

01-01. 선언문

선언문이란 HTML페이지 내에 자바스크립트를 작성하기 위해 “여기서부터는 자바스크립트를 작성할 것이다.”라고 ‘스크립트 개시’를 선언하는 것이다.

만일 선언문을 작성하지 않고 스크립트를 작성하면 브라우저는 자바스크립트 실행 문을 단순히 HTML 태그로 인식하게 되어 문법 오류가 발생할 것이다.

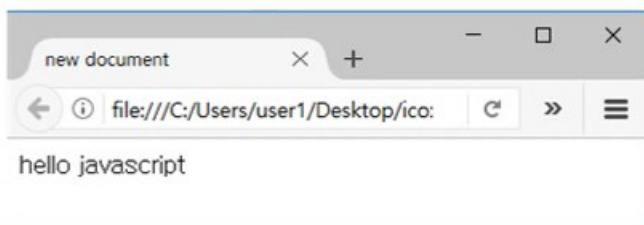
선언문의 문법

```
<script type = "text/javascript" >
    스크립트 실행문;
</script>
```

예제1-1) 간단한 출력

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <title> new document </title>
    <script type="text/javascript">
        //<![CDATA[
            document.write("hello javascript");
        //]]&gt;
    &lt;/script&gt;
&lt;/head&gt;
&lt;body&gt;

&lt;/body&gt;
&lt;/html&gt;</pre>
```



실행문 “document.write();” 부분의 ‘document’를 객체라 부르고, ‘write()’는 메소드(Method)라고 부른다.

한 줄의 실행문을 작성했다면 문장의 마침표에 해당하는 세미콜론(;)으로 마무리를 해 준다.

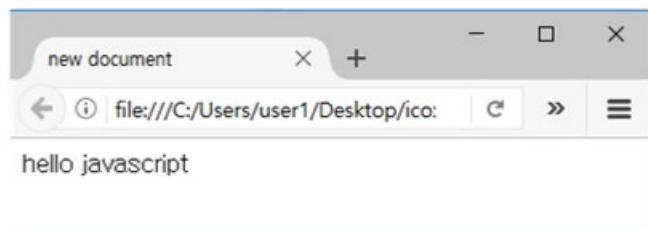
한줄에 하나의 실행문만 내려진다면 오류가 발생하지 않는다.

```
예제 1-1-1) ./js/note.js
```

```
document.write("hello javascript"); // 실행문 내용만 작성
```

```
예제 1-1-2) note.html
```

```
<head>
<script type="text/javascript" src="js/note.js"></script>
// 외부형 js 파일을 소스로 불러온다는 의미의 외부형 선언문
</head>
```



01-02. 변수 및 데이터 형

변수란 “변하는 값(Data)”을 저장하는 공간이라 보면 된다.

단, 변수에는 한 번에 1개의 데이터만 저장할 수 있다.

즉, 새로운 데이터가 입력되면 기존 값은 삭제된다.

```
var 변수명;  
var 변수명1, 변수명2, 변수명3;  
var 변수명 = 데이터 값;  
.....  
var num = 10;           // "num"이라는 변수에 숫자형 데이터 값인 10이 저장되었다  
var num = num /2;      // 변수 "num"에 "num/2"의 결과인 5가 새롭게 저장된다.  
document.write(num);   // 그대로 출력하면 '5'라고 출력된다.
```

1. 변수명 작성 시 주의 사항

① 대 - 소문자 구분

변수명을 소문자로 만들고 출력할 때 대문자로 입력하면 ‘정의되지 않은 변수’로서 오류가 발생한다.

```
var num =50;  
document.write(NUM);  
대 - 소문자 구분으로 인한 오류가 발생한다.
```

② 변수명 맨 앞에는 영문 _, \$가 사용 가능

```
var $num = 10;  
var 3num = 0; //숫자가 맨 앞에 올 수 없다.  
var the% = 100 // %(특수문자)는 포함할 수 없다.
```

예제 1-2)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">  
<head>  
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />  
    <title> new document </title>  
    <script type="text/javascript">  
        var firstNum=100;  
        var nextNum=2;  
        var result=firstNum+nextNum;  
        document.write(result); // 출력값 : 102  
    </script>  
</head>  
  
<body>
```

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

2. 변수에 저장 가능한 데이터의 종류

① 문자(String)형 데이터

다음과 같이 큰 따옴표 또는 작은 따옴표로 둘러싸인 데이터로서, ‘문자’로 인식된다.

```
var txt1 = “자바스크립트” ;  
var txt2 = ‘10’ ;
```

② 숫자(Number)형 데이터

다음과 같이 데이터에 숫자로 정의되는 데이터이다.

```
var num = 10;
```

③ 논리(Boolean)형 데이터

다음과 같이 데이터에 true(참) 또는 false(거짓)로 정의되는 데이터이다.

```
var result = true;
```

④ 널(null)형 데이터

다음과 같이 데이터에 null(비어 있음)으로 정의되는 데이터이다.

```
var result = null;
```

01-3. 연산자(Operator)

1. 산술연산자

피연산자(연산의 대상인 데이터)가 2개 필요하며 ‘이항 연산자’라고도 부른다.

+, -, *, /, %

예제 1-3)

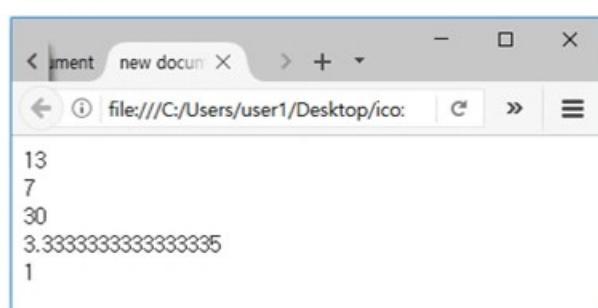
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title> new document </title>
<script type="text/javascript">

    var A=10;
    var B=3;
    var result;
    result=A+B;
    document.write(result+"<br />")
    result=A-B;
    document.write(result+"<br />")
    result=A*B;
    document.write(result+"<br />")
    result=A/B;
    document.write(result+"<br />")
    result=A%B;
    document.write(result+"<br />")

</script>
</head>

<body>

</body>
</html>
```



<유효성 검사 오류를 막는 CDATA 선언>

오류의 원인은, 바로 자바스크립트 내에 들어간
태그이다.

내부형으로 작성된 스크립트에서 html 태그를 포함하게 되면 비록 기능은 정상적으로 작동할지라도 브라우저의 유효성 검사기들은 오류를 지적한다.

결국, 내부형 스크립트에서 thml코드를 포함해야 할 때에는 유효성 검사오류를 막기 위해 다음과 같이 ‘CDATA 선언’을 함께 작성해야 한다.

```
<script type = "text/javascript" >
```

```
//<![CDATA[
```

 실행문 내용들

```
//]]>
```

```
</script>
```

CDATA는 내부형 자바스크립트 내에 HTML태그가 포함되었을 경우, 이들을 모두 문자형 데이터로 인식시켜 이후 문서의 body 부분에 잘 출력되도록 돋는 역할을 한다.

즉, 스크립트가 실행되는 사이에 HTML 태그가 갑자기 끼어드는 현상을 막아주는 것이다.

그러나 앞으로 HTML 내에서 자바스크립트를 작성할 때 태그를 포함해야 한다면 그 실행문 전체를 CDATA선언문으로 감싸주는 습관을 들여야 할 것이다.

2. 비교 연산자

역시 피연산자가 2개 필요하며, 그 2개의 데이터를 비교할 때 사용한다.

비교하여 잘못됐으면 false(거짓)를, 올바르면 true(참)라는 값을 반환한다.

연산자	의미	수식 예	수식의 의미
<	보다 작다	a < b	a가 b보다 작으면 참이다
>	보다 크다	a > b	a가 b보다 크면 참이다
<=	작거나 같다	a <= b	a가 b보다 작거나 같으면 참이다
>=	크거나 같다	a >= b	a가 b보다 크거나 같으면 참이다.
=	같다	a == b	a와 b가 같으면 참이다.
!=	같지 않다	a != b	a가 b와 같지 않으면 참이다.

예제 3-4)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <title> new document </title>
<script type="text/javascript">
//<![CDATA[
    var A=10;
    var B=3;
    var result;
```

```

result=A > B; // A(10)가 B(20)보다 크다 결과 true
document.write(result+"<br />");

result=A <= B; // A(10)가 B(20)보다 작거나 같다 결과 false
document.write(result+"<br />");

result=A == B // A(10)가 B(20)와 같다 결과 false
document.write(result+"<br />");

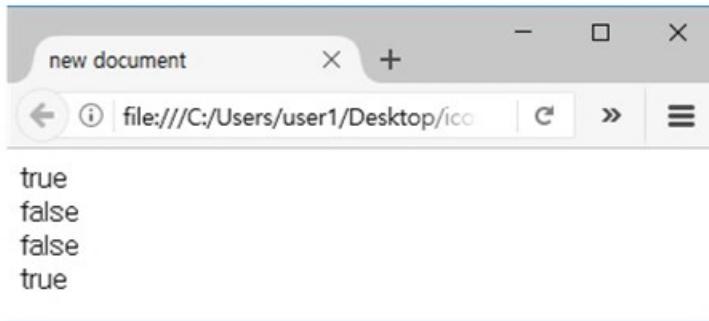
result=A != B; // A(10)가 B(20)와 다르다 결과 true
document.write(result+"<br />");

//]]>
</script>
</head>

<body>

</body>
</html>

```



3. 대입(할당) 연산자

대입 연산자는 연산된 데이터를 지정된 변수에 대입할 때 사용된다.

연산자	단축형	일반
<code>+ =</code>	<code>a += b</code>	<code>a = a + b</code>
<code>- =</code>	<code>a -= b</code>	<code>a = a - b</code>
<code>* =</code>	<code>a *= b</code>	<code>a = a * b</code>
<code>/ =</code>	<code>a /= b</code>	<code>a = a / b</code>
<code>% =</code>	<code>a %= b</code>	<code>a = a % b</code>
<code>& =</code>	<code>a &= b</code>	<code>a = a & b</code>
<code>^ =</code>	<code>a ^= b</code>	<code>a = a ^ b</code>
<code> =</code>	<code>a = b</code>	<code>a = a b</code>
<code><<=</code>	<code>a <<= b</code>	<code>a = a << b</code>
<code>>>=</code>	<code>a >>= b</code>	<code>a = a >> b</code>
<code>>>>=</code>	<code>a >>>= b</code>	<code>a = a >>> b</code>

예제 3-5)

```

!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

```

```

<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title> new document </title>
<script type="text/javascript">
//<![CDATA[
    var num1=10;
    var num2=20;
    num1+=5; //기존에 num1에 저장된 10에 5를 더해서 num1에 저장합니다.
    document.write(num1+"<br />"); //출력 15

    num1-=5; //기존에 num1에 저장된 15에 5를 빼서 num1에 저장합니다.
    document.write(num1+"<br />"); //출력 10

    num1*=5; //기존에 num1에 저장된 10에 5를 곱해서 num1에 저장합니다.
    document.write(num1+"<br />"); //출력 50

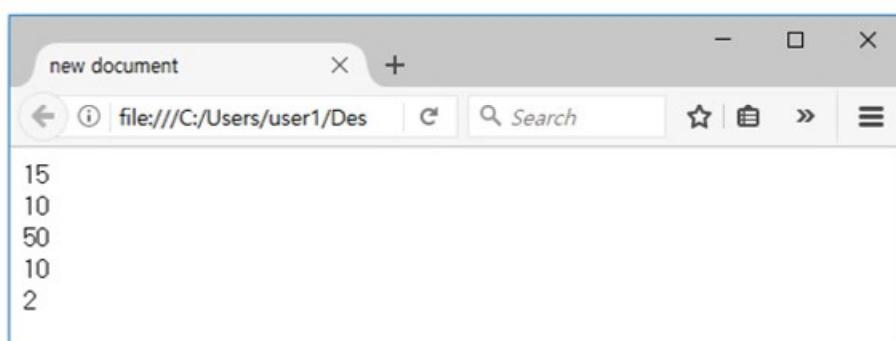
    num1/=5; //기존에 num1에 저장된 50에 5를 나누어 num1에 저장합니다.
    document.write(num1+"<br />"); //출력 10

    num2%=3; //기존에 num2에 저장된 20에 3으로 나눈 나머지값을 num1에 저장합니다.
    document.write(num2+"<br />"); //출력 2
//]]>
</script>
</head>

<body>

</body>
</html>

```



4. 증감 연산자

증가 연산자는 변수에 저장된 값을 1씩 증가한다.

종류	예시	의미
전위형	<code>++a</code>	<code>a = a+1</code> or <code>a += 1</code>
	<code>--a</code>	<code>a= a-1</code> or <code>a -= 1</code>
후위형	<code>a++</code>	<code>a = a+1</code> or <code>a += 1</code>
	<code>a--</code>	<code>a = a-1</code> or <code>a -= 1</code>

연습 3-6)

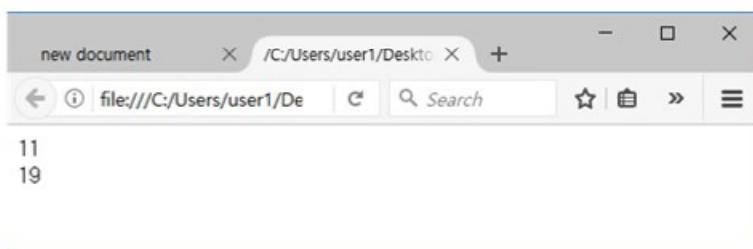
```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<script type="text/javascript">
//<![CDATA[
    var num1=10;
    var num2=20;
    num1++;                      //num1의 10에서 1 더하여 저장
    document.write(num1+"&lt;br /&gt;"); //출력값: 11
    num2--;
    document.write(num2+"&lt;br /&gt;"); //출력값: 19
//]]&gt;
&lt;/script&gt;
&lt;/head&gt;

&lt;body&gt;

&lt;/body&gt;
&lt;/html&gt;
</pre>

```



5. 논리 연산자

피연산자로는 논리데이터인 `true` 또는 `false`가 올수 있다.

연산자로는 이항 연산자인 논리합(`||`), 논리곱(`&&`)이 있으며, 단항 연산자인 논리부정(`!`)이 있다.

연산자	연산자의 기능	결합방향
&&	예) A && B A와 B 모두 true이면 연산결과는 true (논리 AND)	→
	예) A B A와 B 둘 중 하나라도 true이면 연산결과는 true (논리 OR)	→
!	예) !A 연산결과는 A가 true이면 false, A가 false이면 true (논리 NOT)	↔

논리합 연산자는 피연산자에 true가 한 개라도 있으면 true값을 반환하는 반면, 논리곱 연산자는 피연산자 중 false가 한 개라도 있으면 false값을 반환한다.

논리부정은 변수에 true가 저장되어 있으면 false를, false가 저장되어 있으면 true를 결과로 내 놓는다.

문제) 20대 대상의 소개팅을 준비 중이다. 그런데 남성은 다 모집이 되었지만 여성은 아직 모집이 다 되지 않았다. prompt 메서드를 활용하여 성별과 나이를 물어본다. 20대 여성이라면 결과값이 true로, 아니면 결과 값으로 false가 나오게 만든다.

기본형식

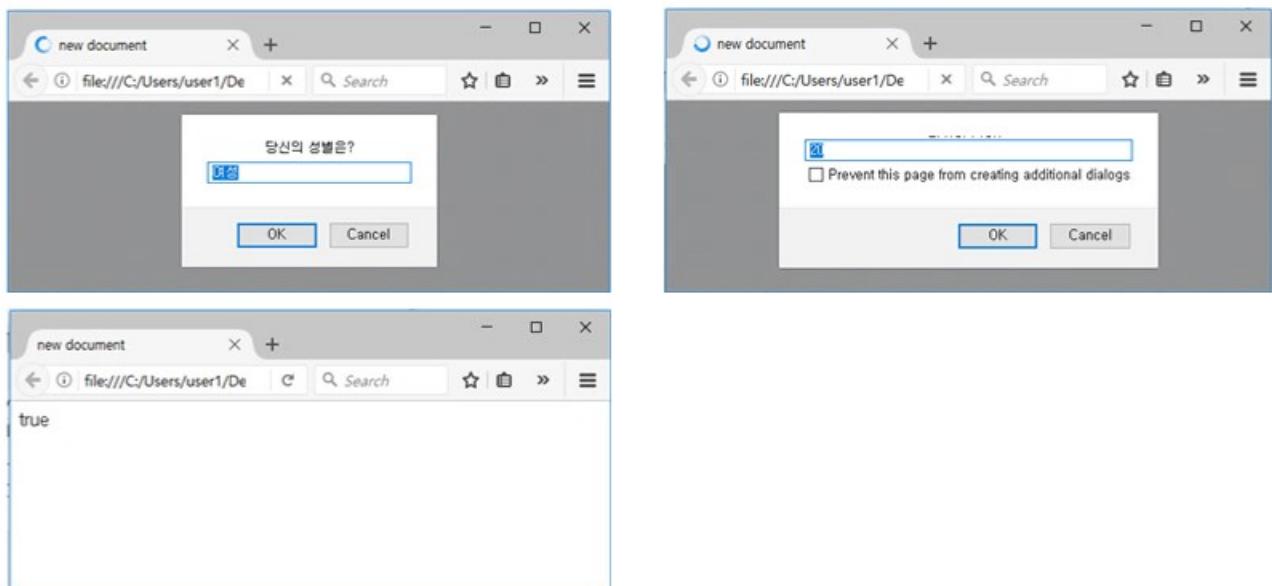
```
prompt("질문내용")처럼 기술하거나 만약 기본적인 답변이 있을 경우라면, prompt("질문내용", "기본답변")
<예> prompt("이름은?", "홍길동")과 같이 기술한다.
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title> new document </title>
<script type="text/javascript">
//<![CDATA[
var gender=prompt("당신의 성별은?","여성");
var age=prompt("당신의 나이는","20");
var result=age&gt;=20 &amp;&amp; age&lt;30 &amp;&amp; gender=="여성";
document.write(result);
//]]&gt;
&lt;/script&gt;
&lt;/head&gt;

&lt;body&gt;

&lt;/body&gt;
&lt;/html&gt;</pre>

```



문제) 국어, 영어, 수학 3과목 평균 및 통과 여부

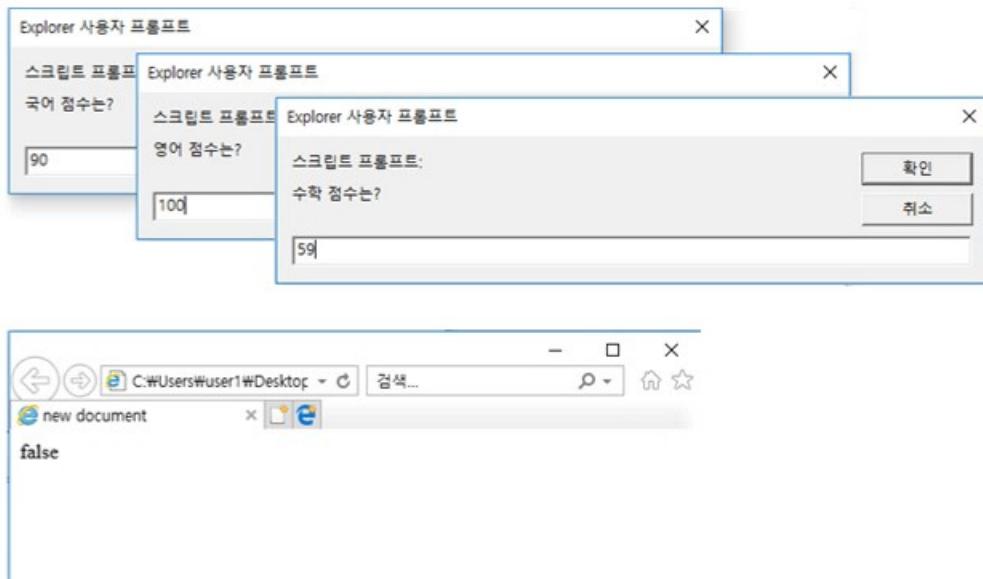
prompt메서드를 활용하여 국어, 영어, 수학 점수를 입력 받으시오. 평균이 70점 이상이고 각 과목별 점n가 60점 이상이면 true를 아니면 false를 출력한다.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title> new document </title>
<script type="text/javascript">
//<![CDATA[
    var korNum=Number(prompt("국어 점수는?", "0"));
    var engNum=Number(prompt("영어 점수는?", "0"));
    var mathNum=Number(prompt("수학 점수는?", "0"));
    var avg=(korNum+engNum+mathNum) / 3;
    var result=avg&gt;=70 &amp;&amp; korNum&gt;=60 &amp;&amp; engNum&gt;=60 &amp;&amp; mathNum&gt;=60;
    document.write(result);
//]]&gt;
&lt;/script&gt;
&lt;/head&gt;

&lt;body&gt;

&lt;/body&gt;
&lt;/html&gt;</pre>

```



6. 문자 결합 연산자

피연산자로는 2개의 데이터가 필요하며, 피연산자(연산대상)에 데이터가 1개 이상 문자형 데이터일 경우에 문자 결합 연산자로 실행한다.

만일 데이터가 피연산자 데이터가 모두 숫자형 데이터라면 산술 연산자로 실행을 하게 된다.

문자 결합 연산자	설명
+	“문자” + “문자”
	“문자” + 숫자
	String(숫자) + 숫자

예제 1-9)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<script type="text/javascript">
//<![CDATA[
    var txt1="아름다운" ;
    var txt2="우리강산" ;
    document.write(txt1+txt2+"&lt;br /&gt;"); //문자결합=문자+문자

    var txt3="은하철도";
    var txt4=999;
    document.write(txt3+txt4+"&lt;br /&gt;"); //문자결합=문자+숫자

    var num1=200;
    var num2=300;</pre>

```

```

        var txt5=String(1000);
        document.write(num1+num2+txt5+"<br />"); // 문자결합=숫자+숫자+문자

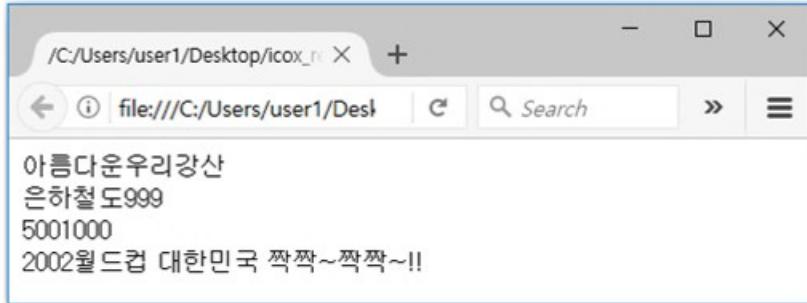
        var addText="2002월드컵 ";
        addText+=" 대한민국"; //기존 문자데이터에 추가 결합 "2002월드컵 대한민국"
        addText+=" 짹쫙~쫙쫙~!!"; //기존 문자데이터에 추가 결합 "2002월드컵 대한민국 짹쫙~쫙
        짹~"
        document.write(addText); //출력: 기존 문자데이터에 추가 결합 "2002월드컵 대한민국"
    //]]>
</script>

</head>

<body>

</body>
</html>

```



7. 조건 연산자

조건 연산자는 조건식의 결과가 true(참) 혹은 false(거짓)일 때 각각 다른 실행문을 수행시킨다.
우선 기본형식을 익힌 후 예제를 통해 정확한 사용법을 익혀보도록 한다.

