

Session 4

JavaScript

유영훈

내용

- **JavaScript** 개요와 기본 문법
- 함수와 내장함수
- **JavaScript Event Handling**
- **JavaScript Object**와 내장 객체
- **BOM**과 **DOM**
- **JavaScript Exception Handling**
- **JavaScript Cookie**

※본 강의자료는 저자의 허락없이 인터넷이나 기타의 방법으로 재배포하거나 수정할 수 없습니다.

Session 04 JavaScript 개요

□배경

- 최초의 스크립트 : 1987년 애플사의 HyperCard
- 발전 계기 : 1990년대초에 MS사에서 VB에서 사용할 수 있는 VBA(VBApplication) 개발

□등장

- 1994년 Netscape사에서 Dynamic Web page를 웹브라우저에서 구현하기 위한 목적으로 **LiveScript** 개발
- 1995년 Netscape는 Sun Microsystems사(Oracle에 인수됨)와 전략적 제휴를 하며 JavaScript로 이름을 바꾸고, HTML과 java와의 Dynamic 연동을 지원
- 1996년 Microsoft는 인터넷익스플로러(3.0)에서 동작되는 **JScript** 개발
- 1997년 Netscape에서 ECMA(European Computer Manufacturer's Association) 인터네셔널에 제출한 표준안이 최초 표준안으로 채택됨(ECMA-262 Edition 1): ECMAScript 1
- 최신 표준 버전은 2017년 발표된 ECMA-8(또는 ES2017)
현재 가장 범용적인 버전은 2009년 발표된 **ECMA-5**



※html/css/javascript framework 통계 동향: <https://bestofjs.org/>

□특징

- 서버가 아닌 클라이언트에서 실행되는 인터프리터 언어(Interpreter Language)
- 객체 기반 언어(Object based Language)
 - ✓ Browser Object : Window, Screen, Location, History, Navigator, Alert 등
 - ✓ Document Object : Nodes, Elements, Attributes, text, css, event
- 프로토타입 기반 언어
 - ✓ 프로토타입 객체를 연결하여 참조: prototype chain
- HTML 문서에 혼합하여 사용
- 변수의 자료형을 지정할 필요 없음(Dynamic Typing & Dynamic Binding)
- 함수형 프로그래밍, 구조적 프로그래밍 지원
- 클래스 지원(ECMAScript 6 이상)

□ JavaScript와 JAVA(for Applet) 비교

- 이름은 유사하지만 전혀 다른 언어임

항목	JavaScript	JAVA (Applet 중심으로)
실행방법	브라우저에 의해 해석됨 (스크립트 언어)	서버에서 컴파일 된 Byte Code를 클라이언트(JVM)에서 해석됨(컴파일 언어)
언어성격	Object-Based Script Language -프로토타입 기반 언어 -클래스(class) 지원(ECMAScript 6이상)	Object-Oriented Compile Language -상속(inheritance)과 다형성(polymorphism)을 지원하는 완전한 OOP 지향 언어 -클래스(class) 지원
사용 형태	HTML 문서 내에 소스코드 기술(HTML과 결합되어 사용)	HTML 문서에 작은 공간을 얻어 수행 (HTML과 별도로 존재할 수도 있음)
변수형 정의	미리 정의할 필요 없음 (Dynamic Typing)	반드시 미리 정의 해야 함 (Strong & Static Typing)
객체참조 시기	실행 시 객체참조 검사 (동적 바인딩/Dynamic Binding)	컴파일 시 객체참조 검사 (정적 바인딩 / Static Binding)
보안	Secure (HTML에 같이 수행되므로). But, 악의적인 목적의 코드 존재	Secure (Local File System에 Write를 못하므로)

JavaScript 사용 용도

- HTML Form의 입력 값을 가공하거나 검사하여 서버 쪽에 보낼 때
- 사용자의 입력을 제어하고 싶을 때
- 웹 브라우저 내의 여러 가지 기능을 조절하고 싶을 때
- 사용자에게 메시지를 보내고 싶을 때(경고나 확인)
- 쿠키(Cookie)를 이용한 페이지 간의 Data 유지
- 동적인 웹 사이트 제작(HTML 요소나 스타일 동적 변경)
- 게임이나 애니메이션(HTML5 API와 연계)
- 페이지 갱신 없이 서버로부터 값을 가져올 때(AJAX)

Session 04 JavaScript 위치

JavaScript 사용 위치

[01_js.html 파일]

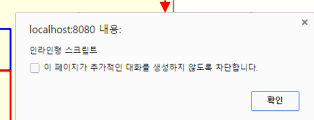
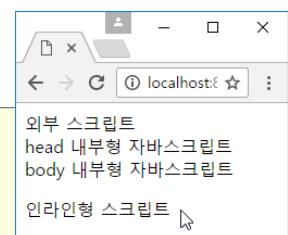
```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>JavaScript 위치</title>
<script src="myscript.js"></script>
<script>
document.write("head 내부형 자바스크립트<br>");
</script>
</head>
<body>
<script>
document.write("body 내부형 자바스크립트<br>");
</script>
<p onclick="alert('인라인형 스크립트')">인라인형 스크립트</p>
</body>
</html>
```

외부형

<head>내부형

<body>내부형

인라인(in-line)형



[myscript.js 파일]

```
document.write("외부 스크립트<br>");
```

Session 04 JavaScript Comments

□ Single Line Comments

```
// document 객체를 통해 브라우저에 html tag 출력하기
document.write("My <b>First</b> Page");
var x = 5; // 변수 선언과 동시에 초기화 하기
```

□ Multi- line Comments

```
/*
    여러 라인에 걸친 주석
*/
document.write("My <span style='color:red'>First</span> Page");
```

➤ 다음의 주석 사용 예 중 올바르게 사용된 것은?

```
/*    /* 주석 */    */    (    )
```

```
/*    // 주석    */    (    )
```

```
//    /*
    주석
*/    (    )
```

```
<body>
<!--    // html 주석 -->
// 주석
</body>
```

Session 04 변수(variables)

□ 변수 : 데이터를 저장하는 기억장소

➤ 명시적 또는 묵시적으로 선언 가능

➤ syntax

```
var a;
var b, c;
var d=1;
var e=d+2;
```

```
a=1;
b=a+2;
```

```
var test = 10;
test = "hello";
test = "hello" + 12;
```

➤ 변수명 생성 규칙

- 영문자, 숫자, _, \$ 를 포함할 수 있음
- 영문자, _, \$ 로 시작할 수 있음(숫자로 시작할 수 없음)
- 대소문자를 구분함
- 예약어(Javascript keywords), 함수명, 객체명은 변수명으로 사용할 수 없음
: var, for, in, if, while, do, function, catch, new, this, const, true, false, NaN 등

선언 예)

```
7abc, !sou, $num, function, this, num2
(    ), (    ), (    ), (    ), (    ), (    )
```

Session 04 자료형 (Data types)

□ 자료형

자료형	의미	사용 예
수치형 (number)	10진, 16진, 8진수의 형태로 표현 가능한 소수점이 없는 정수값.(범위는 시스템에 따라 다름.)	0, 10, 914, - 255, 04(8진수), 0xA(16진수)
	소수점이나 지수형으로 표현되는 실수값	var x = 34.00; var y = 123e5; var z = 123e- 5;
논리형 (boolean)	true 또는 false	var x = false; var x = Boolean(1);
문자열(string)	quotation mark(“” 또는 ‘’) 사이의 문자들의 집합. Character형이 없다.	“12345”, ‘Korea’, ‘a’
undefined	변수 선언 후, 값이 할당된 적이 없는 변수. 또는 type이 지정되지 않은 변수	var x; var y = undefined;
null	아무 값도 없거나, 변수가 선언된 적이 없다는 의미 (예외 처리 목적으로 사용되기도 함) 일부에서는 typeof null을 object로 출력함(JS 버그)	var person = null;
NaN	숫자 또는 숫자형 문자가 아닌 데이터(Not a Number)	isNaN('0') // false isNaN(1) // false isNaN(1/'a') //true

Session 04 자료형 (Data types)

[02_datatype.html 파일]

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset="UTF-8">
5. <title>데이터 자료형</title>
6. <script>
7. var nstr = "10";
8. document.write("typeof nstr : ", typeof nstr, "<br>");
9. nstr=10;
10. document.write("typeof nstr : ", typeof nstr, "<br>");
11. document.write("nstr == 10 : ", nstr == '10', "<br>");
12. //데이터형 구별하지 않고, 값만 같으면 true 반환
13. document.write("nstr === 10 : ", nstr === '10', "<br>");
14. //데이터형과 값이 같으면 true 반환(데이터형을 구별함)
15. document.write("nStr+10+1=" , nstr + 10 + 1, "<br>");
16. document.write("nStr+10+1=" + nstr + 10 + 1, "<br>");
17. document.write("typeof a : ", typeof a, "<br>");
18. document.write("typeof a : ", nstr / 'Apple', "<br>");
19. x=5;
20. document.write("x>=1 : ", x >= 1 , "<br>");
21. </script>
22. </head>
23. <body>
24. </body>
25. </html>

```

데이터 자료형

```

typeof nstr : string
typeof nstr : number
nstr == 10 : true
nstr === 10 : false
nStr+10+1=21
nStr+10+1=10101
typeof a : undefined
typeof : NaN
x>=1 : 2

```

Session 04 연산자(Operators)

□ 연산자와 우선순위

x=5 일 때,

연산자	설명	사용예	결과
() . [] new	객체 참조, 배열 첨자, 괄호	document.write()	
++ -- ! typeof	단항 증감연산	x++;	6
* / %	곱셈, 나눗셈, 나머지 연산	3 / 4	0.75
+ -	덧셈, 뺄셈	"John" + 3	"John3"
<< >> >>>	쉬프트 연산	5 << 1	10
< > <= >=	관계 연산	2 < "12"	true
== != === !==	상등 연산	x !== "5"	false
& ^	비트 논리 연산(AND, OR, XOR)	5 ^ 1	4
&&	논리 연산(AND, OR)	(x < 10 && 3 > 1)	true
?:	조건 연산	(x < 18) ? "y" : "n"	"y"
= += -= *= /= %=	대입연산(산술, 비트논리)	y=3; x %= y; x = x%y;	2
&= ^= =	대입연산(쉬프트)	x >>= 1	2

Session 04 특수문자(Special Characters)

□ 특수문자(Special Characters)

➤ “\” 또는 “\ ” 문자로 특수 문자 표시

특수문자	설명	비고
\n	다음 줄로 이동(new line)	document.write() 에서는 적용 안됨
\b	뒤로 한 칸 이동(backspace)	
\t	탭(tab)	
\r	리턴(return)	
\f	Form feed	
\\	역 슬래시	
\'	따옴표	
\"	큰 따옴표	

```
<script>
var T=100;
var T="My Name";
alert("\\" + T + "\t 홍길동" + "\\");
document.write("\\" + T + "\t 홍길동" + "\\");
</script>
```

이 사이트의 메시지...

My Name

홍길동

☐ 이 페이지에서 추가 메시지를 만들도록 허용하지 않음

확인

□코드라인 연결하기

- 코드 라인상의 길이가 길어질 경우 백슬래쉬(\)로 코드를 다음 줄에 이어서 작성할 수 있음
- 문서 편집기의 문자셋에 따라서 \ 또는 \r로 표현될 수 있음

예제1)

```
document.write("Hello \
World!");
```

예제2)

```
document.write \
("Hello World!");
```

Session **04** JavaScript의 기본 입출력

□입/출력을 위한 기본 내장 객체와 함수

입/출력 구분	선언 내장 객체	함수	설명
출력	window	alert(string);	메시지 전달 용 경고 창을 띄운다. window 객체는 생략 가능
	window	confirm(string);	사용자의 확인을 받기 위한 확인 창을 띄운다. window 객체는 생략 가능
	document	write(string);	웹 브라우저에 출력한다. [주의] HTML문서 전체가 load되고 난 뒤 호출되면, 기존의 문서 내용을 지움
입력	window	prompt(string, initial)	대화 상자에서 입력 받은 내용을 리턴한다.(입력 취소 시 null 리턴) string : 보여주고 싶은 질문 메시지 initial : 초기 값(문자열)

□ 사용 예

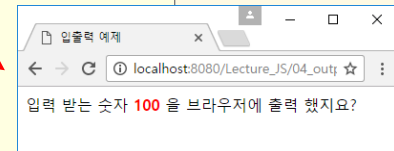
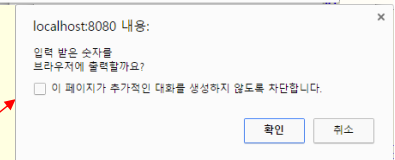
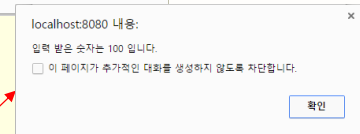
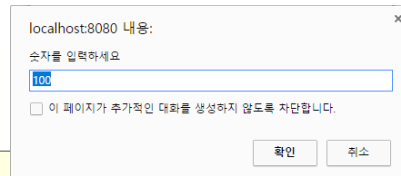
[04_output.html 파일]

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>입출력 예제</title>
<script>
var num = prompt("숫자를 입력하세요", 100);
alert("입력 받은 숫자는 " + num + " 입니다.");
confirm("입력 받은 숫자를 \n브라우저에 출력할까요?");
document.write("입력 받은 숫자 <span style='color:red'><b>"
    + num + "</b></span> 을 브라우저에 출력 했지요?");
</script>
</head>
<body>

</body>
</html>

```



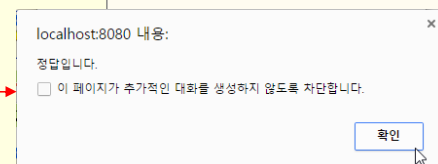
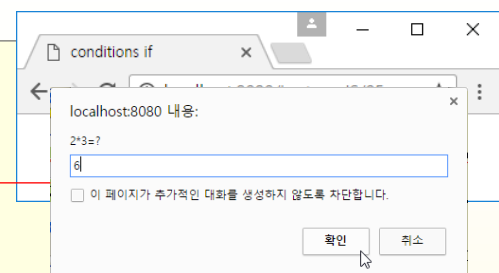
□ if(conditions) else ~

[05_conditions.html 파일]

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset="UTF-8">
5. <title>conditions if</title>
6. <script>
7. var num = prompt("2*3=?", "");
8. if(num == 6) {
9.     alert("정답입니다.");
10. }else {
11.     alert('오답입니다. ');
12. }
13. </script>
14. </head>
15. <body>
16. </body>
17. </html>

