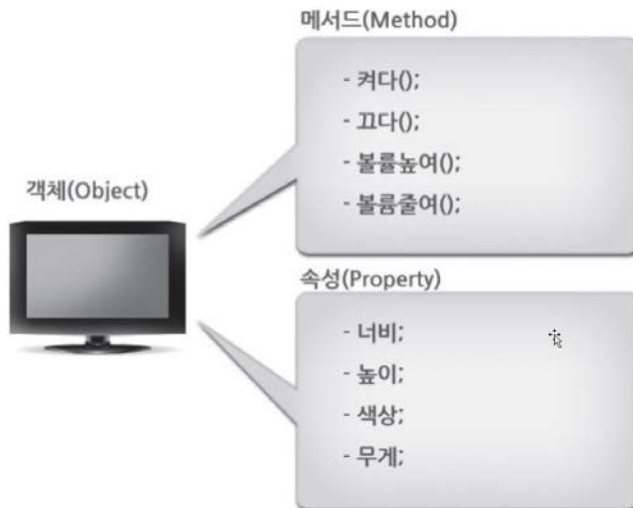


## 4장 객체

### 4.1 객체

#### 4.1.1 객체란?

- 자바스크립트는 객체(Object) 기반 언어이다. 객체는 기능 또는 속성으로 구성되어 있다.



```
//기본형
객체.메서드();
객체.속성 또는 객체.속성=값;

TV.켜다();
TV.끄다();
TV.너비 = "30in";
TV.색상 = "black";
```

#### 4.1.2 객체의 종류

##### (1) 내장 객체

- 자바스크립트 자체 내에서 지원하는 객체를 가리킨다. 내장 객체에는 문자(String), 날짜(Date), 수학(Math), 배열(Array) 객체 등으로 구성되어 있다.

##### (2) 브라우저 객체 모델(BOM)

- 브라우저에 계층적으로 내장되어 있는 객체들을 말한다. 브라우저 객체로는 windows, screen, location, history, navigator 객체 등이 있다.

##### (3) 문서 객체 모델(DOM)

- HTML 문서 구조를 말한다. HTML 문서의 기본 구조는 최상위 객체로 <html>이 있으며, 그 하위 객체로는 <head>와 <body>가 있다.

- 자바스크립트의 문석 객체 모델은 IE8 이하 버전에서 호환성이 떨어지기 때문에 사용이 힘들다는 단점이 있다. 그래서 최근에는 제이쿼리 문서 객체 모델을 많이 사용한다.

## 4.2 내장 객체

- 내장 객체(Built-in Object)는 브라우저의 자바스크립트 엔진에 내장된 객체를 말한다.

### 4.2.1 내장 객체 생성하기

- 객체를 생성할 때에는 new라는 키워드와 생성함수를 사용한다.

```
//기본형  
참조 변수 = new 생성함수();
```

[실습파일: ob1\_test.html, 완성파일: ob1.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="ko" xml:lang="ko">  
<head>  
<meta http-equiv="content-type" content="text/html; charset=utf-8" />  
<title>내장객체</title>  
<script type="text/javascript">  
    //<br/>        var tv = new Object(); //생성된 객체를 변수 tv가 참조<br/><br/>        //tv 객체의 속성 생성함.<br/>        tv.width = "30cm";<br/>        tv.height = "25cm";<br/>        tv.weight = "20kg";<br/>        tv.color = "white";<br/><br/>        //tv 객체의 off 기능(메서드)을 생성함.<br/>        tv.off = function() {<br/>            document.write("전원 off", "&lt;br /&gt;");<br/>        }<br/><br/>        document.write(tv.width, "&lt;br /&gt;"); //tv에 너비 속성을 출력합니다.<br/>        document.write(tv.height, "&lt;br /&gt;"); //tv에 높이 속성을 출력합니다.<br/>        document.write(tv.weight, "&lt;br /&gt;"); //tv에 무게 속성을 출력합니다.<br/>        document.write(tv.color, "&lt;br /&gt;"); //tv에 컬러 속성을 출력합니다.<br/><br/>        tv.off(); //tv에 off 기능을 실행합니다.<br/>    //]]&gt;<br/>&lt;/script&gt;<br/>&lt;/head&gt;<br/>&lt;body&gt;<br/><br/>&lt;/body&gt;<br/>&lt;/html&gt;</pre></div><div data-bbox="115 827 323 845" data-label="Section-Header"><h3>4.2.2 날짜 정보 객체</h3></div><div data-bbox="117 860 864 892" data-label="List-Group"><ul><li>■ 날짜 객체는 현재 날짜 또는 특정 날짜 관련 다양한 정보를 제공한다. 가령, 2020.10.20이 무슨 요일인지 정보를 알 수 있다.</li></ul></div>
```

```
//기본형
참조 변수 = new Date();

var t = new Date();
var t = new Date("2002/5/30");
var t = new Date(2002,4,30);
```

- 날짜 정보 객체에는 날짜의 정보를 가져오는 메서드와 날짜를 수정하는 메서드로 나누어 진다.

날짜 정보를 가져올 때 (GET)		날짜 정보를 수정할 때 (SET)	
getFullYear()	연도 정보를 가져옴	setFullYear()	연도 정보만 수정
getMonth()	월 정보를 가져옴(현재 월-1)	setMonth()	월 정보만 수정
getDate()	일 정보를 가져옴	setDate()	일 정보만 수정
getDay()	요일 정보를 가져옴(일:0~토:6)	요일은 날짜를 바꾸면 자동으로 바뀌므로 setDay는 없음	
getHours()	시 정보를 가져옴	setHours()	시 정보만 수정
getMinutes()	분 정보를 가져옴	setMinutes()	분 정보만 수정
getSeconds()	초 정보를 가져옴	setSeconds()	초 정보만 수정
getMilliseconds()	밀리초 정보를 가져옴(1/1000초 단위)	setMilliseconds()	밀리초 정보만 수정
getTime()	1970년 1월 1일부터 경과된 시간을 밀리초로 표기함.	setTime()	1970년 1월 1일부터 경과된 시간을 밀리초로 수정
toGMTString()	GMT 표준 표기 방식으로 문자형 데이터로 반환함.	toLocaleString()	운영 시스템 표기 방식으로 문자형 데이터로 반환함.

[실습파일: date\_ob1\_test.html, 완성파일: date\_ob1.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="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>날짜 정보 객체</title>
<script type="text/javascript">
    //
    var t = new Date();
    var nowMonth = t.getMonth(); // 현재 월을 반환하며 1적게 나옵니다.
    var nowDate = t.getDate(); // 현재 일자를 반환합니다.
    var nowDay = t.getDay(); // 현재 요일을 반환합니다. (0:일~6:토)
    document.write("현재 월:" + nowMonth, "&lt;br /&gt;");
    document.write("현재 일:" + nowDate, "&lt;br /&gt;");
    document.write("현재 요일:" + nowDay, "&lt;br /&gt;");

    var m = new Date(2002, 4, 31); // 2002(5.31)월드컵 날짜객체 생성

    var theMonth = m.getMonth(); //날짜 객체에서 월에 대한 정보만 가져옵니다.
    var theDate = m.getDate(); //날짜 객체에서 일에 대한 정보만 가져옵니다.
    var theDay = m.getDay(); //날짜 객체에서 요일에 대한 정보만 가져옵니다.

    document.write("2002월드컵 몇 월:" + theMonth, "&lt;br /&gt;");
    document.write("2002월드컵 몇 일:" + theDate, "&lt;br /&gt;");
    document.write("2002월드컵 무슨 요일:" + theDay, "&lt;br /&gt;");
    //]]&gt;
&lt;/script&gt;
&lt;/head&gt;
&lt;body&gt;

&lt;/body&gt;
&lt;/html&gt;</pre>
</div>
<div data-bbox="125 867 500 881" data-label="Text">
<p>[실습파일: date_ob2_test.html, 완성파일: date_ob2.html]</p>
</div>
<div data-bbox="125 892 825 906" data-label="Text">
<pre>&lt;!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-</pre>
</div>
```

```

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>날짜 정보 객체</title>
<script type="text/javascript">
    <![CDATA[
        var t = new Date(); //오늘 날짜 객체를 생성합니다.
        var nowYear = t.getFullYear(); //오늘 날짜의 연도 정보를 가져옵니다.

        //현재 연도에 12월 31 날짜 객체를 생성합니다.
        var theDate = new Date(nowYear, 11, 31);

        var diffDate = theDate - t; //연말까지 남은 기간 = 연말 날짜 - 현재 날짜

        //밀리초를 일로 개산합니다.
        var result = Math.ceil(diffDate / (60 * 1000 * 60 * 24));

        //남은 일수를 출력합니다.
        document.write("현재일로 부터 올해 말일까지 " + result + "일 남았습니다.");
    ]]>
</script>
</head>
<body>

</body>
</html>

```

### 4.2.3 숫자 객체

- 숫자를 표기할 때, 표현 가능한 수의 속성과 숫자 표기법에 대한 메서드를 제공한다.

```

//기본형
var 참조변수 = new Number(값) 또는 var 참조변수 = 값;
Number.속성;

