택 |

<form name='userinfo' ...>

<input type='text' name='id' ... />

</form>인 경우

document.userinfo.id.focus()

document.userinfo.id.value

document.userinfo.id.select()

form 태그의 node 찾기

- <input type=''>: checkbox

| | |
|-----|--|
| 형식 | document.폼객체명.checkbox객체명.속성
document.폼객체명.checkbox객체명.메소드 |
| 속성 | value: input 태그의 value 속성값
checked: 체크여부를 설정하거나 반환 |
| 메소드 | blur(): 입력창에 focus를 제거
focus(): 입력창에 focus를 주어 입력창에 커서를 표시
click(): 체크박스를 클릭 |

<form name='userinfo' ...>

<input type='checkbox' name='interest' ... />

</form>인 경우

document.userinfo.interest.focus()

document.userinfo.interest.checked

document.userinfo.interest.click()

form 태그의 node 찾기

- <input type='': radio

| | |
|-----|---|
| 형식 | document.폼객체명.radio객체명.속성
document.폼객체명.radio객체명.메소드 |
| 속성 | value: input 태그의 value 속성값
checked: 체크여부를 설정하거나 반환
length: 같은 그룹 안에 있는 라디오 버튼의 갯수 |
| 메소드 | blur(): 입력창에 focus를 제거
focus(): 입력창에 focus를 주어 입력창에 커서를 표시
click(): 체크박스를 클릭 |

```
<form name='userinfo' ...>
```

```
  <input type='radio' name='gender' ... />
```

```
</form>인 경우
```

```
document.userinfo.gender.focus()
```

```
document.userinfo.gender.checked
```

```
document.userinfo.gender.click()
```

form 태그의 node 찾기

- <select>

- 콤보박스 내의 값을 다루기 위해서는 `options[]` 배열 사용

| | |
|-----|---|
| 형식 | document.폼객체명.select객체명.속성
document.폼객체명.select객체명.메소드 |
| 속성 | name: name속성
selectedIndex: 선택된 항목의 index번호
options: <option>태그 객체들의 배열
length: 콤보박스내의 리스트 항목 개수 |
| 메소드 | blur(): 입력창에 focus를 제거
focus(): 입력창에 focus를 주어 입력창에 커서를 표시 |

```
<form name='f1' >
  <select name="s1" >
    <option value="menu1" selected>메뉴 1</option>
    <option value="menu2">메뉴 2</option>
    <option value="menu3">메뉴 3</option>
  </select>
</form>

document.f1.s1.length
document.f1.s1.options[document.f1.s1.selectedIndex].value
```

Form 접근

```
<html>
<head>
<title>Form example</title>
</head>
<script>
  function getInfoD() {
    var info;
    var f = document.forms["testForm"];
    info = "f.elements: " + f.elements + "\n"
      + "f.length: " + f.length + "\n"
      + "f.name: " + f.elements + "\n"
      + "f.acceptCharset: " + f.acceptCharset + "\n"
      + "f.action: " + f.action + "\n"
      + "f enctype: " + f enctype + "\n"
      + "f.encoding: " + f.encoding + "\n"
      + "f.method: " + f.method + "\n"
      + "f.target: " + f.target;
    document.forms["testForm"].elements['text'].value = info;
  }
```

Form 접근

```
function getInfo(f) {
    f.method = "GET";
    f.action = "chapter4_4_2.jsp";
    f.name = "totally_new";
}
</script>
<h1>이것이 품입니다!! </h1>
<form name="testForm" id="testForm"
action="chapter4_4_1_2.jsp" method="POST">
    <p/>
    <input type="button" value="정보"    onclick="getInfoD();">
    <input type="button" value="설정"    onclick="getInfo(this.form);">
    <input type="reset" value="Reset">
    <br/>
    <textarea id="text" style="height:15em; width:20em"></textarea>
</p>
</form>
</html>
```