기본형식

조건식? 실행문1 : 실행문2;
-조건식의 결과값이 참(true)이면 실행문1을, 거짓이면 실행문2를 수행한다.

예제1-10)

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="kor" xml:lang="ko">
<head>
<title> new document </title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<script type="text/javascript">

```

```

//<![CDATA[
    var num1=30;
    var num2=20;
    num1>=num2 ? alert("참(true)일 경우") : alert("거짓(false)일 경우");
    // 조건식이 참(true)이므로 "참(true)일 경우" 경고창을 띄웁니다.
//]]>
</script>
</head>

<body>

</body>
</html>

```

예제 1-11)

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="kor" xml:lang="ko">
<head>
<title> new document </title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<script type="text/javascript">
//<![CDATA[
    var result=confirm("Do you like javascript?");
    var theText=result ? "good" : "No-good";
    document.write(theText);
//]]>
</script>

</head>

<body>

</body>
</html>

```

01-4. 제어문(Control Statement)

1. 조건문

조건문은 조건식을 만족(true)했을 때에만 실행문을 수행하는 제어문이다.

① if문

조건식을 만족할 경우 중괄호({})내의 실행문을 수행한다.

기본형식

```
if(조건식){  
    실행문;  
}
```

예제 1-12)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml" lang="kor" xml:lang="ko">  
<head>  
    <title> new document </title>  
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />  
    <script type="text/javascript">  
        //<![CDATA[<br/>            var myAge=25;  
            if(myAge>=20) { //myAge에 저장된 값이 25이므로 20보다 크므로 조건 만족함.  
                document.write("당신은 성인입니다"); // 출력값: "당신은 성인입니다."  
            }  
        //]]>  
    </script>  
  
</head>  
  
<body>  
  
</body>  
</html>
```

예제 1-13)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml" lang="kor" xml:lang="ko">  
<head>  
    <title> new document </title>  
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
```

```

<script type="text/javascript">
//<![CDATA[
    var testNum=prompt("시험점수를 입력하세요?");
    if(testNum&gt;=70){ // 시험점수가 70점 이상일 경우만 조건을 만족(true)함
        document.write("합격입니다~!!");
    }
//]]&gt;
&lt;/script&gt;

&lt;/head&gt;

&lt;body&gt;

&lt;/body&gt;
&lt;/html&gt;
</pre>

```

② if~else문

조건식을 만족(true)했을 때에는 if문의 중괄호에 든 실행문1을 수행하며, 그렇지 않은(false)경우라면 else문의 중괄호 안에 든 실행문2를 수행한다.

그러므로 출력결과가 2가지일 경우에 사용한다.

기본형식
if(조건식){ 실행문; }else{ 실행문2; }

예제1-14) 나이가 20세 이상이고 30세 미만일 경우에는 “통과”라고 출력한다.

그밖에 만족하지 않으면 “비통과”라고 출력하는 if~else문의 예제이다.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="kor" xml:lang="ko">
<head>
<title> new document </title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<script type="text/javascript">
//<![CDATA[
    var myAge=33;
    if(myAge&gt;=20 &amp;&amp; myAge&lt;30){ //20세이상이고 30세미만 조건을 만족하면 실행
        document.write("통과");
    }else{ //조건을 만족하지 않을때 실행
</pre>

```

```

        document.write("비통과");
    }
//]]>
</script>
</head>
<body>

</body>
</html>

```

예제는 나이(myAge)의 값에 따라 “통과” 또는 “비통과”라는 2가지 결과가 출력되는데, 나이(myAge) 변수에 33이 저장되어 있으므로 조건을 만족하지 못하여 else문이 실행되어 “비통과”가 출력된다.

③ if~else if문

맨 위에서 지정한 조건식1부터 아래로 내려가면서 순서대로 조건식 검사를 수행한다.

만일 조건식 검사를 하다가 조건식2에서 만족하게 되면 그 중괄호({})에 든 실행문2를 수행하여 조건문 검사를 마친다.

하지만 만약 조건식을 모두 만족하지 못하면 마지막 else문 중괄호 내에 기술된 실행문4를 수행한다.

그러므로 이 형식의 조건문은 출력 결과가 3가지 이상일 경우에 사용한다.

기본형식

```

if(조건식){
    실행문;
}else if(조건식2){
    실행문2;
}else if(조건식3){
    실행문3;
}else{
    실행문4;
}

```

예제1-15) 학점(num)이 90점 이상이면 “A학점”, 80점 이상이면 “B학점”, 70점 이상이면 “C학점”, 60점 이상이면 “D학점”, 그밖에 60점 미만이면 “F학점”이 출력되는 성적 프로그램을 만들어보자.

현재 학점을 75로 둔다.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <title> new document </title>
<script type="text/javascript">
//<![CDATA[
</pre>

```

```

var num=75;
if(num>=90){
    document.write("A학점");
}else if(num>=80){
    document.write("B학점");
}else if(num>=70){
    document.write("C학점");
}else if(num>=60){
    document.write("D학점");
}else{
    document.write("F학점");
}
//]]>
</script>
</head>

<body>

</body>
</html>

```

2. switch~case문

switch~case 문은 지정한 변수에 저장된 데이터(data)와 정확히 일치되는 경우(case)가 있는지 검사하여 일치하는 데이터가 있을 때에는 해당 실행문을 수행한 뒤 마지막 break문으로 반복 끝냅니다.
하지만 일치되는 값이 없을 때에는 default에 해당하는 실행문을 수행한 뒤 문장을 마치게 된다.

기본형식

```

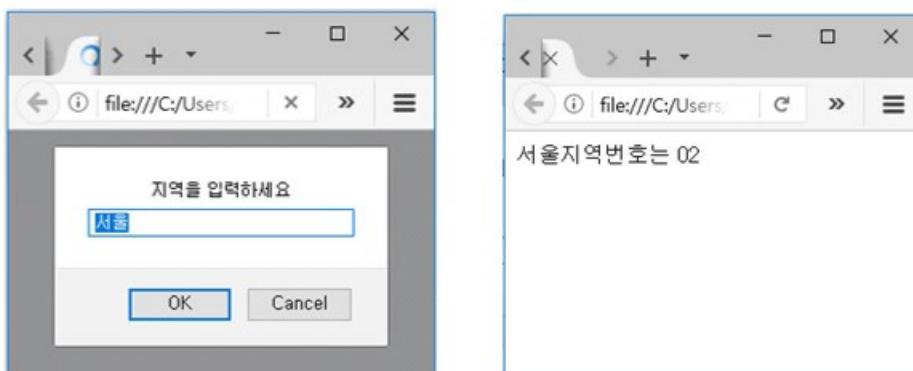
num = 3;
switch(num){
    case 1 :
        실행문 1;
        break;
    case 2 :
        실행문 2;
        break;
    case 3 :
        실행문 3;
        break;
    default :
        실행문 4;
}

```

예제1-16) 응답창(prompt) 기능을 이용하여 지역(서울, 경기, 부산)을 입력한 후에 해당 지역번호를 출력하는 예제이다.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title> new document </title>
<script type="text/javascript">
//<![CDATA[
    var myArea=prompt("지역을 입력하세요", "서울");
    switch(myArea){
        case "서울":           //myArea에 값이 "서울" 인 경우
            areaNum="02";
            break;
        case "경기":           //myArea에 값이 "경기" 인 경우
            areaNum="031";
            break;
        case "부산":           //myArea에 값이 "부산" 인 경우
            areaNum="051";
            break;
        default:               //myArea에 값이 "서울" 인 경우
            alert("등록되지 않은 지역입니다")
    }
    document.write(myArea+"지역 번호는 "+areaNum);
//]]&gt;
&lt;/script&gt;
&lt;/head&gt;

&lt;body&gt;
&lt;/body&gt;
&lt;/html&gt;</pre>
```



3. 반복문

반복문은 해당 조건식을 만족하는 동안 중괄호({}) 안의 실행문을 반복적으로 수행한다.

만약 1부터 10까지 그냥 출력하면 출력문(document.write())을 이용해서 1에서 10까지 총 10번이나 작성해야겠지만, 반복문을 이용하면 이런 번거로움을 줄일 수 있다.

반복문에는 ‘while 문’과 ‘for 문’ 등 2가지 종류가 있다.

① while문

while은 ‘~하는 동안’이라는 뜻의 접속사이다.

즉, 특정 조건식을 만족하는 동안에 지정된 실행문을 반복적으로 수행하게 된다.

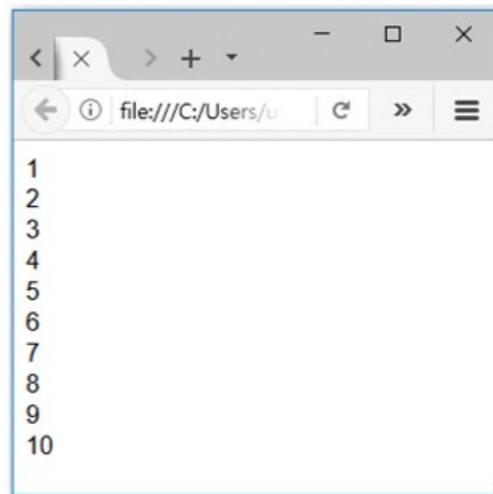
기본형식

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

<예>

```
var i = 1;  
while (i <= 10){  
    document.write(i + "<br />");  
    i++;  
}
```

1. 변수 i에 지정된 초기값은 1이며 while에서 조건식을 검사한다.
 2. ‘i=1’이라는 초기 변수값 지정은 조건식을 만족하므로 중괄호({})의 실행문을 수행하여 1을 출력하고,
로 개행(줄바꿈)이 작동한다.
 3. 그 다음에 기술된 증감식(i++)에 의하여 i값은 증가하여 2가 된다.
 4. 중괄호({})의 마지막 실행문을 마친 후에는 다시 while문 조건식 검사를 반복한다.
- 역시 ‘i=2’가 된 상태로 조건을 만족하므로 실행문을 수행하면 2가 출력되고, 개행이 된다.
5. 그리고 역시 마지막 증감식을 만나면서 i값은 1만큼 증가하여 3이 된다.



예제 1-17) 10부터 1까지 내림차순으로 2의 배수(짝수)만 출력해 보시오.

반복문 조건이 ‘10과 1사이에 있는 2의 배수들’ 이므로 초기값은 10이 된다.

또한 1까지의 ‘내림차순’이 필요하므로 조건식은 ‘1보다 크거나 같다($i \geq 1$)’가 되고 마지막 증감식으로는 1씩 줄어드는 감소 연산자($i--$)가 적당하겠다.

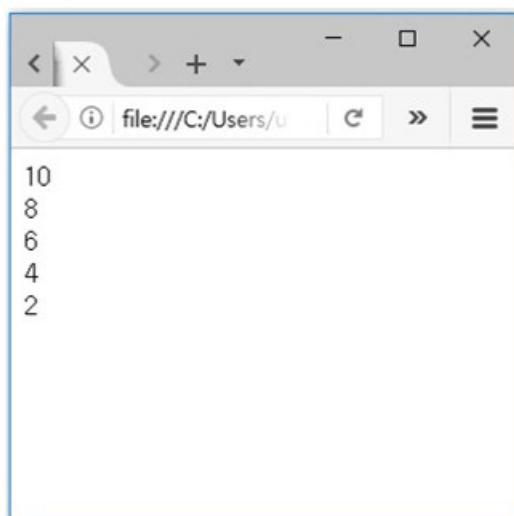
‘2의 배수만 출력하라’는 조건이 있으므로 출력 시 ‘2로 나누었을 때 나머지가 0일 경우($i \% 2 == 0$)’를 if조건문으로 덧붙여주면 된다.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title> new document </title>
<script type="text/javascript">
//<![CDATA[
var i=10;

while(i&gt;=1){
    if(i%2==0) document.write(i+"&lt;br /&gt;");
    i--;
}
//]]&gt;
&lt;/script&gt;
&lt;/head&gt;

&lt;body&gt;

&lt;/body&gt;
&lt;/html&gt;</pre>
```



② for문

기본형식

```
for(var k=초기값; 조건식; 증감식){
```

 실행문

```
}
```

```
for(var k=1;k<=10;k++)
```

```
{
```

 document.write(k+"
");

```
}
```

k의 초기값 1은 10보다 작으므로 조건식 만족

실행문을 수행하여 초기값인 1을 출력

증감식으로 1이 증가한 2를 변수 k에 새로 저장

k값이 여전히 10보다 작으므로 역시 조건식 만족

실행문을 수행하여 현재 저장값인 2를 출력

증감식으로 다시 1이 증가된 3이 k값으로 저장

k값이 증가하다가 11이 되면 조건을 만족하지 않으므로 반복문을 종료하게 된다.

예제1-18) <h*>라는 태그가 작동하게 한 뒤, *에 해당하는 값을 변수로 하여 6에서 1까지 줄여가는 방법일 것이다.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title> new document </title>
<script type="text/javascript">
//<![CDATA[
    for(var k=6; k>=1; k--){
        document.write("<h"+k+">h "+k+" 재미있는 자바스크립트 "+"</h"+k+">");
    }
//]]>
</script>
</head>

<body>

</body>
</html>
```



예제 1-19) 구구단 중 2단을 출력하시오.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title> new document </title>
<script type="text/javascript">
//<![CDATA[
    var result;
    var dan=2;
    document.write("&lt;h1&gt;" + dan + "단&lt;/h1&gt;");
    for(var k=1; k&lt;=9; k++){
        result=k*dan;
        document.write("2X" + k + "=" + result + " &lt;br /&gt;");
    }
//]]&gt;
&lt;/script&gt;
&lt;/head&gt;

&lt;body&gt;

&lt;/body&gt;
&lt;/html&gt;</pre>
```

The screenshot shows a simple web page with a title '2단' (2x table). Below it, the results of the multiplication table for 2 are listed from 2x1=2 up to 2x9=18. The browser interface includes a back button, forward button, address bar, and other standard window controls.

```
2X1=2  
2X2=4  
2X3=6  
2X4=8  
2X5=10  
2X6=12  
2X7=14  
2X8=16  
2X9=18
```

구구단 결과값을 저장할 result 변수를 선언했다. 구구단중 단수를 dan이라는 변수에 저장한다.

단수를 h1태그를 사용하여 출력하였다.

for문을 이용하여 k값이 1부터 9까지 증가하도록 하였다.

단수(dan)에 k(증가변수)값을 곱한 결과값을 result에 다시 저장한다.

출력문을 이용하여 출력하고
태그를 사용해 개방한다.

예제 1-20)prompt 메서드를 활용하여 2단부터 9단까지의 구구단을 순차적으로 출력해 본다.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">  
<head>  
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />  
    <title> new document </title>  
<script type="text/javascript">  
//<![CDATA[<br/>    for(var k=2; k<=9; k++){  
        document.write("<h1>" + k + "단</h1>");  
        for(var m=1; m<=9; m++){  
            document.write(k + "X" + m + "=" + (k*m) + "<br />");  
        }  
    }  
//]]>  
</script>  
</head>  
  
<body>  
  
</body>
```

```
</html>
```

The screenshot shows a Microsoft Word document window titled "new docm". The address bar shows the file path: file:///C:/Users/user1/Desktop/. The document content is as follows:

2단

2×1=2
2×2=4
2×3=6
2×4=8
2×5=10
2×6=12
2×7=14
2×8=16
2×9=18

3단

3×1=3
3×2=6
3×3=9
3×4=12
3×5=15
3×6=18

바깥 for문이 한번 실행할 때 안쪽 for문은 1부터 9까지 9번을 실행한다.

즉, k가 한단을 출력할 때마다 안쪽 for문은 1부터 9까지 곱해주는 역할을 한다.

바깥 for문에서는 제목(*단)이 출력되고 안쪽(for)문에서는 구구단이 출력된다.

한 단 출력이 모두 끝나면 (m이 1에서 9까지 순환하고 나면) 다시 바깥 for문의 증감식으로 가서 k값이 1만큼 증가되고, 그 증가된 k값에 맞는 구구단이 다시 이어진다.

이렇게 반복문의 조건을 만족할 때까지, 계속 실행되면 9단까지 모두 출력된다.

4. break문

반복문의 실행문 들은 수행과정에서 break문을 만나면 해당 반복문을 종료한다.

break문은 반복문을 정지시킬 때 사용한다.

기본 문법

```
<script type = "text/javascript">
    for(var i = 1 : i<=10 ; i++){
        document.write(i+“<br />”);
        if(i==5) break;
    }
</script>
```

5. continue문

반복문이 continue문을 만나면, 더 이상의 실행문을 수행하지 않고 바로 증감식으로 돌아가 실행을 반복하게 된다.

기본 예문

```
<script type =“text/javascript”>
    for(var i = 1 ; i<=10 ; i++){
        if(i%2==1) continue;
        document.write(i+“  
”);
    }
</script>
```

01-5. 객체(javascript에서 있는 API)

* 객체의 생성 - new 연산자

new 연산자는 새로운 객체를 생성할 때 사용하며, 앞에 new를 붙이고 내장 객체의 함수명을 써서 작성한다.

1. Date 객체

기본 형식

```
var todayObj = new Date();
var 변수 = new Date(년,월,일,시,분,초);
```

날짜 정보를 제공해주는 Date객체에 new 연산자를 이용하여 생성한다.

날짜 객체를 사용할 수 있게 변수 선언을 해주면 된다.

① 날짜 정보 제공 메서드

메소드의 종류	설명	기타
getFullYear()	날짜에서 연도에 대한 정보를 제공	
getMonth()	날짜에서 월에 대한 정보를 제공 0~11반환	0=>1월 , 11=>12월
getDate()	날짜에서 일에 대한 정보 제공	
getDay()	날짜에서 요일에 대한 정보 제공	0=>일요일, 6=>토요일
getHours()	시간에서 시(0~23)에 대한 정보를 제공	
getMinutes()	시간에서 분(0~59)에 대한 정보를 제공	
getSeconds()	시간에서 초(0~59)에 대한 정보를 제공	
getMilliseconds()	시간에서 밀리(1/1000)초에 대한 정보 제공	
getTime()	1970년1월1일 이후 경과 시간을 밀리초로 제공	

②날짜 지정 메소드

메소드 종류	설명	기타
setFullYear()	날짜에서 '연도'만 새로 지정	
setMonth()	날짜에서 '월'만 새로 지정	
setDate()	날짜에서 '일'만 새로 지정	
setHours()	'시간'만 새로 지정	
setMinutes()	'분'만 새로 지정	
setSeconds()	'초'만 새로 지정	
setMilliseconds()	'밀리초'만 새로 지정	
setTime()	현재까지 경과 된 밀리초를 새로 지정할	
toGMTString()	날짜를 그리니치 천문대 표준시로 표시	
toString()	날짜를 문자형식으로 표시	

예제 1-21)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<title> new document </title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<script type="text/javascript">
```

```

//<![CDATA[
document.write("<h1>현재날짜/시간 정보</h1>")
var today=new Date();
var nowMonth=today.getMonth()+1;      //현재 월  (0~11)
var nowDate=today.getDate();           //현재 일
var nowDay=today.getDay();           //현재 요일  (0:일요일~6:토요일)
var nowHours=today.getHours();        //현재 시간
var nowMinutes=today.getMinutes();    //현재 분
var nowSeconds=today.getSeconds();   //현재 초
var nowTime=today.getTime();         //1970년1월1일부터 밀리초 경과
된 시간

document.write("월: "+nowMonth+"<br />");
document.write("일: "+nowDate+"<br />");
document.write("요일: "+nowDay+"<br />");
document.write("시: "+nowHours+"<br />");
document.write("분: "+nowMinutes+"<br />");
document.write("초: "+nowSeconds+"<br />");
document.write("경과시간: "+nowTime+"<br />");

document.write("<h1>날짜 바꾸기</h1>")
today.setMonth(11); //월을 12월로 지정
today.setDate(25); //일을 25일로 지정

document.write("월: "+nowMonth+"<br />");
document.write("일: "+nowDate+"<br />");
//]]>
</script>

</head>

<body>

</body>
</html>

```



예제 1-22) 요일별 이미지 출력

이미지를 이용하여 요일별로 출력되도록 만들어 보자.