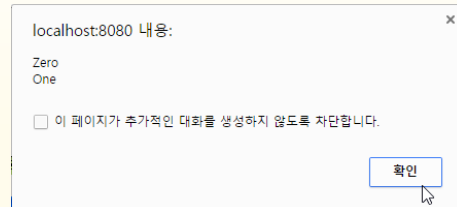
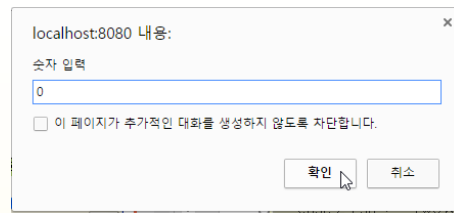
```



Session 04 Conditions

□ switch(expression) case ~ default

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Conditions switch</title>
<script>
var str="";
var num = prompt("숫자 입력", "");
switch( parseInt(num) ) {
    case 0 : str += "Zero\n";
    case 1 : str += "One\n";
               break;
    case 2 : str += "Two\n";
               break;
    case 3 : str += "Three\n";
               break;
    default : str = "잘못 입력";
}
alert(str);
</script>
</head>
<body>
</body>
</html>
```

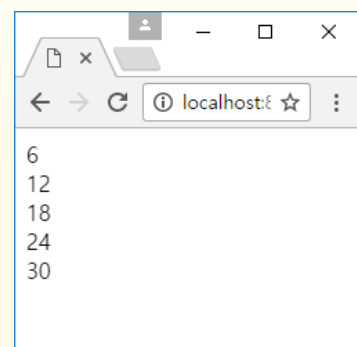


Session 04 Loop

□ while

```
var 변수 = 초기값;
while (조건식) {
    실행문;
    증감식;
}
```

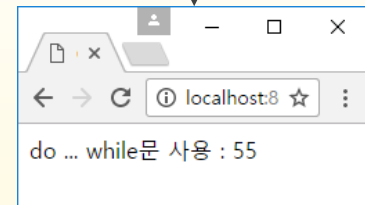
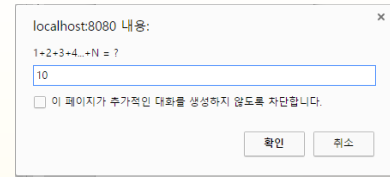
```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>while</title>
<script>
var i=1;
while(i<=30) {
    if(i%2==0 && i%6==0) //2의 배수이면서 6의 배수
        document.write(i, "<br>");
    i++;
}
</script>
</head>
<body>
</body>
</html>
```



do ~ while

```
var 변수 = 초기값;
do{
    실행문;
    증감식;
} while (조건식);
```

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>do while</title>
<script>
var num = prompt("1+2+3+4...+N = ?", "");
sum=0, i=1;
do {
    sum += i;
    i++;
} while( i<=num );
document.write("do ... while문 사용 : " + sum);
</script>
</head>
<body>
</body>
</html>
```



for

```
for (초기식; 조건식; 증감식) {
    code block to be executed;
}
```

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>loop</title>
<script>
document.write("<table border='1' style='border-collapse:collapse'>");
document.write("<tr><td>섭씨온도</td><td>화씨온도</td></tr>");

for (celsius = 0; celsius <= 10; ++celsius) {
    document.write("<tr><td>" + celsius + "</td><td>"
        + ((celsius * 9.0 / 5) + 32) + "</td></tr>");
}

document.write("</table>");
</script>
</head>
<body>
</body>
</html>
```

섭씨온도	화씨온도
0	32
1	33.8
2	35.6
3	37.4
4	39.2
5	41
6	42.8
7	44.6
8	46.4
9	48.2
10	50

Session 04 Loop

중첩 for

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>loop</title>
<script>
var num = prompt("Line Number", "");

for(a=1; a<=num ; a++) {
    for(b=1; b<=a; b++)
        document.write("*");
    document.write("<br>");
}
</script>
</head>
<body>
</body>
</html>
```

localhost:8080 내용:

Line Number

5

확인 취소

```
*
**
***
****
*****
```

Session 04 Break and Continue

break

[11_break.html 파일]

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>break</title>
<script>
while(1)
{
    a = prompt("A입력(A / b)", "");
    b = prompt("B입력(a / B)", "");
    if(a==0 || b==0) break;
    document.write(a + "/" + b + "=" + (a/b) + " ... " + (a%b) + "<br>");
}
</script>
</head>
<body>
</body>
</html>
```

```
break
2/1=2 ... 0
4/3=1.3333333333333333 ... 1
3/1=3 ... 0
```

localhost:8080 내용:

A입력(A / b)

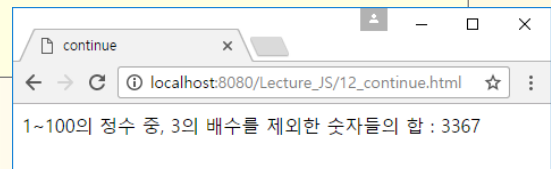
☐ 이 페이지가 추가적인 대화를 생성하지 않도록 차단합니다.

확인 취소

□ continue

[12_continue.html 파일]

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>continue</title>
<script>
var sum=0;
for(a=1; a<=100; ++a) {
    if(a%3 == 0)
        continue;
    sum += a;
}
document.write("1~100의 정수 중, 3의 배수를 제외한 숫자들의 합 : ", sum);
</script>
</head>
<body>
</body>
</html>
```



Session 04 함수(function)

□ JavaScript function

- 함수는 실행될 수 있는 코드들의 집합.
- 함수를 호출하려면 반드시, 호출전에 정의되어 있어야 함.
- 함수를 사용하는 이유는 자주 사용하는 코드를 재 사용할 수 있으며, 프로그래밍 코드를 좀 더 블록화 시킬 수 있기 때문임.
- 주로 이벤트 핸들러에서 호출 사용하지만, 일반 스크립트에서도 사용됨
- 함수의 매개변수엔 자료형을 선언하지 않음
- 함수내의 결과값을 반환할 수 있지만, 반환 자료형은 선언하지 않음

[선언]

```
function 함수명(param1, param2, ..... ) {
    자바스크립트 실행문;
    .....;
    return 변수 또는 상수;
}
```

[호출]

```
함수명(param1, param2, .....);
```

[선언]

```
참조변수 = function () {
    자바스크립트 실행문;
    ...
}
```

[호출]

```
참조변수명();
```

Session 04 함수(function)

□ 사용 예

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>함수 사용 예</title>
<script>
function sum(num1, num2) {
    alert(num1 + num2);
}

var myfun = function (str) {
    alert(str);
}
</script>
</head>
<body>
<p>
<button onclick="sum(20,10)">버튼 합계</button>
<div onclick="myfun('참조변수 함수 호출')">참조변수를 활용한 함수 호출</div>
</p>
</body>
</html>
```

Session 04 JavaScript Scope(변수의 사용범위)

□ 지역변수(local variable)와 전역변수(global variable)

구분	설명
지역 변수	함수 내부에서 var로 선언한 변수, 변수를 선언한 함수 내부에서만 사용 가능
전역 변수	함수 내부에서 선언되지 않은 변수, 또는 함수 내부에서 var로 선언하지 않은 변수, 페이지의 어디에서나 사용 가능

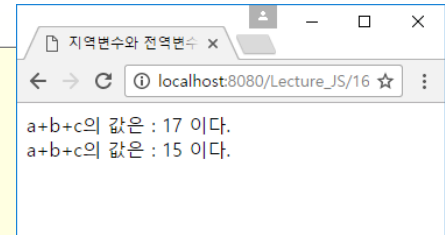
Session 04 JavaScript Scope(변수의 사용범위)

□ 사용 예

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>지역변수와 전역변수</title>
<script>
var a = 2;
b = 7;

function add() {
    var a = 4;
    c = 6;    //전역변수
    document.write("a+b+c의 값은 : " + (a + b + c) + " 이다.", "<br>");
}

add();
document.write("a+b+c의 값은 : " + (a + b + c) + " 이다.", "<br>");
</script>
</head>
<body>
</body>
</html>
```



Session 04 JavaScript Global Functions

□ Global Functions

Function	Description	사용예
decodeURI()	Decodes a URI	
decodeURIComponent()	Decodes a URI component	
encodeURI()	Encodes a URI	
encodeURIComponent()	Encodes a URI component	
<u>eval(str)</u>	Evaluates a string and executes it as if it was script code	eval("5+4") : 9
isFinite(value)	value가 유효한 유효한 숫자인지 검사 +infinity, -infinity, NaN 인 경우 false, 나머지는 true를 반환	isFinite(0) : true isFinite("2017/01") : false
isNaN(value)	value가 유효한 숫자인지 검사 문자가 포함되거나, NaN, undefined일 경우 true 반환	isNaN("5-3") : true isNaN("53") : false
Number(object)	Converts an object's value to a number	Number("5") : 5 Number("5 5") : NaN
<u>parseFloat(str)</u>	str 문자열을 파싱하여 실수 값 출력 (숫자문자와 ".", "e", "E" 이외의 다른 문자를 만날 때까지의 숫자를 실수로 파싱함)	parseFloat("5.12") : 5.12
parseInt(str, radix)	str 문자열을 파싱하여 radix 진법(2~36)에 해당하는 10진수 출력 (지정한 진법에 대하여, 숫자문자 이외의 다른 문자를 만날 때까지의 숫자를 정수로 파싱함. 예를들어, 16진법의 경우, a~f, A~F까지의 문자가 숫자로 간주됨)	parseInt("5.12") : 5
<u>String(object)</u>	Converts an object's value to a string	String(5) : "5"

□ eval()

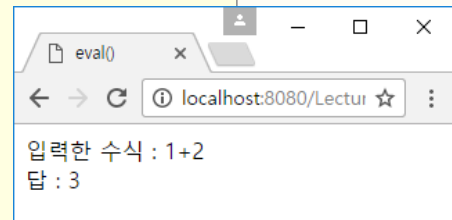
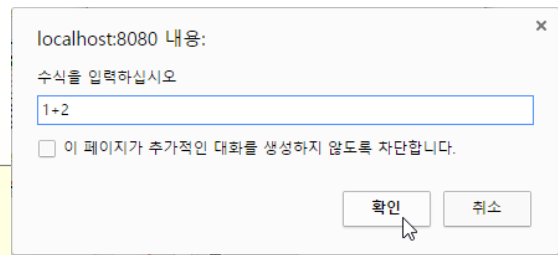
➤ 문자열 수식을 계산함

[17_eval.html 파일]

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>eval()</title>
<script>
var numstr=prompt("수식을 입력하십시오");
document.write("입력한 수식 : "+numstr+"<br>");
document.write("답      : "+eval(numstr));
</script>
</head>
<body>
</body>
</html>

```



□ parseInt(str, radix), parseFloat(str)

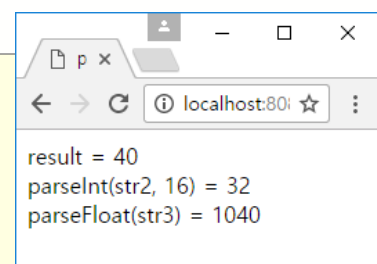
[18_parseInt_Float.html 파일]

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>parseInt와 parseFloat</title>
<script>
var str1, str2, str3;
str1 = "20kr5";
str2 = "20.45";
str3 = "10.4e2k";

result = parseInt(str1) + parseInt(str2);
document.write("result = " + result + "<br>");
document.write("parseInt(str2, 16) = " + parseInt(str2, 16) + "<br>");
document.write("parseFloat(str3) = " + parseFloat(str3));
</script>
</head>
<body>
</body>
</html>

```



□ Event

- 브라우저에서 방문 사용자가 취하는 모든 동작이나 브라우저의 상태
- 이벤트 종류

구분	이벤트	설명
마우스	onmouseover	마우스가 지정한 요소에 올라갔을 때 발생
	onmouseout	마우스가 지정한 요소에서 벗어났을 때 발생
	onmousemove	마우스가 지정한 요소 영역에서 움직였을 때 발생
	onclick	마우스가 지정한 요소를 클릭했을 때 발생
	ondblclick	마우스가 지정한 요소를 빠른 시간 내에 연속 2회 눌렀을 때 발생
키보드	onkeypress	지정한 요소에서 키보드가 눌렀을 때 발생
	onkeydown	지정한 요소에서 키보드를 눌렀을 때 발생
	onkeyup	지정한 요소에서 키보드를 떼었을 때 발생
기타	onfocus	지정한 요소에 포커스가 갔을 때 발생
	onblur	지정한 요소에서 포커스가 다른 요소로 이동되어 잃었을 때 발생
	onchange	지정한 요소에 value 속성값이 바뀌고, 포커스가 이동되었을 때
	onload	지정한 요소의 하위 요소를 모두 로딩했을 때 발생
	onunload	문서를 닫거나 다른 문서로 이동되었을 때 발생
	onsubmit	폼 요소의 전송 버튼을 눌렀을 때 발생
	onreset	폼 요소의 취소 버튼을 눌렀을 때 발생

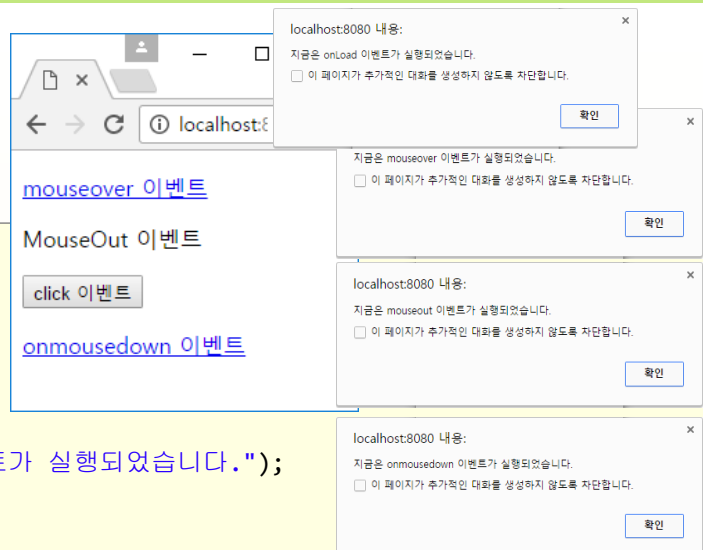
□ 예제 1

[19_event01.html 파일]

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset="UTF-8">
5. <title>Event</title>
6. <script>
7. function showMsg(str) {
8.     alert("지금은 " + str + " 이벤트가 실행되었습니다.");
9. }
10. </script>
11. </head>
12. <body onload="showMsg('onLoad')" onunload="showMsg('onUnload')"
13.     onresize="showMsg('resize')">
14. <p><a href="#" onmouseover="showMsg('mouseover')">mouseover 이벤트</a></p>
15. <div onmouseout="showMsg('mouseout')">MouseOut 이벤트 </div>
16. <p><button onclick="showMsg('click')">click 이벤트</button></p>
17. <p><a href="#" onmousedown="showMsg('mousedown')">mousedown 이벤트</a></p>
18. </body>
19. </html>

