훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 브라우저 제어를 위한 내장 객체

◆ window 객체

* Javascript의 최상위 객체 - 따라서 모든 객체는 window객체의 하위 객체로 존재한다.

★원칙적 내장객체 접근 법

```
window.내장객체이름.함수이름([파라미터]);
```

* 모든 객체가 window 객체안에 내장되어있기 때문에 window 객체 명시를 생략 가능

★window객체 생략 내장객체 접근 법

```
내장객체이름.함수이름([파라미터]);
```

★document 객체 사용 예 (파란색은 생략됨)

```
window.document.getElementById("id value");
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ window 객체

◆ 브라우저의 창 제어 기능

* 새로운 창 열기 - 탭 브라우징을 지원하는 웹 브라우저에서 탭을 여는 기능으로 동작

```
window.open("페이지 URL");
```

```
window.open("페이지 URL", "창이름", "옵션");
```

* 현재 창 닫기 (window: 현재 창 , self : 관계연산자 자기자신)

```
window.close();
```

또는

```
self.close();
```

❖ window 객체

◆ 브라우저의 창 제어 기능

* 팝업창 열기 - 창 이름을 지정한 경우

- 반복 클릭하더라도 하나의 창을 계속 사용

```
window.open("페이지 URL", "창이름", "팝업창 옵션");
```

* 팝업창 열기 - 창 이름을 지정하지 않은 경우

- 반복 클릭하면 다른 팝업창이 열린다.

```
window.open('페이지 URL', '', '팝업창 옵션');
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ window 객체

◆ 브라우저의 창 제어 기능

* 팝업창 열기 옵션값의 종류

옵션값	값 지정법	설 명
toolbar	yes / no	툴바 아이콘의 표시 여부를 설정
location		주소 표시줄의 표시 여부 설정
status		상태 바의 표시 여부 설정
menubar		메뉴 표시줄의 표시 여부 설정
scrollbars		스크롤바의 표시 여부 설정
resizable		창의 크기를 조절 가능하게 할지 여부를 설정
width	픽셀값	창의 폭을 지정
height		창의 높이를 지정

fullscreen 등의 추가적인 용어가 있으며 scrollbars, resizeable 등은 지정을 하지 않을 경우
열려야 하는 창이 팝업사이즈보다 더 클경우 상하 및 좌우스크롤은 자동으로 생성

* 팝업창 열기 옵션 적용 예

```
window.open('http://www.naver.com', 'mywin', 'width=300, height=500, scrollbars=no,
toolbar=no, menubar=no, status=no, location=no);
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ window 객체 이벤트처리

js_window02.html

```
<body>
        <h1>window 객체</h1>
        <h3>open 메소드 확인</h3>
        <div>
                <a href="#" onclick="open1(); return false;">새 창 열기</a>
                <br/>
                <a href="#" onclick="open2(); return false;">팝업 창 열기(1)</a>
                <br/>
                <!-- 파라미터를 포함한 함수의 호출 -->
                <a href="#" onclick="open3('http://www.naver.com'); return false;”>naver 창 열기(2-1)</a>
                <br/>
                <a href="#" onclick="open3('http://www.daum.net'); return false;">Daum 창 열기(2-2)</a>
                <br/>
        </div>
</body>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ window 객체 이벤트처리

js_window02.html

```
<script type="text/javascript">
	function open1() {
		/** 새 창(혹은 탭) 띄우기 */
		window.open('01-open.html');
	}
	function open2() {
		/** 클릭할 때 마다 창이 새로 열리는 팝업창 */
		window.open('01-open.html', '', 'width=300, height=500, scrollbars=no, toolbar=no,
				menubar=no, status=no, location=no');
	}
	function open3(url) {
		/** 한번 생성된 팝업창을 지속적으로 재 사용하는 팝업창 */
		// 팝업의 주소를 파라미터로 전달받는다.
		window.open(url,'mywin','width=500, height=300, scrollbars=no, toolbar=no, menubar=no,
				status=no, location=no');
	}
</script>
```

❖ 브라우저 제어를 위한 내장 객체

◆ location 객체

* 웹브라우저의 주소표시줄에 표시되는 URL로부터 다양한 정보를 추출할 수 있는 속성값들을 가지고있다.