```

[실행파일: number\_ob1\_test.html, 완성파일: number\_ob1.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="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>숫자 객체</title>
<script type="text/javascript">
    <![CDATA[
        var num1 = 3.456789;
        var num2 = 7000000;
        var num3 = "30.5px";
        var num4 = 40;

        document.write("표현 가능한 가장 큰 수:" + Number.MAX_VALUE, "<br />");
        document.write("표현 가능한 가장 작은 수:" + Number.MIN_VALUE, "<br />");
        document.write("숫자가 아닌경우의 표기:" + Number.NaN, "<br />");
        document.write("무한대 수 표기:" + Number.POSITIVE_INFINITY, "<br />");
        document.write("음의 무한대 수 표기:" + Number.NEGATIVE_INFINITY, "<br />");

        document.write(num2.toExponential(1), "<br />");
        document.write(num1.toFixed(2), "<br />");
        document.write(num1.toPrecision(2), "<br />");
        document.write(num1.toString(), "<br />");
        document.write(num4.valueOf(), "<br />");
    ]]>
</script>
</head>
<body>

</body>
</html>

```

```

        document.write(parseInt(num3) + num4, "<br />");
        document.write(parseFloat(num3) + num4, "<br />");
    //]]>
</script>
</head>

<body>

</body>
</html>

```

## [꿀팁] new 키워드와 생성함수 사용하기

- 날짜 객체를 제외한 모든 객체는 변수에 값만 참조하여도 객체가 생성된다.

구분	객체 생성할 때 생성자 함수가 반드시 필요한 객체	객체 생성할 때 생성자 함수가 필요 없는 객체
객체 종류	날짜 객체 (Date Object)	수학 객체(Math Object) 숫자 객체(Number Object) 문자 객체(String Object) 배열 객체(Array Object) 정규 표현 객체(RegExp Object)

## 4.2.4 수학 객체

- 앞서 수를 더하거나 빼는 작업은 산술 연산자를 사용하였다. 하지만 최대값, 반올림 등 수학 관련 기능을 사용하기 위해서는 수학객체를 사용해야 한다.
- 다음 표는 수학 객체의 메서드와 참조하는 상수에 대해 정리한 것이다.

종류	설명
Math.abs(숫자)	숫자의 절대값을 반환한다.
Math.max(숫자1, 숫자2, 숫자3, 숫자4)	숫자 중 가장 큰 값을 반환한다.
Math.min(숫자1, 숫자2, 숫자3, 숫자4)	숫자 중 가장 작은 값을 반환한다.
Math.pow(숫자, 제곱값)	숫자의 거듭제곱한 값을 반환한다.
Math.random()	0~1 사이에 난수를 반환한다.
Math.round(숫자)	소수점 첫째 자리에서 반올림하여 정수를 반환한다.
Math.ceil(숫자)	소수점 첫째 자리에서 무조건 올림해서 정수를 반환한다.
Math.floor(숫자)	소수점 첫째 자리에서 무조건 내림해서 정수를 반환한다.
Math.sqrt(숫자)	숫자의 제곱근 값을 반환한다.
Math.PI	원주율 상수를 반환한다.

[실습파일: math\_ob1\_test.html, 완성파일: math\_ob1.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="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>수학객체</title>
<script type="text/javascript">
    //
        var num = 2.1234;

        var maxNum = Math.max(10, 5, 8, 30); // 최댓값 반환합니다.
        var minNum = Math.min(10, 5, 8, 30); // 최솟값 반환합니다.
        var roundNum = Math.round(num); // 반올림 값을 반환합니다.
        var floorNum = Math.floor(num); // 소수점 무조건 내립니다.
        var ceilNum = Math.ceil(num); // 소수점 무조건 올립니다.
        var rndNum = Math.random(); // 0~1 사이 난수를 발생립니다.
    //]]&gt;
</pre>
</div>
```

```

var piNum = Math.PI; //Math.PI는 원주율 상수를 반환합니다.

document.write(maxNum, "<br />");
document.write(minNum, "<br />");
document.write(roundNum, "<br />");
document.write(floorNum, "<br />");
document.write(ceilNum, "<br />");
document.write(rndNum, "<br />");
document.write(piNum, "<br />");
//]]>
</script>
</head>

<body>

</body>
</html>

```

[실습파일: math\_ob2\_test.html, 완성파일: math\_ob2.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="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>수학객체</title>
<script type="text/javascript">
    <![CDATA[
        document.write("<h1>컴퓨터 가위, 바위, 보 맞추기</h1>");

        var t = prompt("가위, 바위, 보 중 선택하세요?", "가위");
        var tNum;
        switch (t) { //가위, 바위, 보를 숫자 1,2,3으로 바꿉니다.
            case "가위":
                tNum = 1;
                break;
            case "바위":
                tNum = 2;
                break;
            case "보":
                tNum = 3;
                break;
            default:
                alert("잘못 작성하셨습니다.");
                location.reload();
        }
        //1부터 3까지 난수(정수)를 발생시킵니다.
        var com = Math.ceil(Math.random() * 3); //컴퓨터 가위(1), 바위(2), 보(3)

        /*난수를 이용해 컴퓨터 가위 바위 보 이미지가 출력되도록 하였습니다.*/
        document.write("<img src='images/math_img_"+com+".jpg' alt='가위,바위,보' />");

        //사용자가 낸 값과 컴퓨터의 값이 일치하나 조건검사합니다.
        if (tNum == com) {
            alert("맞췄습니다! 축하합니다!!");
        } else {
            alert("틀렸네요! 다음기회 또 도전하세요!!");
        }
    </script>
</head>
<body>
</body>
</html>

```

## 4.2.5 배열 객체

- 앞서 변수에는 한 개의 데이터만 저장할 수 있고, 만일 변수에 새 값이 저장되면 기존에 저장된 값은 제거되었다. 하지만 이번에 배열 객체를 이용하면 여러 개의 데이터를 한 변수에 저장할 수 있다.

```
//기본형
var 참조변수 = new Array();
var 참조변수 = new Array(값1,값2,값3,...,값n);
var 참조변수 = [값1,값2,값3,...,값n];

// 배열 객체 저장된 데이터 불러오기
참조변수[인덱스 번호];
```

[실습파일: array\_ob1\_test.html, 완성파일: array\_ob1.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="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>배열객체</title>
<script type="text/javascript">
    <![CDATA[
        var d = [ 30, "홍길동", true ];
        document.write(d[0], "<br />");
        document.write(d[1], "<br />");
        document.write(d[2], "<br />");

        //for문을 적용함. 3은 배열 객체에 저장된 데이터 개수.
        for (var i = 0; i < 3; i++) {
            document.write(d[i], "<br />");
        }

        /*배열의 length(저장된 데이터의 개수) 속성을 이용해 반복문의 조건식에 적용함.*/
        for (var i = 0; i < d.length; i++) {
            document.write(d[i], "<br />");
        }

        /*for in문을 적용함. 변수 i의 값은 0부터 배열 객체의 데이터 개수만큼 증가됨.*/
        for (i in d) {
            document.write(d[i], "<br />");
        }
    </![CDATA[
</script>
</head>

<body>

</body>
</html>
```

- 다음은 배열 객체에서 사용할 수 있는 메서드 및 속성이다.

종류	설명
join(연결 문자)	배열 객체에 데이터를 연결 문자 기준으로 1개의 문자형 데이터로 반환한다.
reverse()	배열 객체에 데이터의 순서를 거꾸로 바꾼 후 반환한다.
sort()	배열 객체에 데이터를 오름차순으로 정렬한다.
slice(index1,index2)	배열 객체에 데이터 중 원하는 인덱스 구간만큼 잘라서 배열 객체로 가져온다.
splice()	배열 객체에 지정 데이터를 삭제하고 그 구간에 새 데이터를 삽입할 수 있다.
concat()	2개의 배열 객체를 하나로 결합시킨다.

pop()	배열에 저장된 데이터 중 마지막 인덱스에 저장된 데이터를 삭제한다.
push(new data)	배열 객체 마지막 인덱스에 새 데이터를 삽입한다.
shift()	배열 객체에 저장된 데이터 중 첫 번째 인덱스에 저장된 데이터를 삭제한다.
unshift(new data)	배열 객체의 가장 앞의 인덱스에 새 데이터를 삽입한다.
length()	배열에 저장된 총 데이터의 개수를 반환한다.

[실습파일: array\_ob2\_test.html, 완성파일: array\_ob2.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="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>배열객체</title>
<script type="text/javascript">
    <![CDATA[
        var num = [ "사당", "교대", "방배", "강남" ];
        //배열 객체를 출력하였습니다.
        document.write(num, "<br />");
        //배열 객체의 타입을 출력하였습니다.
        document.write(typeof num, "<br />");

        /*기준 문자("-") 기준으로 배열의 값들이 하나의 문자형
        데이터로 결합됩니다.*/
        document.write(num.join("-"), "<br />");
        document.write(typeof num.join("-"), "<br />");

        //배열 객체에 값의 순서를 거꾸로 뒤집고 출력합니다.
        document.write(num.reverse(), "<br />");

        //배열 객체에 값들을 오름차순으로 정렬 후 출력합니다.
        document.write(num.sort(), "<br />");
    ]]>
</script>
</head>

<body>

</body>
</html>
```

[실습파일: array\_ob3\_test.html, 완성파일: array\_ob3.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="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>배열객체</title>
<script type="text/javascript">
    <![CDATA[
        var greenLine = [ "사당", "교대", "방배", "강남" ];
        var yellowLine = [ "미금", "정자", "모란", "수서" ];

        greenLine.splice(2, 1, "서초", "역삼");
        document.write(greenLine, "<br />");

        document.write(greenLine.slice(1, 3), "<br />");
    ]]>
</script>
</head>
<body>
</body>
</html>
```