```

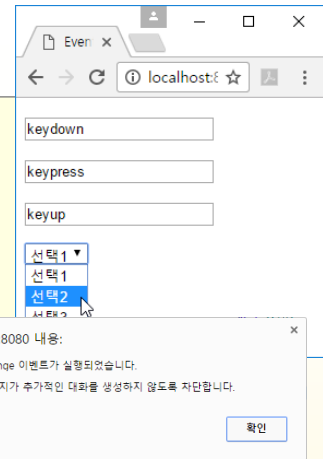


예제2

[19_event02.html 파일]

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset="UTF-8">
5. <title>Event</title>
6. <script>
7. function aaa(str) {
8.     alert("지금은 " + str + " 이벤트가 실행되었습니다.");
9. }
10. </script>
11. </head>
12. <body>
13. <p><input type="text" value="keydown" onkeydown="aaa('keydown')"></p>
14. <p><input type="text" value="keypress" onkeypress="aaa('keypress')"></p>
15. <p><input type="text" value="keyup" onkeyup="aaa('keyup')"
16.                                     onblur="aaa('blur')"></p>
17. <select name="select" onchange="aaa('change')">
18.     <option value="1">선택1</option>
19.     <option value="2">선택2</option>
20.     <option value="3">선택3</option>
21. </select>
22. </body>
23. </html>
    
```



객체 (Object)

- 프로그래밍 대상을 속성과 동작으로 묶어서 정의한 자료형
- 속성(Properties : 변수)과 메소드(Method : 함수)로 이루어짐



TV Object

속성(Property)

- power
- volume

메소드(Method)

- turnOn()
- turnOff()
- volumeUP()
- volumeDown()

JavaScript의 객체 종류

- 사용자 정의 객체(Custom Object)
 - ✓ 사용자가 정의한 객체
- 내장 객체(Built-in Object)
 - ✓ JavaScript 엔진에 내장되어 있는 객체(String, Date, Math, Array, Number 등)
- BOM(Browser Object Model)
 - ✓ 브라우저를 제어할 수 있는 객체(window, screen, location, history, navigator 등)
- DOM(Document Object Model)
 - ✓ Tree구조의 HTML 요소에 접근할 수 있는 객체(document)

□ 사용자 정의 객체

➤ Object Literal을 사용하여 생성하기

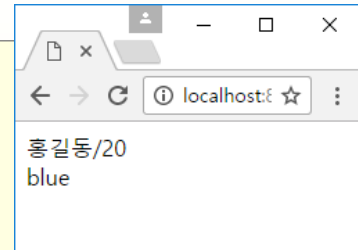
✓ 단일 객체 생성(type이 아님)

```

<script>
var person = {
  name: "홍길동",
  age: 40,
  eyeColor: "blue",
  getEyeColor : function() {
    return this.eyeColor;
  }
};

person.age = 20;
document.write(person.name, "/", person.age, "<br>");
document.write(person.getEyeColor());
</script>

```



➤ new Object()를 사용하여 생성하기

✓ 단일 객체 생성(type이 아님)

```

<script>
var person = new Object();

person.name = "홍길동";
person.age = 40;
person.eyeColor = "blue";
person.getEyeColor = function() {
  return this.eyeColor;
}

person.age = 20;
document.write(person.name, "/", person.age, "<br>");
document.write(person.getEyeColor());
console.log(person);
</script>

```

[크롬에서 F12키]



Session 04 Objects - 사용자 정의 객체

➤ Object constructor를 사용하여 생성하기

- ✓ 동일한 Object Type으로 여러 개의 객체 생성 가능
- ✓ 예약어 function으로 Type을 정의하고, new 연산자로 객체를 생성함
- ✓ 외부 함수 선언

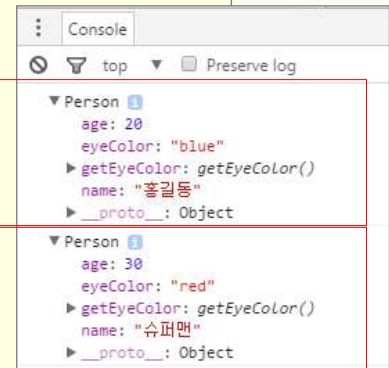
```
<script>
function getEyeColor() {
    return this.eyeColor;
}

function Person(name, age, eyeColor) {
    this.name = name;
    this.age = age;
    this.eyeColor = eyeColor;
    this.getEyeColor = getEyeColor; //외부 함수 선언과 참조
}

var people1 = new Person("홍길동", 40, "blue");

people1.age = 20;
document.write(people1.name, "/", people1.age, "<br>");
document.write(people1.getEyeColor());

var people2 = new Person("슈퍼맨", 30, "red");
console.log(people1);
console.log(people2);
</script>
```



Internet Programming

37

Session 04 Objects - 사용자 정의 객체

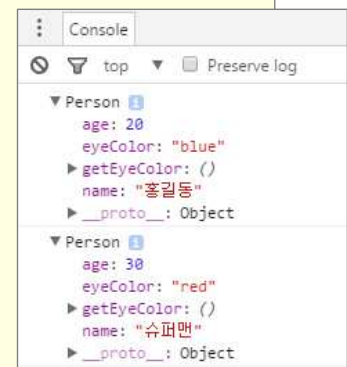
✓ 내부 함수 선언

```
<script>
function Person(name, age, eyeColor) {
    this.name = name;
    this.age = age;
    this.eyeColor = eyeColor;
    this.getEyeColor = function () { //내부 함수 선언
        return this.eyeColor;
    };
}

var people1 = new Person("홍길동", 40, "blue");

people1.age = 20;
document.write(people1.name, "/", people1.age, "<br>");
document.write(people1.getEyeColor());

var people2 = new Person("슈퍼맨", 30, "red");
console.log(people1);
console.log(people2);
</script>
```



[크롬에서 F12키]

Internet Programming

38

Session 04 Objects Properties

□객체 속성 접근

➤syntax

```
objectName.property
```

dot scope operator 접근법

```
objectName["property"]
```

name index 접근법

➤사용 예

```
<script>
var person = { name:"홍길동", age:40, eyeColor:"blue" };
person.age = 20;
person["name"] = "슈퍼맨";
var vname = "eyeColor";
person[vname] = "red";

var people = person;
people.age = 30;      // person.age의 값도 같이 변경됨(이유 : 참조하기 때문).

person.nationality = "English";    //멤버변수(속성) 신규 추가
person.getEyeColor = function () { return this.eyeColor; } // 멤버 메소드 신규 추가
delete person.age;                // 멤버변수(속성) 삭제 (또는, delete person["age"] )
...
</script>
```

Internet Programming

39

Session 04 Object - Prototype

□prototype

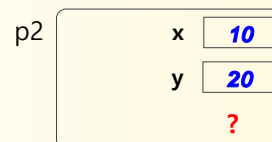
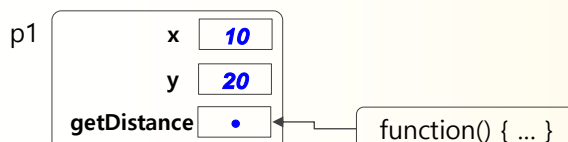
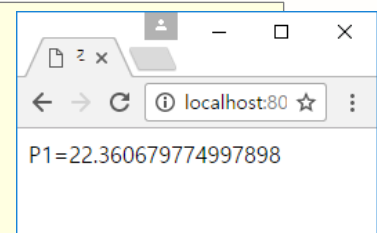
➤JavaScript의 모든 객체들에 존재

➤이미 선언된 객체의 타입에 객체의 멤버들을 추가할 때 사용함

```
<script>
function Point(xpos, ypos) {
    this.x = xpos;
    this.y = ypos;
}
var p1 = new Point(10, 20);
p1.getDistance = function () {
    return Math.sqrt(this.x * this.x + this.y * this.y);
}

var p2 = new Point(10, 30);

document.write("P1=" + p1.getDistance() + "<br>");
document.write("P2=" + p2.getDistance() + "<br>"); //error
</script>
```



Internet Programming

40

Session 04 Object - Prototype

□ prototype 사용 예

```
<script>
function Point(xpos, ypos) {
    this.x = xpos;
    this.y = ypos;
}

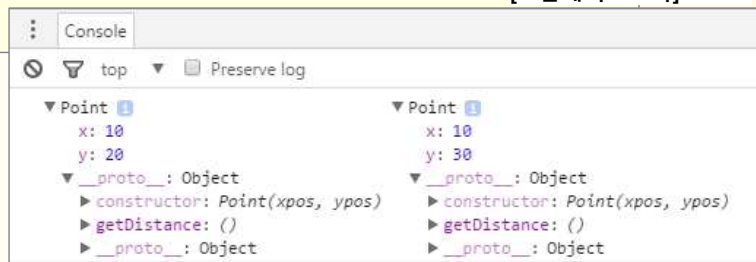
Point.prototype.getDistance = function () {
    return Math.sqrt(this.x * this.x + this.y * this.y);
}

var p1 = new Point(10, 20);
var p2 = new Point(10, 30);

document.write("P1=" + p1.getDistance() + "<br>");
document.write("P2=" + p2.getDistance() + "<br>");
console.log(p1, p2);
</script>
```

P1=22.360679774997898
P2=31.622776601683793

[크롬에서 F12키]



Internet Programming

41

Session 04 Objects - for/in Loop

□ for/in

- object의 속성변수가 없을 때까지 object의 멤버를 반복하여 탐색함
- Array에 대해서도 사용할 수 있음

```
for (속성변수 in 객체) {
    code block to be executed;
}
```

```
<script>
var person = {
    name: "홍길동",
    age: 20,
    eyeColor: "blue",
    getEyeColor: function() {
        return this.eyeColor;
    }
};

for(var vname in person) {
    document.write(vname, " : ", person[vname], "<br>");
}
</script>
```

name : 홍길동
age : 20
eyeColor : blue
getEyeColor : function () { return this.eyeColor; }

Internet Programming

42

Session 04 배열 (array)

□ JavaScript에서의 배열

- 하나의 변수에 여러 값을 저장하기 위해 사용
- 항상, index에 의해 배열의 각 요소에 접근함 (name으로 접근하는 것은 Array Object가 아닌 일반 Object임)
- 선언시에 배열의 크기를 정의하지 않아도 되며, 선언 후 index의 범의를 벗어나도 접근 가능

□ 배열 선언 방법

➤ literal array

```
var array-name = [item1, item2, ...];
```

```
var arr = [];  
arr[0] = 'a';
```

➤ new 연산자에 의한 Array 내장객체 생성

- ✓ 가독성, 실행속도 등이 더 좋은 literal array 생성방법을 권장함

```
var variable-name = new Array(item1, item2, ...);
```

```
var arr = new Array();  
arr[0] = 'a';
```

Session 04 배열 (array) - literal array

□ literal array 생성 예제

```
<!DOCTYPE html>  
<html>  
<head>  
<meta charset="UTF-8">  
<title>array</title>  
<script>  
var arr = ["Bus", 512, "Taxi"]; //literal array 선언  
document.write("typeof arr = ", typeof arr, "<br>");  
  
for(var i=0; i<arr.length; i++) {  
    document.write(i + " ", typeof arr[i], " ", arr[i], "<br>");  
}  
arr[4] = 10000; //undefined형의 arr[3]이 생성됨  
document.write("arr.length=", arr.length, "<br>");  
console.log(arr);  
</script>  
</head>  
<body>  
</body>  
</html>
```

typeof arr = object
0, string, Bus
1, number, 512
2, string, Taxi
arr.length=5

[크롬에서 F12키]

html

Console Animations

top Preserve log

▼ Array[5]

- 0: "Bus"
- 1: 512
- 2: "Taxi"
- 4: 10000
- length: 5
- __proto__: Array[0]

[13_array.html 파일]

□ Array Property and Methods

Property and Method	Description
length	배열의 길이(undefined 포함)
concat(array2, array3, ...)	2개 또는 그 이상의 배열 객체를 하나로 합친 배열 객체를 반환
join(separator)	array의 모든 요소 사이에 separator 문자를 삽입한 결과를 string으로 반환
pop()	배열에 저장된 데이터 중 마지막 인덱스에 저장된 데이터를 반환하고 삭제
push(item1, item2, ...)	배열의 마지막에 새로운 item 요소들을 추가하고, 새로운 배열 length를 반환
reverse()	배열 객체의 데이터 순서를 거꾸로 한 뒤 반환
shift()	배열의 데이터 중 첫번째 인덱스에 저장된 데이터를 반환하고 삭제
slice(start, end)	start와 end index 사이의 배열을 반환(단, end index는 포함시키지 않음)
sort(compareFunction)	배열 데이터에 대하여 문자를 기준으로 오름차순 정렬함 compareFunction에 정렬 기준을 지정할 수 있음
splice(index, howmany, item1,)	배열 index 위치에 howmany만큼 요소를 제거하고, item 요소들을 추가함
unshift(item1, item2, ...)	배열의 가장 앞 위치에 item 요소들을 삽입하고, 새로운 배열의 length를 반환

□ new 연산자에 의한 Array 객체 생성 예

```

<script>
var fruits = new Array("apple","banana","orange");
fruits[fruits.length] = "pear";

for(var idx in fruits) {    //배열에서의 for~in 문 사용
    document.write(fruits[idx], "<br>")
}

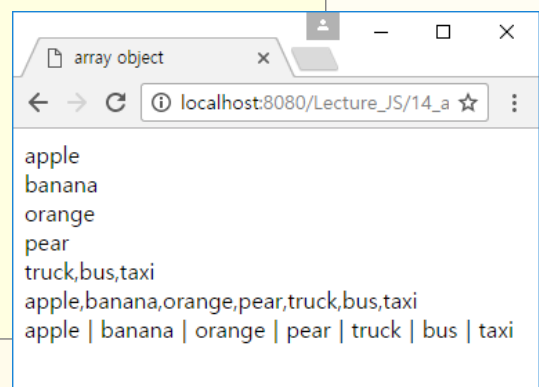
var cars = new Array();
cars[0] = "truck";
cars[1] = "bus";
cars[2] = "taxi";

document.write(cars, "<br>");

var arr = fruits.concat(cars);
document.write(arr, "<br>");
document.write(arr.join(" | "), "<br>");
</script>

```

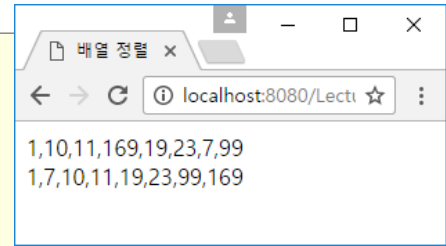
[14_array_object.html 파일]



Session 04 Objects - 내장객체 Array

□ Array 정렬 예

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset="UTF-8">
5. <title>배열 정렬</title>
6. <script>
7. var myArray = [10, 7, 23, 99, 169, 19, 11, 1];
8. myArray.sort() //알파벳 문자 기준 정렬
9. document.write(myArray, "<br>");
10.
11. myArray.sort(
12.     function(a, b) { //정렬 기준 지정
13.         return a-b;
14.     });
15.
16. document.write(myArray, "<br>");
17. </script>
18. </head>
19. <body>
20. </body>
21. </html>
```