우선 요일에 따라 변경되어야 하므로 다음과 같이 `getDay()` 메서드를 이용한다.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <title> new document </title>
<script type="text/javascript">
//<![CDATA[
    var today=new Date();
    var nowDay=today.getDay();
    if(nowDay==1){
        document.write('&lt;img src="images/img1.jpg" alt="월요일" /&gt;');
    }else if(nowDay==2){
        document.write('&lt;img src="images/img2.jpg" alt="화요일" /&gt;');
    }else if(nowDay==3){
        document.write('&lt;img src="images/img3.jpg" alt="수요일" /&gt;');
    }else if(nowDay==4){
        document.write('&lt;img src="images/img4.jpg" alt="목요일" /&gt;');
    }else if(nowDay==5){
        document.write('&lt;img src="images/img5.jpg" alt="금요일" /&gt;');
    }else if(nowDay==6){
        document.write('&lt;img src="images/img6.jpg" alt="토요일" /&gt;');
    }else{
</pre>

```

```

        document.write('');
    }
//]]>
</script>
</head>

<body>

</body>
</html>

```

2. Math 객체

단순하게 수를 더하거나 뺄경우에는 ‘산술연산자’들을 사용한다.

하지만 소수점을 반올림하거나 최대/최소값등 특정 공식을 사용해야 할 경우라면, 수학 객체의 메서드를 사용하는 것이 훨씬 편하다.

수학 객체는 다른 내장 객체와 달리 new연산자를 사용하지 않는 정적인 객체이다.

메서드 종류	설명	기타
max(숫자1,숫자2,숫자3)	숫자 중 최대값을 반환	
min(숫자1,숫자2,숫자3)	숫자 중 최소값을 반환	
round(숫자)	소수점 첫째 자리를 반올림	
ceil(숫자)	소수점 자리를 무조건 올림	
floor(숫자)	소수점 자리를 무조건 절삭	
abs(숫자)	수의 절대값을 반환	
random()	0과 1사이의 난수를 발생	

예제 1-23)

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="kor" xml:lang="ko">
<head>
<title> new document </title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<script type="text/javascript">
//<![CDATA[
var maxNum=Math.max(30,70,5);
document.write("최대값: "+maxNum+"<br />");

var minNum=Math.min(8,10,15);
document.write("최소값: "+minNum+"<br />");

var roundNum=Math.round(3.5);
document.write("반올림: "+roundNum+"<br />");

```

```

        var ceilNum=Math.ceil(5.1);
        document.write("소수점올림: "+ceilNum+"<br />");

        var floorNum=Math.floor(10.5);
        document.write("소수점내림: "+floorNum+"<br />");

        var absNum=Math.abs(-10);
        document.write("절대값: "+absNum+"<br />");

        var ranNum=Math.random();
        document.write("0~1 난수: "+ranNum+"<br />");

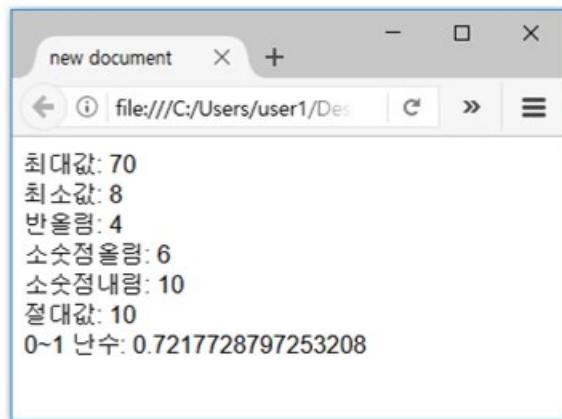
        //]]>
    </script>

</head>

<body>

</body>
</html>

```



예제1-23) 이미지들이 접속할 때마다 랜덤으로 출력되도록 만드시오.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="kor" xml:lang="ko">
<head>
    <title> new document </title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
<script type="text/javascript">
//<![CDATA[

```

```

var imgNum=Math.floor(Math.random()*3)+1;
document.write("<img src='images/img"+imgNum+".jpg' alt='랜덤이미지' />");
//]]>
</script>

</head>

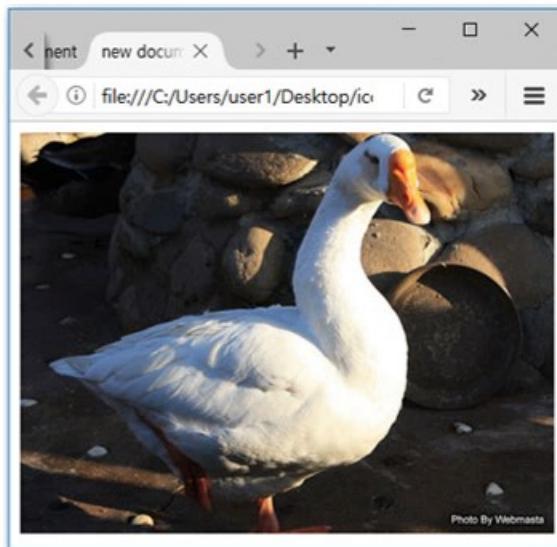
<body>

</body>
</html>

```

난수 값 공식을 이용하여 1부터 3까지의 정수가 무작위로 추출되도록 하였다.

랜덤으로 구한 숫자를 이용해 src경로의 파일명에 숫자 대신 난수가 오도록 변수로 처리한 것이다.



3. String 객체

문자형 데이터를 변수에 저장하여 사용한다. 그 문자형 변수도 또한 String 객체인 것이다.

문자형 변수에는 String 객체가 사용할 수 있는 메서드나 속성들을 동일하게 사용할 수 있다.

① 보편적인 생성법

기본 형식

```
var 변수(인스턴스 변수) = new String( "자바스크립트" );
```

② 간편한 생성법

기본 형식

```
var 변수(인스턴스 변수) = "자바스크립트" ;
```

메서드 종류	설명	결과
bold()	텍스트를 진하게 표현	
link("URL")	텍스트에 링크를 걸어 줌	
length	텍스트의 개수 반환	
toLowerCase()	영문자를 소문자로	
toUpperCase()	영문자를 대문자로	
indexOf("i")	처음부터 "i"를 찾아 최초로 일치하는 인덱스 번호를 반환, 없으면 -1	
lastIndexOf("i"):	끝에서부터 "i"를 찾아 최초로 일치하는 인덱스 번호를 반환, 없으면 -1	
charAt(5)	5번 인덱스에 저장된 문자 데이터를 반환	
substring(4,6)	4번부터 6번 인덱스까지의 문자데이터를 반환	
substr(10,3)	10번 인덱스부터 3글자를 반환	
replase("web","bye")	텍스트 중 "web"을 "bye"로 치환(교체)해서 반환	
slice(7,9)	7번부터 9번 안덱스까지 일부만 잘라서 반환	
concat("good")	끝에 "good"이라는 문자를 결합하여 반환	
split(" ")	" "(공백)을 기준으로 데이터를 분리	

예제 1-25)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<script type="text/javascript">
//<![CDATA[
    var theText="web is hi ciy";
    document.write(theText.bold()+"&lt;br /&gt;");
    document.write(theText.link("http://www.naver.com")+"&lt;br /&gt;");
    document.write(theText.length+"&lt;br /&gt;");
    document.write(theText.toLowerCase()+"&lt;br /&gt;");
    document.write(theText.toUpperCase()+"&lt;br /&gt;");
    document.write(theText.indexOf("i")+"&lt;br /&gt;");
    document.write(theText.lastIndexOf("i")+"&lt;br /&gt;");
    document.write(theText.charAt(5)+"&lt;br /&gt;");
    document.write(theText.substring(4,6)+"&lt;br /&gt;");
    document.write(theText.substr(10,3)+"&lt;br /&gt;");
    document.write(theText.replace("web","bye")+"&lt;br /&gt;");
    document.write(theText.slice(7,9)+"&lt;br /&gt;");
    document.write(theText.concat(" good")+"&lt;br /&gt;");
    document.write(theText.split(" ")+"&lt;br /&gt;");

//]]&gt;
&lt;/script&gt;
&lt;title&gt; new document &lt;/title&gt;</pre>

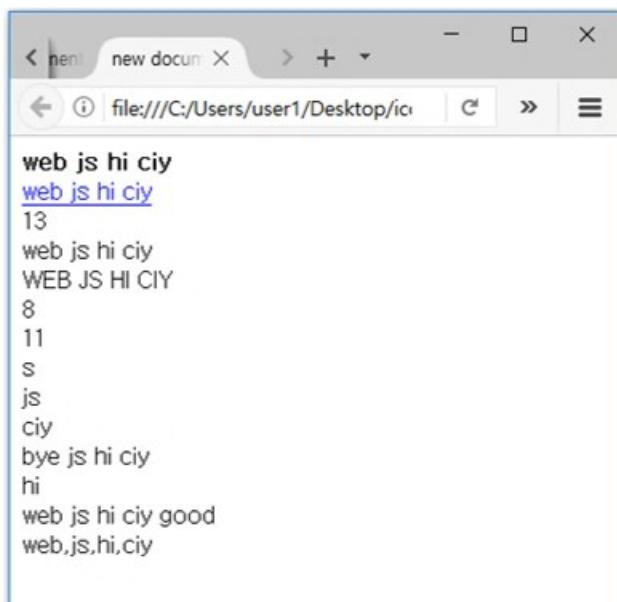
```

```
</head>
```

```
<body>
```

```
</body>
```

```
</html>
```



예제 1-26)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title> 웹표준퍼블리셔 </title>
<style type="text/css">
body,p,h1,h2,h3,h4,h5,h5,u1,ol,li,dl,dt,table,caption,form,fieldset,legend
{margin:0;padding:0;}
li{list-style-type:none;}
img,fieldset{border:0;}
.hdd{font-size:0;width:0;height:0;line-height:0;
position:absolute;left:-9999px;}

</style>

<script type="text/javascript">
<!--
document.write("<h3>문자치환</h3>");
```

```

var theText1="img2_out.jpg";
theText1=theText1.replace("out.jpg", "over.jpg");
document.write(theText1+"<br />");

document.write("<h3>문자추출</h3>");
var theText2="images/img10.jpg";
theText2=theText2.split("/")[1];
theText2=theText2.substring(3);
theText2=parseInt(theText2);
document.write(theText2+"<br />");

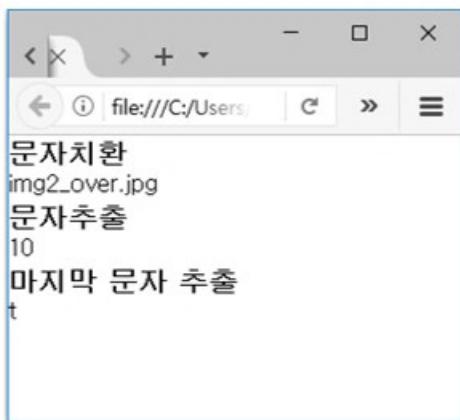
document.write("<h3>마지막 문자 추출</h3>");
var theText3="hello javascript";
var lastIndex=theText3.length-1;
theText3=theText3.charAt(lastIndex);
document.write(theText3+"<br />");

//-->
</script>
</head>

<body>

</body>
</html>

```



4. Array 객체

배열(“Array”)을 이용하면 하나의 변수에 여러개의 데이터를 저장할 수 있다.

하나의 변수에 여러개의 공간으로 이루어진 “배열 객체”를 대입하여 저장하므로, 각 공간마다의 인덱스 (Index)번호가 존재하며 그 인덱스 번호를 사용해 각각의 공간에 저장된 여러 데이터들을 불러올 수 있는 것이다.

① new 연산자를 사용한 배열 객체 생성법

기본형식

```
var arr = new Array();
arr[0] = "값1"
arr[1] = "값2"
arr[2] = "값3"
arr[3] = "값4"
arr[4] = "값5"
```

② 저장 공간을 지정한 배열 객체 생성법

기본형식

```
var arr = new Array(5);
```

배열의 크기가 5인 배열을 만든 것이고 데이터를 5개만 저장할 수 있다.

③ 편리한 배열 객체 생성법

기본형식

```
var arr = [값1,값2,값3,값4,값5];
```

new 연산자를 사용하지 않고 변수에 여러개의 값을 저장하는 방법들이 있다.

예제 1-27)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>웹표준퍼블리셔 </title>
<style type="text/css">
body,p,h1,h2,h3,h4,h5,ul,ol,li,dl,dt,dd,table,caption,form,fieldset,legend
{margin:0;padding:0;}
li{list-style-type:none;}
img,fieldset{border:0;}
.hdd{font-size:0;width:0;height:0;line-height:0;
position:absolute;left:-9999px;}</style>
<script type="text/javascript">
//<![CDATA[
var name=new Array();
name[0]="김부자";
name[1]="정현철";
name[2]="정주영";
name[3]="김민경";</pre>
```

```

name[4]="장영주";
document.write(name[3]+<br />); //3번 인덱스에 저장된 데이터를 출력

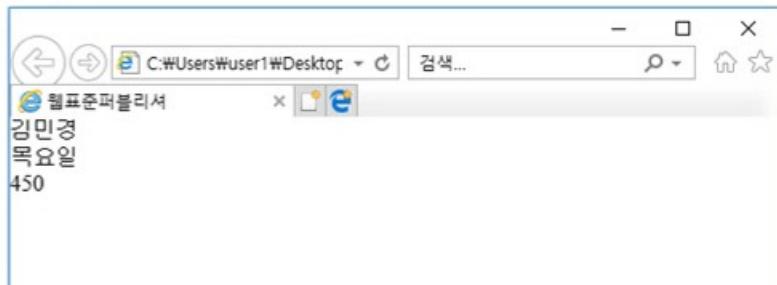
var days=new Array("일","월","화","수","목","금","토");
var today=new Date();
var yoil=days[today.getDay()]; //오늘 요일에 해당하는 값을 반환 일:0 ~ 토:6
document.write(yoil+"요일 <br />"); //오늘의 요일 반환

var money=[100,100,200,50];
var theLen=money.length;
var total=0;
for(var i=0; i<theLen; i++){
    total=total+money[i];
}
document.write(total);
//]]>
</script>
</head>

<body>

</body>
</html>

```



④ 배열 메서드

메서드 종류	설명
reverse()	배열 데이터를 역순으로 반환
slice(번호1, 번호2)	인덱스 번호 1부터 시작하여 2직전의 데이터까지 잘라서 저장
sort()	배열 데이터를 알파벳 혹은 0/1/2..의 오름차순으로 정렬
join("구분자")	배열 데이터를 구분자로 연결하여 결합
concat()	여러개의 배열 데이터들을 하나로 결합

⑤ 배열 데이터 추가 / 삭제 메서드

메서드 종류	설명
shift()	배열에서 첫 인덱스()에 저장된 데이터를 삭제

<code>pop()</code>	배열에서 마지막 인덱스에 저장된 데이터를 삭제
<code>unshift()</code>	배열에서 첫 인덱스(0)에 새 데이터를 추가
<code>push(값)</code>	배열에서 마지막 인덱스에 새 데이터를 중첩하여 추가
<code>concat()</code>	여러개의 배열 데이터들을 하나로 결합

예제 1-28)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title> new document </title>
<script type="text/javascript">
//<![CDATA[
    //1. reverse()-배열 데이터 역순 정렬
    var theFood=["짜장면", "냉면", "갈비탕", "돈까스", "삼겹살"];
    theFood.reverse();
    console.log(theFood)

    //2. sort()-배열 데이터 이름 또는 숫자 정렬
    var theSubway=["2호선", "9호선", "7호선", "4호선", "3호선"];
    theSubway.sort();
    console.log(theSubway);

    //3. concat()-배열들에 데이터를 결합
    var Mountain1=["남산", "도봉산", "관악산", "청계산", "북한산"];
    var Mountain2=["설악산", "지리산", "한라산", "속리산"];
    var joinResult=Mountain1.concat(Mountain2);
    console.log(joinResult);

    //4. slice(시작인덱스, 마지막인덱스)-원하는 데이터만 선택
    var theName=["최준호", "이성은", "기오정", "김민경", "정예선"];
    var result=theName.slice(2,4);
    console.log(result);

    //5. join("구분자")-구분자 기준으로 데이터를 합칩니다.
    var theProduct=["냉장고", "컴퓨터", "선풍기", "밥솥", "핸드폰"];
    var joinResult=theProduct.join("-");
    console.log(theProduct);
//]]>
</script>
</head>
```

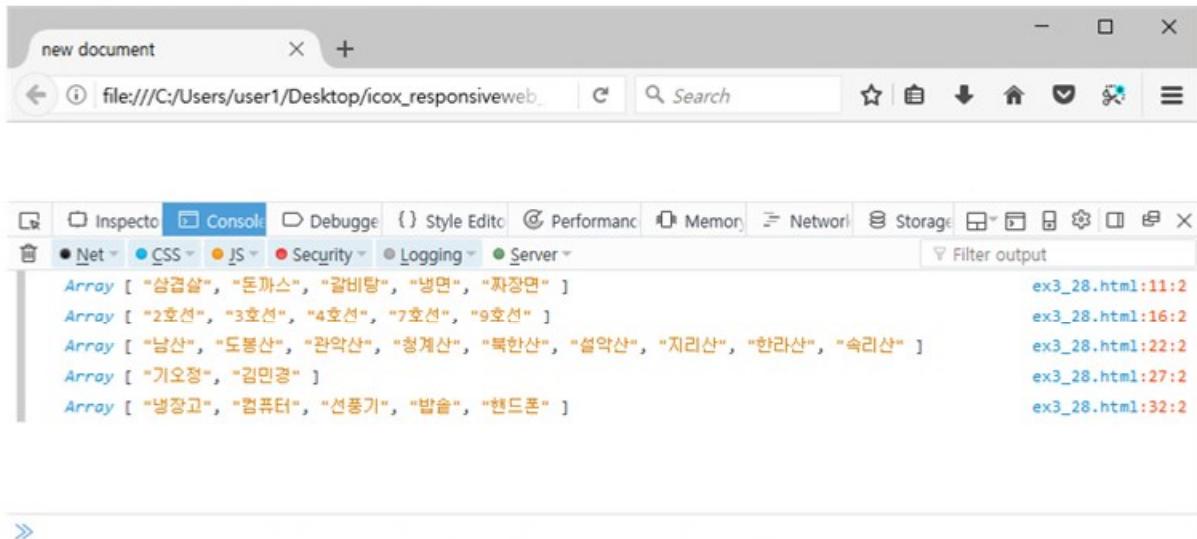
```

<body>

</body>
</html>

```

F12를 누르면 개발자 모드가 생긴다



» reverse()를 이용해서 배열에 저장된 데이터를 역순으로 바꾼다.
 sort()를 이용해서 배열에 저장된 데이터를 오름차순으로 정렬한다.
 concat()를 이용하여 Mountain 과 Mountain2배열을 하나로 합쳐 새로운 배열로 생성한 후 joinResult에 저장한다.
 slice()를 이용하여 배열에 저장된 데이터 중 2번 인덱스부터 4번 인덱스 직전, 즉 3번 인덱스까지 잘라서 새로운 배열로 생성한 후 result에 저장한다.
 join()을 이용하여 배열에 저장된 데이터를 구분자(-)로 연결하여 하나의 데이터로 새로 생성 후 joinResult에 저장한다.

예제 1-29)

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
  <meta http-equiv="content-type" content="text/html; charset=utf-8" />
  <script type="text/javascript">
//<![CDATA[
  var theSubway=["2호선","9호선","7호선","4호선","3호선"];
  var newLine1="1호선";
  var newLine2="분당선";

  theSubway.shift();
</pre>

```

```
console.log(theSubway)
theSubway.unshift(newLine1);
console.log(theSubway);

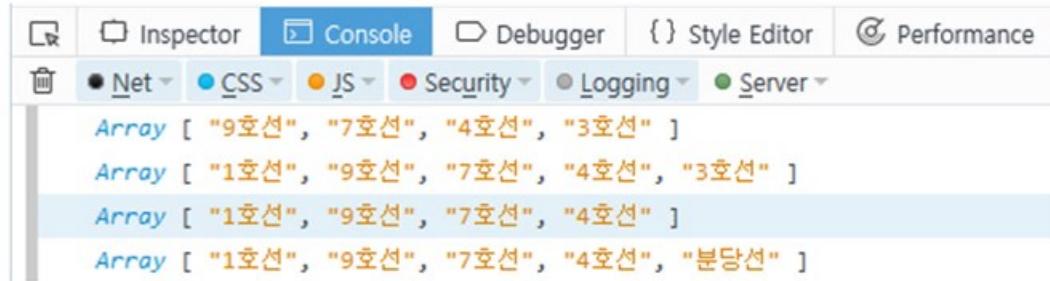
theSubway.pop();
console.log(theSubway)
theSubway.push(newLine2);
console.log(theSubway)

//]]>
</script>
<title> new document </title>

</head>

<body>

</body>
</html>
```



01-6. 함수(Function)

함수를 사용하게 되면 일련의 실행문들을 메모리에 저장했다가 필요할 때 해당 함수의 이름으로 다시 호출하여 사용할 수 있다.

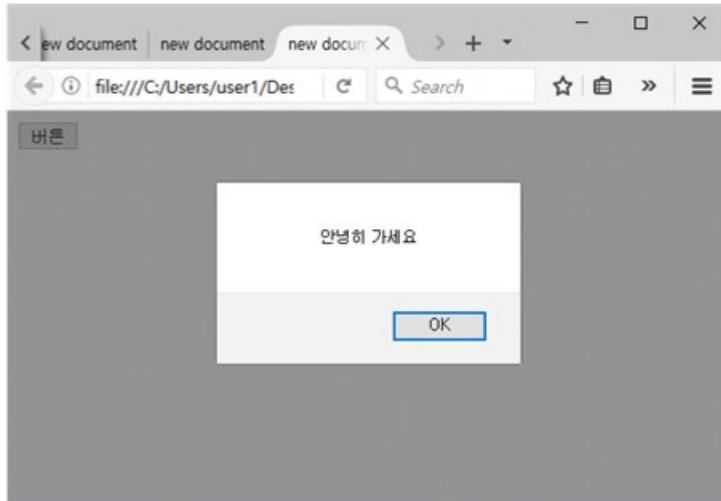
1. 기본적인 함수의 정의와 호출

2. [기본형식]

```
function 함수명() {  
    실행문;  
}  
함수명();
```

function은 함수의 정의문으로서, “다음 줄괄호({})에 입력하는 실행문들은 지금 당장 실행하지 말고, 함수명으로 저장하라”는 의미이다. 저장된 실행문들은 “함수명”으로 메모리에 저장되어 언제든 “함수명”으로 호출할 수 있다.

예제 1-30) 아래 화면처럼 버튼을 누르면 메시지 창이 뜨도록 함수를 만드시오.



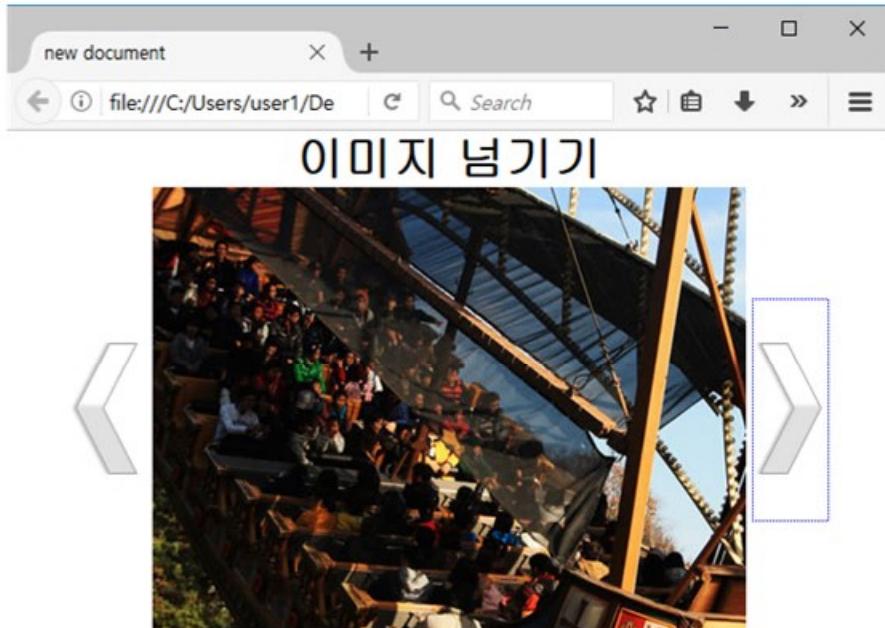
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">  
<head>  
<meta http-equiv="content-type" content="text/html; charset=utf-8" />  
<script type="text/javascript">  
//<![CDATA[<br/>    function greet(){  
        alert("안녕히 가세요 ")  
    }  
//]]>  
</script>  
<title> new document </title>  
</head>  
<body>
```

```

<button onclick="greet()">버튼</button>
</body>
</html>

```

예제 1-31) 이전 / 다음 버튼을 누를 때 그림이 순환(Rolling)되도록 만드시오.