Form접근: 전송방법

- 웹서버로의 데이터 전송 방식: POST/GET
- Get: HyperLink방식
 - url 뒤에 전송데이터 첨부
 - `http://xxx.xxx.xxx/test.jsp?name=kim`
 - 간단, 보안상의 문제, 길이 제약
- Post: 폼방식
 - HTTP헤더에 전송데이터 첨부
 - Get보다 보안안전, 길이 제한 없음

Form접근-GET

- 사용편리, 처리 간단
- 길이 제한(255자), 보안상문제(인코딩으로 보완)
- url 입력란에서 직접 데이터 전송 가능
 - `http://localhost/root.jsp?code=M001&Dept=S001`
- DB에 대한 질의어(where절)에 포함되는 데이터와 같은 단순 정보 전송시 주로 사용

```
<html> <head><title> get 방식 예제 </title></head>
<body><center>
<form name="form1" action="get_my_name.jsp" method="get">
  이 름 : <input type="text" name="uname">
  메 일 : <input type="text" name="mail">
  <p></p>
  <input type="submit" value="전송">
  <input type="reset" value="다시">
</form></center> </body> </html>
```

url입력란에 아래와 같이 입력하여 요청하는 것과 같음

```
get_my_name.jsp?uname=***&mail=****
```


Form접근-Post

- HTTP헤더에 전송데이터 포함해 전송
- 다량의 정보 전송가능, Get보다 안정적
- url만 표시(http://localhost/root.jsp)
- Get보다 처리속도 느림

```
<html> <head><title> POST 방식 예제 </title> </head> <body>
<form name="form1" action="post_form.jsp" method="post">
  이 름 : <input type="text" name="name"> <br/>
  메 일 : <input type="text" name="mail"> <br/><br/>
<hr>
  <input type="submit" value="submit">
  <input type="reset" value="reset">
</form> </body> </html>
```

Event

- 객체에 특정 행위나 변화가 발생
 - 버튼 클릭, 마우스 드래그, 키보드 누름 등
- 이벤트 핸들러
 - 이벤트가 발생되면 미리 설정된 일정한 작업을 수행
- 사용 형식

이벤트	이벤트 핸들러	내용
그림 관련 이벤트	abort(onAbort)	사용자가 이미지를 읽다가 중단할 경우
	error(onError)	웹 페이지가 불러들일 때 에러가 발생한 경우
포커스 관련 이벤트	blur(onBlur)	사용자로부터 입력받는 양식에서 커서가 다른 곳으로 이동한 경우
	focus(onFocus)	사용자의 입력 양식 안으로 커서가 들어왔을 경우
마우스 관련 이벤트	click(onClick)	사용자가 링크나 입력 양식을 클릭한 경우
	dblclick(onDblick)	사용자가 마우스를 더블 클릭한 경우
	mousedown(onMouseDown)	사용자가 마우스 버튼을 누른 경우
	mouseover(onMouseover)	사용자의 마우스가 링크된 곳을 지나가는 경우
	mouseup(onMouseup)	사용자가 마우스를 눌렀다 놓을 경우
	mousemove(onMouseMove)	사용자가 마우스를 움직일 경우
	dragdrop(onDragdrop)	사용자가 마우스를 클릭한 상태에서 움직였을 경우
폼 관련 이벤트	change(onChange)	사용자로부터 입력받는 양식의 값이 바뀌는 경우
	submit(onSubmit)	사용자가 입력한 양식을 서버로 전송할 경우
	select(onSelect)	사용자가 입력 양식의 한 항목을 선택한 경우
키보드 관련 이벤트	keydown(onKeyDown)	사용자가 키를 입력했을 경우
	keypress(onKeyPress)	사용자가 키를 눌렀을 경우
	keyup(onKeyUp)	사용자가 키를 눌렀다 놓았을 경우
윈도우 관련 이벤트	load(onLoad)	네스케이프에서 웹페이지를 열 경우
	move(onMove)	사용자가 윈도우를 움직일 경우
	resize(onResize)	사용자가 윈도우나 프레임의 크기를 변경한 경우
	unload(onUnload)	사용자가 현재의 웹 페이지에서 다른 곳으로 이동하는 경우