[실습파일: array\_ob4\_test.html, 완성파일: array\_ob4.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="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>배열객체</title>
<script type="text/javascript">
    //
        var greenLine = [ "사당", "교대", "방배", "강남" ];
        var yellowLine = [ "미금", "정자", "모란", "수서" ];

        var twoLine = greenLine.concat(yellowLine);
        document.write(twoLine, "&lt;br /&gt;");

        greenLine.pop();
        document.write(greenLine, "&lt;br /&gt;");

        greenLine.push("삼성");
        document.write(greenLine, "&lt;br /&gt;");

        greenLine.shift();
        document.write(greenLine, "&lt;br /&gt;");

        greenLine.unshift("신도림");
        document.write(greenLine, "&lt;br /&gt;");
    //]]&gt;
&lt;/script&gt;
&lt;/head&gt;
&lt;body&gt;
&lt;/body&gt;
&lt;/html&gt;</pre></div><div data-bbox="116 516 534 532" data-label="Section-Header"><h4>[꿀팁] 자바스크립트의 유용한 배열 메소스 사용하기</h4></div><div data-bbox="116 548 384 563" data-label="Text"><p>// <a href="http://bblog.tistory.com/300">http://bblog.tistory.com/300</a></p></div><div data-bbox="116 562 214 624" data-label="List-Group"><ol><li>1. map()</li><li>2. filter()</li><li>3. find()</li><li>4. reduce()</li></ol></div><div data-bbox="116 654 274 672" data-label="Section-Header"><h4>4.2.6 문자 객체</h4></div><div data-bbox="116 687 885 735" data-label="List-Group"><ul><li>■ 문자 객체(String Object)는 문자형 데이터를 객체로 취급하는 것으로, 자바스크립트에서 가장 많이 사용한다. 다음 기본형과 같이 new 키워드와 문자 객체 함수 String 메서드를 사용한다.</li></ul></div><div data-bbox="125 762 433 825" data-label="Text"><pre>//기본형
var 참조변수 = new String(문자형 데이터);

var t = new String("hello javascript");
var t = "hello javascript"; // 문자 객체 생성</pre></div><div data-bbox="116 850 357 867" data-label="List-Group"><ul><li>■ 문자 객체 메서드 및 속성</li></ul></div><div data-bbox="116 880 862 908" data-label="Table"><table><tr><th>종류</th><th>설명</th></tr><tr><td>charAt(index)</td><td>문자열에서 인덱스 번호에 해당하는 문자를 반환한다.</td></tr></table></div>
```

indexOf("찾을 문자")	문자열에서 왼쪽부터 찾을 문자와 일치하는 문자를 찾아 최초로 일치하는 문자의 인덱스 번호를 반환한다. 만일 찾는 문자가 없으면 -1을 반환한다.
lastIndexOf("찾을 문자")	문자열에서 오른쪽부터 찾을 문자와 일치하는 문자를 찾아 최초로 일치하는 문자의 인덱스 번호를 반환한다. 만일 찾는 문자가 없으면 -1을 반환한다.
match("찾을 문자")	문자열에서 왼쪽부터 찾을 문자와 일치하는 문자를 찾아 최초로 찾은 문자를 반환한다. 만일 찾는 문자가 없으면 null을 반환한다.
replace("바꿀 문자", "새 문자")	문자열에서 왼쪽부터 바꿀 문자와 일치하는 문자를 찾아 최초로 찾은 문자를 새 문자로 치환한다.
search("찾을 문자")	문자열에서 왼쪽부터 찾을 문자와 일치하는 문자를 찾아 최초로 일치하는 인덱스 번호를 반환한다.
slice(a,b)	a개의 문자를 자르고 b번째 이후에 문자를 자른 후 남은 문자를 반환한다. 즉 a~b 사이의 문자만 반환한다.
substring(a,b)	a 인덱스부터 b 인덱스 이전 구간의 문자를 반환한다. var str = "hello javascript"; str.substring(3,7); //"lo j"를 반환함.
substr(a,문자개수)	문자열에 a 인덱스부터 지정한 문자 개수만큼 문자열을 반환한다. var str = "hello javascript"; str.substr(3,2); //"lo"를 반환함
split("문자")	지정한 문자를 기준으로 문자 데이터를 나누어 배열에 저장하여 반환한다. var str = "web,mobile,2002"; var arr = str.split(","); //arr[0]="web", arr[1]="mobile", arr[2]="2002"
toLowerCase()	문자열에서 영문 대문자를 모두 소문자로 바꾼다.
toUpperCase()	문자열에서 영문 소문자를 모두 대문자로 바꾼다.
length	문자열에서 문자의 개수를 반환한다. var str = "hello welcome"; str.length; // 13을 반환한다.
concat("새로운 문자")	문자열에서 새로운 문자열을 결합한다.
charCodeAt("찾는 문자")	찾을 문자의 아스키 코드 값을 반환한다.
fromCharCode(아스키 코드 값)	아스키 코드 값이 해당하는 문자를 반환한다.
trim()	문자의 앞 또는 뒤에 공백 문자열을 삭제한다.

[실습파일: string\_ob1\_test.html, 완성파일: string\_ob1.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="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>문자 객체</title>
<script type="text/javascript">
    //
    /*변수 t에 문자형 데이터를 저장했는 것은 다시 말하면
    변수 t에 문자 객체를 참조시킨거나 같다.*/
    var t = "Hello Thank you good luck to you";
    document.write(t.charAt(16), "&lt;br /&gt;");
    document.write(t.indexOf("you"), "&lt;br /&gt;");
    document.write(t.indexOf("you", 16), "&lt;br /&gt;");
    document.write(t.lastIndexOf("you"), "&lt;br /&gt;");
    document.write(t.lastIndexOf("you", 25), "&lt;br /&gt;");
    document.write(t.match("luck"), "&lt;br /&gt;");
    document.write(t.search("you"), "&lt;br /&gt;");
    document.write(t.substr(21, 4), "&lt;br /&gt;");
    document.write(t.substring(6, 12), "&lt;br /&gt;");
    document.write(t.replace("you", "me"), "&lt;br /&gt;");
    document.write(t.toLowerCase(), "&lt;br /&gt;");
    document.write(t.toUpperCase(), "&lt;br /&gt;");
    document.write(t.length, "&lt;br /&gt;");

    var s = t.split(" ");
    document.write(s[0], "&lt;br /&gt;");
    document.write(s[4], "&lt;br /&gt;");

    document.write(String.fromCharCode("A"), "&lt;br /&gt;");
    document.write(String.fromCharCode(65), "&lt;br /&gt;");
    //]]&gt;
&lt;/script&gt;
&lt;/head&gt;
&lt;body&gt;</pre>
</div>
```

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

[실습파일: string\_ob2\_test.html, 완성파일: string\_ob2.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="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>문자 객체</title>
<script type="text/javascript">
    //
        var userName = prompt("당신의 영문 이름은?", "");

        var upperName = userName.toUpperCase(); //영문을 대문자로 바꿉니다.
        document.write(upperName, "&lt;br /&gt;");

        var userNum = prompt("당신의 연락처는?", "");
        var result = userNum.substring(0, userNum.length - 4) + "****";
        document.write(result, "&lt;br /&gt;");
    //]]&gt;
&lt;/script&gt;
&lt;/head&gt;
&lt;body&gt;
&lt;/body&gt;
&lt;/html&gt;</pre></div><div data-bbox="125 442 527 456" data-label="Text"><p>[실습파일: string_ob3_test.html, 완성파일: string_ob3.html]</p></div><div data-bbox="125 467 825 912" data-label="Text"><pre>&lt;!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd"&gt;
&lt;html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko"&gt;
&lt;head&gt;
&lt;meta http-equiv="content-type" content="text/html; charset=utf-8" /&gt;
&lt;title&gt;문자 객체&lt;/title&gt;
&lt;script type="text/javascript"&gt;
    //<![CDATA[
        var userEmail = prompt("당신의 이메일 주소는?", "");
        //이메일 주소에 뒷부분 형식을 배열로 저장하였습니다.
        var arrUrl = [ ".co.kr", ".com", ".net", ".or.kr", ".go.kr" ];

        var check1 = false; //초기 값 저장
        var check2 = false; //초기 값 저장

        /*방문자가 입력한 이메일 주소에 "@"포함되어 있으면 변수
        check1에 true가 저장되도록 하였습니다.*/
        if (userEmail.indexOf("@") &gt; 0)
            check1 = true;

        for (var i = 0; i &lt; arrUrl.length; i++) { //이메일에 배열 데이터 포함여부 검사
            if (userEmail.indexOf(arrUrl[i]) &gt; 0)
                check2 = true;
        }

        //AND(&amp;&amp;)연산자는 모두 피연산자가 모두 true여야 true를 반환합니다.
        if (check1 &amp;&amp; check2) {
            document.write(userEmail);
        } else {
            alert("이메일 형식이 잘못되었습니다.");
        }
    //]]&gt;
&lt;/script&gt;
&lt;/head&gt;
&lt;body&gt;</pre></div>
```

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

#### 4.2.7 정규 표현 객체

- 정규 표현(RegExp) 객체는 입력 요소에 데이터를 규칙에 맞게 작성했는지 판단해서 알려주는 객체이다.