```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<style type="text/css">
    *{margin:0; padding:0;}
    img{vertical-align:top; border:0;}
    #galleryWrap{width:510px; margin:0 auto; text-align:center;}
    #galleryWrap img{vertical-align:middle;}
</style>
<script type="text/javascript">
//<![CDATA[
var num=1;
function nextGallery(){
    num++;
    if(num>7) num=1;
    document.getElementById("gallery").src="images/img"+num+".jpg";
    return false;
}

```

```

function prevGallery(){
    num--;
    if(num<1) num=7;
    document.getElementById("gallery").src="images/img"+num+".jpg";
    return false
}
//]]>
</script>
<title> new document </title>
</head>
<body>
<div id="galleryWrap">
<h1>이미지 넘기기</h1>
<p>
    <a href="#prev" onclick="return prevGallery();">
    </a>
    
    <a href="#next" onclick="return nextGallery();">
    </a>
</p>
</div>
</body>
</html>

```

a. onclick="return prevGallery(); / onclick="return nextGallery(); " “이전”과 “다음” 버튼을 클릭하면 각각 prevGallery()나 nextGallery()함수를 호출한다.

b. if(num>7) num=1;

이미지 파일명 끝의 숫자가 1~7 까지이므로 num이 7보다 커지면 1로 치환

c. document.getElementById("gallery").src="images/img"+num+".jpg";

getElementById(“id명”)메소드는 문서 내에서 특정 id값으로 지정된 요소를 불러와 그 속성에 대한 설정을 사용할 수 있으며 “gallery”라는 id로 지정된 이미지의 “src”값을 ‘num’ 변수가 포함된 값으로 변경하여 미리 이미지 순환 함수로 지정된 prevGallery()와 nextGallery()등이 작동할 수 있도록 돋는다.

d. return false

여기까지라는 의미이다. 최초 함수를 불러낸 “클릭” 시점에서 동작을 멈추기 위함.

2. 전달 인자값이 있는 함수

매개변수 자리에 특정인자값을 전달하고, 그값이 적용된 상태의 함수가 수행되도록 한다.

기본형식

```
function 함수명(매개변수1, 매개변수2, ...){
```

```
    실행문;
```

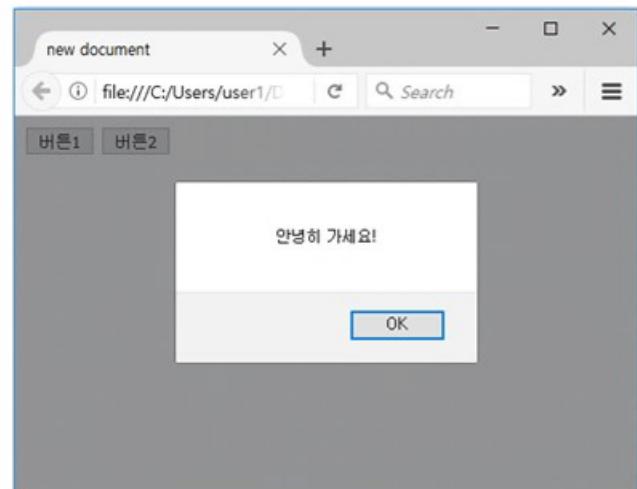
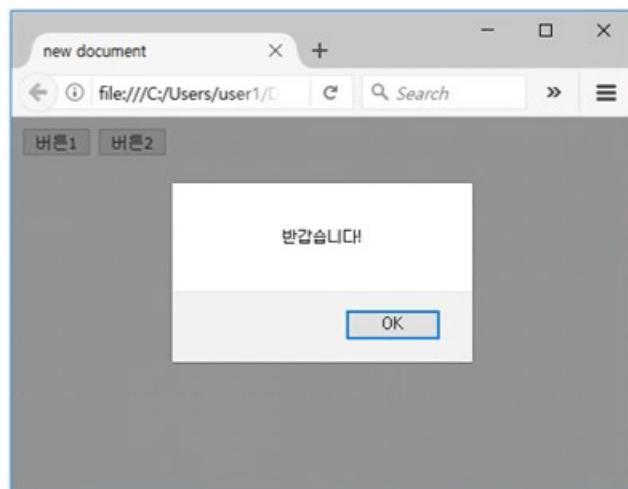
```
}
```

```
함수명(인자값1, 인자값2);
```

위의 기본 형식에서 피 호출함수의 매개변수가 2개라면 호출함수의 인자도 2개이어야 한다.
즉, 매개변수와 인자값이 동일하게 나열되어야 정확한 전달이 가능하다는 의미이다.

예제 1-32)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <title> new document </title>
<script type="text/javascript">
//<![CDATA[
    function greet(theText){
        alert(theText);
    }
//]]&gt;
&lt;/script&gt;
&lt;/head&gt;
&lt;body&gt;
    &lt;button onclick="greet('반갑습니다!')"&gt;버튼1&lt;/button&gt;
    &lt;button onclick="greet('안녕히 가세요 !')"&gt;버튼2&lt;/button&gt;
&lt;/body&gt;
&lt;/html&gt;</pre>
```



3. return

함수의 실행에서 return문을 실행하면 다음 실행문을 수행하지 않고 함수를 마친다.
또 하나의 기능으로는 함수 호출문에게 값을 반환한다.

예제 1-34)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```

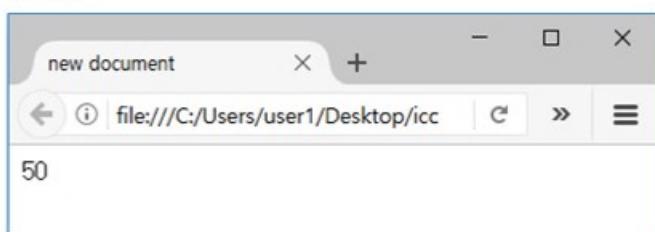
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta name="generator" content="editplus" />
<meta name="author" content="" />
<meta name="keywords" content="" />
<meta name="description" content="" />
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title> new document </title>
<script type="text/javascript">
//<![CDATA[
    var num1=10;
    function theFnc(num2) {
        var sum=num1+num2;
        var mul=num1*num2;

        return sum;
        document.write("return문 뒤에 실행문");
    }

    var result=theFnc(40);
    document.write(result);
//]]>
</script>
</head>
<body>

</body>
</html>

```



4. 지역변수(Local Variable)와 전역변수(Global Variable)

지역변수란 변수가 정의된 함수내에서만 사용이 가능한 변수이며, 전역변수는 현재페이지 내 자바스크립트 어디서든지 사용이 가능한 변수이다.

① 지역변수의 정의

함수내에서 선언하며, 반드시 앞에 var키워드를 붙여서 선언한다.

예제 1-36)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title> new document </title>
<script type="text/javascript">
//<![CDATA[
function theFnc(){
    var num;
    num=30;
}
theFnc();
console.log(num); //지역변수 선언된 변수에 값을 불러 올 수 있을까요?
//]]>
</script>

</head>

<body>

</body>
</html>
```

지역변수로 선언된 변수는 함수 밖에서 사용할 수 없다.

현재 콘솔창에서의 오류는 “정의되지 않은변수”라는 의미이다.

② 전역변수

변수를 함수 바깥에 선언하거나 함수내에서 var연산자를 붙이지 않고 선언한다.

외부에서 선언할 경우 var를 붙이지 않아도 실행에는 문제가 없으나 좋은 습관은 아니므로 항상 var를 붙여서 선언하는 것이 좋다.

예제 1-37)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title> new document </title>
<script type="text/javascript">
//<![CDATA[
var num; // 전역변수 num 선언
```

```

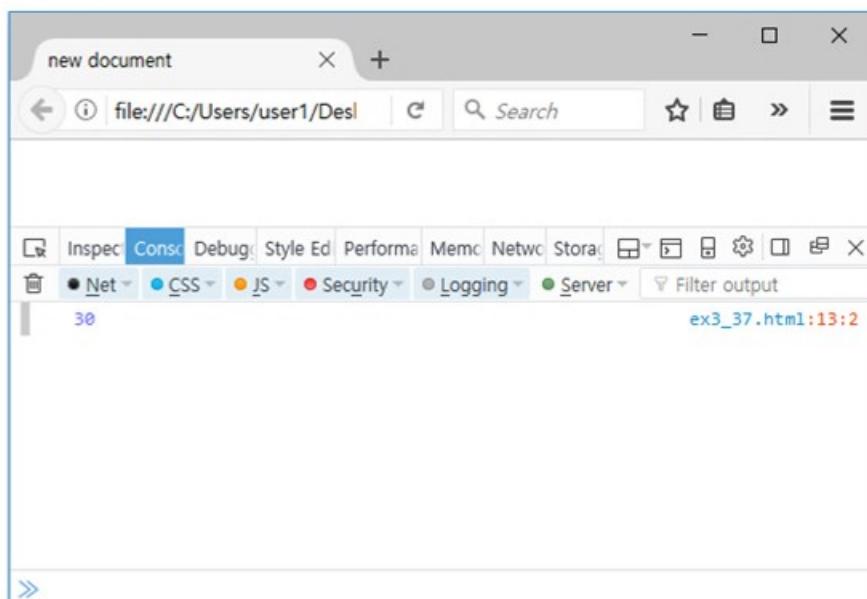
        function theFnc() {
            num=30;          // 지역변수 num 선언
        }
        theFnc();
        console.log(num); // 지역변수로 선언된 num의 값을 불러 올 수 있을까요?
    //]]>
</script>

</head>

<body>

</body>
</html>

```



예제 1-38)

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title> new document </title>
<script type="text/javascript">
//<![CDATA[
var i=10;
var num=20;
function theTest(){
</pre>

```

```

        var num=i+200;
        alert(num);
    }
    theTest();

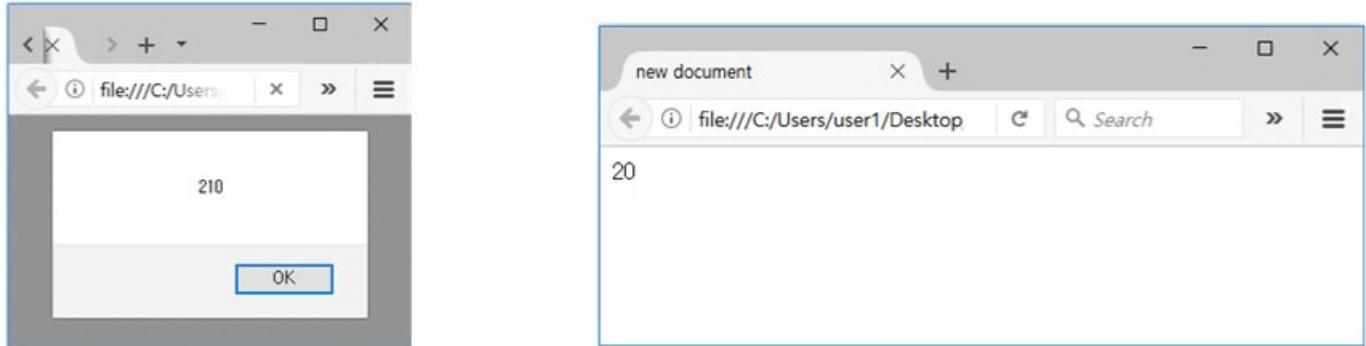
    function theFnc(){
        document.write(num);
    }
    theFnc();
//]]>
</script>

</head>

<body>

</body>
</html>

```



theTest함수는 내부에 지역변수 num을 포함하므로, theTest함수를 호출하면 전역변수가 아닌 지역변수 num(i+200)이 우선 적용이 되어 '210' 이 적힌 경고창이 뜬다.

내부의 지역변수 없이 num이라는 변수를 쓴 theTest함수가 호출되면, theTest함수의 지역변수가 아닌 전역변수 20이 적용된다.

6. 내장함수

자바스크립트는 사용자의 편리를 위해서 다양한 내장 함수를 제공한다.

내장 함수 종류	설명
eval()	문자열을 자바스크립트 문장으로 변환(수식은 계산, 문자는 변수로)
parseInt()	문자형 숫자를 정수로 변환
parseFloat()	문자형 숫자를 실수로 변환
isNaN()	"Not a Number"의 약자로 숫자가 아닌 문자가 포함되면 true를 반환
isFinite()	주어진 값이 유리수인지 아닌지 판단
Number()	문자를 숫자형으로 변환
String()	숫자를 문자형으로 변환

escape()	문자를 16진수 아스키코드 값으로 반환
unescape()	escape()로 변환된 값을 다시 되돌려 변환

예제 1-39)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <title> new document </title>
<script type="text/javascript">
//<![CDATA[
    var result1=eval("10+20");
    document.write(result1+"&lt;br /&gt;");

    var num=100;
    document.write("num"+10+"&lt;br /&gt;");
    document.write(eval("num")+10+"&lt;br /&gt;");

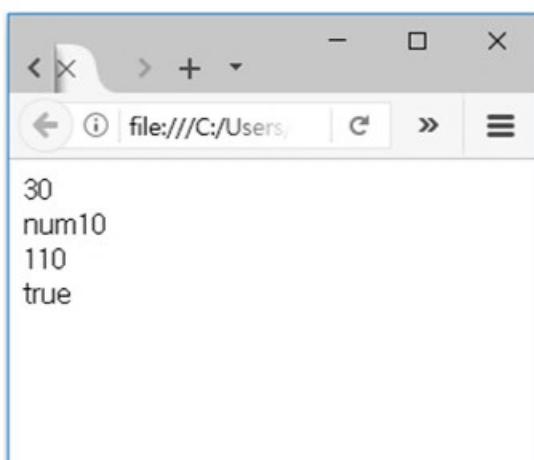
    var result2 isNaN("02-235-1620");
    document.write(result2+"&lt;br /&gt;");

//]]&gt;
&lt;/script&gt;
&lt;/head&gt;

&lt;body&gt;

&lt;/body&gt;
&lt;/html&gt;</pre>

```



7. 생성자

생성자 함수를 이용하면 특정 속성과 메서드를 가진 객체를 쉽게 생성할 수 있다.

`new` 연산자를 이용하여 해당 함수명을 호출하면 미리 정의된 속성과 메서드를 지닌 새로운 객체가 만들어 진다.

기본형식

```
function 함수명(매개변수1, 매개변수2, ..., 매개변수 n){  
    this.속성1 = 매개변수1;  
    this.속성2 = 매개변수2;  
    this.속성3 = 매개변수3;  
    this.속성4 = 매개변수4;  
    this.메서드1=function(){  
        실행문;  
    }  
}
```

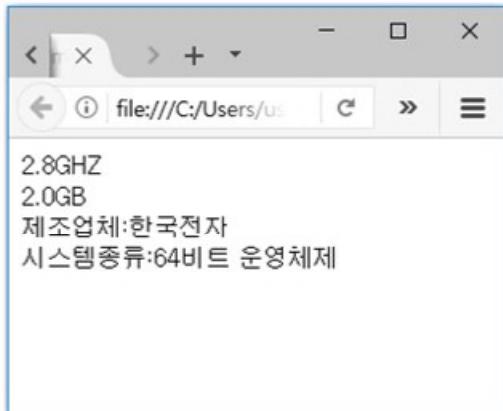
```
var 변수명(객체 변수) = new 함수명(값1,값2,...,값n)
```

예제 1-40)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">  
<head>  
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />  
    <title> new document </title>  
<script type="text/javascript">  
//<![CDATA[  
function comFactory(cpuInfo,ramInfo,hddInfo){  
    this.cpu=cpuInfo;  
    this.ram=ramInfo;  
    this.hdd=hddInfo;  
  
    this.totInfo=function(){  
        document.write("제조업체:한국전자 <br />");  
        document.write("시스템종류:64비트 운영체제 <br />");  
    }  
}  
  
var speedCom=new comFactory("2.8GHZ","2.0GB","1TB");  
document.write(speedCom.cpu+"<br />");  
document.write(speedCom.ram+"<br />");  
  
speedCom.totInfo();
```

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

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



a. function comFactory(cpuInfo,ramInfo,hddInfo)

매개변수가 3개인 생성자 함수이다. 여기서 this란 생성될 객체를 의미한다.

b. var speedCom=new comFactory("2.8GHZ","2.0GB","1TB");

new 연산자를 사용하여 새 객체를 생성하고 객체명으로 'speedCom'을 부여한다.

만일 새로운 객체를 하나 더 생성하려면 new 연산자를 통해 한번 더 함수를 호출하고 생성 객체명을 부여하면 된다.

c. speedCom.totInfo();

생성자 함수에서 지정한 totInfo() 메서드를 실행한다.

이때 각 매개변수는 새로운 객체인 'speedCom'을 생성할 때 넣은 인자값("2.8GHZ","2.0GB","1TB")에 맞게 출력된다.

01-7. 이벤트

브라우저들은 사용자가 띄운 웹 문서를 그 구성과 내용에 맞게 객체화하여 각 요소별(버튼, 이미지 등)로 구조화한다.

각 요소들은 상, 하위 구조나 병렬구조 등으로 체계화되는데 이렇게 정리된 체계를 DOM(Document Object Model, 문서 객체 구조)이라고 부른다.

자바스크립트는 브라우저가 생성한 이 DOM구조를 통해 각 요소와의 상호작용을 가능하게 함으로써 웹 페이지가 보다 인터랙티브하게 작동하도록 돋는 것이다.

DOM구조와의 상호작용에서 가장 빈번하게 사용되는 방법이 “이벤트”이다. 웹 브라우저에서 일어나는 모든 행동들을 “이벤트(Event)”라고 한다.

특정 이벤트가 발생하는 시점에서 함수를 호출하거나 스크립트를 실행하게 하는 요소를 ‘이벤트 핸들러’라고 부른다.

이벤트	설명
onload	문서를 모두 불러오고 나서 발생
onunload	문서를 종료할 때 발생
ondblclick	대상을 더블클릭했을 때 발생
onclick	대상을 클릭했을 때 발생
onkeypress	키보드를 눌렀을 때 발생
onkeydown	키보드를 눌렀을 때 발생
onkeyup	키보드를 누르고 떼었을 때 발생
onmouseover	대상에 마우스를 올렸을 때 발생
onfocus	대상에 포커스가 생겼을 때 발생
onmouseout	대상에서 마우스가 벗어났을 때 발생
onblur	대상에서 포커스를 잃었을 때 발생
onsubmit	전송(submit)버튼을 눌렀을 때 발생
onreset	취소(reset)버튼을 눌렀을 때 발생
onchange	입력요소의 값이 바뀌고, 포커스가 옮겨졌을 때 발생
onmousemove	마우스를 움직일 때마다 발생
onresize	브라우저의 창을 조절할 때마다 발생
onerror	문서나 이미지를 불렀을 때 에러가 나면 발생
onabort	문서를 불러오다가 중단되면 발생

1. 이벤트 핸들러 사용법

“행 입력형” 방식 : 이벤트 핸들러를 사용하는 방법에는 태그에 직접 입력해서 사용

“외부 분리형” 방식 : 완전히 분리해서 사용

① 이벤트 행 입력 방식

<body>

```
<button onclick="alert('javascript')>버튼 </button>
```

</body>

② 이벤트 분리형

■ 수정 전(잘못된 사용법)

```
<script type="text/javascript">
    function theFnc(){
        alert("javascript");
    }
    var myBtn = document.getElementById("btn");
    myBtn.onclick = theFnc;
</script>
</head>
<body>
    <button id = "btn1" > 버튼</button>
</body>
```

a. var myBtn = document.getElementById("btn");

현재 html문서에 자바스크립트 선언문이 <body>보다 위에 있으므로 getElementById 선택자를 이용해서 요소를 가지고 올 수 없다.

b. myBtn.onclick = theFnc;

선택한 버튼요소를 클릭했을 때 theFnc함수를 호출한다.

■ 수정 후(올바른 사용법)

```
<script type="text/javascript">
    function theFnc(){
        alert("javascript");
    }
    window.onload=function(){
        var myBtn = document.getElementById("btn");
        myBtn.onclick = theFnc;
    }
</script>
</head>
<body>
    <button id = "btn1" > 버튼</button>
</body>
```

window.onload=function(){실행문;}

현재 브라우저의 모든 요소들을 로딩한 후에 함수문을 실행하라는 의미이다.

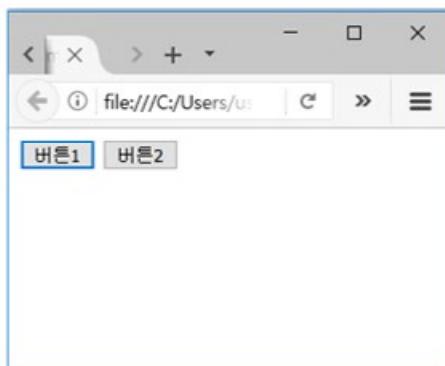
② 기존 방식(DOM Level 0)이벤트의 문제점

객체 하나와 이벤트 하나를 연결하여 작동하므로 한 객체에 동시에 같은 이벤트를 두 번 이상 적용할 수 없다는 문제를 안고 있다.

예제 1-41)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title> new document </title>
<script type="text/javascript">
//<![CDATA[
function theFnc(){
    alert("javascript");
}
window.onload=function(){
    var myBtn=document.getElementById("btn1"); //1. window 객체에 onload이벤트 적용(작동x)
    myBtn.onclick=theFnc;
}

window.onload=function(){
    var myBtn=document.getElementById("btn2"); //1. window 객체에 onload이벤트 적용(작동o)
    myBtn.onclick=theFnc;
}
//]]&gt;
&lt;/script&gt;
&lt;/head&gt;
&lt;body&gt;
&lt;button id="btn1"&gt;버튼1&lt;/button&gt;
&lt;button id="btn2"&gt;버튼2&lt;/button&gt;
&lt;/body&gt;
&lt;/html&gt;</pre>
```



버튼1은 작동하지 않고 버튼2만 작동한다.

'window' 객체에 'onload' 이벤트 핸들러가 2번 적용되었다. 그러므로 마지막에 적용된 이벤트만 실행된다.

③ DOM Level 2방식 이벤트

a. 표준방식(W3C) 브라우저(파이어폭스/크롬/사파리)에 대한 이벤트 등록 방법

이벤트대상.addEventListener("이벤트", 함수명, false);

특정 '버튼'을 '클릭' 했을 때 'theFnc'라는 함수를 호출하고 싶다면 이벤트대상은 '버튼'이고 이벤트명은 'click'이며, 호출할 함수명은 'theFnc'가 된다. 마지막 인자값 false는 해당이벤트를 끝내고 다음 이벤트로 넘기는 역할을 한다.

인자값이 false가 포함되면 해당 이벤트가 발생한 위치에서 멈추고 아래로 진행한다, 이를 버블업 방식이라고 부른다.

b. 비표준 방식 브라우저(IE8 이하)에 대한 이벤트 등록방법

이벤트 대상.attachEvent("on" + "이벤트명",함수명)

여기서 주의할 점은 인자값에 이벤트명을 입력할 때 'on'을 붙여야 한다.

예제 1-42)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="kor" xml:lang="ko">
<head>
<title> new document </title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<script type="text/javascript">
//<![CDATA[
function theFnc(){
    alert("javascript");
}
function clickBtn1(){
    var myBtn1=document.getElementById("btn1");
    myBtn1.onclick=theFnc;
}
function clickBtn2(){
    var myBtn2=document.getElementById("btn2");
    myBtn2.onclick=theFnc;
}

if(window.addEventListener){ //1. 표준 브라우저일 경우 실행
    window.addEventListener("load",clickBtn1,false);
    window.addEventListener("load",clickBtn2,false);
} else if(window.attachEvent){ //2. 비표준 브라우저(IE)일 경우 실행
    window.attachEvent("onload",clickBtn1)
    window.attachEvent("onload",clickBtn2)
}
//]]&gt;</pre>
```

```

</script>

</head>

<body>
<button id="btn1">버튼1</button>
<button id="btn2">버튼2</button>
</body>
</html>

```

4. 이벤트 객체 생성 및 속성

“이벤트 객체”란 이벤트 핸들러가 구동될 때 생성되는 객체로서 ‘클릭’이나 ‘키보드 입력’ 등의 동작 자체를 의미한다.

① 이벤트 객체 생성

```

선택대상.onclick = function(매개변수){
    매개변수(이벤트 객체).속성
}

```

특정 이벤트 핸들러를 구동하면, 함수의 매개변수에 이벤트 객체가 자동으로 생성된다.

단, 브라우저가 W3C의 표준안을 따르는 브라우저일 경우에만 적용된다.