<... on이벤트이름="이벤트 핸들러" ...>

Event

- onAbort(): 이미지 로딩중일 때 중단하면
 - 관련 태그:
- onBlur(): 포커스가 다른 곳으로 이동하면

```
<script>
  function upperCase()
  {
    var x=document.getElementById("myName").value;
    document.getElementById("myName").value=x.toUpperCase();
  }
</script>
```

Enter your name:

```
<input type="text" id="myName" onblur="upperCase()" />
```

Event

- onChange(): 값이 변경되면
 - 관련 태그: <select>, <input type="text">, <textarea>

```
<script>
function onChangeUpperCase(a)
{
    var myID = document.getElementById(a).value;
    document.getElementById(a).value= myID.toUpperCase();
}
</script>
```

Enter your name:

```
<input type="text" value="test" id="myName"
      onchange="onChangeUpperCase(this.id)" />
```

Event

- onClick(): 어떠한 객체를 클릭하면

```
<script>
  function copyString(){
    var second = document.getElementById('string2');
    var first = document.getElementById('string1');
    second.value = first.value;
  }
</script>
Field1: <input type="text" id="string1" value="황사가 왜이리 심하지?"><br/>
Field2: <input type="text" id="string2"><br/><br/>
<button onclick="copyString()">글자 복사</button>
```

Event

- onFocus(): 객체에 포커스가 설정되면
 - 포커스: 선택된 해당 객체가 활성화되는 것

onBlur활용 수정해보기!

```
<script>
  function setColor(color)
  {
    document.getElementById(color).style.background="red";
  }
</script>
오늘 저녁에 먹을 음식은?:
<input type="text" onfocus="setColor(this.id)" id="first"><br />
디저트로 는 무엇 먹을거죠?:
<input type="text" onfocus="setColor(this.id)" id="name">
```

Event

- onKeyDown(): 객체에서 키보드의 키를 누르면

```
<script>
function numberCheckNoNumber(eEvent)
{
    var num; var char; var numCheck;
    if(window.event) {num = eEvent.keyCode; } //키보드의 키값
    else if(eEvent.which) {num = eEvent.which;} //눌린키의 ASCII값
    char = String.fromCharCode(num); // num이 65면 'A'반환
    numCheck = /Wd/; // 정규표현식
    return !numCheck.test(char); // 정규표현식 객체의 test함수
}
</script>
```

Event

- onLoad(): 문서가 웹브라우저에 로드될 때
 - 지원태크: <body>, <frame>, <frameset>, <iframe>, , <link>, <script>

```
<script>
```

```
function loadComplete()
```

```
{
```

```
    window.status = "페이지가 전부 로딩 되었습니다!";
```

```
}
```

```
</script>
```

```
<body onload="loadComplete()">
```

```
</body>
```


Event

- onMouseDown(): 마우스 버튼을 누르면

```
<script>
function choiceRange(eEvent){
    if (!eEvent) { var eEvent = window.event; }
    if (eEvent.target) { choice = eEvent.target; }
    else if (eEvent.srcElement){choice = eEvent.srcElement; }
    if (choice.nodeType == 3){choice = choice.parentNode; }
    var tname;
    tname = choice.tagName;
    alert(" 마우스를 눌렀네요?? " + tname + " 영역을 누르셨네요~");
}
</script>
<body onMousedown="choiceRange(event)">
<h2> H2 영역 입니다.~</h2>
<p> Paragraph 영역 입니다. ~ </p>

```

Event

- onMouseOver()/onMouseOut()
 - 객체 위에 마우스포인트를 옮길 때/객체 박스 밖으로 옮길 때

```
<script>
function OverFunction(){ document.myImage.src ="chapter5-7-8-1.jpg";}
function OutFunction(){ document.myImage.src ="chapter5-7-8-2.jpg";}
</script>
<a href="http://www.naver.com" target="_blank" onmouseover="OverFunction()"
    onmouseout="OutFunction()">

</a>
```