```
//기본형
var 참조변수 = new RegExp(패턴, 검색 옵션);
var 참조변수 = /패턴/검색 옵션;
```

- 정규 표현 검색 옵션

종류	설명
*	0회 이상 일치하는 문자를 검사할 때 사용한다.
+	1회 이상 일치하는 문자를 검사할 때 사용한다.
i	찾는 문자가 영문일 경우 대.소문자를 구분하지 않는다.
?	0회 또는 1회 규칙에 맞는 문자가 있는지 검사한다.
g	옵션에 g가 없으면 데이터의 왼쪽부터 규칙에 일치하는 문자 한개만 찾지만 g를 넣으면 규칙에 맞는 모든 문자를 찾는다.
m	데이터의 행이 바뀌어도 규칙에 맞는 문자를 찾는다.
\$	문자열 끝부분에 규칙에 맞는 문자를 찾는다.
^	문자열 앞에서부터 규칙에 일치하는 문자를 찾는다. 또는 부정을 나타낸다.
\d	숫자 하나를 찾는다.
\D	숫자가 아닌 문자 하나를 찾는다.
\s	공백 문자(스페이스)를 찾는다.
[0-9][A-Z][a-z]	숫자, 대문자 영문, 소문자 영문을 찾는다.
\w	알파벳, 숫자, 밑줄(_) 기호를 찾는다.
\W	알파벳, 숫자, 밑줄(_) 기호를 제외한 문자를 찾는다.
a b	a 또는 b인지 검사할 때 사용한다.

- 정규 표현 메서드

종류	설명
test()	지정된 규칙에 맞는 단어가 있으면 true 값을 반환한다.
exec()	지정된 규칙에 맞는 단어가 있으면 해당 단어를 반환한다.

[실습파일: reg\_ob1\_test.html, 완성파일: reg\_ob1.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="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>정규표현 객체</title>
<script type="text/javascript">
    //<![CDATA[
    var str = "Html Css JQuery ";
    var reg1 = /css/; // var reg1=new RegExp("css")

    /*변수 str에 데이터가 reg1에 정규표현 형식을 잘
    지켰으면 true를 반환하고 안지켰을 경우엔 false를 반환합니다*/
    var result_1 = reg1.test(str);
    document.write(result_1, "<br />");

    /*옵션에 'i'를 입력하면 영문 대소문자를 구분하지 않습니다*/
    var reg2 = /css/i; // var reg2=new RegExp("css","i")
```

```

        var result_2 = reg2.test(str);
        document.write(result_2, "<br />");
    //]]>
</script>
</head>

<body>

</body>
</html>

```

[실습파일: reg\_ob2\_test.html, 완성파일: reg\_ob2.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="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>정규표현 객체</title>
<script type="text/javascript">
    //
        var userName = prompt("당신의 이름은?");
        reg1 = /^[가-힣]{2,5}$/;
        while (true) {
            if (reg1.test(userName))
                break;

            alert("이름 입력 형식이 잘못되었습니다!");
            userName = prompt("당신의 이름은?");
        }
        var userCell = prompt("당신의 핸드폰 번호는?");
        reg2 = /^(010|016|011)\d{3,4}\d{4}$/;

        while (true) {
            if (reg2.test(userCell))
                break;
            alert("핸드폰 입력 형식이 잘못되었습니다!");
            userCell = prompt("당신의 핸드폰 번호는?");
        }

        var userEmail = prompt("당신의 이메일은?");
        reg3 = /^[w]{5,12}@[a-z]{2,10}[\.][a-z]{2,3}[\.]?[a-z]{0,2}$/;

        while (true) {
            if (reg3.test(userEmail))
                break;
            alert("이메일 입력 형식이 잘못되었습니다!");
            userEmail = prompt("당신의 이메일은?");
        }

        document.write(userName, "&lt;br /&gt;");
        document.write(userCell, "&lt;br /&gt;");
        document.write(userEmail, "&lt;br /&gt;");
    //]]&gt;
&lt;/script&gt;
&lt;/head&gt;

&lt;body&gt;

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

## [실습] 현재 월(月)에 해당하는 달력 출력하기

1. [calendar\_test.html] 예제 파일을 열어 다음과 같이 입력한다.

[실습파일: calendar\_test.html, 완성파일: calendar.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="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>달력만들기</title>
<script type="text/javascript">
//
//① 9~17 현재 날짜와 현재 달에 1일의 날짜 객체를 생성합니다.
var date=new Date(); //날짜 객체 생성
var y=date.getFullYear(); //현재 연도
var m=date.getMonth(); //현재 월
var d=date.getDate(); //현재 일

/* 현재 월의 1일에 요일을 구합니다.
   그럼 그달 달력에 첫 번째 줄 빈칸의 개수를 구할 수 있습니다.*/
var theDate=new Date(y, m, 1);
var theDay=theDate.getDay();

//② 20~26 현재 월에 마지막 일을 구해야 합니다.

//1월부터 12월까지 마지막 일을 배열로 저장함.
var last=[31,28,31,30,31,30,31,31,30,31,30,31];
/*현재 연도가 윤년(4년 주기이고 100년 주기는 제외합니다.
또는 400년 주기)일경우 2월에 마지막 날짜는 29가 되어야 합니다.*/
if(y%4 &amp;&amp; y % 100 !=0 || y%400 == 0) lastDate=last[1]=29;

var lastDate=last[m]; //현재 월에 마지막이 몇일인지 구합니다.

/*③ 29~26 현재 월의 달력에 필요한 행의 개수를 구합니다.
var row(행의 개수)= Math.ceil( (theDay(빈 칸)+lastDate(월의 전체 일수)) / 7)*/

var row=Math.ceil((theDay+lastDate)/7); //필요한 행수

//④ 달력 년도/월 표기 및 달력 테이블 생성
document.write("&lt;h1&gt;"+y+"."+m+"&lt;/h1&gt;");
//문자결합 연산자를 사용해 요일이 나오는 행을 생성합니다.
var calendar="&lt;table border='1'&gt;";
calendar+="&lt;tr&gt;";
calendar+="&lt;th&gt;일&lt;/th&gt;";
calendar+="&lt;th&gt;월&lt;/th&gt;";
calendar+="&lt;th&gt;화&lt;/th&gt;";
calendar+="&lt;th&gt;수&lt;/th&gt;";
calendar+="&lt;th&gt;목&lt;/th&gt;";
calendar+="&lt;th&gt;금&lt;/th&gt;";
calendar+="&lt;th&gt;토&lt;/th&gt;";
calendar+="&lt;/tr&gt;";

var dNum=1; // 달력에 표기되는 일의 초기값
//앞에서 배우 이중 for문을 이용해 달력 테이블을 만들겠습니다.
for(var i=1; i&lt;=row; i++){//행 생성 (tr 태그 생성)
    calendar+="&lt;tr&gt;";

    for(var k=1; k&lt;=7; k++){//열 생성 (td 태그 생성)

        /*행이 첫 줄이고 현재 월의 1일의 요일 이전은 모두 빈열로
        표기하고 날짜가 마지막 일보다 크면 빈열로 표기됩니다.*/
        if(i==1 &amp;&amp; k&lt;=theDay || dNum&gt;lastDate){
            calendar+="&lt;td&gt; &amp;nbsp; &lt;/td&gt;";
        }else{</pre></div>
```

```

        calendar+="<td>"+dNum+"</td>";
        dNum++;
    }

}

calendar+="<tr>";
}

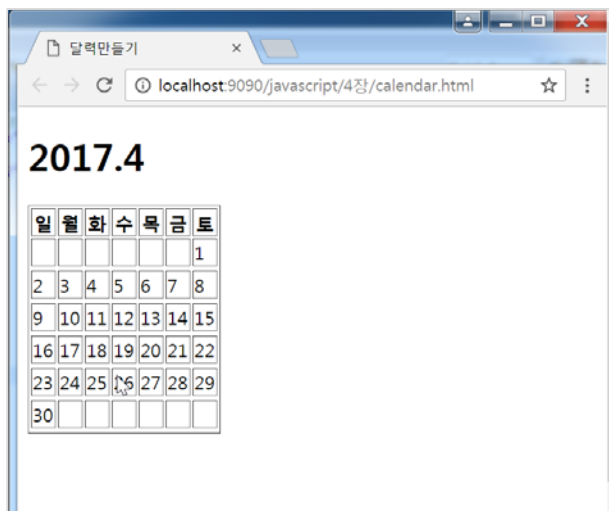
//⑤ 문자로 결합된 달력 테이블을 문서에 출력합니다.
document.write(calendar);
//]]>
</script>
</head>

<body>

</body>
</html>

```

2. 작성한 문서를 저장한 후 브라우저에서 확인해 본다.



## 4.3 브라우저 객체 모델(BOM)

### 4.3.1 브라우저 객체란?

- 브라우저에 내장된 객체를 가리킨다. 브라우저 객체의 최상위 객체는 window 객체이며 그 하위로는 document, screen, location, history, navigator 등의 계층적 구조로 이루어져 있는 것을 일컬어 브라우저 객체 모델(BOM: Browser Object Model)이라고 한다.

Window - document - body, div, img, a, form, input, ...

- screen
- location
- history
- navigator

### 4.3.2 windows 객체

## ■ windows 객체 메서드 종류

종류	설명
open()	새 창을 열 때 사용한다.
alert()	경고 창을 띄운다.
prompt()	질의응답 창을 띄운다.
confirm()	확인/취소 창을 띄운다.
moveTo()	창의 위치를 이동시킬 때 사용한다.
resizeTo()	창의 크기를 변경시킬 때 사용한다.
setInterval()	일정 간격으로 지속적으로 실행문을 실행시킬 때 사용한다.
setTimeout()	일정 간격으로 한 번만 실행문을 실행시킬 때 사용한다.

### (1) open()

#### ■ 새 브라우저 창을 띄울 때 사용한다.