기본형식

```

선택대상.onclick = function(e){
    theEvent = e ? e : window.event;
    console.log(theEvent.속성);
}
.....
```

<예>

```

document.onclick = function(e){
    theEvent = e ? e : window.event;
    console.log(theEvent.clientX);
}

```

② 이벤트 객체 속성

이벤트	속성	설명
keydown, keypress	altKey	키보드의 [Alt]키가 눌렸을 때 true값 변환
	shiftKey	키보드의 [Shift]키가 눌렸을 때 true값 변환
	ctrlKey	키보드의 [Ctrl]키가 눌렸을 때 true값 변환
	keyCode	입력된 문자키의 고유한 유니코드값을 반환한다.
click, mousemove, mouseover, mouseup,	clientX	현재문서 기준, 이벤트가 발생한 X좌표(스크롤바 너비 계산 x)
	clientY	현재문서 기준, 이벤트가 발생한 Y 표(스크롤바 높이 계산 x)
	screanX	모니터 기준, 이벤트가 발생한 X좌표
	pageX	현재문서 기준, 이벤트가 발생한 Y 표(스크롤바 높이 계산 o)
	pageY	현재문서 기준, 이벤트가 발생한 Y 표(스크롤바 높이 계산 o)

mousedown	layerX	position 속성값이 'absolute'인 레이어 안에서의 상대 x 좌표
	layerY	position 속성값이 'absolute'인 레이어 안에서의 상대 Y 좌표
	target	이벤트가 발생한 대상 요소
	button	마우스버튼(왼쪽/오른쪽/휠)의 클릭된 상태 체크
all	type	이벤트의 종류

예제 1-43)

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="kor" xml:lang="ko">
<head>
<title> new document </title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<script type="text/javascript">
//<![CDATA[
document.onmousemove=function(e){
    var theEvent=e?e:window.event; // 1. 이벤트 객체 생성
    var myForm=document.form1;

    //2. 이벤트 속성 - 문서 기준으로 좌표를 구함.
    myForm.clt_x.value="clientX: "+theEvent.clientX;
    myForm.clt_y.value="clientY: "+theEvent.clientY;

    //3. 이벤트 속성 - 스크린(모니터)기준으로 좌표를 구함.
    myForm.src_x.value="screenX: "+theEvent.screenX;
    myForm.src_y.value="screenY: "+theEvent.screenY;

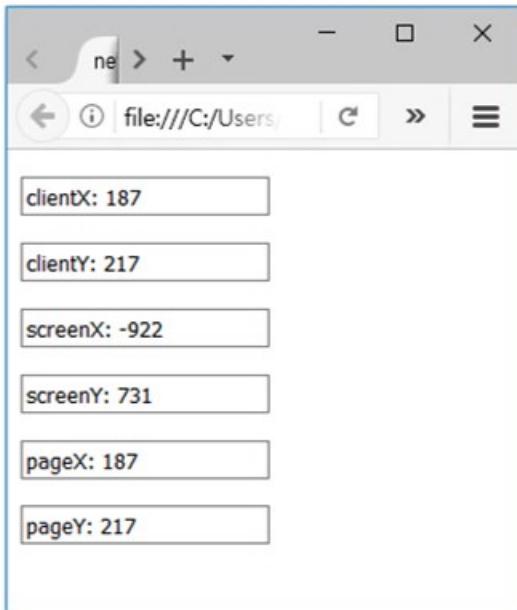
    //4. 이벤트 속성 - 문서 기준으로 스크롤바 높이까지 계산해 좌표를 구함.
    myForm.pg_x.value="pageX: "+theEvent.pageX;
    myForm.pg_y.value="pageY: "+theEvent.pageY;
}

//]]&gt;
&lt;/script&gt;
&lt;/head&gt;

&lt;body&gt;
&lt;form name="form1" action="#"&gt;
    &lt;p&gt;&lt;input type="text" name="clt_x" id="clt_x" /&gt;&lt;/p&gt;
    &lt;p&gt;&lt;input type="text" name="clt_y" id="clt_y" /&gt;&lt;/p&gt;
    &lt;p&gt;&lt;input type="text" name="src_x" id="src_x" /&gt;&lt;/p&gt;
    &lt;p&gt;&lt;input type="text" name="src_y" id="src_y" /&gt;&lt;/p&gt;
    &lt;p&gt;&lt;input type="text" name="pg_x" id="pg_x" /&gt;&lt;/p&gt;
    &lt;p&gt;&lt;input type="text" name="pg_y" id="pg_y" /&gt;&lt;/p&gt;
&lt;/form&gt;
</pre>

```

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



a. theEvent.clientX와 theEvent.pageY 차이점

페이지에서 가로/세로 스크롤바가 발생했을 때 client는 스크롤바의 높이를 계산하지 않고 page는 스크롤바 높이까지 계산하여 각각 위치값을 구한다.

b. theEvent.screenX

모니터 좌측 상단을 기준으로 좌표값을 구한다.

비 IE계열 브라우저(파이어폭스, 크롬 등)들만의 적용 속성으로, 만약 IE에서 실행하면 pageX,pageY의 값은 모두 'undefined(확인되지 않음)'로만 출력된다.

예제 1-44) 특정 키보드 값을 추적하여 이를 활용한 단축키(조합)의 작동을 구현한 예제

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml" lang="kor" xml:lang="ko">  
<head>  
    <title> new document </title>  
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />  
    <script type="text/javascript">  
        //<![CDATA[<br/>        document.onkeydown=function(e){  
            var theEvent=e?e:window.event;           // 1. 이벤트 객체 생성  
            var theKey1=theEvent.keyCode;           // 2. 키코드 생성  
            var theKey2=theEvent.shiftKey;          // 3. 키보드 상의 [Shift]키의 눌림 여부  
  
            if(theKey1==72&&theKey2){           // 4. [Shift]+[H]누르면 조건문 실행  
                alert("단축키 완성")  
            }  
        }  
        //]]>  
    </script>  
</head>  
<body>  
    <h1>Hello World</h1>  
</body>
```

```
}
```

```
}
```

```
//]]>
```

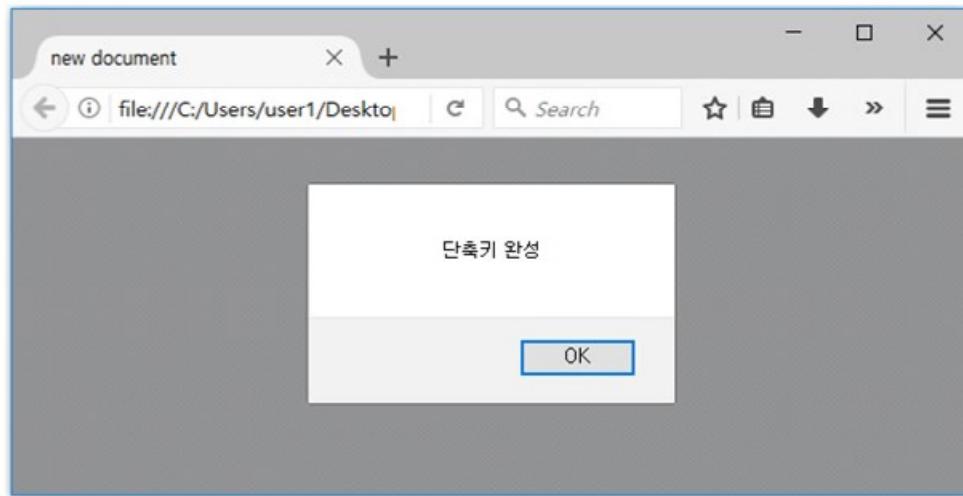
```
</script>
```

```
</head>
```

```
<body>
```

```
</body>
```

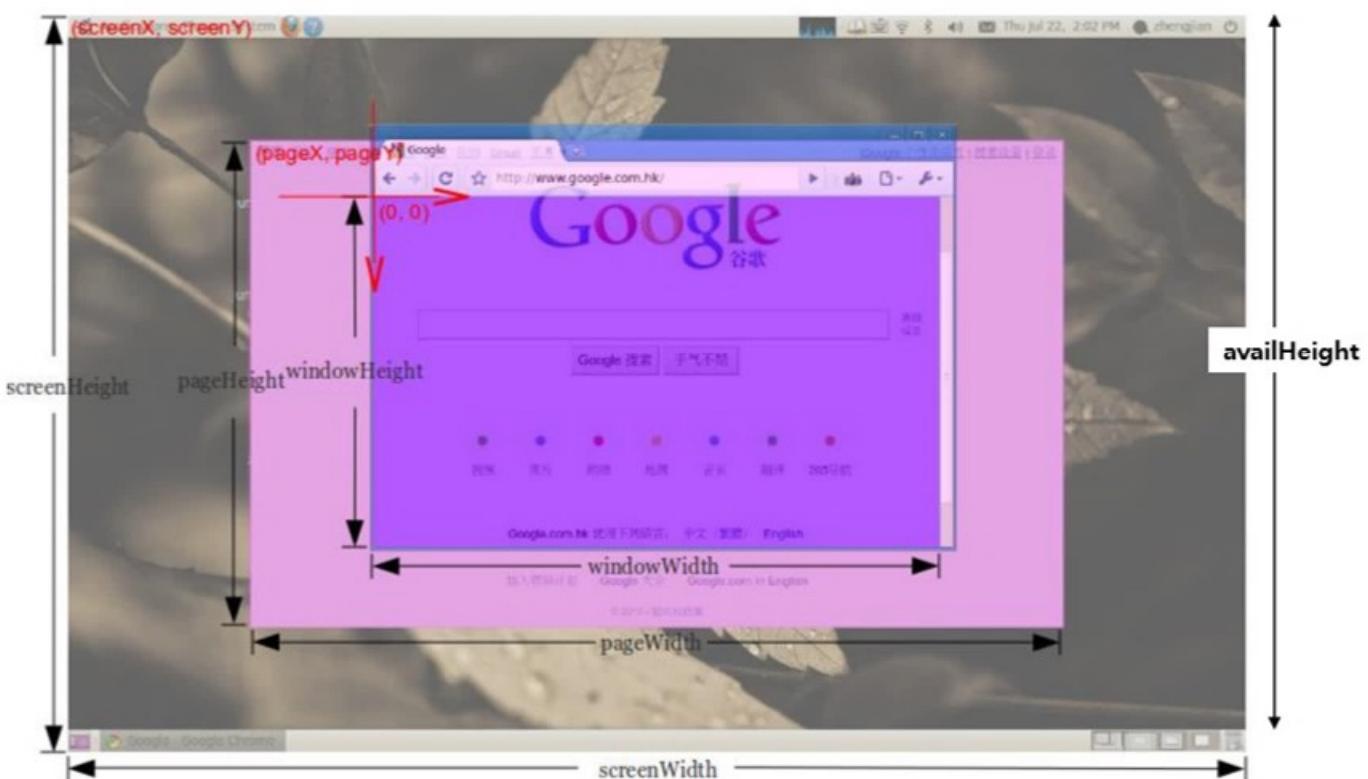
```
</html>
```



01-8. 브라우저 객체(Browser Object Model)

BOM(브라우저 객체 모델)이란 브라우저 창에 포함된 모든 객체 요소들을 의미한다.

브라우저에는 URL주소에 대한 정보를 제공하는 'location' 객체, 현재 실행 중인 브라우저에 대한 정보(브라우저명, 코드명 등)를 제공하는 'navigator' 객체, 브라우저의 방문 기록에 대한 정보를 제공하는 'history' 객체 등이 있다.



브라우저 객체의 종류

브라우저 객체	
window	document
	navigator
	location
	screan
	history

1. window

모든 객체의 최상위 객체이며, 새 창을 생성하는 역할 등을 합니다.

메서드	설명
open("경로","창이름","*옵션")	새 창으로 문서를 열 때 사용
alert("문자")	경고창을 열 때 사용
prompt("질문","기본응답")	질문/응답창을 열 때 사용
confirm("질문")	확인/취소 창을 열 때 사용
setTimeout(실행문,간격시간)	해당 간격시간 뒤에 실행문을 한 번만 수행
setInterval(실행문,간격시간)	해당 간격으로 실행문을 반복적으로 수행

*open 메소드의 옵션 종류

메서드	설명
width	새 창의 너비(px, 픽셀 단위)
height	새 창의 높이(px, 픽셀 단위)
left	새 창을 열 때 스크린 기준 좌측 위치(px, 픽셀 단위)
top	새 창을 열 때 스크린 기준 상측 위치(px, 픽셀 단위)
scrollbars	스크롤바의 생성여부 (값은 yes or no)
menubar	메뉴 바의 노출 여부 (값은 yes or no)
toolbar	도구 상자의 노출 여부(값은 yes or no)
location	주소 표시줄의 노출 여부(값은 yes or no)
status	상단 표시줄의 노출 여부(값은 yes or no)
resizable	창의 사이즈를 조정 가능 여부(값은 yes or no)

예제 1-45) open 메서드를 활용한 팔업창 띄우기

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="kor" xml:lang="ko">
<head>
<title> new document </title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<script type="text/javascript">
//<![CDATA[
window.open("popup1.html", "naver", "width=300px, height=400px, left=200px,
top=20px, scrollbars=no, toolbar=no, location=no");
//]]&gt;
&lt;/script&gt;
&lt;/head&gt;
&lt;body&gt;
&lt;h1&gt;오프너페이지&lt;/h1&gt;
&lt;/body&gt;
&lt;/html&gt;</pre>
```

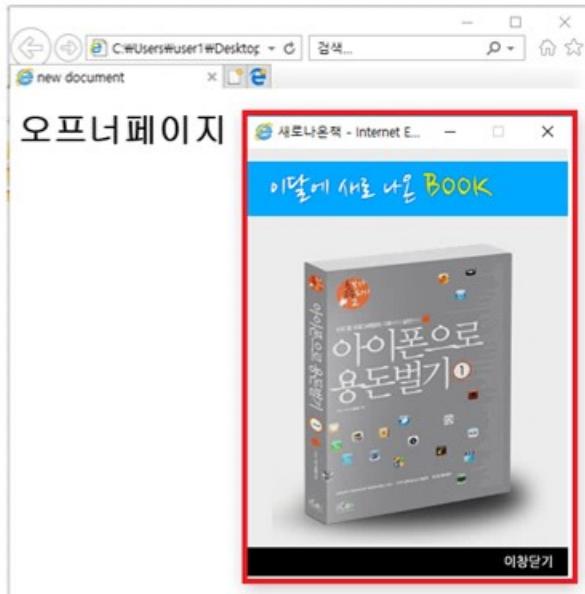
popup1.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="kor" xml:lang="ko">
<head>
<title> 새로나온책 </title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<style type="text/css">
*{margin:0; padding:0;}
</style>
```

```

</head>
<body>
<p></p>
<map name="pop" id="pop">
<area shape="rect" coords="228,371,294,399" href="#" onclick="window.close();" alt="창닫기"/>
</map>
</body>
</html>

```



2. navigator

navigator는 브라우저에 대한 정보를 제공하는 객체이다.

속성	설명
appCodeName	브라우저의 코드명을 제공
appName	브라우저의 이름을 제공
appVersion	브라우저의 기본 버전과 실행 중인 플랫폼 정보를 제공
userAgent	브라우저의 전체적인 정보를 제공

예제 1-46) navigation 객체의 속성들을 사용

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="kor" xml:lang="ko">
<head>
<title> new document </title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<script type="text/javascript">
//<![CDATA[
    document.write("코드명: "+navigator.appCodeName+"<br />")
    document.write("브라우저명: "+navigator.appName+"<br />")

```

```

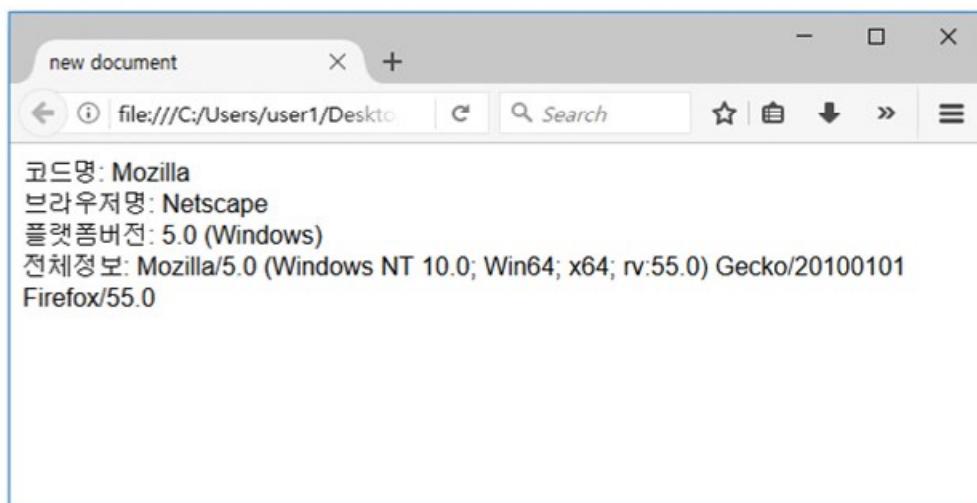
        document.write("플랫폼버전: "+navigator.appVersion+"<br />")
        document.write("전체정보: "+navigator.userAgent+"<br />")
    //]]>
</script>

</head>

<body>

</body>
</html>

```



예제 1-47) userAgent속성을 이용하면 현 운영체제에 대한 정보를 얻을 수 있으므로, 현재 사용자의 기기가 데스크탑 PC인지 모바일인지를 파악할 수 있다.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="kor" xml:lang="ko">
<head>
<title> new document </title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<script type="text/javascript">
//<![CDATA[
var myAgent=navigator.userAgent.toLowerCase();
var mobile=["iphone","ipod","android","blackberry","window
mini","sonyericsson","opera mobi","iemobile"];
for(var i=0; i&lt;mobile.length; i++){
if(myAgent.indexOf(mobile[i]) &gt;=0 ){
    location.href="http://m.naver.com";
    break;
}
</pre>

```

```

}
//]]>
</script>
</head>
<body>
  <h1>PC버전 페이지</h1>
</body>
</html>

```

3. location

location 객체는 브라우저의 현재 URL주소값에 대한 정보를 제공한다.

속성	설명
location.hash	주소창에 URL에서 #뒤에 오는 문자열을 반환
location.host	주소창에 URL에서 도메인명을 반환
location.hostname	주소창에 URL에서 도메인명을 반환
location.href	주소창에 URL에서 전체 정보 반환
location.pathname	주소창에 URL에서 도메인명 다음에 오는 경로를 반환
location.port	주소창에 URL에서 포트번호를 반환
location.protocol	주소창에 URL에서 프로토콜을 반환

예제 1- 48)

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="kor" xml:lang="ko">
<head>
<title> new document </title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<script type="text/javascript">
//<![CDATA[
  document.write("&lt;h1&gt;location.hash: "+location.hash+"&lt;/h1&gt;") ;
  document.write("&lt;h1&gt;location.host: "+location.host+"&lt;/h1&gt;") ;
  document.write("&lt;h1&gt;location.hostname: "+location.hostname+"&lt;/h1&gt;") ;
  document.write("&lt;h1&gt;location.href: "+location.href+"&lt;/h1&gt;") ;
  document.write("&lt;h1&gt;location.pathname: "+location.pathname+"&lt;/h1&gt;") ;
  document.write("&lt;h1&gt;location.protocol: "+location.protocol+"&lt;/h1&gt;") ;
//]]&gt;
&lt;/script&gt;

&lt;/head&gt;

&lt;body&gt;
&lt;/body&gt;
&lt;/html&gt;
</pre>

```

4. screen

screen 객체는 스크린(모니터)에 대한 속성을 제공하는 객체이다.

속성	설명
availWidth	사용 가능한 스크린 가로 너비 값을 반환
availHeight	사용 가능한 스크린 세로 높이 값을 반환
availTop	창의 위치할 수 있는 최상위의 위치값을 반환 (IE에서는 사용 안됨)
availLeft	창의 위치할 수 있는 가장 좌측의 위치값을 반환 (IE에서는 사용 안됨)

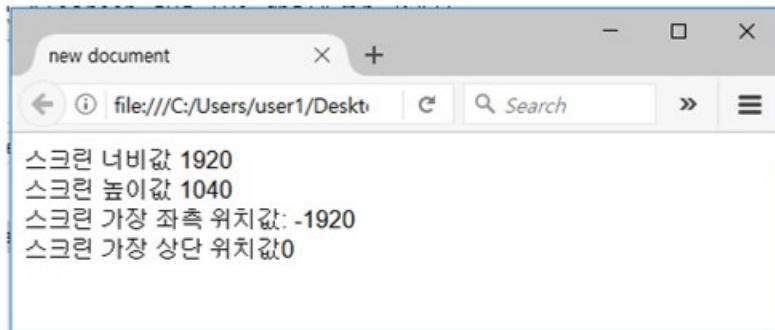
예제 1-49) 각각의 속성에 해당하는 값을 실습해본다.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="kor" xml:lang="ko">
<head>
    <title> new document </title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
<script type="text/javascript">
//<![CDATA[
    document.write("스크린 너비값 "+screen.availWidth+"&lt;br /&gt;");
    document.write("스크린 높이값 "+screen.availHeight+"&lt;br /&gt;");
    document.write("스크린 가장 좌측 위치값: "+screen.availLeft+"&lt;br /&gt;");
    document.write("스크린 가장 상단 위치값"+screen.availTop+"&lt;br /&gt;");
//]]&gt;
&lt;/script&gt;

&lt;/head&gt;

&lt;body&gt;

&lt;/body&gt;
&lt;/html&gt;</pre>
```



availHeight(사용가능한 스크린 높이 값)는 작업표시 줄을 제한한 높이를 반환한 것이다.

5.history

브라우저로 방문한 페이지들의 정보를 제공하여, 메서드를 사용하여 ‘이전’ 및 ‘다음’과 같은 페이지 이동을 위한 버튼을 만들 수 있다.

메서드	설명
go(값)	값만큼 다음 또는 이전 페이지로 이동한다.
back()	이전 페이지로 이동한다.
forward()	다음 페이지로 이동한다.