Event

- onUnload(): 페이지를 다른 페이지로 이동할 때, IE에서 실행됨
 - 관련태그: <body>, <frameset>

```
<html>
<head>
</head>
<body onunload="alert('페이지를 이동하는 이벤트 발생!!')">
  OnUnload Sample!!
</body>
</html>
```

onabort: null
onblur: null
oncancel: null
oncanplay: null
oncanplaythrough: null
onchange: null
onclick: null
onclose: null
oncontextmenu: null
oncuechange: null
ondblclick: null
ondrag: null
ondragend: null
ondragenter: null
ondragleave: null
ondragover: null
ondragstart: null
ondrop: null
ondurationchange: null

onemptied: null
onended: null
onerror: null
onfocus: null
oninput: null
oninvalid: null
onkeydown: null
onkeypress: null
onkeyup: null
onload: null
onloadeddata: null
onloadedmetadata: null
onloadstart: null
onmousedown: null
onmouseenter: null
onmouseleave: null
onmousemove: null
onmouseout: null
onmouseover: null
onmouseup: null
onmousewheel: null

onpause: null
onplay: null
onplaying: null
onprogress: null
onratechange: null
onreset: null
onresize: null
onscroll: null
onseeked: null
onseeking: null
onselect: null
onshow: null
onstalled: null
onsubmit: null
onsuspend: null
ontimeupdate: null
ontoggle: null
onvolumechange: null
onwaiting: null

예외처리

- java의 try~catch문과 유사

```
try{  
    실행소스(예외가 발생할 것으로 예상되는) 작성  
}catch(err){  
    예외 핸들링 소스(예외가 발생하면 처리하는 소스) 작성  
}
```

```
<html><head><title>Try Catch 문 Sample!!</title>  
<script type="text/javascript">  
function myMsg()  
{  
    adddlert("어서오세요!");  
}  
</script></head>  
<body>  
<input type="button" value="메세지 보기" onclick="myMsg()" />  
</body> </html>
```

예외발생! 이곳을 수정해야 함

예외처리

```
<html><head><title>Try Catch 문 Sample!!</title>
```

```
<script>
```

```
function msg()
```

```
{
```

```
  var str;
```

```
  try{
```

```
    adddlert("어서오세요!");
```

```
  }catch(e){
```

```
    str="에러가 발생했습니다. \n\n";
```

```
    str+="Error description: " + e.description + "\n\n";
```

```
    str+="계속 하시려면 OK 버튼을 눌러 주세요!\n\n";
```

```
    alert(str);
```

```
  }
```

```
}
```

```
</script></head>
```

```
<body> <input type="button" value="메세지 보기" onclick="msg()" />
```

```
</body></html>
```

e: 에러 정보 객체

예외처리

```
<html> <head> <title>Try Catch 문 Sample!!</title>
<script>
function msg()
{
    var str;
    try{
        adddlert("어서오세요!");
    }catch(e){
        str="이 페이지 에서 에러가 발생했습니다. \n\n";
        str+="확인을 누르실 경우 해당 페이지를 계속 보여줍니다.\n\n";
        str+="취소를 누를경우 네이버 페이지로 이동 합니다. \n\n";
        if(!confirm(str)){
            document.location.href="http://www.naver.com/";
        }
    }
}
</script></head>
<body> <input type="button" value="메세지 보기" onclick="msg()" />
</body></html>
```

예외처리

- 예외 발생 유도: throw문

- catch문으로 예외객체를 던짐, 예외처리 요구
- throw(exception);

```
<html> <head> <title>Throw문 예제 Sample!!</title> </head><body>
<script>
var value = prompt("0부터 10까지중의 숫자를 입력하세요~","");
try
{
    if(value > 10)    throw "myError1";
    else if(value < 0)    throw "myError2";
} catch(e) {
    if(e=="myError1") alert("너무 큰 숫자를 입력 하였습니다.!");
    if(e == "myError2") alert("너무 작은 숫자를 입력 하였습니다.!");
}
</script></body></html>
```