```
//기본형
window.open("url경로", "창이름", "옵션설정")
```

#### ■ open() 메서드의 속성

종류	설명
width	창의 너비를 설정한다.
height	창의 높이를 설정한다.
left	창의 좌.우 수평 위치를 설정한다.
top	창의 상.하 수직 위치를 설정한다.
location	창의 url 주소 입력 영역 노출 여부를 결정한다.(숨김=no, 노출=yes)
status	창의 상태 영역 노출 여부를 결정한다.(숨김=no, 노출=yes)
scrollbars	창의 스크롤 막대 노출 여부를 결정한다.(숨김=no, 노출=yes)
toolbars	창의 도구상자 노출 여부를 결정한다.(숨김=no, 노출=yes)

[실습파일: winpopup\_test.html, 완성파일: winpopup.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="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title> window 객체 </title>
<style type="text/css">
*{margin:0;padding:0;}
</style>
</head>
<body>
<p>

<map name="intro" id="intro">
<area shape="rect" coords="230,368,280,390" href="#" alt="창 닫기"
onclick="window.close();" />
</map>
</p>
</body>
</html>
```

[실습파일: win\_ob1\_test.html, 완성파일: win\_ob1.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="ko" xml:lang="ko">
```



```

<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>window 객체</title>
<script type="text/javascript">
    <![CDATA[
        window.open("winpopup.html", "pop1",
            "width=300,height=400,left=300,top=10,scrollbars=no,toolbars=no,location=no");
    </script>
</head>
<body>
    <h1>이지스 퍼블리싱</h1>
</body>
</html>

```

## (2) alert()

- 경고 창을 띄울 때 사용한다.

```

//기본형
alert("경고 메시지")

alert("잘못 입력했습니다") // 경고 창을 띄운다.

```

## (3) prompt()

- 질의응답 창을 띄울 때 사용한다.

```

//기본형
prompt("질의 내용", "기본 답변");

prompt("당신의 연령은?", "0"); // 질의응답 창을 띄운다.

```

## (4) confirm()

- 확인/취소 창을 띄울 때 사용한다.

```

//기본형
confirm("질의 내용");

confirm("정말로 삭제하시겠습니까?"); // 확인, 취소 창을 띄운다.

```

## (5) moveTo()

- 브라우저 창의 위치를 이동시킬 때 사용한다.

```

//기본형
moveTo(x의 위치값, y의 위치값);

moveTo(100,200);

```

## (6) `resizeTo()`

- 브라우저 창의 너비와 높이를 바꿀 때 사용한다.

```
//기본형
resizeTo(너비값, 높이값);

resizeTo(200, 300); // 창의 너비는 200px, 높이는 300px로 바꾼다.
```

[실습파일: win\_ob2\_test.html, 완성파일: win\_ob2.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="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>window 객체</title>
<script type="text/javascript">
    <![CDATA[
        //변수 pop가 팝업 창을 참조합니다.
        var pop = window.open("winpopup.html", "pop1",
                                "width=300,height=400,left=300,top=10,scrollbars=no,toolbars=no,location=no");
    ]]>
</script>
</head>
<body>
    <p>
        <button onclick="pop.moveTo(200,300); pop.focus();">브라우저 창 위치 변경</button>
    </p>
    <p>
        <button onclick="pop.resizeTo(200,300); pop.focus();">브라우저 창 크기 변경</button>
    </p>
</body>
</html>
```

## (7) `setInterval()/clearInterval()`

- 일정한 시간 간격으로 스크립트 실행문을 반복하여 실행시킬 때 사용한다.

```
//기본형
var 참조변수 = setInterval("스크립트 실행문", 시간 간격);
clearInterval(참조변수);

var t = setInterval("i++",3000); // 3초마다 변수 i의 값이 1씩 증가한다.
clearInterval(t); // t에 참조되어 있던 setInterval()를 삭제한다.
```

[실습파일: win\_ob3\_test.html, 완성파일: win\_ob3.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="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>window 객체</title>
<script type="text/javascript">
    <![CDATA[
```

```

        var i = 0;
        var auto = setInterval("console.log(i++);", 3000);
        //]]>
</script>
</head>
<body>
    <!-- 버튼을 누르면 auto가 참조하는 setInterval을 삭제합니다. -->
    <button onclick="clearInterval(auto)">정지버튼</button>
</body>
</html>

```

## (8) setTimeout()/clearTimeout()

- 일정한 간격으로 스크립트 실행문을 한 번만 실행시킨다. 하지만 setTimeout() 메서드도 재귀함수 호출을 이용하면 setInterval() 메서드처럼 사용할 수 있다.

```

//기본형
var 참조변수 = setTimeout("스크립트 실행문", 시간간격);
clearTimeout(참조변수);

var t = setTimeout("console.log(++i)", 5000); // 5초 간격 이후 단 한 번만 변수 i의 값을 1 증가시킨다.
clearTimeout(t); // t에 참조되어 있던 setTimeout()을 삭제한다.

```

[실습파일: win\_ob4\_test.html, 완성파일: win\_ob4.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="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>window 객체</title>
<script type="text/javascript">
    <![CDATA[
        var i = 0; //초깃값 지정;
        var auto = setTimeout("console.log(++i);", 3000);
        //]]>
    </script>
</head>
<body>
</body>
</html>

```

### 4.3.3 screen 객체

- screen객체는 스크린(모니터) 너비, 높이 등 정보(속성)를 제공한다.

```

//기본형
screen.속성;

screen.width; // 모니터의 너비값을 반환한다.

```

[실습파일: scr\_ob1\_test.html, 완성파일: scr\_ob1.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="ko" xml:lang="ko">

```

```

<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<script type="text/javascript">
    //
    document.write("width: " + screen.width, "&lt;br /&gt;");
    document.write("height: " + screen.height, "&lt;br /&gt;");
    document.write("availWidth: " + screen.availWidth, "&lt;br /&gt;"); // 작업 표시줄을 제외한 너비값
    document.write("availHeight: " + screen.availHeight, "&lt;br /&gt;"); // 작업 표시줄을 제외한 높이값
    document.write("colorDepth: " + screen.colorDepth, "&lt;br /&gt;"); // 모니터 표현 가능 컬러 bit
    //]]&gt;
&lt;/script&gt;
&lt;title&gt;screen 객체&lt;/title&gt;
&lt;/head&gt;
&lt;body&gt;
&lt;/body&gt;
&lt;/html&gt;
</pre>
</div>
<div data-bbox="115 307 313 325" data-label="Section-Header">
<h4>4.3.4 location 객체</h4>
</div>
<div data-bbox="115 340 716 356" data-label="List-Group">
<ul>
<li>■ location 객체는 브라우저에 url 관련 정보(속성) 및 메서드를 제공한다.</li>
</ul>
</div>
<div data-bbox="124 384 506 434" data-label="Text">
<pre>
//기본형
location.속성 또는 location.메서드();

location.href; // 주소 영역에 참조 주소(url)를 반환한다.
</pre>
</div>
<div data-bbox="115 461 374 477" data-label="Text">
<p>location 객체 속성/메서드 종류</p>
</div>
<div data-bbox="117 475 862 618" data-label="Table">
<table>
<tr>
<th>종류</th><th>설명</th></tr>
<tr>
<td>location.href</td><td>주소 영역에 참조 주소를 설정하거나 URL을 반환한다.</td></tr>
<tr>
<td>location.hash</td><td>URL에 해시값(#에 명시된 값)을 반환한다.<br/>예) http://www.easyspub.com#hello</td></tr>
<tr>
<td>location.hostname</td><td>URL에 호스트 이름을 설정하거나 반환한다.</td></tr>
<tr>
<td>location.host</td><td>URL에 호스트 이름과 포트 번호를 반환한다.</td></tr>
<tr>
<td>location.port</td><td>URL에 포트 번호를 반환한다.</td></tr>
<tr>
<td>location.protocol</td><td>URL에 프로토콜을 반환한다.</td></tr>
<tr>
<td>location.search</td><td>URL에 쿼리(요청 값)을 반환한다.<br/>예) http://www.easyspub.com?pageNum=1&amp;sort=DESC</td></tr>
<tr>
<td>location.reload()</td><td>새로 고침이 일어난다.</td></tr>
</table>
</div>
<div data-bbox="124 633 231 646" data-label="Text">
<p>[loca_ob1.html]</p>
</div>
<div data-bbox="124 658 825 892" data-label="Text">
<pre>
&lt;!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"&gt;
&lt;html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko"&gt;
&lt;head&gt;
&lt;meta http-equiv="content-type" content="text/html; charset=utf-8" /&gt;
&lt;script type="text/javascript"&gt;
    //<![CDATA[
    document.write("href : " + location.href, "&lt;br /&gt;");
    document.write("hash : " + location.hash, "&lt;br /&gt;");
    document.write("hostname : " + location.hostname, "&lt;br /&gt;");
    document.write("host : " + location.host, "&lt;br /&gt;");
    document.write("protocol : " + location.protocol, "&lt;br /&gt;");
    //]]&gt;
&lt;/script&gt;
&lt;title&gt;location 객체&lt;/title&gt;
&lt;/head&gt;
&lt;body&gt;
&lt;/body&gt;
&lt;/html&gt;
</pre>
</div>
```

### 4.3.5 history 객체

- history객체는 브라우저에 방문 기록에 대한 정보(속성) 및 메서드를 제공한다. 마치 브라우저에 이전, 다음버튼과 같은 기능을 제공한다.