예제 1-50)

- ex3_50.html -

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="kor" xml:lang="ko">
<head>
<title> new document </title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
</head>
<body>
<h1>첫 페이지</h1>
<a href="ex3_51_2.html">두번째페이지 이동</a>
</body>
</html>
```

- ex3_50_2.html -

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="kor" xml:lang="ko">
<head>
<title> new document </title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
</head>
<body>
<h1>두번째 페이지</h1>
<p><a href="ex3_52_3.html">마지막페이지로 이동</a></p>
<input type="button" value="이전페이지로 이동" onclick="history.back();"/>
<input type="button" value="다음페이지로 이동" onclick="history.go(1);"/>
<input type="button" value="forward버튼" onclick="history.forward();"/>
</body>
</html>
```

- ex3_50_3.html -

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="kor" xml:lang="ko">
<head>
<title> new document </title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
</head>
<body>
<h1>마지막페이지</h1>
<p><input type="button" value="이전페이지로 이동" onclick="history.go(-1)" /></p>
<p><input type="button" value="첫페이지로 이동" onclick="history.go(-2)" /></p>
</body>
</html>
```

01-9. 문서 객체(Document Object Model)

웹페이지의 구조는 <html>이 모든 태그들을 감싸고 있으며, 그 내부는 다시 각종 설정과 선언으로 이뤄진 <head> 요소와 본문 내용이 모두 담긴 <body>요소로 나누어진다.

<body>요소 안에는 <div>, <p>, , <form>, <table> 등의 요소들이 배치된다.

1. 선택자

특정문서 요소(객체)에 스타일을 적용하거나 삭제하려면 그 문서 요소에 접근하는 방법을 알아야한다.

- * 원거리 선택자 : 요소에 포함되는 id속성 또는 요소(태그)명 등을 이용하여 직접 접근하는 방식
- * 근거리 선택자 : 선택된 요소를 기준으로 그 가까운 요소를 선택하고 싶을 때

일반적으로 요소에 접근할 때에는 우선 원거리 선택자를 이용하여 접근한 후 근거리 선택자를 이용하여 그 관련 요소들을 선택한다.

2. 엘리먼트 찾기

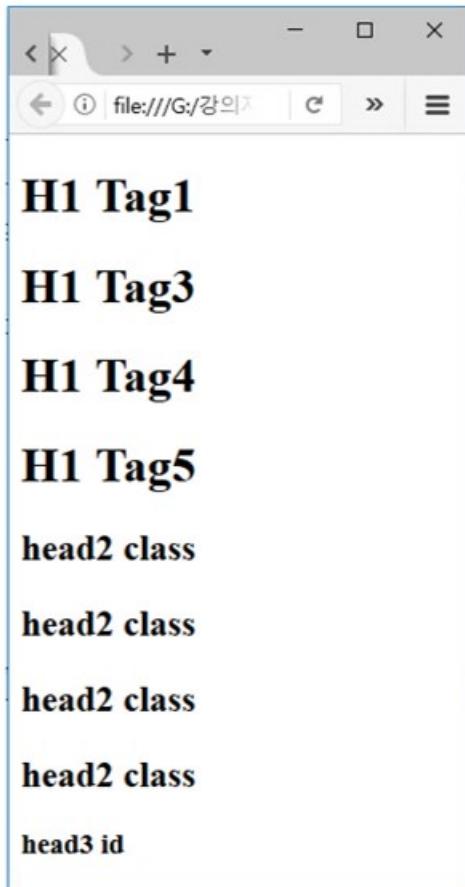
메서드	설명
document.getElementById("아이디")	특정 아이디의 엘리먼트를 찾는다.
document.getElementsByTagName("태그명")	특정 태그의 엘리먼트를 찾는다.
document.getElementsByClassName("클래스명")	특정 클래스의 엘리먼트를 찾는다.
document.querySelector()	css 셀렉터와 같은 역할을 한다.(IE8 이하는 사용 못 함)

예제 1-51)

```
<!DOCTYPE html>
<html>
  <body>
    <h1>H1 Tag1</h1>
    <h1>H1 Tag3</h1>
    <h1>H1 Tag4</h1>
    <h1>H1 Tag5</h1>
      <h2 class="head2">head2 class</h2>
      <h2 class="head2">head2 class</h2>
      <h2 class="head2">head2 class</h2>
      <h2 class="head2">head2 class</h2>
        <h3 id="head3">head3 id</h3>
  </body>
<script>
  var elemTag = document.getElementsByTagName("h1");
  var elemClass = document.getElementsByClassName("head2");
  /*아이디 자체가 하나의 요소로 될 수 있다.*/
  //head3.getElementById("head3");
  var elemId = document.getElementById("head3");

  console.log(elemTag);
```

```
        console.log(elemClass);
        console.log(elemId);
    </script>
</html>
```



예제 1-52)

```
<!DOCTYPE html>
<html>
    <body>
        <h2>Element Control</h2>
        <h3>query selector</h3>
        <p>쿼리 셀렉터는 css 의 셀렉터를 차용 한 것 입니다.</p>
        <p class='ex'>하지만 복수개의 엘리먼트는 수용 하지 못합니다.</p>
        <span id="sample"></span><br>
        <a href="http://www.w3schools.com/jsref/met_document_queryselector.asp"
target="_blank">
            관련 내용은 여기서 보세요
        </a>
        <div>
            <p>특정 요소의 자식 요소를 선택 할 수도 있습니다.</p>
        </div>
    </body>
</html>
```

```
<button>p태그</button>
<button>p의 ex클래스</button>
<button>아이디</button>
<br>
<button>자식요소</button>
<button>속성</button>
</body>
<script>

console.log(document.getElementsByTagName('button'));
console.log(document.querySelector('p'));

var btn = document.getElementsByTagName('button');

btn[0].addEventListener('click',function(){
    document.querySelector('p').style.backgroundColor='red';
});

btn[1].addEventListener('click',function(){
    document.querySelector('.ex').style.backgroundColor='blue';
});

btn[2].addEventListener('click',function(){
    document.querySelector('#sample').innerHTML='그래서 하나의 아이디나 첫번째 요
소를 지목 할때 유용 합니다.';
});

/*복수개를 가져오기 위해서는 querySelectorAll() 사용*/
btn[3].addEventListener('click',function(){
    document.querySelector('div p').style.backgroundColor='gray';
});

btn[4].addEventListener('click',function(){
    document.querySelector('a[target="_blank"]').style.border='10px solid red';
});
</script>
</html>
```

/G:/강의자료/JS_DOM_Ve X +

← ⓘ file:///G:/강의자료/JS_DOM_ve ↵ ↷ ⌂

Element Control

query selector

쿼리 셀렉터는 css 의 셀렉터를 사용 한 것 입니다.
하지만 복수개의 엘리먼트는 수용 하지 못합니다.

[관련 내용은 여기서 보세요](#)

특정 요소의 자식 요소를 선택 할 수도 있습니다.

3. 엘리먼트 및 속성 변경

메서드	설명
element.innerHTML=	html 엘리먼트를 변경
element.getAttribute	html 속성변경
element.setAttribute(attribute,value)	html 속성변경
element.style.property=	html 스타일 변경
element.getAttribute(attribute)	html 속성 가져오기

예제 1-53)

```
<!DOCTYPE html>
<html>
    <body>
        <p id="p1"></p>
        <button id="btn">innerHTML</button><br/>
        <br/>
        <button id="imgchange">이미지 변경</button>
    </body>
    <script>
        var elem = document.getElementById("p1");
        var btn = document.getElementById("btn");

        btn.addEventListener("click", function(){
            elem.innerHTML = "새로운 텍스트 등장";
        });

        var elem2 = document.getElementById("image");
    </script>

```

```

var btn2 = document.getElementById("imgchange");
btn2.addEventListener("click", function(){
    //elem2.src = "img02.jpg";
    elem2.setAttribute("src", "img02.jpg");
});
</script>
</html>

```

예제 1-54)

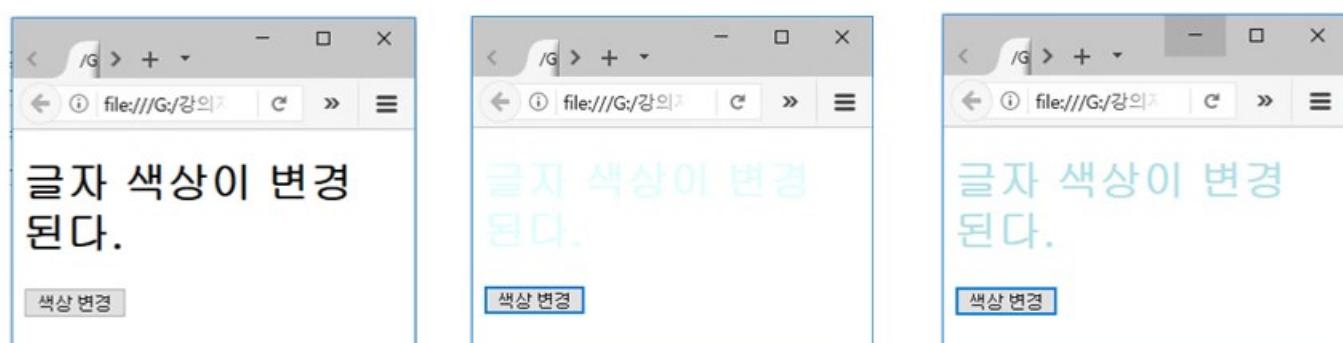
```

<!DOCTYPE html>
<html>
    <body>
        <h1 id="p1">글자 색상이 변경 된다.</h1>
        <button id="btn">색상 변경</button><br/>

    </body>
    <script>
        var elem = document.getElementById("p1");
        var btn = document.getElementById("btn");
        var fontColor =
            ["#e0ffff", "#b0e0e6", "#afeeee", "#add8e6", "#87ceeb", "#87cefa", "#00bfFf"];
        var i = -1;

        btn.addEventListener("click", function(){
            i++;
            if(i==fontColor.length){
                i = 0;
            }
            console.log(i);
            elem.style.color = fontColor[i];
        });
    </script>
</html>

```



예제 1-55)

```
<html>
    <head>
        <body>
            <input type="text" value="아무 내용이나..."/>
        </body>
        <script>
            var tags = document.getElementsByTagName("input");

            for(var i=0;i<tags.length;i++){
                console.log(tags[i].getAttribute("type"));
                console.log(tags[i].getAttribute("value"));
                console.log(tags[i].type);
                console.log(tags[i].value);
            }

            //tag[0].type="button";
            //tag[0].value="click";
            tag[0].setAttribute("type", "button");
            tag[0].setAttribute("value", "click");

            //onclick="alert('hi')"
            //tag[0].onclick="alert('hi');//안됨
            tag[0].setAttribute("onclick", "alert('hi')");

        </script>
    </html>
```

예제 1-56)

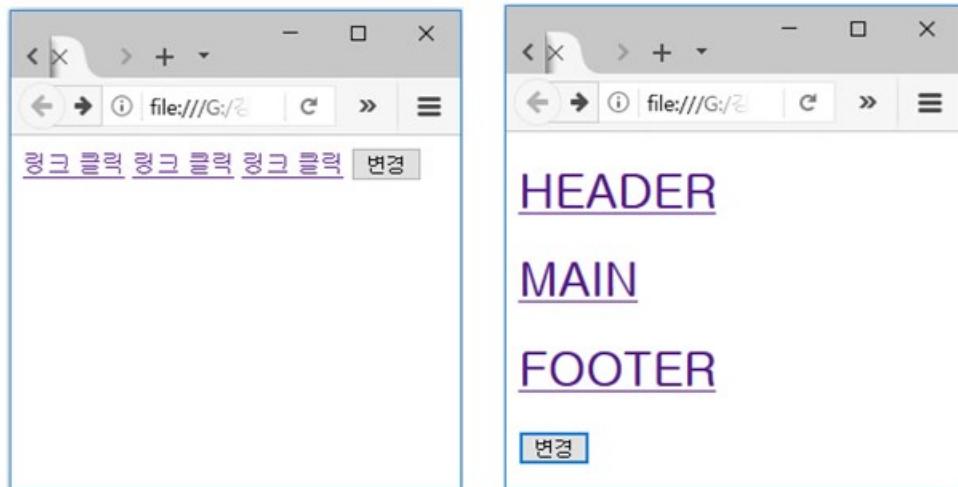
```
<html>
    <head>
        <head>
        <body>
            <a href="#" data-role="header">링크 클릭</a>
            <a href="#" data-role="main">링크 클릭</a>
            <a href="#" data-role="footer">링크 클릭</a>
            <button>변경</button>
        </body>
        <script>
            var tags = document.getElementsByTagName("a");
            var btn = document.querySelector("button");
```

```

btn.addEventListener("click",function(){
    for(var i=0;i<tags.length;i++){
        var attr = tags[i].getAttribute("data-role");
        if(attr == "header"){
            tags[i].innerHTML=<h1>HEADER</h1>;
        }
        if(attr == "main"){
            tags[i].innerHTML=<h1>MAIN</h1>;
        }
        if(attr == "footer"){
            tags[i].innerHTML=<h1>FOOTER</h1>;
        }
    }
});

</script>
</html>

```



예제 1-57)

```

<html>
    <head>
        <title>입력한 값 보여주기</title>
    </head>
    <body>
        당신이 입력한 값 : <span></span>
        <br>
        <input type="text" value="" onkeyup="keyEvt()"/><span>(0자 입력)</span>
    </body>
    <script>

```

```

//요소 추출
var msg = document.getElementsByTagName("span");
var txt = document.getElementsByTagName("input");

function keyEvt(){
    var val = txt[0].value;//동적만 됨

    if(val.length>=0 && val.length<5){
        msg[1].style.color="black";
    }

    if(val.length>=5 && val.length<10){
        msg[1].style.color="orange";
    }

    if(val.length>=11){
        val = val.substring(0,10);//0 번 부터 보여주고 10번부터 잘라낸다.
        alert("더 이상 입력 할 수 없습니다.");
        msg[1].style.color="red";
        txt[0].value = val;
        msg[0].value= val;
    }

    msg[0].innerHTML=val;
    msg[1].innerHTML="("+val.length+"자 입력)";
}
}

// 키업 이벤트 발생 시 value 값 받아 오기
/*
txt[0].addEventListener("keyup",function(){
    var val = txt[0].value;//동적만 됨
    msg[0].innerHTML=val;
    msg[1].innerHTML="("+val.length+"자 입력)";
    console.log(typeof val.length);

    if(val.length>=5 && val.length<10){
        msg[1].style.color="orange";
    }

    if(val.length>=0 && val.length<5){
        msg[1].style.color="black";
    }

    if(val.length>=10){
        val = val.substring(0,10);
        alert("더 이상 입력 할 수 없습니다.");
        msg[1].style.color="red";
        txt[0].value = val;
        msg[0].value= val;
    }

    msg[0].innerHTML=val;
    msg[1].innerHTML="("+val.length+"자 입력)";
})
}

```

```

        alert("더 이상 입력 할 수 없습니다.");
        msg[1].style.color="red";
    }
});
*/
</script>
</html>

```



4. 엘리먼트 추가 및 삭제

메서드	설명
document.createElement()	엘리먼트 만들기
document.removeChild()	자식요소 지우기
document.appendChild()	자식요소 추가
document.replaceChild()	자식요소 교체

예제 1-58)

```

<!DOCTYPE html>
<html>
    <body>
        <input id="val" type="text" /><button onclick="addList()">추가</button>
        <br>
        <ul id="myList">
            <li>Coffee</li><li>Tea</li><li>Milk</li>
        </ul>
        <button id="btn1">엘리먼트 추가</button>
        <button id="btn2">자식요소 지우기</button>
        <button id="btn3">자식요소 교체</button>

    </body>
    <script>

```

```
function addList(){
    var tNode = document.getElementById("val").value;
    var myList = document.getElementById("myList");
    var list = document.createElement("li");
    var txt = document.createTextNode(tNode);
    list.appendChild(txt);
    myList.appendChild(list);
}

document.getElementById("btn1").addEventListener("click",function(){

    var t = document.createTextNode("CLICK ME");
    var btn = document.createElement("BUTTON");

    btn.appendChild(t);
    document.body.appendChild(btn);

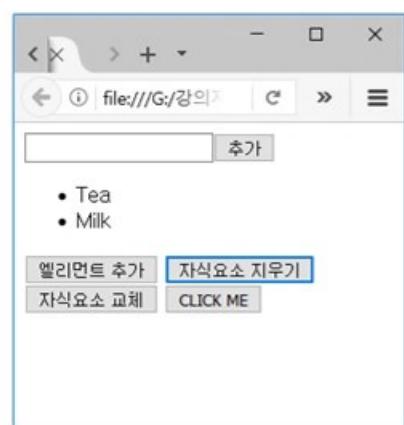
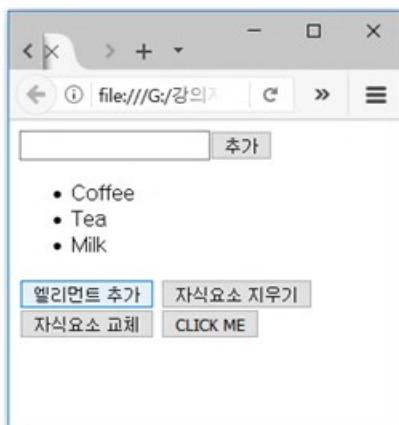
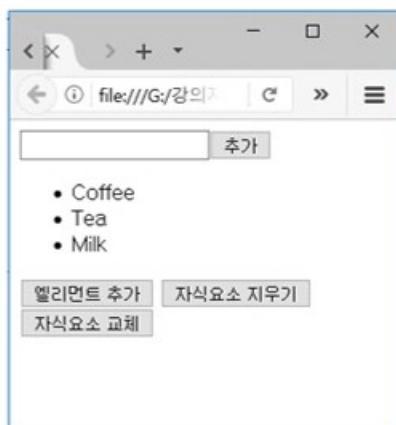
    /* 테스트
    var myList = document.getElementById("myList");
    var list = document.createElement("li");
    var t = document.createTextNode("CLICK ME");
    list.appendChild(t);
    myList.appendChild(list);
    */

});

//요소 지우기
document.getElementById("btn2").addEventListener("click",function(){
    var list = document.getElementById("myList");
    console.log(list);
    console.log(list.childNodes);
    list.removeChild(list.childNodes[1]);
});

//요소 변경 하기
document.getElementById("btn3").addEventListener("click",function(){
    var textnode = document.createTextNode("Water");
    var item = document.getElementById("myList").childNodes[1];
    item.replaceChild(textnode, item.childNodes[0]);
});

</script>
</html>
```



01-10. BOM의 추가 기능

1. Cookie 생성

Cookie - 인터넷 웹 페이지의 저장소, 웹페이지의 사용 흔적

헨젤과 그레텔이 지나온 길을 표시하기 위해

쿠키 조각을 떨어뜨리며 표시했다는 이야기에서 따온 것

- 쿠키값은 생성하거나 읽을 수 있다.
- 쿠키값의 수명은 만드는 순간 지정 할 수 있다.
- 쿠키 값은 사용자의 클라이언트에 저장된다.
- 사용자의 로그인 여부 또는 이벤트 수행 여부등을 판별 할 때 사용 한다.

Cookie 생성

```
document.cookie= "쿠키에 입력 할 값; expires=쿠키 만료 일시; path=저장경로";  
document.cookie="username=John Doe; expires=Thu, 18 Dec 2013 12:00:00 UTC; path=/";
```

예제 1-59)

```
<html>  
  <head>  
    <meta charset="utf-8">  
    <title>쿠키 예제 01</title>  
  </head>  
  <body>  
    ID: <input id="userId" type="text"/>  
    <br/>  
    PW: <input id="userPass" type="text"/>  
    <br/>  
    <button id="setCookie">쿠키 생성</button>  
    <button id="getCookie">쿠키 확인</button>  
    <button id="delCookie">쿠키 삭제</button>  
  </body>  
  <script>  
    //요소 가져오기  
    var id = document.getElementById("userId");  
    var pw = document.getElementById("userPass");  
    var setC = document.getElementById("setCookie");  
    var getCookie = document.getElementById("getCookie");  
    var delC = document.getElementById("delCookie");  
    //쿠키를 담을 변수  
    var coo;  
    //쿠키 만들기  
    setC.addEventListener("click",function(){  
      //var userId = id.getAttribute("valule");
```

```

var userId = id.value;
var userPass = pw.value;
console.log(userId+"/"+userPass);
//쿠키 만들기
//document.cookie= "쿠키에 입력 할 값; expires=쿠키 만료 일시;";
//만료 날짜 만들기
var expire = new Date() ;//1. 날짜 객체 만들기
var cDay = 1 ;//2. 만료 날짜 지정

//expire.setDate(expire.getDate()+cDay);
//expire.setTime(expire.getTime()+(5*60*1000));

coo = document.cookie="id="+userId+"; pw="+userPass+"; expires="+expire.toGMTString()+"";
console.log(coo);

});

//쿠키 사용하기
getC.addEventListener("click",function(){
    //쿠키 값을 ; 으로 나눈 값을 넣은 배열
    var cookieArr = coo.split(";");
    //나눠진 값을 다시 = 나누어진 값을 넣은 배열
    var cValue=[];

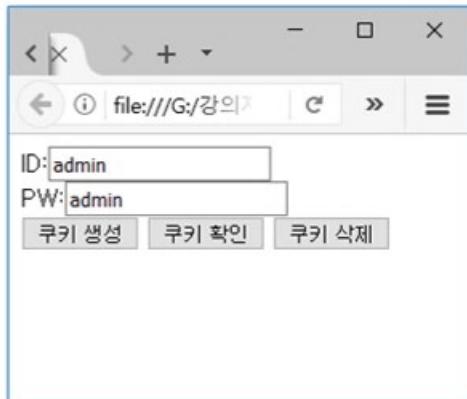
    for(var i=0;i<cookieArr.length;i++){
        //쿠키의 개별 항목 값을 = 단위로 나눈다.
        var val = cookieArr[i].split("=");
        //항목과 값 중 값만 배열에 넣는다.
        cValue.push(val[1]);
    }
    //불필요한 마지막 공백 값을 삭제 한다.
    cValue.pop();
    console.log(cValue);

    if(cValue[0]=="admin" && cValue[1]=="admin"){
        alert("관리자 로그인 되었습니다.");
    }else{
        alert("관리자 아이디가 아닙니다.");
    }
});

```

```
</script>
```

```
</html>
```



예제 1- 60) 쿠키를 읽어 오는 방식

```
<!DOCTYPE html>
<html>
    <body>
        <script>
            var c;
            var regId;
            var regPass;

            function setCookie() {
                var id = document.getElementById("regId").value;
                var pw = document.getElementById("regPass").value;
                var expire = new Date();
                var cDay = 1;

                expire.setDate(expire.getDate() + cDay);
                //expire.setTime(expire.getTime() + (cDay*24*60*60*1000));
                c =
                document.cookie="id="+id+";pw="+pw+";expires="+expire.toGMTString()+";";
                var cArray = c.split(' ');
                for(var i = 0; i<cArray.length; i++){
                    if(cArray[i] != ""){
                        var val = cArray[i].split('=');
                        console.log(val);
                        if(val[0] == "id"){
                            regId = val[1];
                        }
                        if(val[0] == "pw"){

```

```

                regPass = val[1];
            }
        }
    }

    console.log(regId+"/"+regPass);
    regTable = document.getElementById("regist");
    loginTable = document.getElementById("login");
    regTable.style.display="none";
    loginTable.style.display="block";
}

function login(){
    var id = document.getElementById("userId").value;
    var pw = document.getElementById("usePass").value;

    console.log(regId+"/"+regPass);
    console.log(id+"/"+pw);

    if(id==regId && pw==regPass){
        alert("로그인 완료")
    }else{
        alert("아이디 또는 패스워드를 확인하세요!!")
    }
}

</script>
<table id="regist" style="display:block;">
    <tr>
        <td>ID</td>
        <td><input id="regId" type="text"></input></td>
    </tr>
    <tr>
        <td>PW</td>
        <td><input id="regPass" type="text"></input></td>
    </tr>
    <tr>
        <td><button onclick="setCookie()">회원가입</button></td>
    </tr>
</table>
<table id="login" style="display:none;">
    <tr>

```

```

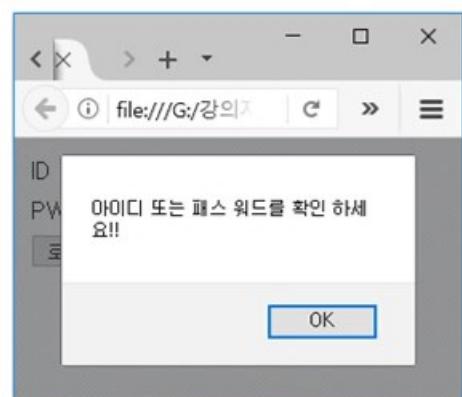
        <td>ID</td>
        <td><input id="userId" type="text"></input></td>
    </tr>
    <tr>
        <td>PW</td>
        <td><input id="usePass" type="text"></input></td>
    </tr>
    <tr>
        <td><button onclick="logIn()">로그인</button></td>
    </tr>
</table>
</body>
</html>

```

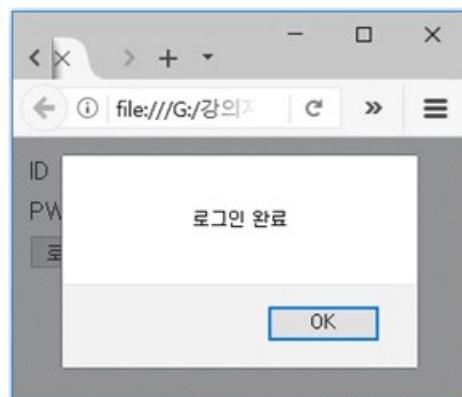
ID: highland0
PW: 123456
회원가입

ID: PW: 로그인

ID: gfusgfugwsf
PW: usvfusufusw
로그인



ID: highland0
PW: 123456
로그인



2. alert, confirm, prompt

alert - 가장 자주 쓰는 경고창

confirm - 액션 실행전 확인(취소 버튼이 존재)

prompt - 유저에게 입력을 받는 창

예제 1-61) alert, confirm, prompt 예제

```
<!DOCTYPE html>
<html>
<body>

<button onclick="popup()">alert</button>
<button onclick="conf()">confirm</button>
<button onclick="prompt()">prompt</button>

<script>
function popup() {
    alert("경고창입니다.");
}

function conf() {
    var con = confirm("정말로 삭제 하시겠습니까?");
    if(con == true){
        alert("삭제 완료 되었습니다.");
    }
}

function prompt() {
    var name = prompt("당신의 이름은?");

    if(name=="" || name==null || name == " "){
        alert("이름을 입력 해 주세요!");
    }else{
        alert(name+"님 반갑습니다.");
    }
}
</script>

</body>
</html>
```

3. setInterval(), clearInterval(), setTimeout()

setInterval() - 특정 시간간격으로 특정 함수를 계속해서 실행

clearInterval() - setInterval 특성을 삭제

setTimeout() - 특정 시간이 지난 후 특정 함수를 실행

1,000 = 1seconds

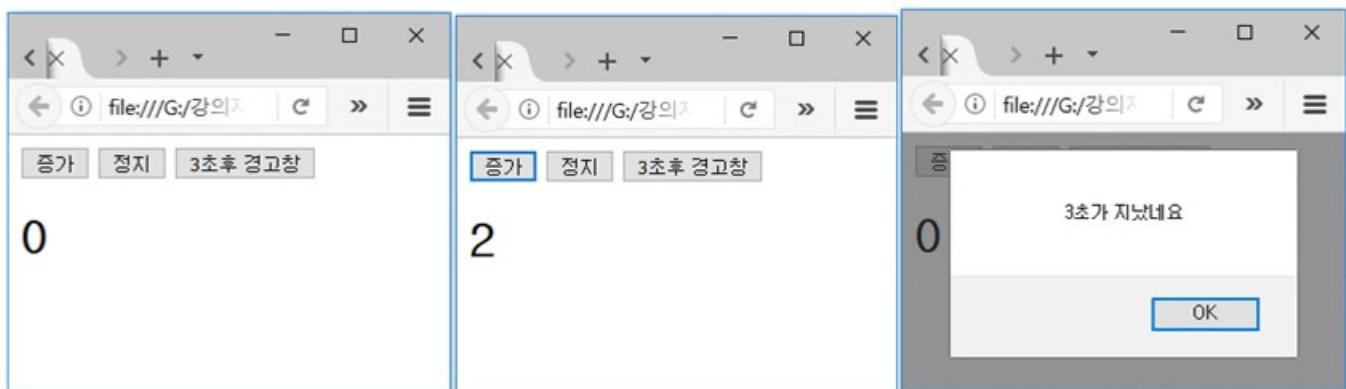
예제 1-62) setInterval, clearInterval, setTimeout에 대한 예제.