[26_array_sort.html 파일]

Session 04 Objects - 내장객체 Number

□ Number

➤new 예약어로 선언되지 않더라도 객체가 될 수 있음

Properties and Method	Description
MAX_VALUE	Returns the largest number possible in JavaScript
MIN_VALUE	Returns the smallest number possible in JavaScript
NEGATIVE_INFINITY	Represents negative infinity (returned on overflow)
POSITIVE_INFINITY	Represents infinity (returned on overflow)
isFinite()	Checks whether a value is a finite number
isInteger()	Checks whether a value is an integer
isNaN()	Checks whether a value is Number.NaN
isSafeInteger()	Checks whether a value is a safe integer
toExponential(x)	Converts a number into an exponential notation
toFixed(x)	Formats a number with x numbers of digits after the decimal point
toPrecision(x)	Formats a number to x length
toString()	Converts a number to a string
valueOf()	Returns the primitive value of a number

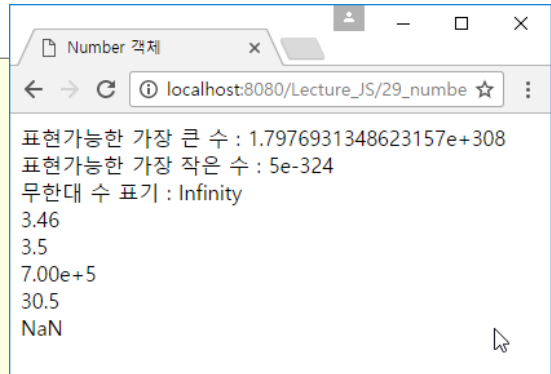
□ 사용 예

[29_number.html 파일]

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>Number 객체 </title>
<script>
var num1=3.456789;
var num2=700000;
var num3=new Number("30.5");
```

```
document.write("표현가능한 가장 큰 수 : " + Number.MAX_VALUE, "<br>");
document.write("표현가능한 가장 작은 수 : " + Number.MIN_VALUE, "<br>");
document.write("무한대 수 표기 : " + Number.POSITIVE_INFINITY, "<br>");

document.write(num1.toFixed(2), "<br>"); //소숫점이하 숫자개수 지정
document.write(num1.toPrecision(2), "<br>"); //전체 숫자 자리수 지정
document.write(num2.toExponential(2), "<br>"); //지수에서의 소수점 자리수 지정
document.write(num3, "<br>");
document.write(Number("10.0px"), "<br>");
</script>
</head>
<body>
</body>
</html>
```



□ Math

➤ Properties and Methods

Property and Method	Description
E	Euler's number (2.718)
LN10	자연 로그(밑수 10) (2.302)
PI	파이 상수 (3.14)
abs(x)	x에 대한 절대값 반환
acos(x), asin(x), atan(x)	x에 대한 아크 삼각함수 값 반환, 단 x는 radian 임($radian = degree \times \frac{\pi}{180}$)
cos(x), sin(x), tan(x)	x에 대한 삼각함수 값 반환, 단, x는 radian임
exp(x)	Returns the value of E^x
log(x)	Returns the natural logarithm (base E) of x
ceil(x)	무조건 올림한 정수를 반환
floor(x)	무조건 버림한 정수를 반환
max(x, y, z, ..., n)	Returns the number with the highest value
min(x, y, z, ..., n)	Returns the number with the lowest value
pow(x, y)	Returns the value of x to the power of y
random()	0~1 사이의 난수를 반환
round(x)	반올림 한 정수를 반환

Session 04 Objects - 내장객체 Math

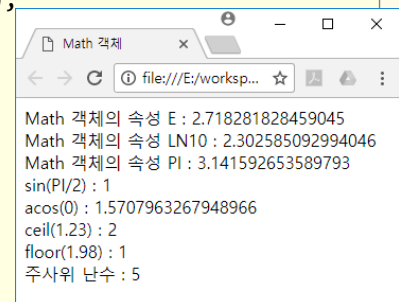
➤ 사용 예

[28_Math.html 파일]

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Math 객체</title>
<script>
// referenced by MDN web docs (https://developer.mozilla.org)
function getRandom(min, max) {
    return Math.floor(Math.random()*(max-min+1)+min);
}

document.write("Math 객체의 속성 E      : " + Math.E + "<br>");
document.write("Math 객체의 속성 LN10 : " + Math.LN10 + "<br>");
document.write("Math 객체의 속성 PI   : " + Math.PI + "<br>");
document.write("sin(PI/2) : " + Math.sin(Math.PI/2) + "<br>");
document.write("acos(0) : " + Math.acos(0) + "<br>");
document.write("ceil(1.23) : " + Math.ceil(1.23) + "<br>");
document.write("floor(1.98) : " + Math.floor(1.98) + "<br>");

// 1 ~ 6 사이의 주사위 난수 구하기
document.write("주사위 난수 : ", getRandom(1, 6), "<br>");
</script>
</head>
<body>
</body>
</html>
```



51

Session 04 Objects - 내장객체 Date

□ Date

➤ Syntax

```
var d = new Date(); //현재 날짜 정보를 알아냄
var d = new Date(milliseconds); //1970년1월1일 이후의 millisec 설정
var d = new Date(dateString); // "월 일, 년 시:분:초" 설정
var d = new Date(year, month, day, hours, minutes, seconds, milliseconds);
```

➤ Methods

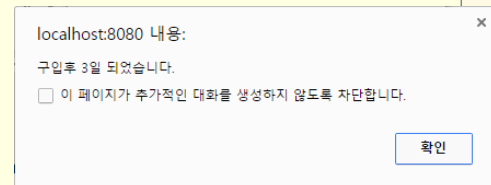
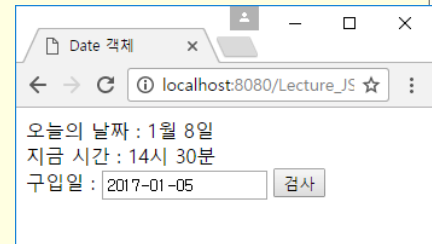
get Method	set Method	기능
getFullYear()	setFullYear(x)	두자리 숫자의 연도를 리턴 또는 설정
getMonth()	setMonth(x)	0에서 11 사이의 두자리 숫자 월을 리턴 또는 설정(0 : 1월)
getDate()	setDate(x)	1에서 31 사이의 날짜 리턴 또는 설정
getDay()	setDay(x)	0에서 6사이의 요일 리턴 또는 설정(0: 일요일)
getHours()	setHours(x)	0에서 23 사이의 숫자로 시간 리턴 또는 설정
getMinutes()	setMinutes(x)	0에서 59 사이의 분 리턴 또는 설정
getSeconds()	setSeconds(x)	0에서 59 사이의 초 리턴 또는 설정
getTime()	setTime(x)	1970년 1월 1일 이후의 시간을 1000분의 1초 단위로 출력 또는 설정

Session 04 Objects - 내장객체 Date

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset="UTF-8">
5. <title>Date 객체</title>
6. <script>
7. var today = new Date();
8. document.write("오늘의 날짜 : " + (today.getMonth()+1)+ "월 "
9.     + today.getDate() + "일<br>");
10. document.write("지금 시간 : " + today.getHours() + "시 "
11.     + today.getMinutes() + "분<br>");
12.
13. function checkDate() {
14.     var p = document.getElementById("pdate").value;
15.     var pdate = new Date(p);
16.     var diff = today.getTime() - pdate.getTime();
17.     var days = Math.floor(diff / (1000*60*60*24));
18.
19.     alert("구입후 " + days + "일 되었습니다.");
20. }
21. </script>
22. </head>
23. <body>
24. 구입일 : <input type="date" id="pdate">
25. <button onclick="checkDate()">검사</button>
26. </body>
27. </html>

```



Session 04 Objects - 내장객체 String

□ String

➤ new 예약어로 선언되지 않더라도 객체가 될 수 있음

➤ Properties and Methods

Property and Method	Description
length	문자열의 길이
charAt(index)	지정한 index 위치의 문자를 반환
concat(str2, str3, ...)	둘 또는 그 이상의 str 문자열을 결합하여, 결합된 문자열을 반환
indexOf(value, start)	지정한 value 문자열이 있는 index를 반환. start index 생략시 0번부터 검색함
lastIndexOf(value, start)	지정한 value 문자열이 있는 index를 반환, start index 생략시 문자열 맨 마지막 부터 검색함
replace(searchvalue, newvalue)	지정한 searchvalue(또는 regular expression) 문자열을 찾아서, newvalue 문자열로 바꾼 문자열을 반환
search(searchvalue)	지정한 searchvalue(또는 regular expression) 문자열을 찾아서, 위치 index를 반환
slice(start, end)	start와 end index 사이의 문자열을 반환, 단, end index 제외
split(separator, limit)	지정한 separator를 기준으로 문자열을 잘라서, array로 반환. limit는 separate 할 개수를 지정함(생략시 전체 문자열)
substr(start, length)	start index부터 length 길이의 문자열을 반환
substring(start, end)	start와 end index 사이의 문자열을 반환, 단, end index 제외
toLowerCase()	Converts a string to lowercase letters
toUpperCase()	Converts a string to uppercase letters
trim()	문자열 양 끝의 빈공백 문제들을 제거

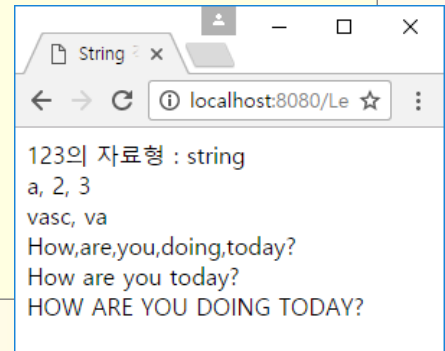
□ 사용 예

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset="UTF-8">
5. <title>String 객체</title>
6. <script>
7. var nst = String(123);
8. document.write(nst, "의 자료형 : ", typeof nst, "<br>");
9. var str1 = "javascript";
10. document.write(str1.charAt(3), ", ", str1.indexOf("v"), ", ",
11. str1.lastIndexOf("a"), "<br>");
12. document.write(str1.substr(2, 4), ", ", str1.substring(2, 4), "<br>");
13.
14. var str2 = new String("How are you doing today?");
15. document.write(str2.split(" "), "<br>");
16. document.write(str2.replace("doing", ""), "<br>");
17. document.write(str2.toUpperCase(), "<br>");
18. </script>
19. </head>
20. <body>
21. </body>
22. </html>

```

[30_string.html 파일]



□ RegExp

- 문자열에 대한 패턴을 정의하는 객체
- 패턴 매칭(pattern- matching)을 통해 입력 형식을 제한할 수 있음
- Syntax

- ✓ RegExp 객체를 사용한 패턴 정의

```
var patt=new RegExp(pattern, modifiers);
```

- ✓ 문자열(string)을 사용한 패턴 정의

```
var patt=/pattern/modifiers;
```

- 수정자(Modifiers)

Modifier	설명
i	대소문자 구분 없이 매칭, 없으면 대소문자 구분함
g	처음부터 전체에 걸쳐서 모든 일치하는 것 찾음
m	여러 라인에 걸쳐서 매칭

➤ 괄호(Brackets)

Expression	설명
[abc]	대괄호 내의 문자들을 찾음
[^abc]	대괄호 내의 문자들을 제외한 문자들을 찾음
[0-9]	0에서 9까지의 숫자문자를 찾음
[A-Z]	대문자 A부터 Z까지의 문자를 찾음
[a-z]	소문자 a부터 z까지의 문자를 찾음
[A-z]	대문자 A부터 소문자 z까지의 문자를 찾음
(red blue green)	지정된 선택문자들 중 하나를 찾음

➤ 메타문자(Metacharacters)

Metacharacter	설명
.	줄 바꿈 또는 라인 끝 표시문자를 제외하고 단일 문자를 찾음
\w (소문자 w)	영문/숫자 문자를 찾음
\W (대문자 W)	영문/숫자 문자가 아닌 특수문자, 빈 공백 등의 문자를 찾음
\d	숫자를 찾음
\D	숫자가 아닌 문자를 찾음
\s (소문자 s)	공백 문자를 찾음
\S (대문자 S)	공백이 아닌 문자를 찾음
\b	단어의 시작 또는 끝 부분에 일치하는 것 찾음
\B	단어의 시작 또는 끝 부분이 아닌 부분에서 일치하는 것 찾음. 즉, 단어의 처음이나 끝부분이 아닌 곳에서 지정한 문자와 일치하는 문자를 찾음

➤ 수량 한정자(Quantifiers)

Quantifier	설명
n+	적어도 하나의 n을 포함하는 문자열을 일치
n*	0 개 이상의 n이 포함 된 문자열을 일치시킴
n?	0 또는 1번 포함하는 문자열을 일치시킴
n{X}	n을 X번 포함하는 문자열을 일치시킴
n{X,Y}	n이 X번에서 Y번 포함하는 문자열을 일치시킴
n{X,}	n이 최소 X번 이상 포함하는 문자열을 일치시킴
n\$	끝에 n을 가지는 문자열을 일치시킴
^n	n으로 시작하는 문자열을 일치시킴

➤ RegExp 객체 메소드(Methods)

Method	설명
exec()	문자열에서 일치하는지를 검사하고, 일치된 첫 번째를 반환
test()	문자열에서 일치하는지를 검사하고 true 또는 false를 반환
toString()	regular expression의 문자열 값을 반환

➤ String 객체 메소드(Methods)

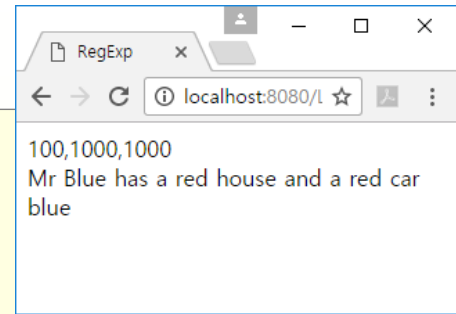
Method	설명
match(regex)	regex에 매치되는 문자열을 배열로 반환, 매치되는 문자열이 없으면 null을 반환
search(searchValue)	searchValue(또는 regex) 문자열과 일치되는 position index를 반환
replace(searchValue, newValue)	searchValue(또는 regex)와 일치되는 문자열을 newValue로 바꿈