* location 객체 기본 속성

속성이름	설 명
href	문서의 URL주소(주소 표시줄에 표시되는 URL 전체를 의미)
host	주소표시줄의 URL에서 호스트이름과 포트를 조회 (예] www.iedu.or.kr:8080)
hostname	호스트 이름을 조회한다. 일반적으로 URL에서 도메인을 조회할 수 있다.
hash	앵커이름을 조회한다. URL에 “#” 기호와 함께 표시되는 단어를 의미
pathname	디렉터리 이하 경로를 조회
port	포트 번호를 조회
protocol	프로토콜의 종류를 조회 - 웹 주소에서 “http:” 혹은 “https:”를 조회
search	URL에 포함된 파라미터를 조회

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ location 객체 이벤트처리

js_location01.html

```
<head>
<meta charset="utf-8" />
<script type="text/javascript">
        document.write("<p>문서의 URL주소: " + location.href + "</p>");
        document.write("<p>호스트 이름과 포트: " + location.host + "</p>");
        document.write("<p>호스트 컴퓨터 이름: " + location.hostname + "</p>");
        document.write("<p>앵커이름: " + location.hash + "</p>");
        document.write("<p>디렉토리 이하 경로: " + location.pathname + "</p>");
        document.write("<p>포트번호 부분: " + location.port + "</p>");
        document.write("<p>프로토콜 종류: " + location.protocol + "</p>");
        document.write("<p>URL 조회부분: " + location.search + "</p>");
</script>
</head>
<body>
        <a href="js_location.html?a=1&b=2#top">이 링크로 다시 실행하세요.</a>
</body>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ location 객체 이벤트처리

js_location02.html

```
<head>
<meta charset="utf-8" />
<script type="text/javascript">
	function goNaver() {
		if (confirm("정말 네이버로 이동하시겠습니까?")) {
			location.href= "http://www.naver.com";
		}
	}
</script>
</head>
<body>
	<input type="button" value="네이버로 이동하기" onclick="goNaver()" />
</body>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ location 객체 이벤트처리

js_location03.html

```
<head>
<meta charset="utf-8" />
<script type="text/javascript">
	/** 두 수 사이의 난수를 리턴하는 함수 */
	function random(n1, n2) {
		return parseInt(Math.random() * (n2 - n1 + 1)) + n1;
	}
	/** 5자리의 인증번호를 id값이 “auth”인 요소에게 출력 */
	function authNo() {
		var value = "";
		for (var i = 0; i < 5; i++) {
			value += random(0, 9);
		}
		document.getElementById("auth").innerHTML = value;
	}
	/** 페이지 새로 고침 */
	function refresh() {
		location.reload();
	}
</script>
</head>
<!-- 페이지 최초 로딩시, authNo() 함수 호출 -->
<body onload="authNo()">
	<p>
		<!-- strong 태그 안이 자바스크립트 출력 부분 -->
		고객님의 인증번호는 <strong id="auth">00000</strong>입니다.
	</p>
	<!-- 페이지 새로고침 이벤트 호출 -->
	<input type="button" value="인증번호 새로 받기" onclick="refresh()"/>
</body>
```

❖ 브라우저 제어를 위한 내장 객체

◆ history 객체

* 브라우저의 '뒤로', '앞으로' 버튼의 기능을 수행하는 객체로 back() 함수와 forward() 함수를 내장

* 이전 페이지로 이동

```
history.back();
```

* 다음 페이지로 이동

```
histroy.forward();
```

❖ history 객체 이벤트처리

js_history01.html

```
<body>
	<h1>History 객체</h1>
	<a href="js_history02.html">페이지 이동</a><br />
	<a href="#" onclick="history.forward(); return false;">앞 페이지로 이동</a>
</body>
```

js_history02.html

```
<body>
	<h1>History 객체</h1>
	<a href="#" onclick="history.back(); return false;">이전 페이지로 이동</a>
</body>
```

❖ 브라우저 제어를 위한 내장 객체

◆ navigator 객체

* 브라우저의 정보를 조회하기 위한 속성들을 가지고 있는 객체
* 하나의 웹 서비스로 다양한 브라우저를 지원할 수 있도록 처리하기 위한 기준이 되는 정보를 추출할 수 있다.