```
<!DOCTYPE html>
<html>
    <body>
        <button onclick="interval()">증가</button>
        <button onclick="stop()">정지</button>
        <button onclick="setTime()">3초후 경고창</button>
        <h1 id="msg">0</h1>
        <script>
            var myTime;
            var x = 0;
            var msg = document.getElementById("msg");

            function interval(){
                myTime = setInterval(function(){
                    x++;
                    msg.innerHTML = x;
                }, 1000);
            }

            function stop(){
                clearInterval(myTime);
                x = 0;
                msg.innerHTML = x;
            }

            function setTime(){
                setTimeout(function(){
                    alert("3초가 지났네요");
                }, 3000);
            }
        </script>
    </body>
</html>
```



예제 1-63)

```
<html>
    <head>
        <meta charset="utf-8">
        <title>java scr ipt</title>
        <style>
            #obj{
                width:30px;
                height:30px;
                position: absolute;
                background-color: cadetblue;
                top:50px;
                left: 0px;
            }
        </style>
    </head>
    <body>
        <button onclick="left()">◀</button>
        <button onclick="right()">▶</button>
        <button onclick="stop()">■</button>
        <div id="obj"></div>
    </body>
    <script>

        var obj = document.getElementById("obj");
        var btn = document.getElementsByTagName("button");
        var x=0;
        var move;

        function right(){
            var i=0;

```

```
move = setInterval(function() {
    if(i<=100) {
        i++;
        x = x+1;
        btn[0].disabled=true;
        btn[1].disabled=true;
    }else{
        stop();
        i=0;
    }
    obj.style.left = x;
},10);
}

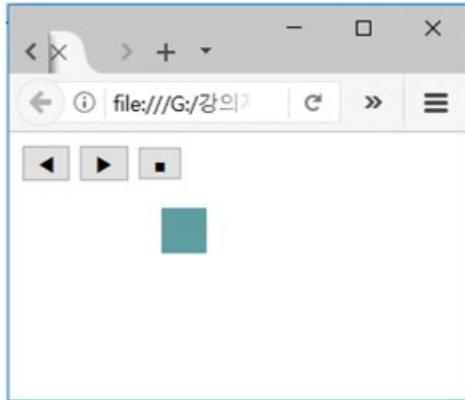
function left(){
var i=0;

if(x<=0){
    alert("이동 할 수 없습니다.");
}else{
    move = setInterval(function() {
        if(i<=100){
            i++;
            x = x-1;
            btn[0].disabled=true;
            btn[1].disabled=true;
        }else{
            stop();
            i=0;
        }
        obj.style.left=x;
    },10);
}
}

function stop(){
clearInterval(move);
btn[0].disabled=false;
btn[1].disabled=false;
}
```

```
</script>
```

```
</html>
```



예제 1-64)

```
<!DOCTYPE html>
<html>
    <head>
        <title>타이밍 예제</title>
        <style>
            #count{
                color: red;
            }
        </style>
    </head>
    <body>
        location.href = "http://www.google.com";
        <h1><span id="count">10</span> 초 후 페이지가 이동 됩니다.</h1>
        <script>
            var cnt = 10;
            var msg = document.getElementById("count");

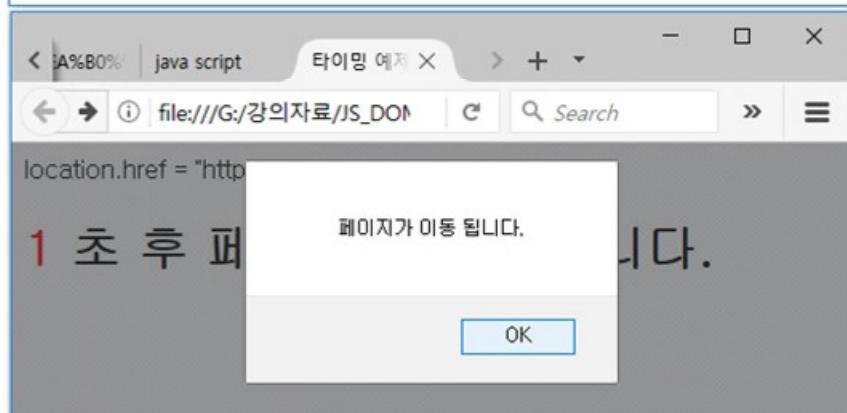
            setInterval(function(){
                cnt--;
                msg.innerHTML = cnt;
            }, 1000);

            setTimeout(function(){
                alert("페이지가 이동 됩니다.");
                location.href = "http://www.google.com";
            }, 10000);

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

```
</body>
```

```
</html>
```



01-11. JSON

JavaScript Object Notation

자바스크립트 객체 표기법

- > 데이터가 가볍다.
- > 그래서 전송이 빠르다.
- > 모든 프로그래밍 언어에서 사용 가능 하다.
- > 보안이 강조되지 않는 데이터의 경우 Restful 을 이용한 데이터 전송 시 JSON을 주로 사용 한다.

1. JSON이란

JSON : JavaScript Object Notation

영어 그대로 해석하자면 자바스크립트를 객체형식으로 표현하는 것입니다.

JSON은 XML보다 빠르고 변환하기도 쉬워 많이 대체해서 쓰고 있습니다.

적은양의 데이터가 송수신 되므로 모바일쪽에서 쓰기에도 부담이 적습니다.

- > 데이터가 가볍다.
- > 그래서 전송이 빠르다.
- > 모든 프로그래밍 언어에서 사용 가능 하다.
- > 보안이 강조되지 않는 데이터의 경우 Restful 을 이용한 데이터 전송 시 JSON을 주로 사용 한다.

2. JSON 표기법

JSON의 표기법은 다음과 같습니다.

JSON 데이터는 key/value형식으로 구성되어 있습니다.

"firstName" : "김 "

알아보기 쉽게 자바스크립트 문장으로 바꾸면 다음과 같습니다.

firstName = "김 "

이중에서 value에는 다음과 같은 형식을 넣을 수 있습니다.

- a number(integer or float)
- a string(큰 따옴표 포함)
- a boolean(true or false)
- an object(중괄호 포함)
- an array(대괄호 포함)
- null

[a number]

integer형이나 float형을 넣을 수 있습니다.

"clickNumber" : 1

[a string]

string형을 표현하고 싶을 때는 큰 따옴표로 묶어줍니다.

"name" : "홍길동"

[a boolean]

true 나 false를 넣을 수 있습니다.

"bResult" : true

[an object]

여러개의 key/value를 입력하여 중괄호로 둘러싸여집니다.

{"firstName" : "길동", "lastName" : "홍"}

[an array]

여러개의 object를 대괄호를 통해 둘러싸여집니다.

```
{  
  "employees" : [  
    {"firstName" : "길동", "lastName" : "홍"},  
    {"firstName" : "철수", "lastName" : "김"},  
    {"firstName" : "영순", "lastName" : "박"}  
  ]  
}  
  
var firstName = employees[0].firstname;  
var lastName = employees[0].lastname
```

JSON은 자바스크립트 문법을 사용하기 때문에, JSON으로 작업할 때 별도의 소프트웨어가 필요없습니다.

자바스크립트를 사용해서 다음과 같이 생성할 수 있습니다.

```
var employees = [  
  {"firstName" : "길동", "lastName" : "홍"},  
  {"firstName" : "철수", "lastName" : "김"},  
  {"firstName" : "영순", "lastName" : "박"}  
];
```

1-65) JSON 예제

```
<html>  
  <header></header>  
  <body>  
    Name: <span id="jname"></span><br />  
    Age: <span id="jage"></span><br />  
    Address: <span id="jstreet"></span><br />  
    Phone: <span id="jphone"></span><br />  
  </body>  
  
<script type="text/javascript">  
  var employees = {
```

```

    "name": "홍길동",
    "street": "서울시 서초구 서초동",
    "age": 33,
    "phone": "010-1234-5678"} ;
document.getElementById("jname").innerHTML=employees.name
document.getElementById("jage").innerHTML=employees.age
document.getElementById("jstreet").innerHTML=employees.street
document.getElementById("jphone").innerHTML=employees.phone
</script>
</html>
```

JSON 텍스트의 MIME 타입은 “application/json”입니다.

예제 1-66)

```

<html>
  <head>
    <meta charset="utf-8"/>
    <title>제이슨 만들기</title>
  </head>
  <body>
    <textarea id="jText" cols="40" rows="10">
    {
      "employee": [
        {"firstName": "Jhone", "lastName": "Doe"},
        {"firstName": "Anna", "lastName": "Smith"},
        {"firstName": "Peter", "lastName": "Jhones"}
      ]
    }
    </textarea>
    <!--http://json.parser.online.fr-->
    <br>
    <button id="jWrite">입력</button>
    <br>
    <input id="eName" type="text" value="" />
    번째 사원 이름
    <button id="jRead">추출</button>
    <br>
    <h2 id="result"></h2>
  </body>
  <script src="jsonData.js"></script>
</html>
```

예제 1-67)

```
<!doctype html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>JSON</title>
</head>
<body>
    오브젝트 : employee<br/>
    firstName : <input id="fn" type="text"/><br/>
    lastName : <input id="ln" type="text"/><br/>
    <button id="add">리스트 추가</button>
    <p id="result"></p>

</body>
<script>

var addBtn = document.getElementById("add");
//최상위 오브젝트
var obj={};

//employee 속성에 대한 값 배열
var person = [];
var jsonObj;

addBtn.addEventListener("click",addVal);

function addVal(){

    //하위 오브젝트
    var info = {};

    var first = document.getElementById("fn").value;
    var last = document.getElementById("ln").value;

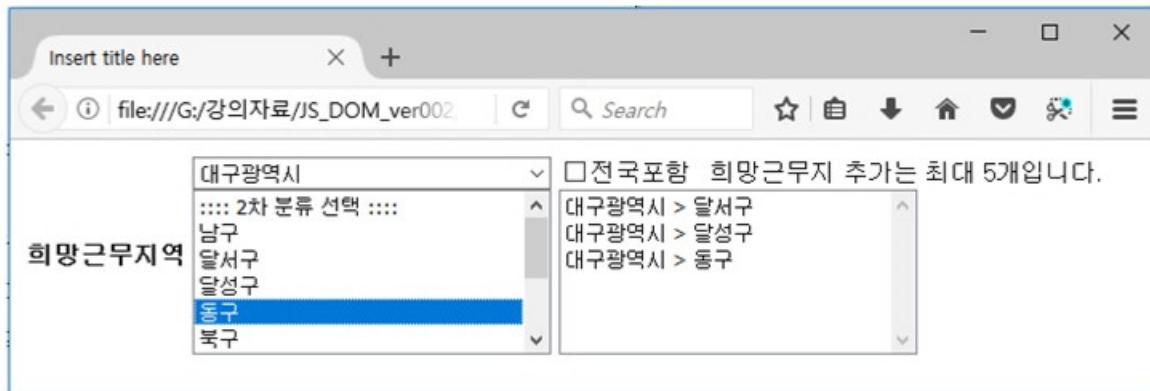
    info.firstName = first;
    info.lastName = last;
    person.push(info);

    obj.employee = person;
    console.log(obj);
    jsonObj = JSON.stringify(obj);

}
```

```
        document.getElementById("result").innerHTML = jsonObj ;
    }
</script>
</html>
```

추가 예제 1)



```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<script src="http://code.jquery.com/jquery-1.5.js"></script>
<script>
model.info = new Array(
    new Array(
        new Array("::::: 2차 분류 선택 ::::: ,-1 )
    ),
    new Array(
        new Array("::::: 2차 분류 선택 ::::: ,-1 ),
        new Array("강남구", 1),
        new Array("강동구",2),
        new Array("강서구", 3),
        new Array("관악구", 4),
        new Array("광진구", 5),
        new Array("구로구", 6),
        new Array("금천구", 7),
        new Array("노원구", 8),
        new Array("도봉구", 9),
        new Array("동대문구", 10),
        new Array("동작구", 11),
        new Array("마포구", 12),
        new Array("서대문구",13),
        new Array("서초구", 14),
        new Array("성동구", 15),
        new Array("성북구", 16),
        new Array("송파구", 17),
        new Array("양천구", 18),
    )
);
```

```
        new Array("영등포구", 19),
        new Array("용산구", 20),
        new Array("은평구", 21),
        new Array("종로구", 22),
        new Array("중구", 23),
        new Array("중랑구", 24),
        new Array("서울전지역", 25)
    ),
    new Array(
        new Array("::::: 2차 분류 선택 :::::", -1),
        new Array("광산구", 1),
        new Array("남구", 2),
        new Array("동구", 3),
        new Array("북구", 4),
        new Array("서구", 5),
        new Array("광주전지역", 6)
    ),
    new Array(
        new Array("::::: 2차 분류 선택 :::::", -1),
        new Array("남구", 1),
        new Array("달서구", 2),
        new Array("달성구", 3),
        new Array("동구", 5),
        new Array("북구", 4),
        new Array("서구", 6),
        new Array("수성구", 11),
        new Array("중구", 7),
        new Array("대구전지역", 8)
    ),
    new Array(
        new Array("::::: 2차 분류 선택 :::::", -1),
        new Array("대덕구", 1),
        new Array("동구", 2),
        new Array("서구", 3),
        new Array("유성구", 4),
        new Array("중구", 5),
        new Array("대전전지역", 6)
    )
);
```

```
new Array(
    new Array("::::: 2차 분류 선택 ::::: ,-1 ),
    new Array("강서구", 1),
    new Array("금정구", 2),
    new Array("기장군", 3),
    new Array("남구", 4),
    new Array("동구", 5),
    new Array("동래구", 6),
    new Array("부산진구", 7),
    new Array("북구", 8),
    new Array("사상구", 2),
    new Array("사하구", 3),
    new Array("서구", 4),
    new Array("수영구", 5),
    new Array("연제구", 6),
    new Array("영도구", 7),
    new Array("중구", 8),
    new Array("해운대구", 7),
    new Array("부산전지역", 8)
),

new Array(
    new Array("::::: 2차 분류 선택 ::::: ,-1 ),
    new Array("남구", 1),
    new Array("동구", 2),
    new Array("북구", 3),
    new Array("율진군", 4),
    new Array("중구", 5),
    new Array("울산전지역", 6)
),

new Array(
    new Array("::::: 2차 분류 선택 ::::: ,-1 ),
    new Array("강화군", 1),
    new Array("계양구", 2),
    new Array("남구", 3),
    new Array("남동구", 4),
    new Array("동구", 5),
    new Array("부평구", 6),
    new Array("서구", 7),
    new Array("연수구", 8),
    new Array("옹진군", 9),
)
```

```
    new Array("중구", 10),
    new Array("인천전지역", 11)
),

new Array(
    new Array("::::: 2차 분류 선택 ::::", -1),
    new Array("가평군", 1),
    new Array("고양시 덕양구", 2),
    new Array("고양시 일산동구", 3),
    new Array("고양시 일산서구", 4),
    new Array("과천시", 5),
    new Array("광명시", 6),
    new Array("광주시", 7),
    new Array("구리시", 8),
    new Array("군포시", 9),
    new Array("김포시", 10),
    new Array("남양주시", 11),
    new Array("동두천시", 12),
    new Array("부천시 소사구", 13),
    new Array("부천시 원미구", 14),
    new Array("부천시 오정구", 15),
    new Array("성남시 분당구", 16),
    new Array("성남시 수정구", 17),
    new Array("성남시 중원구", 18),
    new Array("수원시 권선구", 19),
    new Array("수원시 장안구", 20),
    new Array("수원시 팔달구", 21),
    new Array("수원시 영통구", 22),
    new Array("시흥시", 23),
    new Array("안산시 단월구", 24),
    new Array("안산시 상록구", 25),
    new Array("안성시", 26),
    new Array("안양시 동안구", 27),
    new Array("안양시 만안구", 28),
    new Array("양주시", 29),
    new Array("양평군", 30),
    new Array("여주군", 31),
    new Array("연천군", 32),
    new Array("오산시", 33),
    new Array("용인시 기흥구", 34),
    new Array("용인시 수지구", 35),
    new Array("용인시 처인구", 36),
```

```
        new Array("의왕시", 37),
        new Array("의정부시", 38),
        new Array("이천시", 39),
        new Array("파주시", 40),
        new Array("평택시", 41),
        new Array("포천시", 42),
        new Array("하남시", 43),
        new Array("화성시", 44),
        new Array("경기도전지역", 45)
    ),

    new Array(
        new Array("::::: 2차 분류 선택 :::::", -1),
        new Array("강릉시", 1),
        new Array("고성군", 2),
        new Array("동해시", 3),
        new Array("삼척시", 4),
        new Array("속초시", 5),
        new Array("양구군", 6),
        new Array("양양군", 7),
        new Array("영월군", 8),
        new Array("원주시", 9),
        new Array("인제군", 10),
        new Array("정선군", 11),
        new Array("철원군", 12),
        new Array("춘천시", 13),
        new Array("태백시", 14),
        new Array("평창군", 15),
        new Array("홍천군", 16),
        new Array("화천군", 17),
        new Array("횡성군", 18),
        new Array("강원전지역", 19)
    ),

    new Array(
        new Array("::::: 2차 분류 선택 :::::", -1),
        new Array("거제시", 1),
        new Array("거창군", 2),
        new Array("고성군", 3),
        new Array("김해시", 4),
        new Array("남해군", 5),
        new Array("밀양시", 6),
```

```
    new Array("사천시", 7),
    new Array("산청군", 8),
    new Array("양산시", 9),
    new Array("의령군", 10),
    new Array("진주시", 11),
    new Array("창녕군", 12),
    new Array("창원시 마산함포구", 13),
    new Array("창원시 마산회원구", 14),
    new Array("창원시 성산구", 15),
    new Array("창원시 의창구", 16),
    new Array("창원시 진해구", 17),
    new Array("통영시", 18),
    new Array("하동군", 19),
    new Array("함안군", 20),
    new Array("함양군", 21),
    new Array("함천군", 22),
    new Array("경남전지역", 23)
),

new Array(
    new Array("::::: 2차 분류 선택 :::::", -1),
    new Array("경산시", 1),
    new Array("경주시", 2),
    new Array("고령군", 3),
    new Array("구미시", 4),
    new Array("군위군", 5),
    new Array("김천시", 6),
    new Array("문경시", 7),
    new Array("봉화군", 8),
    new Array("상주시", 9),
    new Array("성주군", 10),
    new Array("양동시", 11),
    new Array("영덕군", 12),
    new Array("영양군", 13),
    new Array("영주시", 14),
    new Array("영천시", 15),
    new Array("예천군", 16),
    new Array("울릉군", 17),
    new Array("울진군", 18),
    new Array("의성군", 19),
    new Array("청도군", 20),
    new Array("청송군", 21),
```

```
        new Array("칠곡군", 22),
        new Array("포항시 남구", 23),
        new Array("포항시 북구", 24),
        new Array("경북전지역", 25)
    ),
    new Array(
        new Array("::::: 2차 분류 선택 :::::", -1),
        new Array("강진군", 1),
        new Array("고흥군", 2),
        new Array("곡성군", 3),
        new Array("광양시", 4),
        new Array("구례군", 5),
        new Array("나주시", 6),
        new Array("담양군", 7),
        new Array("목포시", 8),
        new Array("무안군", 9),
        new Array("보성군", 10),
        new Array("순천시", 11),
        new Array("신안군", 12),
        new Array("여수시", 13),
        new Array("영광군", 14),
        new Array("영암군", 15),
        new Array("완도군", 16),
        new Array("장성군", 17),
        new Array("장흥군", 18),
        new Array("진도군", 19),
        new Array("함평군", 20),
        new Array("해남군", 21),
        new Array("화순군", 22),
        new Array("전남전지역", 23)
    ),
    new Array(
        new Array("::::: 2차 분류 선택 :::::", -1),
        new Array("고창군", 1),
        new Array("군산시", 2),
        new Array("김제시", 3),
        new Array("남원시", 4),
        new Array("무주군", 5),
        new Array("부안군", 6),
        new Array("순창군", 7),
```

```
    new Array("완주군", 8),
    new Array("익산시", 9),
    new Array("임실군", 10),
    new Array("장수군", 11),
    new Array("전주시 덕진군", 12),
    new Array("전주시 완산구", 13),
    new Array("정읍시", 14),
    new Array("진안군", 15),
    new Array("전북전지역", 16)
),

new Array(
    new Array("::::: 2차 분류 선택 :::::", -1),
    new Array("공주시", 1),
    new Array("금산군", 2),
    new Array("논산시", 3),
    new Array("당진시", 4),
    new Array("보령시", 5),
    new Array("부여군", 6),
    new Array("서산시", 7),
    new Array("서천군", 8),
    new Array("아산시", 9),
    new Array("연기군", 10),
    new Array("예산군", 11),
    new Array("천안시 동남구", 12),
    new Array("천안시 서북구", 13),
    new Array("청양군", 14),
    new Array("태안군", 15),
    new Array("홍성군", 16),
    new Array("계룡시", 17),
    new Array("충남전지역", 18)
),

new Array(
    new Array("::::: 2차 분류 선택 :::::", -1),
    new Array("괴산군", 1),
    new Array("단양군", 2),
    new Array("보은군", 3),
    new Array("영동군", 4),
    new Array("옥천군", 5),
    new Array("음성군", 6),
    new Array("제천시", 7),
```

```
        new Array("진천군", 8),
        new Array("청원군", 9),
        new Array("청주시 상당구", 10),
        new Array("청주시 흥덕구", 11),
        new Array("충주시", 12),
        new Array("증평군", 13),
        new Array("충북전지역", 14)
    ),
    new Array(
        new Array("::::: 2차 분류 선택 :::::", -1),
        new Array("서귀포시", 1),
        new Array("제주시", 2),
        new Array("제주전지역", 3)
    ),
    new Array(
        new Array("::::: 2차 분류 선택 :::::", -1),
        new Array("개성공단전지역", 1)
    ),
    new Array(
        new Array("::::: 2차 분류 선택 :::::", -1),
        new Array("금강산특구전지역", 1)
    ),
    new Array(
        new Array("::::: 2차 분류 선택 :::::", -1),
        new Array("전국", 1)
    ),
    new Array(
        new Array("::::: 2차 분류 선택 :::::", -1)
    ),
    new Array(
        new Array("::::: 2차 분류 선택 :::::", -1),
        new Array("북경", 1),
        new Array("상해", 2),
        new Array("중경", 3),
        new Array("홍콩", 4),
        new Array("마카오", 5),
        new Array("감숙성", 6),
        new Array("강서성", 7),
        new Array("길림성", 8),
        new Array("북건성", 9),
        new Array("사천성", 10),
        new Array("산동성", 11),
        new Array("제주도", 12)
    )
);
```

```
        new Array("산서성", 12),
        new Array("섬서성", 13),
        new Array("안휘성", 14),
        new Array("요녕성", 15),
        new Array("운남성", 16),
        new Array("절강성", 17),
        new Array("청해성", 18),
        new Array("하남성", 19),
        new Array("하북성", 20),
        new Array("해남성", 21),
        new Array("호남성", 22),
        new Array("호북성", 23),
        new Array("흑룡강성", 24),
        new Array("기타", 25),
        new Array("중국전지역", 26)
    ),
    new Array(
        new Array("::::: 2차 분류 선택 ::::", -1),
        new Array("보스턴", 1),
        new Array("필라델피아", 2),
        new Array("뉴욕", 3),
        new Array("시카고", 4),
        new Array("로스앤젤레스", 5),
        new Array("워싱턴", 6),
        new Array("네바다주", 7),
        new Array("네브라스타주", 8),
        new Array("노스다코타주", 9),
        new Array("노스캐롤라이나주", 10),
        new Array("뉴멕시코주", 11),
        new Array("뉴욕주", 12),
        new Array("뉴저지주", 13),
        new Array("뉴햄프셔주", 14),
        new Array("델라웨어주", 15),
        new Array("로드아일랜드주", 16),
        new Array("루지애나주", 17),
        new Array("메릴랜드주", 18),
        new Array("메사추세츠주", 19),
        new Array("메인주", 20),
        new Array("몬타나주", 21),
        new Array("미네소타주", 22),
        new Array("미시건주", 23),
        new Array("미시시피주", 24),
```

```
    new Array("미주리주", 25),
    new Array("버먼트주", 26),
    new Array("버지니아주", 27),
    new Array("웨스트버지니아주", 28),
    new Array("사우스다코타주", 29),
    new Array("사우스캐롤라이나주", 30),
    new Array("아이다호주", 31),
    new Array("아이오와주", 32),
    new Array("알래스카주", 33),
    new Array("알칸사주", 34),
    new Array("애리조나주", 35),
    new Array("앨라배마주", 36),
    new Array("오레건주", 37),
    new Array("오클라호마주", 38),
    new Array("오하이오주", 39),
    new Array("와이오밍주", 40),
    new Array("워싱턴주", 41),
    new Array("위스콘신주", 42),
    new Array("유타주", 43),
    new Array("인디애나주", 44),
    new Array("일리노이주", 45),
    new Array("조지아주", 46),
    new Array("캔사스주", 47),
    new Array("캘리포니아주", 48),
    new Array("켄터키주", 49),
    new Array("코네티컷주", 50),
    new Array("콜로라도주", 51),
    new Array("테네시주", 52),
    new Array("텍사스주", 53),
    new Array("팬실베니아주", 54),
    new Array("플로리다주", 55),
    new Array("하와이주", 56),
    new Array("미국전지역", 57)
),
new Array(
    new Array("::::: 2차 분류 선택 :::::", -1),
    new Array("도쿄", 1),
    new Array("오오사카", 2),
    new Array("쿄토", 3),
    new Array("가가와", 4),
    new Array("가고시마", 5),
    new Array("가나가와", 6),