□ 사용 예 1

[31_RegExp.html 파일]

```

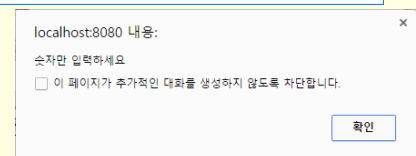
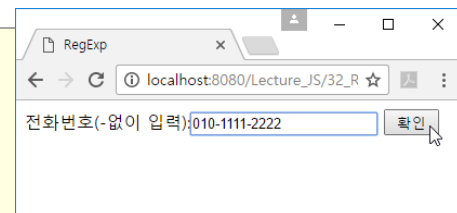
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset="UTF-8">
5. <title>RegExp</title>
6. <script>
7. var str="100, 1000 or 10000?";
8. var patt1=/\d{3,4}/g;           //3~4자리 숫자 찾기
9. document.write(str.match(patt1), "<br>"); // 100, 1000, 1000
10.
11. var str = "Mr Blue has a blue house and a blue car";
12. var res1 = str.replace(/blue/g, "red");
13. document.write(res1, "<br>");
14.
15. var patt2 = /blue/g;
16. var res2 = patt2.exec(str);
17. document.write(res2, "<br>");
18. </script>
19. </head>
20. <body>
21. </body>
22. </html>
    
```



□ 사용 예2

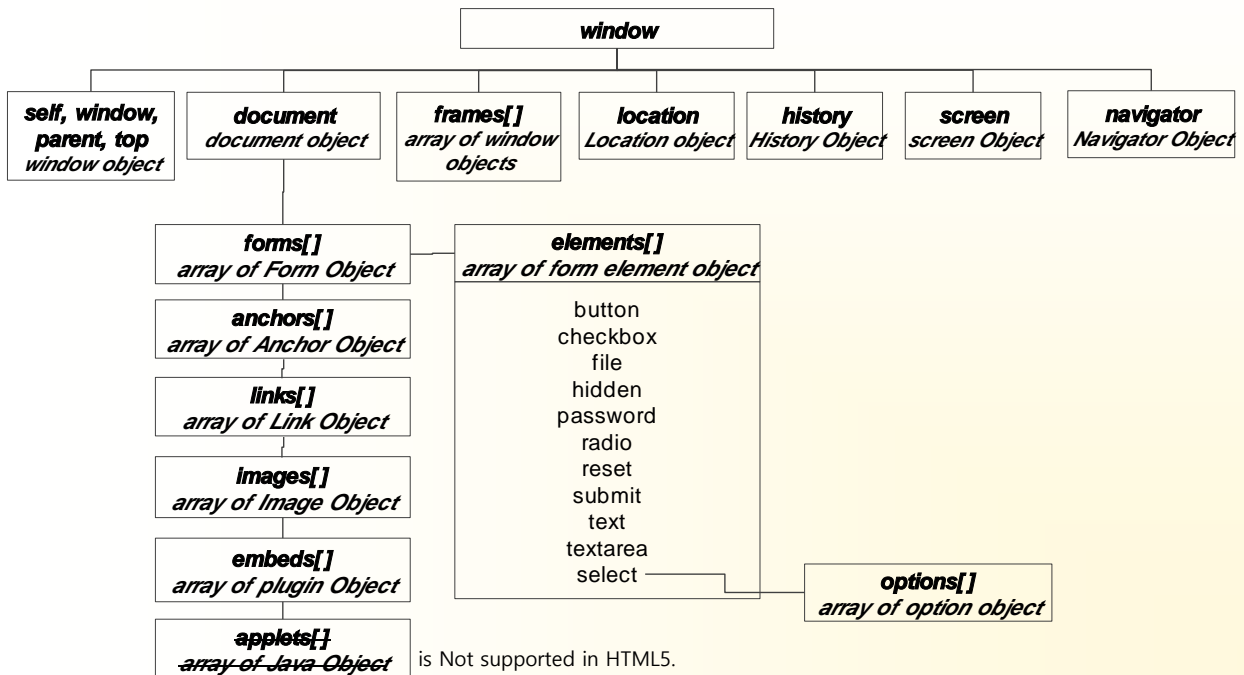
```

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset="UTF-8">
5. <title>RegExp</title>
6. <script>
7. function checkNumeric(msg) {
8.     var exp = /^[0-9]+$ /g;
9.     var obj = document.getElementById("phone");
10.    if(obj.value.match(exp)) {
11.        return true;
12.    } else {
13.        alert(msg);
14.        obj.focus();
15.        return false;
16.    }
17. }
18. </script>
19. </head>
20. <body>
21. <form>
22. 전화번호(-없이 입력):<input type="text" id="phone">
23. <input type="button" onClick="checkNumeric('숫자만 입력하세요')" value=" 확인 ">
24. </form>
25. </body>
26. </html>
    
```



□ BOM(Browser Object Model)

- 브라우저에 내장된 객체로 브라우저 기능 제어 가능
- 최상위 객체는 window이며, 생략 가능함



□ window 객체

- 브라우저의 최상위 객체
- Properties

Property	기능
status	브라우저의 상태 표시줄에 나타나는 문자열
defaultStatus	status의 초기 값 문자열
self	window 자기 자신 객체를 가리킨다.
parent	window 객체간의 계층 구조가 생길 때의 상위 객체
top	window 객체간의 계층 구조가 생길 때의 최상위 객체
frames	현재 window 내의 모든 <iframe> 요소를 반환
opener	open() 메소드로 오픈한 윈도우의 부모 윈도우
innerHeight, innerWidth	윈도에서 내용이 나타나는 영역의 높이와 폭
outerHeight, outerWidth	윈도의 전체 높이와 넓이
pageXOffset, pageYOffset	현재 브라우저의 x 위치, y위치
screen	window에 대한 screen 객체 반환
name	window의 이름 반환하거나 설정
document	window에 대한 Document 객체 반환

➤Methods

Method	Description
alert(msg)	"확인" 버튼과 msg 문자열을 가지는 경고창 출력
confirm(msg)	"확인"과 "취소" 버튼, msg 문자열을 가지는 경고창 출력
prompt(text, defaultText)	사용자로부터 입력 받는 질의/응답 창을 출력
open(URL, name, specs, replace)	새로운 브라우저를 연다.
close()	열린 브라우저를 닫는다.
setTimeout(str, time)	지정된 시간 동안 기다린 후 명령 실행, ex) var tid = setTimeout("a()", 1000)
clearTimeout(tid)	setTimeout() 해제, ex) clearTimeout(tid)
setInterval(str, time)	주기적인 명령 실행, ex) var tid = setInterval('함수()', 1/1000초단위시간)
clearInterval(tid)	setInterval() 해제, ex) clearInterval(tid)
moveBy(x, y)	윈도의 위치에 상대적으로 이동한다.
moveTo(x, y)	윈도의 위치에 절대적으로 이동한다.
resizeBy(width, height)	윈도의 상대적 크기를 바꾼다.
resizeTo(width, height)	윈도의 절대적 크기를 바꾼다.
scrollBy(xnum, ynum)	윈도에 보여지는 부분을 상대적으로 스크롤한다.
stop()	window loading을 중지시킴(브라우저의 stop버튼효과), IE 미지원
print()	윈도에 포함된 내용을 출력한다.

□ window.open() 메소드

➤syntax

```
window.open(URL, name, specs, replace)
```

- URL : 오픈할 페이지의 URL
- name : window 이름, 또는 _blank, _parent, _self, _top
- replace : history list에 현재 문서를 URL로 갱신할지(true), 새로운 entry로 URL을 등록할지(false) 여부 설정
- specs : 윈도우 모양 설정, 콤마(,)로 값 구분

specs options	description
fullscreen=yes no 1 0	Whether or not to display the browser in full-screen mode. Default is no. A window in full-screen mode must also be in theater mode. IE only
height=pixels	The height of the window. Min. value is 100
width=pixels	The width of the window. Min. value is 100
left=pixels	The left position of the window. Negative values not allowed
top=pixels	The top position of the window. Negative values not allowed
location=yes no 1 0	Whether or not to display the address field. Opera only
menubar=yes no 1 0	Whether or not to display the menu bar
resizable=yes no 1 0	Whether or not the window is resizable. IE only
scrollbars=yes no 1 0	Whether or not to display scroll bars. IE, Firefox & Opera only
status=yes no 1 0	Whether or not to add a status bar
titlebar=yes no 1 0	Whether or not to display the title bar. Ignored unless the calling application is an HTML Application or a trusted dialog box
toolbar=yes no 1 0	Whether or not to display the browser toolbar. IE and Firefox only

Session 04 Objects - BOM : window

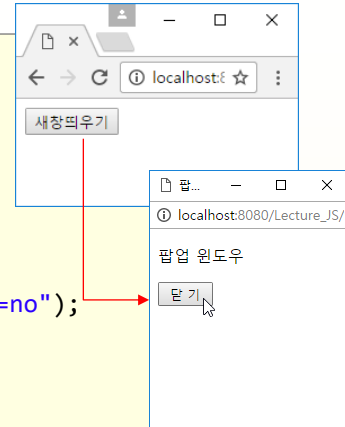
□ window.open() / close() 사용 예

[33_window.html 파일]

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset="UTF-8">
5. <title>새창 띄우기</title>
6. <script>
7. function myPopup() {
8.     window.open("34_window_popup.html", "mypopup",
9.         "height=200,width=200, toolbar=no, location=no");
10. }
11. </script>
12. </head>
13. <body onload="myPopup()">
14. <input type="button" onclick="myPopup()" value="새창띄우기">
15. </body>
16. </html>

```



[34_window_popup.html 파일]

```

...
7. <body>
8. <p>팝업 윈도우</p>
9. <input type="button" onclick="self.close()" value=" 닫 기 ">
10. </body>
11. </html>

```

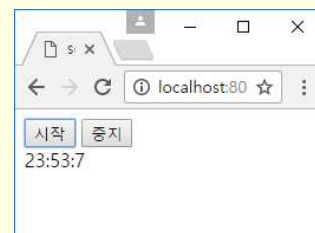
Session 04 Objects - BOM : window

□ setInterval() / clearInterval() 사용 예

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset="UTF-8">
5. <title>setInterval</title>
6. <script>
7. var pid;
8. function showTime() {
9.     var today = new Date();
10.    h = today.getHours();
11.    m = today.getMinutes();
12.    s = today.getSeconds();
13.    obj = document.getElementById("time");
14.    obj.innerHTML = h + ":" + m + ":" + s;
15. }
16. function startTime() {
17.    pid = window.setInterval("showTime()", 1000);
18. }
19. function stopTime() {
20.    window.clearInterval(pid);
21. }
22. </script>
23. </head>
24. <body>
25. <input type="button" value="시작" onclick="startTime()">
26. <input type="button" value="중지" onclick="stopTime()">
27. <div id="time"></div>
28. </body>
29. </html>

```



□ screen 객체

➤ 접속한 사용자 모니터의 해상도나 크기에 관한 정보를 가지고 있음

➤Property

Property	Description
availHeight	Windows Taskbar를 제외한 화면의 높이
availWidth	Windows Taskbar를 제외한 화면의 너비
colorDepth	이미지를 출력하기 위한 칼라 팔레트의 bit 깊이
height	화면의 전체 높이
pixelDepth	화면의 칼라 해상도(pixel당 bits)
width	화면의 전체 너비

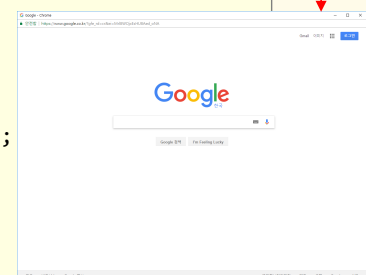
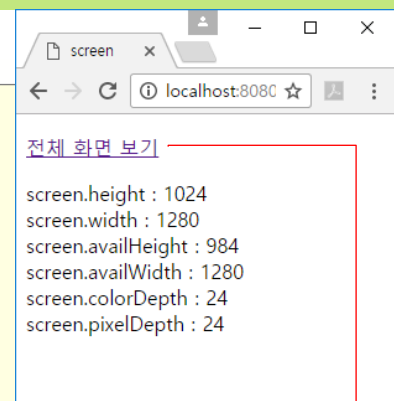
Session 04 Objects - BOM : screen

➤사용 예

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset="UTF-8">
5. <title>screen</title>
6. <script>
7. function maxopen(url, winattributes) {
8.     var maxwindow = window.open(url, "", winattributes);
9.     maxwindow.moveTo(0, 0);
10.    maxwindow.resizeTo(screen.availWidth, screen.availHeight);
11. }
12. </script>
13. </head>
14. <body>
15. <p>
16. <a href="#" onClick="maxopen('http://www.google.com', 'resize=1, scrollbars=1, status=1')">전체 화면 보기</a>
17. </p>
18. <script>
19. document.write("screen.height : ", window.screen.height, "<br>");
20. document.write("screen.width : ", window.screen.width, "<br>");
21. document.write("screen.availHeight : ", screen.availHeight, "<br>");
22. document.write("screen.availWidth : ", screen.availWidth, "<br>");
23. document.write("screen.colorDepth : ", screen.colorDepth, "<br>");
24. document.write("screen.pixelDepth : ", screen.pixelDepth, "<br>");
25. </script>
26. </body>
27. </html>

```



location 객체

➤ 현재 URL에 대한 정보를 가지고 있는 객체

➤ Properties and Methods

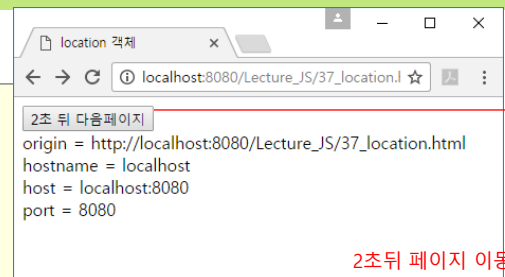
Property	Description
hash	Sets or returns the anchor part (#) of a URL
host	Sets or returns the hostname and port number of a URL
hostname	Sets or returns the hostname of a URL
href	Sets or returns the entire URL
origin	Returns the protocol, hostname and port number of a URL
pathname	Sets or returns the path name of a URL
port	Sets or returns the port number of a URL
protocol	Sets or returns the protocol of a URL
search	Sets or returns the querystring part of a URL
assign()	Loads a new document
reload()	Reloads the current document
replace()	Replaces the current document with a new one

➤ 사용 예

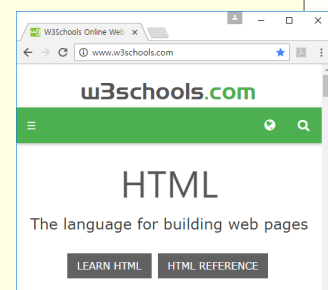
```