```
//기본형
history.메서드 또는 history.속성;

history.back(); // 바로 이전 페이지로 이동한다.
```

[his\_ob1.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="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>history 객체</title>
</head>
<body>
    <h1>첫 페이지</h1>
    <a href="his_obp2.html">2페이지 이동</a>
</body>
</html>
```

[his\_ob2.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="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>history 객체</title>
<style type="text/css">
#nextBtn {
    display: none;
}
</style>
</head>
<body>
    <h1>두번째 페이지</h1>
    <a href="his_obp3.html">3페이지 이동</a>
    <p>
        <button id="prevBtn" onclick="history.back();">이전 페이지</button>
        <button id="nextBtn" onclick="history.forward();">다음 페이지</button>
    </p>
</body>
</html>
```

[his\_ob3.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="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title> history 객체 </title>
</head>
<body>
```

```

<h1>세번째 페이지</h1>
<p>
  <button onclick="history.go(-1)">1단계 이전 페이지</button><br />
  <button onclick="history.go(-2)">2단계 이전 페이지</button>
</p>
</body>
</html>

```

### 4.3.6 navigator 객체

- navigator 객체는 방문자의 브라우저 정보와 운영체제 정보를 제공한다.

```

[nav_ob1.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="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>navigator 객체</title>
<script type="text/javascript">
  <![CDATA[
    document.write("appCodeName: " + navigator.appCodeName, "<br />");
    document.write("appName: " + navigator.appName, "<br />");
    document.write("appVersion: " + navigator.appVersion, "<br />");
    document.write("language: " + navigator.language, "<br />");
    document.write("product: " + navigator.product, "<br />");
    document.write("platform: " + navigator.platform, "<br />");
    document.write("userAgent: " + navigator.userAgent, "<br />");
  ]]>
</script>
</head>
<body>

</body>
</html>

```

### [실습] 브라우저 객체 모델을 사용해 운영체제와 스크린 정보 얻기

1. 실습 파일(os\_info\_test.html)을 열고 주석 설명에 유의하여 다음 자바스크립트 코드를 작성한다.

```

[os_info.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="ko" xml:lang="ko">
<head>
<title> 운영체제 및 스크린 정보 </title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<meta name="viewport" content="width=device-width"/>
<style type="text/css">
  #info_wrap{width:300px;height:100px;background-color:#eee;overflow:hidden;padding:10px;}
</style>
<script type="text/javascript">
  <![CDATA[

/*① [운영 체제 정보]버튼을 누를 때마다 실행되는 함수입니다*/
function info1(){
  /* 브라우저 및 운영체제 종합정보를 반환후 모두 소문자로 바꿉니다.*/

```

```

var os=navigator.userAgent.toLowerCase();

/*요소 중 id의 값이 "info_wrap"은 요소 객체를 가져옵니다.*/
var info_wrap=document.getElementById("info_wrap");

/*navigator객체를 이용해 반환받은 정보에 다음과 같은 문자가 포함되어 있으면 운영체제 명을 지정한 요소
사이에 텍스트로 출력합니다.*/
if(os.indexOf('windows')>=0){
    info_wrap.innerHTML="윈도우";
}else if(os.indexOf('macintosh')>=0){
    info_wrap.innerHTML="메킨토시";
}else if(os.indexOf('iphone')>=0){
    info_wrap.innerHTML="아이폰";
}else if(os.indexOf('android')>=0){
    info_wrap.innerHTML="안드로이드";
}

//3초 후에 운영체제 정보를 지웁니다.
setTimeout("info_wrap.innerHTML=''",3000);
}

/*② [스크린 정보]버튼을 누를 때마다 실행되는 함수입니다*/
function info2(){
    /*스크린 너비와 높이 값을 가져와 각각의 변수에 저장합니다.*/
    var sc_w=screen.width;
    var sc_h=screen.height;

    /*요소 중 id의 값이 "info_wrap"은 요소 객체를 가져옵니다.*/
    var info_wrap=document.getElementById("info_wrap");

    /*스크린 너비와 높이 값을 지정한 요소에 텍스트로 출력합니다. */
    info_wrap.innerHTML=sc_w+"X"+sc_h;

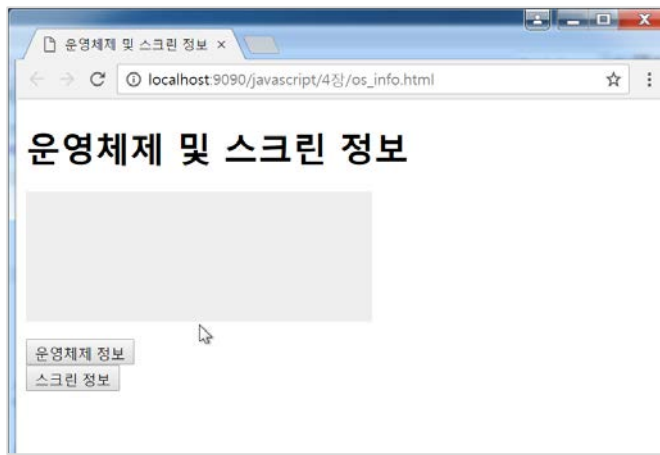
    //3초 후에 스크린 정보를 지웁니다.
    setTimeout("info_wrap.innerHTML=''",3000);
}
//]]>
</script>
</head>
<body>
    <h1>운영체제 및 스크린 정보</h1>
    <p id="info_wrap"></p>

    <!-- 아래 버튼을 누를 때 마다 info1 함수의 중괄호 사이의 자바스크립트 실행문 코드가 실행됩니다. -->
    <button onclick="info1();">운영체제 정보</button> <br />

    <!-- 아래 버튼을 누를 때 마다 info2 함수의 중괄호 사이의 자바스크립트 실행문 코드가 실행됩니다. -->
    <button onclick="info2();">스크린 정보</button> <br />
</body>
</html>

```

2. 작성한 코드를 저장한 후 PC와 스마트폰에서 접속해 본다.



## 4.4 문서 객체 모델(DOM)

### 4.4.1 문서 객체 모델

- 문서 객체 모델이란 HTML 문서의 구조를 가리킨다. 가령, <img> 태그에는 src라는 속성과 그림을 표현하는 기능을 가지고 있다. 이렇듯 HTML 태그는 다른 말로 문서 객체라 부른다.
- 문서 객체 모델을 배우는 목적은 자바스크립트를 사용하여 문서 객체를 선택하고 속성 또는 스타일(CSS)을 적용하기 위해서이다.

### 4.4.2 선택자

- 선택자는 자바스크립트를 이용해 HTML에 문서 객체를 선택할 때 사용한다. 스타일(CSS)에 선택자와 비슷하다고 생각하면 된다.

```
// CSS 선택자 = javascript 선택자
// CSS
#box{ color:red; }
#box + p { color:red; }

//javascript
document.getElementById("box").style.color=red;
document.getElementById("box").nextSibling.style.color=red;
```

### 4.4.3 인접 요소 관계 파악하기

- <html>은 모든 문서 객체를 감싸고 있어 문서 객체의 최상위 객체 또는 루트 객체라 한다.
- <head> 태그와 <body> 태그는 <html> 태그의 자식 요소이다. 그렇다면 <html> 태그는 이들의 부모 요소가 된다.

### 4.4.4 선택자 종류

구분	종류	설명
직접 선택자	document.getElementById("아이디명")	아이디를 이용해 요소를 선택해 온다.
	document.getElementsByTagName("요소(태그)명")	요소의 이름을 이용해 요소를 선택해 온다.



	document.formName.inputName	폼 요소에 name 속성을 이용해 요소를 선택한다.
인접 관계 선택자	parentNode	선택한 요소의 부모 요소를 선택해 온다.
	childNodes	선택한 요소의 모든 자식 요소를 선택해 온다. 선택한 모든 요소가 배열 객체로 저장된다.
	children	선택한 요소의 자식 요소인 태그만 선택해 온다. 선택한 모든 요소가 배열 객체로 저장된다.
	firstChild	선택한 요소의 첫 번째 자식 요소만 선택해 온다.
	previousSibling	선택한 요소의 이전에 오는 형제 요소만 선택해 온다.
	nextSibling	선택한 요소의 다음에 오는 형제 요소만 선택해 온다.

[꿀팁] parentNode, firstChild, previousSibling, nextSibling은 공백 문자 인식 호환성에 문제가 있다.