```

```
new Array("고오찌", 7),
new Array("구마모또", 8),
new Array("군마", 9),
new Array("기후", 10),
new Array("나가노", 11),
new Array("나가사끼", 12),
new Array("나라", 13),
new Array("니이까따", 14),
new Array("나라", 15),
new Array("니이가따", 16),
new Array("도꾸시마", 17),
new Array("도야마", 18),
new Array("도찌끼", 19),
new Array("돗도리", 20),
new Array("미야끼", 21),
new Array("미야자끼", 22),
new Array("미에", 23),
new Array("사가", 24),
new Array("사이따마", 25),
new Array("시가", 26),
new Array("시마네", 27),
new Array("시즈오까", 28),
new Array("아끼따", 29),
new Array("아오모리", 30),
new Array("아이지", 31),
new Array("야마까따", 32),
new Array("야마구찌", 33),
new Array("야마나시", 34),
new Array("에히메", 35),
new Array("오끼야마", 36),
new Array("오끼나와", 37),
new Array("오오이따", 38),
new Array("와까야마", 39),
new Array("이바라끼", 40),
new Array("이시까와", 41),
new Array("이와떼", 42),
new Array("지바", 43),
new Array("홋카이도", 44),
new Array("효고", 45),
new Array("후꾸시마", 46),
new Array("후꾸오까", 47),
new Array("후꾸미", 48),
```

```
new Array("히로시마", 49),
new Array("일본전지역", 50)
),

new Array(
    new Array("::::: 2차 분류 선택 ::::", -1),
    new Array("대만", 1),
    new Array("말레이시아", 2),
    new Array("몽골", 3),
    new Array("미얀마", 4),
    new Array("방글라데시", 5),
    new Array("베트남", 6),
    new Array("사우디아라비아", 7),
    new Array("스리랑카", 8),
    new Array("싱가포르", 9),
    new Array("인도", 10),
    new Array("인도네시아", 11),
    new Array("캄보디아", 12),
    new Array("태국", 13),
    new Array("필리핀", 14),
    new Array("네팔", 15),
    new Array("몽골모르", 16),
    new Array("라오스", 17),
    new Array("레바논", 18),
    new Array("몰디브", 19),
    new Array("바레인", 20),
    new Array("부탄", 21),
    new Array("브루나이", 22),
    new Array("시리아", 23),
    new Array("아랍에미리트연합국", 24),
    new Array("아프가니스탄", 25),
    new Array("예멘", 26),
    new Array("오만", 27),
    new Array("요르단", 28),
    new Array("이라크", 29),
    new Array("이란", 30),
    new Array("이스라엘", 31),
    new Array("카타르", 32),
    new Array("쿠웨이트", 33),
    new Array("파키스탄", 34),
    new Array("그루지아", 35),
    new Array("우즈베키스탄", 36),
```

```
        new Array("카자흐스탄", 37),
        new Array("카르기즈스탄", 38),
        new Array("카지키스탄", 39),
        new Array("카르기즈스탄", 40),
        new Array("타지키스탄", 41),
        new Array("투르크메니스탄", 42),
        new Array("기타", 43),
        new Array("아시아중동전지역", 44)
    ),
    new Array(
        new Array("::::: 2차 분류 선택 :::::", -1),
        new Array("멕시코", 1),
        new Array("캐나다", 2),
        new Array("그레나다", 3),
        new Array("그린란드", 4),
        new Array("도미니카공화국", 5),
        new Array("바베이도스", 6),
        new Array("바하마", 7),
        new Array("세인트루시아", 8),
        new Array("세인트빈센트그레나딘", 9),
        new Array("세인트카츠네비스", 10),
        new Array("코스타리카", 11),
        new Array("푸에르토리코", 12),
        new Array("기타", 13),
        new Array("북아메리카전지역", 14)
    ),
    new Array(
        new Array("::::: 2차 분류 선택 :::::", -1),
        new Array("과테말라", 1),
        new Array("베네수엘라", 2),
        new Array("브라질", 3),
        new Array("아르헨티나", 4),
        new Array("에콰도르", 5),
        new Array("온두라스", 6),
        new Array("우루과이", 7),
        new Array("칠레", 8),
        new Array("콜롬비아", 9),
        new Array("쿠바", 10),
        new Array("파나마", 11),
        new Array("페루", 12),
        new Array("가이아나", 13),
        new Array("아프리카전지역", 14)
    )
);
```

```
    new Array("니카라과", 14),
    new Array("도미니카연방", 15),
    new Array("벨리즈", 16),
    new Array("볼리비아", 17),
    new Array("수리남", 18),
    new Array("아이티", 19),
    new Array("앤티가바부다", 20),
    new Array("엘살바도르", 21),
    new Array("자메이카", 22),
    new Array("트리니다토바고", 23),
    new Array("파라과이", 24),
    new Array("기타", 25),
    new Array("남아프리카전지역", 26)
),

new Array(
    new Array("::::: 2차 분류 선택 ::::: ",-1),
    new Array("그리스", 1),
    new Array("네델란드", 2),
    new Array("노르웨이", 3),
    new Array("덴마크", 4),
    new Array("독일", 5),
    new Array("러시아", 6),
    new Array("벨기에", 7),
    new Array("스웨덴", 8),
    new Array("스위스", 9),
    new Array("스페인", 10),
    new Array("영국", 11),
    new Array("이탈리아", 12),
    new Array("체코", 13),
    new Array("터키", 14),
    new Array("포루투칼", 15),
    new Array("폴란드", 16),
    new Array("프랑스", 17),
    new Array("핀란드", 18),
    new Array("헝가리", 19),
    new Array("네팔", 20),
    new Array("루마니아", 21),
    new Array("룩셈부르크", 22),
    new Array("리투아니아", 23),
    new Array("리히텐슈타이", 24),
    new Array("마케도니아", 25),
```

```
    new Array("모나코", 26),
    new Array("몰도바", 27),
    new Array("몰타", 28),
    new Array("바티칸시국", 29),
    new Array("벨라루스", 30),
    new Array("보스니아헤르체고비나", 31),
    new Array("불가리스", 32),
    new Array("사이포러스", 33),
    new Array("산마리노", 34),
    new Array("세르비아몬테네그로", 35),
    new Array("슬로베니아", 36),
    new Array("아르메니아", 37),
    new Array("아이슬란드", 38),
    new Array("아일랜드", 39),
    new Array("아제르바이잔", 40),
    new Array("안도라", 41),
    new Array("알바니아", 42),
    new Array("에스토니아", 43),
    new Array("오스트리아", 44),
    new Array("우크라이나", 41),
    new Array("크로아티아", 42),
    new Array("기타", 43),
    new Array("유럽전지역", 44)
),
new Array(
    new Array("::::: 2차 분류 선택 :::::", -1),
    new Array("괌", 1),
    new Array("뉴질랜드", 2),
    new Array("파푸아뉴기니", 3),
    new Array("피지", 4),
    new Array("호주", 5),
    new Array("나우루공화국", 6),
    new Array("마셜", 7),
    new Array("미크로네시아", 8),
    new Array("바누아투", 9),
    new Array("사모아", 10),
    new Array("슬로몬제도", 11),
    new Array("키리바시", 12),
    new Array("퉁가", 13),
    new Array("투발루", 14),
    new Array("팔라우", 15),
    new Array("기타", 16),
```

```
    new Array("오세아니아전지역", 17)
),
new Array(
    new Array(":::: 2차 분류 선택 ::::", -1),
    new Array("가나", 1),
    new Array("가봉", 2),
    new Array("나이지리아", 3),
    new Array("남아프리카공화국", 4),
    new Array("리비아", 5),
    new Array("모로코", 6),
    new Array("수단", 7),
    new Array("알제리", 8),
    new Array("에티오피아", 9),
    new Array("우간다", 10),
    new Array("이집트", 11),
    new Array("케냐", 12),
    new Array("탄자니아", 13),
    new Array("튀니지", 14),
    new Array("감비아", 15),
    new Array("기니", 16),
    new Array("기니비사우", 17),
    new Array("나미비아", 18),
    new Array("니제르", 19),
    new Array("라이베리아", 20),
    new Array("레소토", 21),
    new Array("르완다", 22),
    new Array("마다가스카르", 23),
    new Array("말라위", 24),
    new Array("말리", 25),
    new Array("모리셔스", 26),
    new Array("모리타니아", 27),
    new Array("모잠비크", 28),
    new Array("베냉", 29),
    new Array("보츠와나", 30),
    new Array("부룬디", 31),
    new Array("부르키나파소", 32),
    new Array("상투메프린시페", 33),
    new Array("세네갈", 34),
    new Array("세이셸", 35),
    new Array("소말리아", 36),
    new Array("스와질란드", 37),
    new Array("시에라리온", 38),
```

```

        new Array("앙골라", 39),
        new Array("에르트레아", 40),
        new Array("잠비아", 41),
        new Array("적도기니", 42),
        new Array("중앙아프리카공화국", 43),
        new Array("지부티", 44),
        new Array("짐바브웨", 45),
        new Array("차드", 46),
        new Array("카메룬", 47),
        new Array("카보베르데", 48),
        new Array("코모로", 49),
        new Array("코트디부아르", 50),
        new Array("콩고", 46),
        new Array("콩고민주공화국", 47),
        new Array("토고", 48),
        new Array("기타", 49),
        new Array("아프리카전지역", 50)
    )
);

```

```

function fillSelectFromArray1(selectCtrl, itemArray, goodPrompt, badPrompt, defaultItem) {
    var i, j;
    var prompt;
    // empty existing items
    var WORK_DISTRICT_NAME1 = document.frm.WORK_DISTRICT_NAME1.value;
    for (i = selectCtrl.options.length; i >= 0; i--) {
        selectCtrl.options[i] = null;
    }
    prompt = (itemArray != null) ? goodPrompt : badPrompt;
    if (prompt == null) {
        j = 0;
    } else {
        selectCtrl.options[0] = new Option(prompt);
        j = 1;
    }
    if (itemArray != null) {
        // add new items
        for (i = 0; i < itemArray.length; i++) {
            selectCtrl.options[j] = new Option(itemArray[i][0]);
            if (itemArray[i][1] != null) {
                selectCtrl.options[j].value = WORK_DISTRICT_NAME1 +

```

```

    " " +itemArray[i][0];
    }
    j++;
}
// select first item (prompt) for sub list
selectCtrl.options[0].selected = true;
}

}

function test1(){
//alert(selIndex);
//text1=$("#WORK_DISTRICT_NAME2").val().replace(/\s/g, ""); //앞,뒤,중간 모든 공백 없앰
text1=$("#WORK_DISTRICT_NAME2").val();
//alert(text1);
text11 = text1.split(' ');
//alert(text11[0]);
//alert(text11[1]);
if(text11[1]==":::: 2차 분류 선택 ::::"){
    return false;
}
var form = document.frm;
selIndex = form.select_box2.length;
if(selIndex <= 4){
    for(i = 0 ; i<selIndex; i++){
        //alert($("#select_box2 option:eq("+i+")").val());
        if($("#select_box2 option:eq("+i+")").val() == text1 ){
            return false;
        }
    }
    if(text1.length>0){
        $("#select_box2").append("<option value='"+text1+"'" +
data-image='../images/mypage/bt_de12.gif' >" +text11[0] + " > " + text11[1]+ "</option>");
    }
} else{
    alert("희망근무지역은 최대 5개까지 선택가능합니다.");
}
}

function del1() {
    var selIndex = $("#select_box2 option").index($("#select_box2 option:selected")); //선택된 값의
index
    $("#select_box2 option:eq("+selIndex+")").remove();
}

```

```

</script>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
<title>Insert title here</title>
</head>
<body>
<form name="frm">
    <table>
        <tr>
            <th>희망근무지역</th>
            <td>
                <select style="width: 230px;" id="svSelect"
n a m e = " W O R K _ D I S T R I C T _ N A M E 1 "
onChange="fillSelectFromArray1(this.form.WORK_DISTRICT_NAME2,((this.selectedIndex == -1) ? null :
model.info[this.selectedIndex]));">
                    <option value = -1>-- 1차 분류 선택
                    <!--</option>
                    <option value = 서울특별시>서울특별
시</option>
                    <option value = 광주광역시>광주광역
시</option>
                    <option value = 대구광역시>대구광역
시</option>
                    <option value = 대전광역시>대전광역
시</option>
                    <option value = 부산광역시>부산광역
시</option>
                    <option value = 울산광역시>울산광역
시</option>
                    <option value = 인천광역시>인천광역시
</option>
                    <option value = 경기도>경기도
</option>
                    <option value = 강원도>강원도
</option>
                    <option value = 경상남도>경상남도
</option>
                    <option value = 경상북도>경상북도
</option>
                    <option value = 전라남도>전라남도
</option>
                    <option value = 전라북도>전라북도
</option>

```

```

</option>
<option value = "충청남도" >충청남도
</option>
<option value = "충청북도" >충청북도
</option>
<option value = "제주특별자치도" >제주
특별자치도</option>
<option value = "개성공업지구" >개성공
업지구</option>
<option value = "금광산관광특구" >금광
산관광특구</option>
<option value = "전국" >전국</option>
<option value = "중국·홍콩" >중국·홍
콩</option>
<option value = "미국" >미국</option>
<option value = "일본" >일본</option>
<option value = "아시아·중동" >아시아·
중동</option>
<option value = "북아메리카" >북아메리
카</option>
<option value = "남아메리카" >남아메리
카</option>
<option value = "유럽" >유럽</option>
<option value = "오세아니아" >오세아니
아</option>
<option value = "아프리카" >아프리카
</option>
</select>
<input type="checkbox"
name="WORK_DISTRICT_NAME" value="전국"/><label>전국포함</label>&nbsp;
< s p a n > < i m g
src="../../images/person_info/p_blt04.gif" alt="" />희망근무지 추가는 최대 5개입니다.<span><br />
<select style="width:230px;" class="f_left"
name="WORK_DISTRICT_NAME2" id="WORK_DISTRICT_NAME2" size="6" onChange="test1()" >
</select>
<select style="width:230px;" 
class="bodcon_list" size = "6" name= "select_box2" id="select_box2" onChange="del1()"></select>

```

```

        </td>
    </tr>
</table>
</form>
</body>
</html>

```

추가 예제 2)

```

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<script>
//개인회원 구직정보에서 가족 추가
function family_add(){
    var Row = fam_tb.insertRow();
    Row.onmouseover=function(){fam_tb.clickedRowIndex=this.rowIndex};

    var Cell1 = Row.insertCell();
    Cell1.align = "left";//cell의 align를 사용할 수 있습니다. (left, center, right)

    Cell1.innerHTML = "<input type=text name=FAMILY_RELATION maxlength=10 class=input_res style=width:50px;>";

    var Cell2 = Row.insertCell();
    Cell2.align = "left";//cell의 align를 사용할 수 있습니다. (left, center, right)

    Cell2.innerHTML = "<input type=text name=FAMILY_NAME maxlength=10 class=input_res style=width:100px;>";

    var Cell3 = Row.insertCell();
    Cell3.align = "left";//cell의 align를 사용할 수 있습니다. (left, center, right)

    Cell3.innerHTML = "<input type=text name=FAMILY_AGE maxlength=3 class=input_res style=width:50px;>";

    var Cell4 = Row.insertCell();
    Cell4.align = "left";//cell의 align를 사용할 수 있습니다. (left, center, right)

    Cell4.innerHTML = "<input type=text name=FAMILY_JOB maxlength=20 class=input_res style=width:100px;>";

    var Cell5 = Row.insertCell();
}

```

```

Cell15.align = "left";//cell의 align를 사용할 수 있습니다. (left, center, right)

Cell15.innerHTML = "<select name=LIVE_TOGETHER_CHECK><option value=Y selected>동거</option><option value=N>비동거</option></select>";

var Cell16 = Row.insertCell();
Cell16.align = "left";//cell의 align를 사용할 수 있습니다. (left, center, right)

Cell16.innerHTML = "<img src=../../images/mypage/bt_del.gif / onclick=rowdel() ;>";
}

//개인회원 구직정보에서 가족 삭제
function rowdel()
{
    fam_tb.deleteRow(fam_tb.clickedRowIndex);
}
</script>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=EUC-KR">
<title>Insert title here</title>
</head>
<body>

    <table width="580" border="0" cellspacing="0" cellpadding="0" class="family">
        <tr>
            <td width="65" class="ttl">관계</td>
            <td width="110" class="ttl">이름</td>
            <td width="65" class="ttl">나이</td>
            <td width="110" class="ttl">직업</td>
            <td class="ttl">동거 여부</td>
            <td width="80" class="ttl">가족없음
                <input type="checkbox" name="nofam" onclick="no_fam() ;" />
            </td>
        </tr>
        </table>
        <table width="580" border="0" cellspacing="0" cellpadding="0" id="fam_tb" class="family">
            <tr>
                <td width="65"><input type="text" name="FAMILY_RELATION" maxlength="10" class="input_res" style="width:50px;"></td>
                <td width="110"><input type="text" name="FAMILY_NAME" maxlength="10" class="input_res" style="width:100px;"></td>
            </tr>
        </table>
    </div>

```

```

        <td width="65"><input type="text" name="FAMILY_AGE" maxlength="3"
class="input_res" style="width:50px;"></td>
        <td width="110"><input type="text" name="FAMILY_JOB"
maxlength="20" class="input_res" style="width:100px;"></td>
        <td>
            <select name="LIVE_TOGETHER_CHECK">
                <option value="Y">동거</option>
                <option value="N">비동거</option>
            </select>
        </td>
        <td width="80"></td>
    </tr>

</table>

<div id="fam_add" style="display: ;">
    가족사항 추가시 [추가]버튼을 눌러주세요. <input type = button
align="absmiddle" border="0" onclick="family_add(); value= "추가" />
</div>

```

</body>

</html>

관계	이름	나이	직업	동거여부	가족없음
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="동거"/> <input type="button" value="동거"/>	<input type="checkbox"/>

가족사항 추가시 [추가]버튼을 눌러주세요.

추가