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset="UTF-8">
5. <title>location 객체</title>
6. <script>
7. function nextPage() {
8.     location.href = "http://www.w3schools.com";
9. }
10. </script>
11. </head>
12. <body>
13. <input type="button" value="2초 뒤 다음페이지"
14.     onClick="window.setTimeout('nextPage()', 2000)"><br>
15. <script>
16. document.write("href = ", location.href, "<br>");
17. document.write("hostname = ", location.hostname, "<br>");
18. document.write("host = ", location.host, "<br>");
19. document.write("port = ", location.port, "<br>");
20. </script>
21. </body>
22. </html>

```



2초 뒤 페이지 이동



[37_location.html 파일]

□history 객체

- 브라우저가 방문했던 URL 리스트 정보를 저장해 둔 객체
- Property와 Method

Property와 Method	설명
length	히스토리 리스트에 저장된 주소의 개수
back()	히스토리 리스트에서 한 단계 뒤의 주소로 이동
forward()	히스토리 리스트에서 한 단계 앞의 주소로 이동
go(number)	지정된 number 만큼 이동한 단계의 주소로 이동 (number가 음수일 경우 이전 단계 주소로, 양수일 경우 앞 단계 주소로 이동)

➤ 사용 예

```
<body>
<input type="button" value="이전" onclick="history.back()">
<input type="button" value="이전" onclick="history.go(-1)">
<input type="button" value="다음" onclick="history.forward()">
<input type="button" value="다음" onclick="history.go(1)">
</body>
```

□navigator 객체

- 방문 사용자의 브라우저 정보를 담고 있는 객체

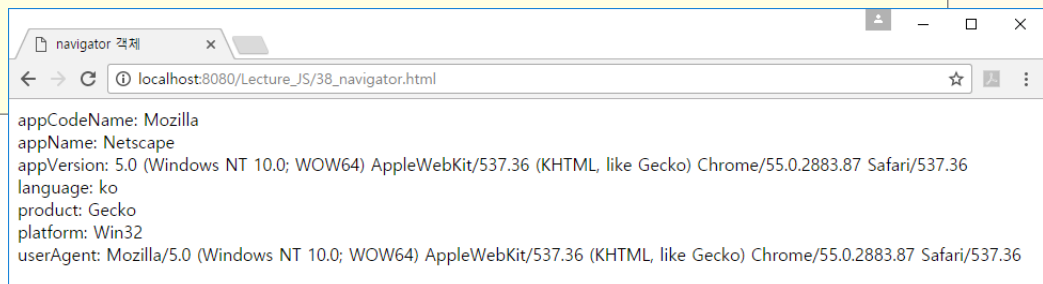
Property / Method	Description
appCodeName	브라우저의 code name 반환
appName	브라우저 이름 반환
appVersion	브라우저 버전 정보 반환
cookieEnabled	cookie가 브라우저에서 사용가능한지 여부(true : 사용가능 함)
geolocation	사용자의 지리적 위치가 저장된 Geolocation 객체 반환
language	브라우저의 사용 언어 반환
onLine	브라우저가 온라인 상태인지의 여부(true : 온라인상태)
platform	브라우저가 컴파일된 플랫폼 반환
product	브라우저의 engine name 반환
userAgent	브라우저에 의해 user-agent header가 서버로 전송되는 값 반환
javaEnabled()	브라우저가 java를 지원하는 지의 여부(true : 자바 지원)

Session 04 Objects - BOM : navigator

▶ 사용 예

[38_navigator.html 파일]

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset="utf-8" />
5. <title> navigator 객체 </title>
6. <script>
7. document.write("appName: "+navigator.appName,"<br>");
8. document.write("appVersion: "+navigator.appVersion,"<br>");
9. document.write("language: "+navigator.language,"<br>");
10. document.write("product: "+navigator.product,"<br>");
11. document.write("platform: "+navigator.platform,"<br>");
12. document.write("userAgent: "+navigator.userAgent,"<br>");
13. </script>
14. </head>
15. <body>
16. </body>
17. </html>
```

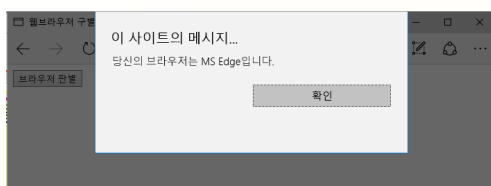


Session 04 Objects - BOM : 브라우저 구별

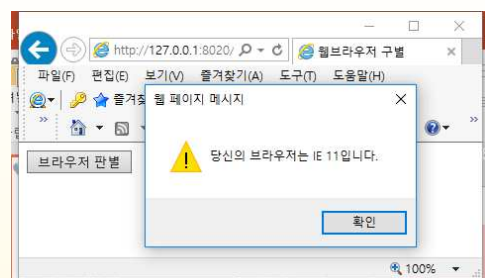
□ 브라우저 구별

▶ **navigator.userAgent**의 문자열 내에 특정한 키워드가 있는지의 여부로 구별

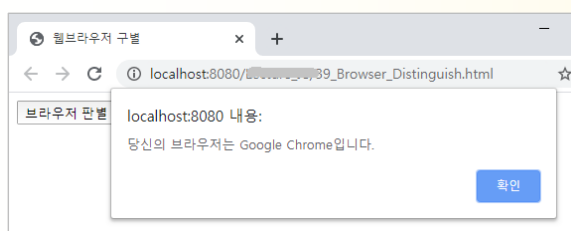
✓ Firefox, MSIE, Trident, edge, Opr, Chrome, Safari



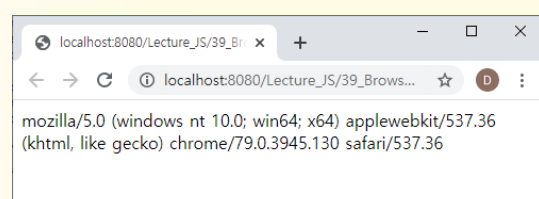
[Microsoft Edge]



[인터넷 익스플로러 v11]



[크롬]



Session 04 Objects - BOM : 브라우저 구별 (예제)

□브라우저 구별 예제

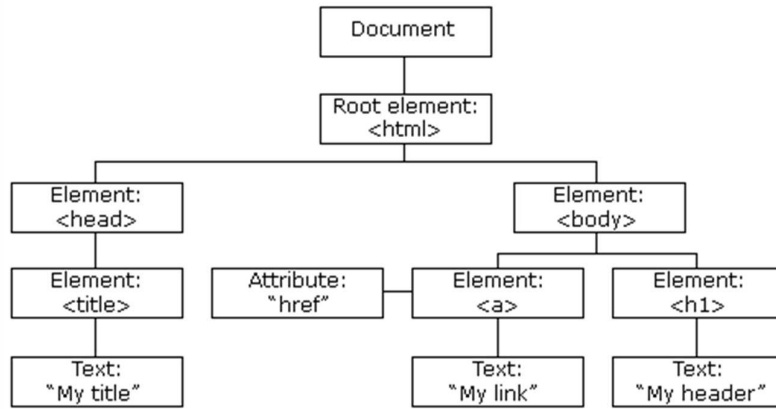
```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset="UTF-8">
5. <title>웹브라우저 구별</title>
6. <script>
7. function showBrowser() {
8.     var agent = navigator.userAgent.toLowerCase();
9.
10.    if( agent.search("msie") != -1 || agent.search("trident") != -1 ||
11.        agent.search("edge") != -1) {
12.        if( agent.search("msie") != -1) {
13.            browser="Internet Explorer 10 이하";
14.        } else {
15.            if( agent.search("trident") != -1)
16.                browser="IE 11";
17.            if( agent.search("edge") != -1)
18.                browser="MS Edge";
19.        }
20.    }
21. }
```

Session 04 Objects - BOM : 브라우저 구별 (예제)

```
20.     } else if( agent.search("safari") != -1) {
21.         if( agent.search("opr") != -1)
22.             browser="Opera";
23.         else if( agent.search("chrome") != -1)
24.             browser = "Google Chrome";
25.         else
26.             browser = "Apple Safari";
27.     } else if( agent.search("firefox") != -1) {
28.         browser = "Firefox";
29.     }
30.     alert("당신의 브라우저는 " + browser + "입니다.");
31.     document.write(agent, "<br>");
32. }
33. </script>
34. </head>
35. <body>
36. <button onClick="showBrowser()">브라우저 판별</button>
37. </body>
38. </html>
```

DOM(Document Object Model)

- 웹 페이지가 로드될 때, 브라우저에 의해 생성된 문서 객체 모델
- HTML DOM 모델은 객체에 대한 Tree로 구성됨
- HTML DOM은 HTML elements와 attributes를 접근하고, 변경시키고, 추가하고, 삭제하기 위한 표준 방법임



Finding HTML Elements

Method	Description
<code>document.getElementById(id)</code>	id 선택자로 지정된 element 객체 반환
<code>document.getElementsByTagName(name)</code>	지정한 tag name에 해당하는 모든 element들을 array로 반환
<code>document.getElementsByClassName(name)</code>	지정한 class name에 해당하는 모든 element들을 array로 반환

Changing HTML Elements

Method	Description
<code>element.innerHTML = new html content</code>	element의 내부 content를 얻거나 설정
<code>element.attribute = new value</code>	element의 attribute 값을 얻거나 설정
<code>element.setAttribute(attribute, value)</code>	element의 attribute와 값을 설정
<code>element.style.property = new style</code>	element의 style property를 설정(CSS의 property)

□DOM을 사용하기 위한 호출 위치

➤기본 개념 : 호출되기 전에 브라우저에 의해 객체가 인식되어야 함

방법1

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
  HTML 태그 작성
  <script>
    calling JavaScript DOM;
  </script>
</body>
</html>
```

방법2

```
<!DOCTYPE html>
<html>
<head>
  <script>
    window.onload = function() {
      calling JavaScript DOM;
    }
  </script>
</head>
<body>
  HTML 태그 작성
</body>
</html>
```

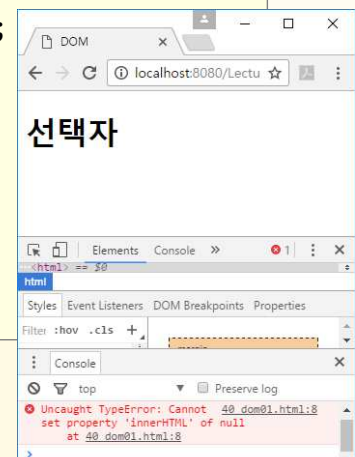
방법3

```
<!DOCTYPE html>
<html>
<head>
  <script>
    function myEvent() {
      calling JavaScript DOM;
    }
  </script>
</head>
<body>
  HTML 태그 작성
  <div onClick="myEvent()">
</body>
</html>
```

□사용 예

[40_dom01.html 파일]

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset="UTF-8">
5. <title>DOM</title>
6. <script>
7. var obj = document.getElementById("mydiv");
8. obj.innerHTML = "<a href='http://www.w3schools.com'>링크</a>";
9.
10. document.getElementById("myh").style.color = "red";
11.
12. </script>
13. </head>
14. <body>
15. <div id="mydiv"></div>
16. <h1 id="myh">선택자</h1>
17. </body>
18. </html>
```



ERROR~!!

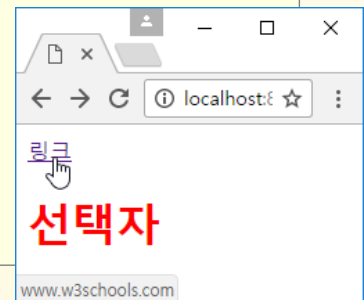
□ 사용 예

[40_dom02.html 파일]

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset="UTF-8">
5. <title>DOM</title>
6. <script>
7. window.onload = function() {
8.     var obj = document.getElementById("mydiv");
9.     obj.innerHTML = "<a href='http://www.w3schools.com'>링크</a>";
10.
11.    document.getElementById("myh").style.color = "red";
12. }
13. </script>
14. </head>
15. <body>
16. <div id="mydiv"></div>
17. <h1 id="myh">선택자</h1>
18. </body>
19. </html>

```



□ Adding and Deleting Elements

Method	Description
document.createElement(<i>element</i>)	Create an HTML element
element.removeChild(<i>element</i>)	Remove an HTML element
element.appendChild(<i>element</i>)	Add an HTML element
element.replaceChild(<i>element</i>)	Replace an HTML element

```

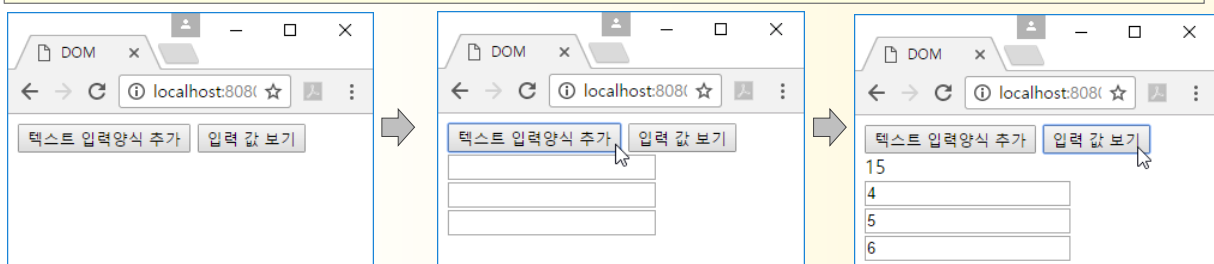
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset="UTF-8">
5. <title>DOM</title>
6. <script>
7. var divGroup;
8. function addText() {
9.     divGroup = document.getElementById("divGroup");
10.    var divObj = document.createElement("div");
11.    var inputObj = document.createElement("input");
12.    inputObj.setAttribute("type", "text");
13.    inputObj.setAttribute("name", "num");
14.    divObj.appendChild(inputObj);
15.
16.    divGroup.appendChild(divObj);
17. }
18.

```

```

19. function showData() {
20.    var arrObj = document.getElementsByName("num");
21.    var str="";
22.    for(var idx=0 ; idx < arrObj.length; ++idx) {
23.        str += parseInt(arrObj[idx].value);
24.    }
25.    document.getElementById("userData").innerHTML = str;
26. }
27. </script>
28. </head>
29. <body>
30. <input type="button" value="텍스트 입력양식 추가" onClick="addText()">
31. <input type="button" value="입력 값 보기" onClick="showData()">
32. <div id="userData"></div>
33. <div id="divGroup"></div>
34. </body>
35. </html>

```



□ Changing CSS

➤ Syntax