예외처리

- onError 이벤트
 - 스크립트에 에러가 발생하면 발생하는 이벤트
 - 이벤트 핸들러: 3가지 인자
 - msg: 에러 메시지
 - url: 에러 발생 페이지 url
 - line: 에러가 발생한 행의 번호

```
onError=myErrorHandle;
```

```
function myErrorHandle(msg, url, line){  
  처리  
  return true or false;  
}
```


예외처리

```
<html> <head> <script>
onerror=myErrorHandle;
function myErrorHandle(msg,url,line)
{
    var txt="";
    txt="이 페이지 에서 에러가 발생했습니다. \n\n";
    txt+="에러메세지: " + msg + "\n";
    txt+="경로: " + url + "\n";
    txt+="에러 발생 행: " + line + "\n\n";
    txt+="계속 하시려면 OK 버튼을 눌러주세요!\n\n";
    alert(txt);
    return false;
}
                                true이면 에러 메시지 표시안됨!
function my_message()
{
    addlert("어서오세요!");
}
</script></head>
<body> <input type="button" value="메세지 보기" onclick="my_message()" />
</body></html>
```

특수문자 삽입

입력	의미	
\'	single quote	따옴표
\''	double quote	쌍따옴표
\&	ampersand	앰퍼샌드
\	backslash	역슬래쉬
\n	new line	개행, 줄바꿈
\r	carriage return	캐리지 리턴
\t	tab	탭
\b	backspace	백스페이스
\f	form feed	폼피드

특수문자 삽입

```
<html>
<head>
</head>
<body>
<h2>특수 문자 예제</h2>
<br/>
<script>
    var txt="아 휴가가 가고 싶다~~ W"휴가W" 정말 좋은 말이야~ <br/>";
    document.write(txt);
    var txt="근데 휴가 가면 모하지?~~ W&모하지?W& ~";
    document.write(txt);
</script>
</body>
</html>
```

브라우저 확인

- 접속한 유저의 브라우저 버전 확인
 - 크로스 브라우징 코딩

```
<html><head><script>
function checkMyBrowser(){
    var browser      = navigator.appName;
    var b_version    = navigator.appVersion;
    var version      = parseFloat(b_version);
    var txt;
    if ((browser=="Netscape" || browser=="Microsoft Internet Explorer")
        && (version>=4)){
        txt = "당신의 브라우저 버전은 충분합니다. !! \n";
        txt = txt + browser + " ; " + b_version + " ; " + version;
        alert(txt);
    }else {
        txt = "브라우저 버전을 업그레이드 해주세요!! \n";
        txt = txt + browser + " ; " + b_version + " ; " + version;
        alert(txt);
    }
}
</script></head> <body onload="checkMyBrowser()">
<h2> 브라우저 버전 Check!! </h2></body></html>
```

Cookie처리

- 쿠키: 방문자 컴퓨터에 저장하는 변수
 - 개인 설정 정보, 보안상 문제, 바이러스 침투

```
<html> <head><title>Cookie!!</title></head>
<script>
function setCookie(cookie_name, value, expiredays){
    var tempdate=new Date();
    var cookieStr;
    tempdate.setDate(tempdate.getDate()+expiredays);
    cookieStr = cookie_name+ "=" +escape(value)
                +((expiredays==null) ? "" : ";expires="+tempdate.toGMTString());
    document.cookie=cookieStr;
    document.write(cookieStr);
}
</script> <body>
<h2> Cookie 를 생성하였습니다. </h2>
<script language="javascript">
    setCookie("myAjaxCookie", "2008", "80");
</script></body></html>
```