#### 4.4.5 선택자 적용 위치와 방법

##### ■ 선택자 사용시 잘못된 적용 위치

```
<html>
<head>
  자바스크립트 선언 // HTML 문단 태그가 로딩되기 전에 실행된다.
</head>
<body>
  HTML 문단 태그 작성
</body>
</html>
```

##### ■ 선택자 사용시 올바르게 적용된 위치 (두가지 방식)

```
<html>
<head>
</head>
<body>
  HTML 문단 태그 작성
  자바스크립트 선언 // HTML 문단 태그가 로딩된 후에 실행된다.
</body>
</html>
```

```
<html>
<head>
<script type="text/javascript">
  window.onload=function(){
    스크립트 실행문; // HTML 문단 태그가 로딩된 후에 실행된다.
  }
</script>
</head>
<body>
  HTML 문단 태그 작성
</body>
</html>
```

#### 4.4.6 직접 선택자를 사용한 문석 객체 CSS 적용법

```
// 기본형
```

```
document.직접선택자(선택 메서드).style.속성=값;

<h1 id="title">웹표준</h1>
document.getElementById("title").style.color="blue";
```

[dom\_ob2.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="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>DOM 객체</title>
</head>
<body>
    <div>
        <h1>선호하는 음식</h1>
        <div id="food_1">
            <h2 id="title">한식</h2>
            <ol>
                <li><a href="#">김치</a></li>
                <li>불고기</li>
                <li>비빔밥</li>
            </ol>
        </div>
        <div id="food_2">
            <h2>양식</h2>
            <ol id="list">
                <li>돈까스</li>
                <li id="pizza">피자</li>
                <li>파스타</li>
            </ol>
        </div>
    </div>
    <script type="text/javascript">
        //<![CDATA[
        document.getElementsByTagName("h1")[0].style.color = "green";
        document.getElementById("title").style.color = "red";

        var myList = document.getElementById("food_1").getElementsByTagName("li")[1];
        myList.style.backgroundColor = "red";
        //]]>
    </script>
</body>
</html>
```

#### 4.4.7 인접 관계 선택자로 문서 객체에 스타일 적용하기

- 직접 선택자를 사용하여 원거리에 요소까지 선택한 후, 인접 관계 선택자를 사용해 앞에서 선택한 요소를 기준으로 근거리에 있는 요소를 선택한다.

```
// 기본형
document.직접선택자(선택 메서드).인접관계선택자.style.속성=값;

<h1 id="title">웹표준</h1>
document.getElementById("title").children[0].style.backgroundColor="#ff0";
```

[dom\_ob3.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="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>DOM 객체</title>
</head>
<body>
    <div id="wrap"> <!-- 줄 바꿈(공백 문자 생성됨) -->
        <p>내용1</p> <!-- 줄 바꿈(공백 문자 생성됨) -->
        <p>내용2</p>
        <p>내용3</p>
        <p>내용4</p>
    </div>
    <script type="text/javascript">
        /*<![CDATA[
        /*위에 요소 중 id의 값이 "wrap"인 <div>를 선택합니다. <div>내에
        첫 번째 요소인 <p>내용1</p>가 선택이 되어야 합니다. 하지만
        IE(8이하)의 브라우저에서는 요소가 아닌 공백 문자가 선택됩니다.*/
        document.getElementById("wrap").firstChild.style.color = "red";

        /*위에 <p> 요소 중 두번째 문서 객체를 변수 p에 참조시켰습니다. 그리고
        다음 요소 선택자(nextSibling)를 사용해 두번째 <p>에 다음에 오는 요소
        <p>내용3</p>를 선택되어야 합니다. 이때도 역시 IE(8이하)의 브라우저는
        공백이 선택됩니다*/
        var p = document.getElementsByTagName("p")[1];
        p.nextSibling.style.backgroundColor = "yellow";
        //]]>
    </script>
</body>
</html>

```

- 위의 예제를 실행하면 IE8 이하 버전에서는 인접 관계 선택자가 잘 적용되어 스타일이 정상적으로 적용된다. 하지만 파이어폭스, 크롬, 사파리 등의 브라우저에서는 정상적으로 적용되지 않는다. 결과가 다르게 나오는 이유는 공백 문자의 인식 여부 차이 때문이다. IE8 이하의 브라우저는 HTML 코드에 공백이 있거나 줄 바꿈이 있어도 문자로 인식되지 않는다.

#### 4.4.8 노드 타입으로 인접 관계 선택자 문제 해결하기

- DOM에서 인접 관계 선택자를 사용하면 IE 8 이하 브라우저와 그 외의 브라우저에 호환성 문제(공백문자 처리)가 생긴다. 이 호환성 문제는 노드 타입(nodeType)을 이용하면 해결할 수 있다.

```

<a href="http://w3.org" id="m">W3C</a>
document.getElementById("m").nodeType; // 1을 반환한다. '<a>'
document.getElementById("m").getAttributedNode("href").nodeType; // 2을 반환한다. 'href'
document.getElementById("m").firstChild.nodeType; // 3을 반환한다. 'W3C'

<div id="wrap">
    <p>내용1</p> index 0
    <p>내용1</p> index 1
    <p>내용1</p> index 2
    <p>내용1</p> index 3
</div>
document.getElementById("wrap").firstChild.nodeType; // IE8 이하인 경우, 노드 타입 값이 1을 반환한다.
document.getElementById("wrap").firstChild.nodeType; // 파이어폭스, 크롬, 사파리 등 --> 3을 반환 (공백 문자가 선택된다.)

// 비표준 인접 관계 선택자 해결 방법
document.getElementById("wrap").children[0].nodeType;
var p=document.getElementsByTagName("p")[1];
var nextObj=p.nextSibling;
while(nextObj.nodeType !=1){
    nextObj=nextObj.nextSibling;
}

```

```
}
```

[실습파일: dom\_ob4\_test.html, 완성파일: dom\_ob4.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="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title> DOM 객체 </title>
</head>
<body>
<div id="wrap">
  <p>내용1</p>
  <p>내용2</p>
  <p>내용3</p>
  <p>내용4</p>
</div>
<script type="text/javascript">
//
  document.getElementById("wrap").children[0].style.color="red";

  var p=document.getElementsByTagName("p")[1];
  var nextObj=p.nextSibling;
  while(nextObj.nodeType !=1){
    nextObj=nextObj.nextSibling;
  }
  nextObj.style.backgroundColor="yellow";
//]]&gt;
&lt;/script&gt;
&lt;/body&gt;
&lt;/html&gt;</pre></div><div data-bbox="115 519 594 537" data-label="Section-Header"><h4>4.4.9 요소의 속성값 변경, 생성 및 불러오는 방법</h4></div><div data-bbox="115 552 348 568" data-label="Section-Header"><h5>(1) 문서 객체의 속성 메서드</h5></div><div data-bbox="125 595 647 684" data-label="Text"><pre>&lt;h1&gt;&lt;img src="images/logo.gif" id="top_logo" alt="이지스 퍼블리싱" /&gt;&lt;/h1&gt;

document.getElementById("top_logo").src; // "images/logo.gif"
document.getElementById("top_logo").getAttribute("src"); // "images/logo.gif"

document.getElementById("top_logo").src = "images/symbol.jpg";
document.getElementById("top_logo").getAttribute("src","images/symbol.jpg");</pre></div><div data-bbox="115 725 466 741" data-label="Section-Header"><h5>(2) 문서 객체의 하위 요소(innerHTML) 속성</h5></div><div data-bbox="125 768 600 807" data-label="Text"><pre>&lt;h1 id="title"&gt;&lt;a href="#"&gt;이지스 퍼블리싱&lt;/a&gt;&lt;/h1&gt;

document.getElementById("title").innerHTML = "&lt;strong&gt;Hello&lt;/strong&gt;";</pre></div><div data-bbox="115 848 514 866" data-label="Section-Header"><h4>4.4.10 문서 객체 이벤트 핸들러 적용하기</h4></div><div data-bbox="115 881 248 898" data-label="Section-Header"><h5>■ 이벤트 종류</h5></div>
```

이벤트	설명
onclick	선택한 요소를 마우스로 클릭했을 때 이벤트가 발생한다.
onmouseover	선택한 요소에 마우스를 올렸을 때 이벤트가 발생한다.
onmouseout	선택한 요소에 마우스가 벗어났을 때 이벤트가 발생한다.
submit	선택한 폼에 전송이 일어났을 때 이벤트가 발생한다.