```
document.getElementById(id).style.property = new style
```

➤ Style Object Properties

Property	Description
backgroundColor	Sets or returns the background-color of an element
border	Sets or returns borderWidth, borderStyle, and borderColor in one declaration
color	Sets or returns the color of the text
display	Sets or returns an element's display type
font	Sets or returns fontStyle, fontVariant, fontWeight, fontSize, lineHeight, and fontFamily in one declaration
margin	Sets or returns the margins of an element (can have up to four values)
padding	Sets or returns the padding of an element (can have up to four values)
position	Sets or returns the type of positioning method used for an element (static, relative, absolute or fixed)
visibility	Sets or returns whether an element should be visible

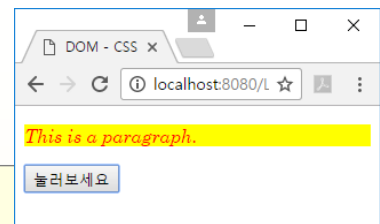
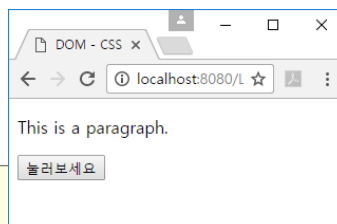
□ 사용 예

[42_dom03.html 파일]

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset="UTF-8">
5. <title>DOM - CSS</title>
6. <script>
7. function changeStyle() {
8.     document.getElementById("p1").style.color = "red";
9.     document.getElementById("p1").style.backgroundColor = "yellow";
10.    document.getElementById("p1").style.fontFamily = "Century Schoolbook";
11.    document.getElementById("p1").style.fontStyle = "italic";
12. }
13. </script>
14. </head>
15. <body>
16. <p id="p1">This is a paragraph.</p>
17. <input type="button" onClick="changeStyle()" value="눌러보세요">
18. </body>
19. </html>

```



Session 04 Objects - DOM : HTML Object

□ HTML DOM Select & Option Object

- <select> element와 <option> element 객체를 나타냄
- Select Object Properties and Method

Property	Description
disabled	Sets or returns whether the drop-down list is disabled, or not
length	Returns the number of <option> elements in a drop-down list
multiple	Sets or returns whether more than one option can be selected from the drop-down list
selectedIndex	Sets or returns the index of the selected option in a drop-down list
size	Sets or returns the value of the size of a drop-down list
value	Sets or returns the value of the selected option in a drop-down list
add()	Adds an option to a drop-down list
remove()	Removes an option from a drop-down list

Session 04 Objects - DOM : HTML Object

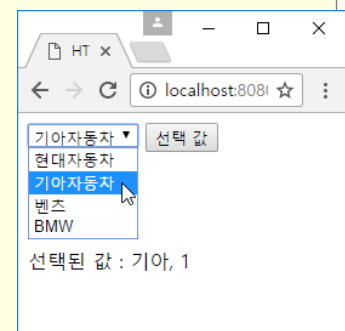
□ 사용 예

[45_select_obj.html 파일]

```

1. <!DOCTYPE html>
2. <html>
3. <head><meta charset="UTF-8">
4. <title>HTML Object</title>
5. <script>
6. function show() {
7.     var obj = document.getElementById("car");
8.     document.getElementById("result").innerHTML = obj.value + ", "
9.                                             + obj.selectedIndex
10. }
11. </script>
12. </head>
13. <body>
14. <form>
15. <select name="car" id="car">
16.     <option value="현대">현대자동차</option>
17.     <option value="기아">기아자동차</option>
18.     <option value="벤츠">벤츠</option>
19.     <option value="BMW">BMW</option>
20. </select>
21. <button type="button" onclick="show()">선택 값</button>
22. <br><br><br><br>
23. <p>선택된 값 : <span id="result"></span></p>
24. </form>
25. </body></html>

```



Session 04 Objects - DOM : HTML Object

➤Option Object Properties and Method

Property	Description
defaultSelected	Returns the default value of the selected attribute
disabled	Sets or returns whether an option is disabled, or not
form	Returns a reference to the form that contains the option
index	Sets or returns the index position of an option in a drop-down list
label	Sets or returns the value of the label attribute of an option in a drop-down list
selected	Sets or returns the selected state of an option
text	Sets or returns the text of an option

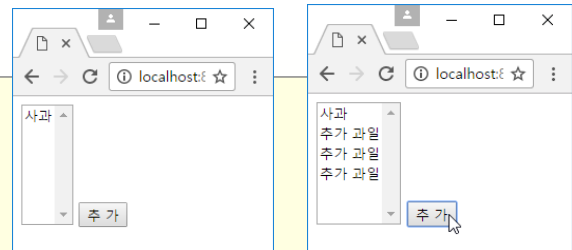
Session 04 Objects - DOM : HTML Object

□ 사용 예

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset="UTF-8">
5. <title>HTML Object</title>
6. <script>
7.   function add() {
8.     var optionObj = document.createElement("option");
9.
10.    optionObj.setAttribute("value", "my fruit");
11.    var t = document.createTextNode("추가 과일");
12.    optionObj.appendChild(t);
13.    document.getElementById("mySelect").appendChild(optionObj);
14.  }
15. </script>
16. </head>
17. <body>
18. <select size="6" id="mySelect">
19.   <option value="apple">사과</option>
20. </select>
21. <button type="button" onclick="add()">추 가</button>
22. </body>
23. </html>

```



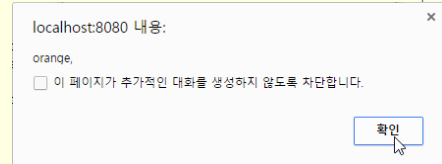
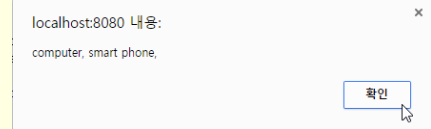
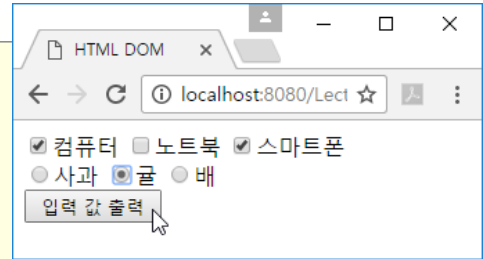
Session 04 Objects - DOM : HTML Object

□ 사용 예

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset="UTF-8">
5. <title>HTML DOM</title>
6. <script>
7.   function show() {
8.     var arr = document.getElementsByName("fav");
9.     var str="";
10.    for(var i in arr) {
11.      if(arr[i].checked)
12.        str += arr[i].value + ", ";
13.    }
14.    alert(str);
15.
16.    str="";
17.    arr = document.getElementsByName("fruit");
18.    for(var i in arr) {
19.      if(arr[i].checked)
20.        str += arr[i].value + ", ";
21.    }
22.    alert(str);
23.  }

```



Session 04 Objects - DOM : HTML Object

```

1.   return false;
2. }
3. </script>
4. </head>
5. <body>
6. <form action="#" onSubmit="return show()">
7.   <div>
8.     <input type="checkbox" name="fav" value="computer">컴퓨터
9.     <input type="checkbox" name="fav" value="notebook">노트북
10.    <input type="checkbox" name="fav" value="smart phone">스마트폰
11.  </div>
12.  <div>
13.    <input type="radio" name="fruit" value="apple">사과
14.    <input type="radio" name="fruit" value="orange">귤
15.    <input type="radio" name="fruit" value="pear">배
16.  </div>
17.  <div><input type="submit" value=" 입력 값 출력 "></div>
18. </form>
19. </body>
20. </html>

```

DOM Node Navigation

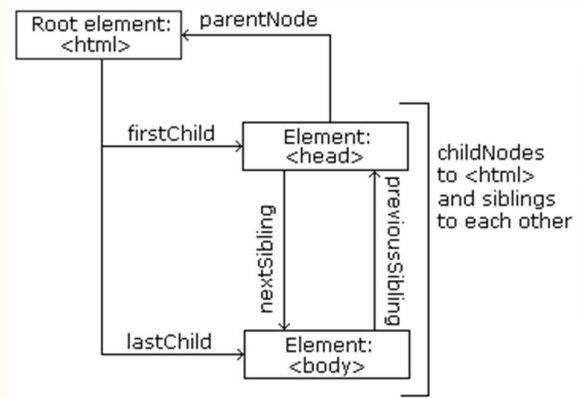
- 전체 문서는 document 노드임
- 모든 HTML element(tag)는 element 노드임
- HTML elements 내의 텍스트는 text 노드임
- 모든 HTML 속성은 attribute 노드임
- 모든 주석은 comment 노드임

```
<html>

<head>
  <title>DOM Tutorial</title>
</head>

<body>
  <h1>DOM Lesson one</h1>
  <p>Hello world!</p>
</body>

</html>
```



Properties

- ✓ text, comment, whitespace, [Enter]를 하나의 node로 간주하는 속성

Property / Method	Description
<i>element.childNodes</i>	element의 자식 노드에 대한 collection 반환
<i>element.firstChild</i>	element의 첫번째 자식 노드 반환
<i>element.lastChild</i>	element의 마지막 자식 노드 반환
<i>element.parentNode</i>	element의 부모 노드 반환
<i>element.nextSibling</i>	element의 형제 노드에서 다음 노드 반환
<i>element.previousSibling</i>	element의 형제 노드에서 이전 노드 반환

- ✓ text, comment, whitespace, [Enter]를 하나의 node로 간주하지 않고, element만을 지시하는 속성

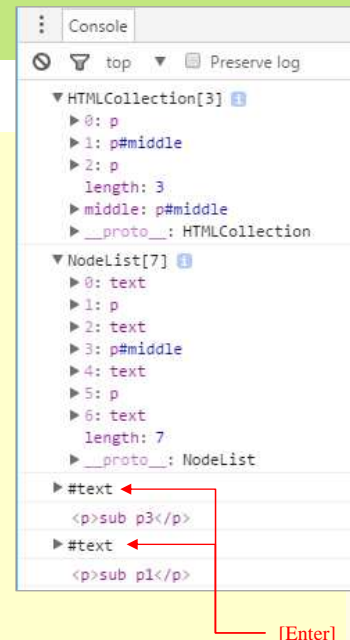
Property / Method	Description
<i>element.children</i>	element의 자식 노드에 대한 collection 반환
<i>element.firstChild</i>	element의 첫번째 자식 element 반환
<i>element.lastChild</i>	element의 마지막 자식 element 반환
<i>element.parentNode</i>	element의 부모 element 노드 반환
<i>element.nextSibling</i>	element의 형제 element에서 다음 element 반환
<i>element.previousElementSibling</i>	element의 형제 element에서 이전 element 반환

□ 사용 예

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <title>DOM 테스트 node와 Element</title>
5. </head>
6. <body>
7.   <div id='myid'>
8.     <p>sub p1</p>
9.     <p id="middle">sub p2</p>
10.    <p>sub p3</p>
11.  </div>
12. <script>
13.  var arr1 = document.getElementById("myid").children;
14.  console.log(arr1);
15.  var arr2 = document.getElementById("myid").childNodes;
16.  console.log(arr2);
17.
18.  console.log(document.getElementById("myid").firstChild); //Enter
19.  console.log(document.getElementById("myid").lastElementChild);
20.  console.log(document.getElementById("middle").previousSibling); //Enter
21.  console.log(document.getElementById("middle").previousElementSibling);
22. </script>
23. </body>
24. </html>

```



□ HTML Object Collections

➤ properties

Property	Description
document.anchors	Returns a collection of all <a> elements in the document that have a name attribute
document.embeds	Returns a collection of all <embed> elements the document
document.forms	Returns a collection of all <form> elements in the document
document.images	Returns a collection of all elements in the document
document.links	Returns a collection of all <a> and <area> elements in the document that have a href attribute
document.scripts	Returns a collection of <script> elements in the document

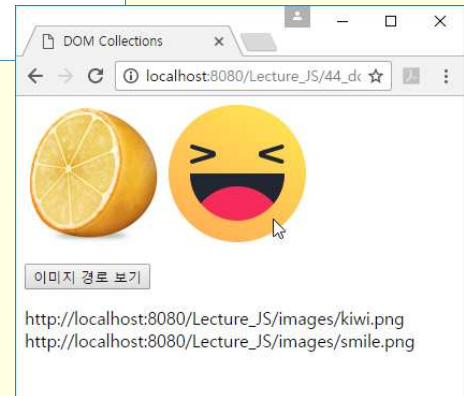
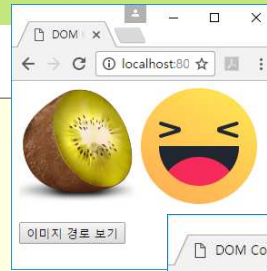
Session 04 Objects - DOM

□ 사용 예

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset="UTF-8">
5. <title>DOM Collections</title>
6. <script>
7. function change() {
8.     document.images[0].src = "images/orange.png";
9. }
10. function showSrc() {
11.     var arrObj = document.images;
12.     var sourceObj = document.getElementById("source");
13.     for(var i=0; i<arrObj.length; i++){
14.         sourceObj.innerHTML += arrObj[i].src + "<br>";
15.     }
16. }
17. </script>
18. </head>
19. <body>
20. 
21. 
22. <p>
23. <button type="button" onclick="showSrc()">이미지 경로 보기</button>
24. </p>
25. <div id="source"></div>
26. </body>
27. </html>

```



Session 04 DOM Events Handlers(1)

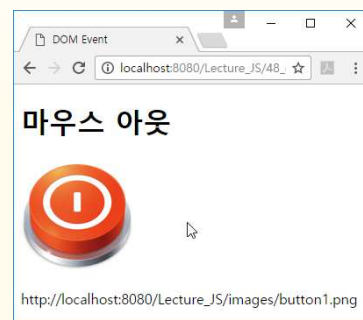
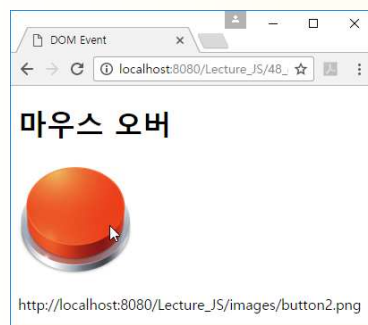
□ DOM Event

➤ Syntax

```

document.getElementById(id).onclick = function(){
    code
}

```



```

1. <!DOCTYPE html>
2. <html>
3. <head><meta charset="UTF-8">
4. <title>DOM Event</title>
5. <script>
6. window.onload = function () {
7.     var imgObj = document.getElementById("myImg");
8.     document.getElementById("myImg").onmouseover = function () {
9.         imgObj.src = "images/button2.png";
10.        document.getElementById("title").innerHTML = "마우스 오버";
11.        document.getElementById("img_src").innerHTML = imgObj.src;
12.    }
13.    document.getElementById("myImg").onmouseout = function () {
14.        imgObj.src = "images/button1.png";
15.        document.getElementById("title").innerHTML = "마우스 아웃";
16.        document.getElementById("img_src").innerHTML = imgObj.src;
17.    }
18. }
19. </script>
20. </head>
21. <body>
22. <h1 id="title">마우스 아웃</h1>
23. 
24. <p id="img_src"></p>
25. </body>
26. </html>

```

Session 04 EventListener(1)

□ EventListener

- HTML 요소 또는 window에 임의로 이벤트를 추가하거나 제거할 경우 사용
- 한 요소에 같은 이벤트를 중복으로 등록할 경우 사용
- syntax

```
element.addEventListener(event, function);
```