* navigator 객체의 주요 속성

속성이름	설 명
appName	브라우저의 이름
appCodeName	브라우저의 코드명
platform	브라우저가 설치된 시스템(클라이언트의)의 환경
userAgent	웹 브라우저의 종류와 버젼 (가장 포괄적)
appVersion	웹 브라우저의 버젼

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ navigator 객체

js_navigator01.html

```
<script type="text/javascript">
        var info = "<h1>웹 브라우저 정보 확인</h1>";
        info += "<p>브라우저 이름 : " + navigator.appName + "</p>";
        info += "<p>브라우저 코드명 : " + navigator.appCodeName + "</p>";
        info += "<p>플랫폼 정보 : " + navigator.platform + "</p>";
        info += "<p>사용자 정보 : " + navigator.userAgent + "</p>";
        info += "<p>브라우저 버전 : " + navigator.appVersion + "</p>";
        document.write(info);
</script>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ navigator 객체

js_navigator02.html

```
<script type="text/javascript">
        /** 모바일 브라우저이면 true, 그렇지 않으면 false를 리턴하는 사용자 정의 함수 */
        function isMobile() {
                var tmpUser = navigator.userAgent;
                var isMobile = false;
                // userAgent값에 iPhone, iPad, iPod, Android 라는 문자열이 하나라도 검색되면, 모바일로 간주한다.
                if (tmpUser.indexOf("iPhone") > 0 || tmpUser.indexOf("iPad") > 0 ||
                                                tmpUser.indexOf("iPod") > 0 || tmpUser.indexOf("Android ") > 0) {
                        isMobile = true;
                }
                return isMobile;
        }
        var isMobileWeb = isMobile();
        if (isMobileWeb) {
                document.write("<h1>모바일 웹 브라우저로 접속하셨습니다.</h1>");
        } else {
                document.write("<h1>PC용 웹 브라우저로 접속하셨습니다.</h1>");
        }
</script>
```

❖ navigator 객체

js_navigator03.html

```
<script type="text/javascript">
        /* 브라우저의 이름을 리턴하는 함수 */
        function getWebBrowserName() {
                // userAgent값을 모두 소문자로 변환하여 변수에 대입
                var agt = navigator.userAgent.toLowerCase();
                // 특정 브라우저의 이름이 검색되는지 여부를 판별하여
                // 특정 이름이 검색될 경우, 브라우저 이름을 의미하는 문자열을 리턴
                if (agt.indexOf("chrome") != -1) {
                        return 'Chrome';
                } else if (agt.indexOf("opera") != -1) {
                        return 'Opera';
                } else if (agt.indexOf("firefox") != -1) {
                        return 'Firefox';
                } else if (agt.indexOf("safari") != -1) {
                        return 'Safari';
                } else if (agt.indexOf("skipstone") != -1) {
                        return 'SkipStone';
                } else if (agt.indexOf("msie") != -1 || agt.indexOf("trident") != -1) {
                        return 'Internet Explorer';
                } else if (agt.indexOf("netscape") != -1) {
                        return 'Netscape';
                } else {
                        return "Unknown";
                }
        }
        document.write("<h1>" + getWebBrowserName() + "</h1>");
</script>
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 브라우저 제어를 위한 내장 객체

◆ screen 객체

* 장치의 디스플레이 정보를 조회할 수 있는 변수들을 내장한 객체
* 함수를 내장하고 있지 않다.

* navigator 객체의 주요 속성

속성이름	설 명
availHeight	화면 높이
availWidth	화면 너비
colorDepth	색상 수
height	픽셀당 높이
width	픽셀당 너비
pixelDepth	픽셀당 비트수 (IE9 미만 버전은 지원 안함)

❖ screen 객체

js_screen01.html

```
<script type="text/javascript">
	/** screen 객체의 정보 조회 */
	document.write("<h1>화면 높이 : " + screen.availHeight + "</h1>");
	document.write("<h1>화면 너비 : " + screen.availWidth + "</h1>");
	document.write("<h1>색상  수  : " + screen.colorDepth + "</h1>");
	document.write("<h1>픽셀당 높이 : " + screen.height + "</h1>");
	document.write("<h1>픽셀당 너비 : " + screen.width + "</h1>");
	document.write("<h1>픽셀당 비트수(IE는 지원안함) : " + screen.pixelDepth + "</h1>");
</script>
```

js_screen02.html

```
<script type="text/javascript">
	/* 화면의 중심 좌표를 가로/세로 형태의 배열로 리턴하는 함수 */
	function getCenterPixcel() {
		var center = new Array(
			parseInt(screen.availWidth/2), parseInt(screen.availHeight/2)
		);
		return center;
	}
	var screenCenter = getCenterPixcel();
	document.write("<h1>모니터 화면의 중심 좌표: x=" + screenCenter[0] + "px, y=" +
screenCenter[1] + "px</h1>");
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ 브라우저 제어를 위한 내장 객체

◆ document 객체

* Javascript에서 가장 핵심적인 객체
* HTML 문서의 구조나 내용을 제어하기 위한 기본 기능을 가지고 있다.
* 하위 객체로 image, form 등을 가지고 있다.

❖ 특정 HTML 요소를 객체 형태로 가져오기