```
<a href="#" id="btn1"> 버튼1 </a>
```

```
document.getElementById("btn1").onclick=function(){
    alert("welcome");
}
```

[실습파일: dom\_ob5\_test.html, 완성파일: dom\_ob5.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="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title> DOM 객체 </title>
</head>
<body>
<h1 id="title">마우스 아웃</h1>
<a href="#" id="btn"></a>
<p id="img_src"></p>
<script type="text/javascript">
//
    var theBtn=document.getElementById("btn");
    var s=document.getElementById("img_src");
    theBtn.onmouseover=function(){
        document.getElementById("title").innerHTML="마우스 오버";

        theBtn.firstChild.src="images/btn_over.gif";
        s.innerHTML=theBtn.firstChild.src;
    }

    theBtn.onmouseout=function(){
        document.getElementById("title").innerHTML="마우스 아웃";

        theBtn.firstChild.src="images/btn_out.gif";
        s.innerHTML=theBtn.firstChild.src;
    }
//]]&gt;
&lt;/script&gt;
&lt;/body&gt;
&lt;/html&gt;</pre>
</div>
<div data-bbox="115 727 334 744" data-label="Section-Header">
<h4>4.4.11 폼 요소 선택자</h4>
</div>
<div data-bbox="115 760 297 775" data-label="Section-Header">
<h5>■ 폼 요소 선택 방식</h5>
</div>
<div data-bbox="117 790 878 881" data-label="Table">
<table>
<tr>
<th>구분</th><th>종류</th><th>설명</th></tr>
<tr>
<td data-kind="parent" data-rs="2">입력 요소<br/>선택자</td><td>document.getElementById("아이디명")</td><td>폼 요소를 아이디로 선택할 때 사용한다.</td></tr>
<tr>
<td data-kind="ghost"></td><td>document.폼이름.입력요소이름</td><td>폼 요소를 이름으로 선택할 때 사용한다.</td></tr>
<tr>
<td data-kind="parent" data-rs="2">select option<br/>선택자</td><td>document.폼이름.입력요소이름.options[index]</td><td>&lt;select&gt;에 &lt;option&gt;을 선택할 때 사용한다.</td></tr>
<tr>
<td data-kind="ghost"></td><td>var i=document.폼이름.입력요소이름.selectedIndex;<br/>document.폼이름.입력요소이름.options[i];</td><td>&lt;select&gt;에 선택된 &lt;option&gt;을 선택할 때 사용합니다.</td></tr>
</table>
</div>
<div data-bbox="125 895 486 909" data-label="Text">
<p>[실습파일: dom_ob6_test.html, 완성파일: dom_ob6.html]</p>
</div>
```

```

<!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> DOM 객체 </title>
<style type="text/css">
    #terms{width:400px;height:50px;
    overflow:auto;border:1px solid #bbb;}
    form{width:550px;}
</style>
</head>
<body>
<form action="#" method="get" name="f1">
<fieldset>
    <legend>폼 요소1</legend>
    <p>
    <label for="user_pw1">비밀번호</label>
    <input type="password" name="user_pw1" id="user_pw1" />
    </p>
    <p>
    <label for="user_pw2">비밀번호 확인</label>
    <input type="text" name="user_pw2" id="user_pw2" />
    <input type="submit" value="확인" />
    </p>
</fieldset>
</form>
<form action="#" method="get" name="f2">
<fieldset>
    <legend>폼 요소2</legend>
    <div id="terms">
        회사는 법령이 정하는 바에 따라 "회원"의 개인정보를 보호하기
        위해 노력합니다. 개인정보의 보호 및 사용에 대해서는 관련법
        및 회사의 개인정보취급방침이 적용됩니다.
    </div>
    <p>
    <input type="checkbox" name="accept" id="accept" />
    <label for="accept">약관 동의합니다.</label></p>
    <p>
    <input type="checkbox" name="allChk" id="allChk" />
    <label for="allChk">전체선택</label><br />
    <input type="checkbox" name="subject1" id="subject1" value="s1" />
    <label for="subject1">과목1</label><br />
    <input type="checkbox" name="subject2" id="subject2" value="s2" />
    <label for="subject2">과목2</label><br />
    <input type="checkbox" name="subject3" id="subject3" value="s3" />
    <label for="subject3">과목3</label>
    </p>
    <p>
    <input type="submit" value="등록 완료" />
    <input type="reset" value="등록 취소" />
    </p>
</fieldset>
</form>
<script type="text/javascript">
//
// 첫 번째 폼요소에 전송버튼 눌렀을 때 실행
document.f1.onsubmit=function(e){
    var pw1=document.f1.user_pw1;
    var pw2=document.f1.user_pw2;

    pw2.value=pw1.value;
    pw2.disabled=true;
    return false;
}
//전체선택 체크박스를 클릭했을 때
</pre>
</div>
```

```

document.f2.allChk.onclick=function(){
    if(this.checked){ //체크되어 있으면 true 반환
        document.f2.subject1.checked=true; //첫 번째 체크박스 체크됨
        document.f2.subject2.checked=true; //두 번째 체크박스 체크됨
        document.f2.subject3.checked=true; //세 번째 체크박스 체크됨
    }else{
        document.f2.subject1.checked=false; //첫 번째 체크박스 체크 해제됨
        document.f2.subject2.checked=false; //두 번째 체크박스 체크 해제됨
        document.f2.subject3.checked=false; //세 번째 체크박스 체크 해제됨
    }
}
// 두 번째 폼요소에 전송버튼 눌렀을 때 실행
document.f2.onsubmit=function(){
    var act=document.f2.accept;
    if(act.checked==false){ //체크되어 있지 않으면 false
        alert("약관 동의해야 합니다!");
        act.focus();
        return false;
    }
}
//]]>
</script>
</body>
</html>

```

## [실습] 문서 객체 모델을 사용하여 자동차 견적 미리보기 페이지 만들기

1. 다음은 문서 객체에 이벤트 핸들러를 적용하여 문서 객체 속성과 하위 요소를 변경한 것이다. estimate\_test.html 파일을 열고 주석 설명에 유의하여 페이지를 완성한다.

```

[estimate.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>자동차 견적서</title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<style type="text/css">
* {
    margin: 0;
    padding: 0;
}

body {
    font: 12px/1.5 dotum, "돋움", gulim, "굴림", sans-serif;
}

li {
    list-style: none;
}

table {
    border-collapse: collapse;
    border-spacing: 0;
}

h1 {
    text-align: center;
}

#carZone {
    width: 600px;
    margin: 0 auto;
}

```



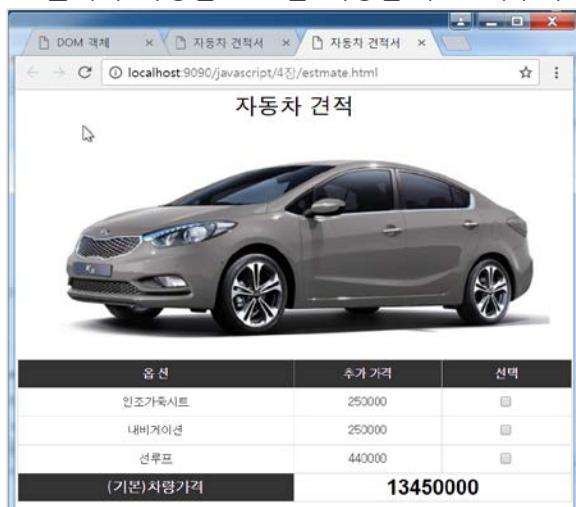


```

        <col width="160px" />
        <col width="*" />
    </colgroup>
    <thead>
        <tr>
            <th scope="row">옵 션</th>
            <th scope="row">추가 가격</th>
            <th scope="row">선택</th>
        </tr>
    </thead>
    <tfoot>
        <tr>
            <th scope="col">( 기본 )차량가격</th>
            <td colspan="2"><input type="text" name="total" id="total"
                value="13450000" readonly="readonly" /></td>
        </tr>
    </tfoot>
    <tbody>
        <tr>
            <td><label for="opt1">인조가죽시트</label></td>
            <td>250000</td>
            <td>
                <!-- 옵션 체크박스에 클릭할 때 마다 car()에 저장된
                일련의 실행문을 실행합니다. --> <input type="checkbox" name="opt1" id="opt1"
                    value="250000" onclick="car();" />
            </td>
        </tr>
        <tr>
            <td><label for="opt2">내비게이션</label></td>
            <td>250000</td>
            <td><input type="checkbox" name="opt2" id="opt2"
                value="250000" onclick="car();" /></td>
        </tr>
        <tr>
            <td><label for="opt3">선루프</label></td>
            <td>440000</td>
            <td><input type="checkbox" name="opt3" id="opt3"
                value="440000" onclick="car();" /></td>
        </tr>
    </tbody>
</table>
</div>
</body>
</html>

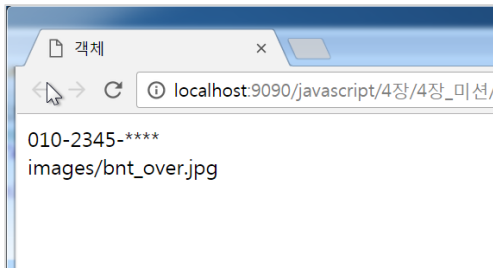
```

2. 앞에서 작성한 코드를 저장한 후 브라우저에서 실행한다.



## [과제] 도전 Mission!

1. 문자 객체를 이용해 다음 결과 화면과 같이 출력되도록 코드를 완성하라.



[실습파일: ob\_test2\_test.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="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title> 객체 </title>
<script type="text/javascript">
//
    var num="010-2345-1234"
    var result1=num.substring(
    document.write(result1,"**** &lt;br /&gt;");

    var imgSrc="images/bnt_out.jpg";
    var result2=imgSrc.replace(
    document.write(result2, "&lt;br /&gt;");
//]]&gt;
&lt;/script&gt;
&lt;/head&gt;
&lt;body&gt;

&lt;/body&gt;
&lt;/html&gt;</pre></div><div data-bbox="115 593 581 609" data-label="List-Group"><ol><li>2. 배열 객체와 수학 객체를 이용해 다음 코드를 완성하라.</li></ol></div><div data-bbox="125 636 867 662" data-label="Text"><p>점심 시간에 "짜장면", "돈가스", "된장국", "김치찌개", "회덮밥" 중 어떤 메뉴로 선택할지 고민된다면 랜덤으로 선택되도록 만들어 보자.</p></div><div data-bbox="125 660 782 687" data-label="List-Group"><ul><li>● 배열 객체를 이용하여 메뉴 5개 데이터를 변수 menu에 참조시킨다.</li><li>● Math.random()과 배열 데이터 개수(length)를 이용하여 배열 인덱스 값이 랜덤으로 나오도록 한다.</li></ul></div><div data-bbox="120 714 460 842" data-label="Image"><img alt="Screenshot of a web browser showing the output of a JavaScript script. The address bar shows 'localhost:9090/javascript/4장/4장_미션/'. The page content displays '된장국'." data-bbox="120 714 460 842"/></div><div data-bbox="126 857 332 872" data-label="Text"><p>[실습파일: ob_test4_test.html]</p></div><div data-bbox="124 883 825 910" data-label="Text"><pre>&lt;!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"&gt;</pre></div>
```

```
<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> 객체 </title>
<script type="text/javascript">
//
var menu=
var menuNum=Math.floor(
var result=menu[menuNum];
document.write(result);
//]]&gt;
&lt;/script&gt;
&lt;/head&gt;
&lt;body&gt;
&lt;/body&gt;
&lt;/html&gt;</pre></div>
```