- *event* : on을 제거한 이벤트 이름
- *function* : 이벤트가 발생했을 때 호출하고자 하는 함수

방법1)

```
element.addEventListener("click", function(){ alert("Hello World!"); });
```

방법2)

```

function myFunction() {
    alert ("Hello World!");
}

element.addEventListener("click", myFunction);

```

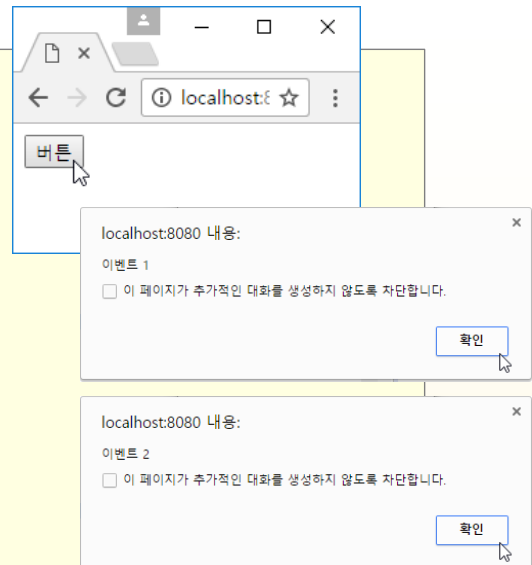
Session 04 EventListener(2)

□ 사용 예

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>event listener</title>
<script>
function fnc() {
    alert("이벤트 1");
}

window.onload = function () {
    var myBtn = document.getElementById("btn");

    myBtn.addEventListener("click", fnc);
    myBtn.addEventListener("click",
        function () {
            alert("이벤트 2");
        }
    );
}
</script>
</head>
<body>
<button id="btn">버튼</button>
</body>
</html>
```



[49_eventListener.html 파일]

Internet Programming

101

Session 04 Event Object(1)

□ Event Object

➤ Syntax

```
document.getElementById(id).onclick = function(e){
    var eventObj = e ? e : window.event; //IE8 이하에서는 window.event 객체 사용
    code;
}
```

➤ Event Object Attributes

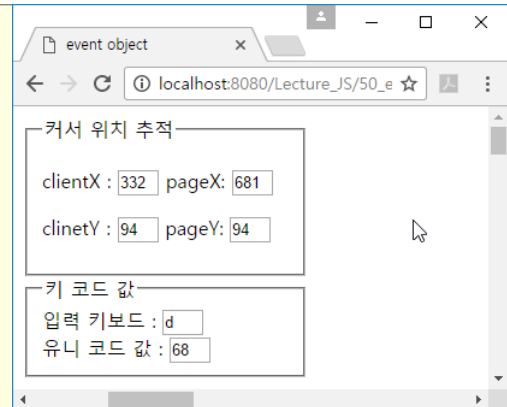
구분	Property	Description
마우스 이벤트	clientX	마우스 이벤트가 발생한 X좌표(수평 스크롤 이동 너비 무시)
	clientY	마우스 이벤트가 발생한 Y좌표(수직 스크롤 이동 너비 무시)
	pageX	마우스 이벤트가 발생한 X좌표(수평 스크롤 이동 너비 포함)
	pageY	마우스 이벤트가 발생한 Y좌표(수직 스크롤 이동 너비 포함)
	screenX	모니터 기준으로 마우스 이벤트가 발생한 X좌표
	screenY	모니터 기준으로 마우스 이벤트가 발생한 Y좌표
키보드 이벤트	keyCode	키보드에 눌린 유니코드 값
	altKey	[Alt] 키가 눌리면 true, 아니면 false
	ctrlKey	[Ctrl] 키가 눌리면 true, 아니면 false
전체 이벤트	target	this와 같이 이벤트가 발생한 요소를 반환

Session 04 Event Object(2)

예제

[50_eventObject.html 파일]

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset="UTF-8">
5. <title>event object</title>
6. <style type="text/css">
7. body {width:2000px; height:2000px}
8. input {width:30px}
9. #myField {position:fixed;width:60%}
10. #key{ime-mode:disabled;}
11. </style>
12. <script>
13. window.onload = function() {
14.     var eventObj;
15.     document.onmousemove = function(e) {
16.         eventObj = e?e:window.event;    //IE8에서는 window.event;
17.         var cx = eventObj.clientX;
18.         var cy = eventObj.clientY;
19.
20.         document.getElementById("client_x").value = cx;
21.         document.getElementById("client_y").value = cy;
22.
23.         var px = eventObj.pageX;
24.         var py = eventObj.pageY;
```

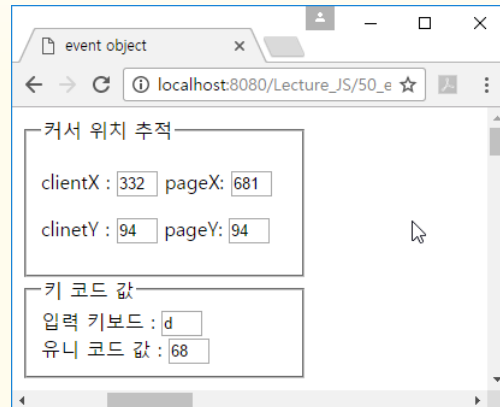


Session 04 Event Object(3)

```
25.     document.getElementById("pg_x").value = px;
26.     document.getElementById("pg_y").value = py;
27. };
28.
29. document.getElementById("key").onkeydown = function(e){
30.     eventObj = e?e:window.event;    //IE8에서는 window.event;
31.     document.getElementById("code").value = eventObj.keyCode;
32. };
33. }
34. </script>
35. </head>
36. <body>
37. <div id="myField">
38. <fieldset>
39.     <legend>커서 위치 추적</legend>
40.     <p>
41.         <label for="client_x">clientX : </label>
42.         <input type="text" name="client_x" id="client_x">
43.         <label for="pg_x">pageX: </label>
44.         <input type="text" name="pg_x" id="pg_x">
45.     </p>
46.     <p>
47.         <label for="client_y">clientY : </label>
48.         <input type="text" name="client_y" id="client_y">
49.         <label for="pg_y">pageY: </label>
50.         <input type="text" name="pg_y" id="pg_y">
51.     </p>
```

Session 04 Event Object(4)

```
52. </fieldset>
53. <fieldset class="myField">
54.   <legend>키 코드 값</legend>
55.   <label for="key">입력 키보드 : </label>
56.   <input type="text" name="key" id="key"><br>
57.   <label for="code">유니 코드 값 : </label>
58.   <input type="text" name="code" id="code">
59. </fieldset>
60. </div>
61. </body>
62. </html>
```



Session 04 JavaScript Exception Handling(1)

□ 실행시간 오류가 발생되었을 때의 예외를 제어

□ Syntax

```
try {
    Block of code to try;
}
catch(err) {
    Block of code to handle errors
}
finally {
    //생략 가능
    Block of code to be executed regardless of the try / catch result
}
```

□ Error Object Properties

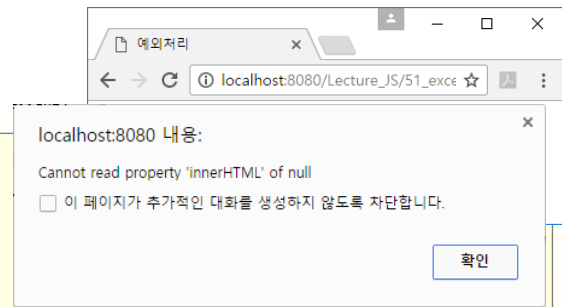
Property	Description
name	Sets or returns an error name
message	Sets or returns an error message (a string)

□ 예제

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset="UTF-8">
5. <title>예외처리</title>
6. <script>
7.   try {
8.     var str = document.getElementById("demo").innerHTML;
9.     alert(str);
10.  }
11.  catch(err) {
12.    alert(err.message);
13.  }
14. </script>
15. </head>
16. <body>
17. <p id="demo">DOM 객체는 문서가 로드된 뒤 사용되어야 함</p>
18. </body>
19. </html>

```



[51_exception01.html 파일]

□ throw

- try~ catch절에서 개발자가 error 내용을 생성하여 프로그램을 제어
- error 자료형은 String, Number, Boolean, Object가 될 수 있음

```

try {
  throw object; // 예외 발생시 object를 throw함
}
catch(err) {      //throw된 데이터를 catch함
  Block of code to handle errors
}

```

```

throw "Too big"; // throw a text
throw 500;       // throw a number

```

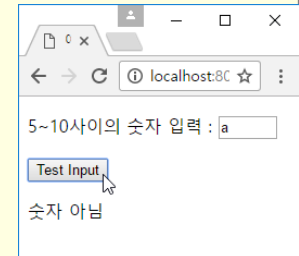
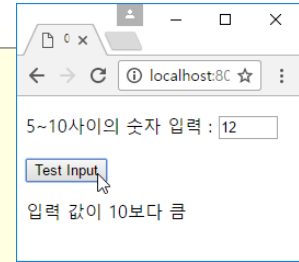
□ 예제

[52_exception02.html 파일]

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset="UTF-8">
5. <title>예외처리</title>
6. <script>
7. function myFunction() {
8.     var message = document.getElementById("message");
9.     message.innerHTML = "";
10.    var x = document.getElementById("demo").value;
11.    try {
12.        if(x == "") throw "empty";
13.        if(isNaN(x)) throw "숫자 아님";
14.        x = Number(x);
15.        if(x < 5) throw "입력 값이 5보다 작음";
16.        if(x > 10) throw "입력 값이 10보다 큼";
17.    } catch(err) {
18.        message.innerHTML = err;
19.    }
20. }
21. </script>
22. </head>
23. <body>
24. <p>5~10사이의 숫자 입력 : <input id="demo" type="text" size="3"></p>
25. <button type="button" onClick="myFunction()">Test Input</button>
26. <p id="message"></p>
27. </body>
28. </html>

```



□ Cookie

- 저장이 필요한 정보가 생길 때 클라이언트 쪽에 일정한 형식의 데이터를 저장하고 필요한 정보가 생길 때마다 서버에 저장된 정보를 전송
- 클라이언트에 300개까지의 Cookie 생성 가능.
- 각 Cookie별로 Name 부여 가능하며, 4KBytes까지 저장가능
- Syntax

```
document.cookie = newCookie
```

➤Cookie 속성

Name=Value; expire=DATE; path=PATH; domain=DOMAIN_NAME; secure

- Name=Value : 쿠키를 구분하기 위함. 반드시 필요
- expire=DATE : 쿠키 유효 날짜 설정 (Mon, DD-MM-YY, HH:MM:SS GMT)
- path=PATH : 쿠키가 유효한 경로 설정
- domain=DOMAIN_NAME : 쿠키가 유효한 도메인 설정
- secure : 연결이 안전할 때만 쿠키 전달(SSL을 사용하는 HTTPS 연결)

□ Cookie 보관 위치

➤ [폴더 옵션]에서 “보호된 운영 체제 파일 숨기기”를 check 해제하고, 숨김 파일, 폴더 및 드라이브 표시를 선택해야 함.

➤ Internet Explorer

✓ Windows 2000 & XP

C:\ Documents and Settings\ <User ID>\ Cookies

✓ Windows Vista & 7

C:\ Users\ <User ID>\ AppData\ Roaming\ Microsoft\ Windows\ Cookies

✓ Windows 8 & 8.1 & 10

C:\ Users\ <User ID>\ AppData\ Local\ Microsoft\ Windows\ INetCookies

➤ Chrome

✓ C:\ Users\ <User ID>\ AppData\ Local\ Google\ Chrome\ User Data\ Default

✓ 쿠키를 읽기 위해서는 SQLite Editor 필요

✓ 크롬 브라우저 주소입력 줄에 “chrome://settings/siteData” 입력하여 확인

□ 사용 예

➤ cookie.js 파일(js 디렉토리에 저장된 것으로 가정)

```

1. function setCookie(cname, cvalue, exDays) {
2.     var d = new Date();
3.     // d.setTime(d.getTime() + (exDays*1000*60*60*24)); //쿠키 유효시간 : 1일
4.     d.setTime(d.getTime() + (exDays*1000*60)); //쿠키 유효시간 1분
5.     var expires = "expires="+d.toUTCString();
6.     document.cookie = cname + "=" + cvalue + ";" + expires + ";path=/";
7. }
8.
9. function getCookie(cname) {
10.     var name = cname + "=";
11.     var ca = document.cookie.split(';');
12.     for(var i = 0; i < ca.length; i++) {
13.         var c = ca[i];
14.         while (c.charAt(0) == ' ') {
15.             c = c.substring(1);
16.         }
17.         if (c.indexOf(name) == 0) { //쿠키name을 찾았다면
18.             return c.substring(name.length, c.length);
19.         }
20.     }
21.     return "";
22. }

```



```

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset="UTF-8">
5. <title>쿠키 사용 예제</title>
6. <style>
7. h2 {text-align: center;}
8. td.left {text-align:left}
9. </style>
10. <script src="js/cookie.js"></script>
11. <script>
12. window.onload = function() {
13.     var user_id = getCookie("user_id");
14.     if(user_id != "") {
15.         document.getElementById("user_id").value = user_id;
16.         document.getElementById("user_check").checked = true;
17.     }
18. };
19.
20. function setUserCookie() {
21.     var idObj = document.getElementById("user_id");
22.     var chkObj = document.getElementById("user_check");
23.

```

```

24.     try {
25.         if(idObj.value.length == 0) throw "idEmpty";
26.
27.         if (chkObj.checked) {
28.             setCookie("user_id", idObj.value, 1);
29.         } else {
30.             setCookie("user_id", "", -1);
31.         }
32.     } catch(err) {
33.         if(err == "idEmpty") {
34.             alert("User ID를 입력하세요!");
35.             chkObj.checked = false;
36.             idObj.focus();
37.             return false;
38.         } //end-of-if
39.     } //end-of-try-catch
40. }
41. </script>
42. </head>
43. <body>
44. <h2> 쿠키(Cookie)로 이용자의 ID 기억하기 </h2><hr>
45. <form action="#" name="fm1" method="post">
46. <table>
47. <tr>
48.     <th>ID</th>
49.     <td class="left"><input type="text" size="10" name="user_id" id="user_id"></td>
50. </tr>

```

Session 04 JavaScript Cookie(6)

```

51. <tr>
52.     <th>Password</th>
53.     <td class="left">
54.         <input type="password" size="10" name="passwd" id="passwd">
55.         <input type="submit" name="ok" value=" 확인 " >
56.     </td>
57. </tr>
58. <tr>
59.     <td colspan="2" style="text-align:center">
60.         <input type="checkbox" id="user_check" onClick="setUserCookie()">User ID 기억
61.     </td>
62. </tr>
63. </table>
64. </form>
65. </body>
66. </html>

```

[최초 화면]

[입력 후 check 화면]

[웹페이지 reflash 시 화면(F5)]

Session 04 Q & A