```
var 변수이름 = document.getElementById(“아이디이름”);
var 변수이름 = document.getElementsByClassName(‘클래스이름’); - 배열로 반환
. . . css 선택자를 사용할 수 있다.
```

❖ 특정 HTML 요소 내용 제어

```
변수이름 = document.getElementById(‘아이디이름’).innerHTML; // 내용 읽기
document.getElementById(‘아이디이름’).innerHTML = 데이타;      // 내용 쓰기
```

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ document 객체

◆ HTML 요소의 CSS 제어

```
document.getElementById("아이디이름").style.CSS속성 = " 속성값 ";
```

❖ 배경 관련 속성

- **Javascript의 CSS 대응 속성 이름은 CSS 속성에서 "-" 가 붙는 경우 빼고 뒷 단어 첫글자만 대문자로 해준다. (카멜(camel)표기법을 따른다.)**

Javascript 속성이름	CSS 속성이름	설 명
backgroundColor	**background-color**	배경 색상 지정
backgroundImage	**background-image**	배경 이미지 경로 지정
backgroundPosition	**background-position**	배경 이미지 위치 지정
backgroundRepeat	**backgroudn-repeat**	배경 이미지 반복 속성 지정

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ document 객체

◆ image 객체

❖ image 객체 기본 속성

속성이름	설 명
src	객체가 표시하고 있는 이미지 파일의 경로 설정
width	객체가 표시하고 있는 이미지의 가로 너비 지정
height	객체가 표시하고 있는 이미지의 세로 높이 지정

훈련교사 : 전 은 석
email : euns_jun@naver.com

❖ image 객체

js_image01.html

```
<style type="text/css">
        /** 목록 정의 초기화 및 목록 박스 좌측 배치 */
        .thumbnail {
                padding: 0; margin: 0;
                list-style: none;
                width: 120px; float: left;
        }
        /** 목록의 각 항목에 대한 크기 및 여백 설정 */
        .thumbnail li {
                width: 100px; height: 100px; padding: 5px 10px;
        }
        /** 썸네일 이미지의 크기 설정 */
        .thumbnail img {
                width: 100px; height: 100px;
        }
        /** 큰 이미지 영역의 배치와 크기, 여백 설정 */
        .view {
                float: left; width: 500px; height: 320px; padding: 5px 0;
        }
        /** 큰 이미지의 크기 설정 */
        .view img {
                width: 500px; height: 320px;
        }
</style>
</head>
<body>
        <!-- 각 이미지를 포함하는 링크를 클릭했을 경우, setImage 함수 호출 -->
        <ul class="thumbnail">
                <li><a href="#" onclick="setImage(0); return false;"><img src="img/1.jpg" /></a></li>
                <li><a href="#" onclick="setImage(1); return false;"><img src="img/2.jpg" /></a></li>
                <li><a href="#" onclick="setImage(2); return false;"><img src="img/3.jpg" /></a></li>
        </ul>
        <div class="view">
                <img src="img/1.jpg" id="target" />
        </div>
</body>
```

❖ image 객체

js_image01.html

```
<script type="text/javascript">
        /** 링크에 의해서 호출될 함수 */
        function setImage(index) {
                // 이미지의 경로를 담고 있는 배열
                var image_list = [
                        'img/1.jpg',
                        'img/2.jpg',
                        'img/3.jpg'
                ];
                // 이미지 요소의 객체화
                var image = document.getElementById("target");
                // 객체의 src속성에 배열의 값들 중에서 파라미터로 전달된 위치의 값을 설정한다.
                image.src = image_list[index];
        }
</script>
```

❖ document 객체

◆ form 객체

* 입력 태그들의 입력 제어

❖ form 객체의 접근은 name 값으로 하므로 name속성을 지정해주자.

* id 값에 의한 객체 획득 방법

```
변수 = document.getElementById("id이름");
```

* name값에 의한 객체 획득 방법

```
변수 = document.form-name값;
```