Cookie처리

- cookieStr 정보
 - myAjaxCookie=2008;expires=Sat, 11 Apr 2020 02:39:04 GMT

C:\Users\사용자계정\AppData\Roaming\Microsoft\Windows\Cookies

C:\Users\사용자계정\AppData\Local\Google\Chrome\User Data\Default\Cache

Cookie처리

- 쿠키값 읽기

```
<html> <head> <title>Cookie!!</title></head>
<script>
function getCookie(cookie_name){
  if (document.cookie.length>0){
    c_start=document.cookie.indexOf(cookie_name + "=");
    if (c_start!=-1){
      c_start=c_start + cookie_name.length+1;
      c_end=document.cookie.indexOf(";",c_start);
      if (c_end==-1) c_end=document.cookie.length;
      return unescape(document.cookie.substring(c_start,c_end));
    }
  }
  return "";
}
</script><body> <h2> Cookie 를 가져 옵니다. </h2>
<script language="javascript">
  var rtnValue;
  document.write("쿠키 데이터 : " + document.cookie);
  rtnValue = getCookie("myAjaxCookie");
  alert(rtnValue);
</script></body></html>
```

Cookie처리-예

```
<html> <head> <title>Cookie!!</title></head>
<script>
function setCookie(cookie_name,value,expiredays){
    var tempdate=new Date();
    var cookieStr;
    tempdate.setDate(tempdate.getDate()+expiredays);
    cookieStr = cookie_name+ "=" +escape(value)
                + ((expiredays==null) ? "" : ";expires="+tempdate.toGMTString());
    document.cookie=cookieStr;
    document.write(cookieStr);
}
function getCookie(cookie_name){
    if (document.cookie.length>0){
        c_start=document.cookie.indexOf(cookie_name + "=");
        if (c_start!=-1){
            c_start=c_start + cookie_name.length+1;
            c_end=document.cookie.indexOf(";",c_start);
            if (c_end==-1) c_end=document.cookie.length;
            return unescape(document.cookie.substring(c_start,c_end));
        }
    }
    return "";
}
```


Cookie처리-예

```
function checkCookie()  
{  
  myAge=getCookie('myAge');  
  if (myAge!=null && myAge!=""){  
    alert('당신의 나이는 '+myAge+' 살 입니다.~!');  
  } else {  
    myAge=prompt('당신의 나이를 넣어주세요~:', "");  
    if (myAge!=null && myAge!=""){  
      setCookie('myAge',myAge,365);  
    }  
  }  
}  
  
</script><body onLoad="checkCookie()">  
<h2> Cookie 확인! </h2></body></html>
```

Validation

- 입력값의 유효성 체크

```
<html> <head> <title>Validation!!</title> </head>
<script>
function validate_check(alerttxt){
    var d = document.testForm.myName;
    if (d.value==null || d.value==""){ alert(alerttxt); return false; }
    else { return true; }
}
</script> <body> <h2> Validation Check! </h2>
<form name="testForm" action="test.html" method="post">
<input type="text" size="20" name="myName">
<input type="button" value="유효성 체크"
    onclick="javascript:validate_check('이름을 넣어 주세요!')">
</form></body></html>
```

Validation

```
<html> <head> <title>Validation!!</title> </head>
<script>
function validate_check(alerttxt){
  var d = document.testForm.myName;
  if (d.value==null||d.value==""){ alert(alerttxt);return false; }
  else{ validate_email(); }
}
function validate_email(){
  var d    = document.testForm.myName;
  a        = d.value.indexOf("@");
  b        = d.value.lastIndexOf(".");
  if (a < 1 || b-a < 2) { alert("Email 형태로 넣어 주세요!"); return false; }
  else{ return true; }
}
</script><body><h2> Validation Check! </h2>
<form name="testForm" action="test.html" method="post">
<input type="text" size="20" name="myName">
<input type="button" value="유효성 체크"
  onclick="javascript:validate_check('이름을 넣어 주세요!')">
</form></body></html>
```