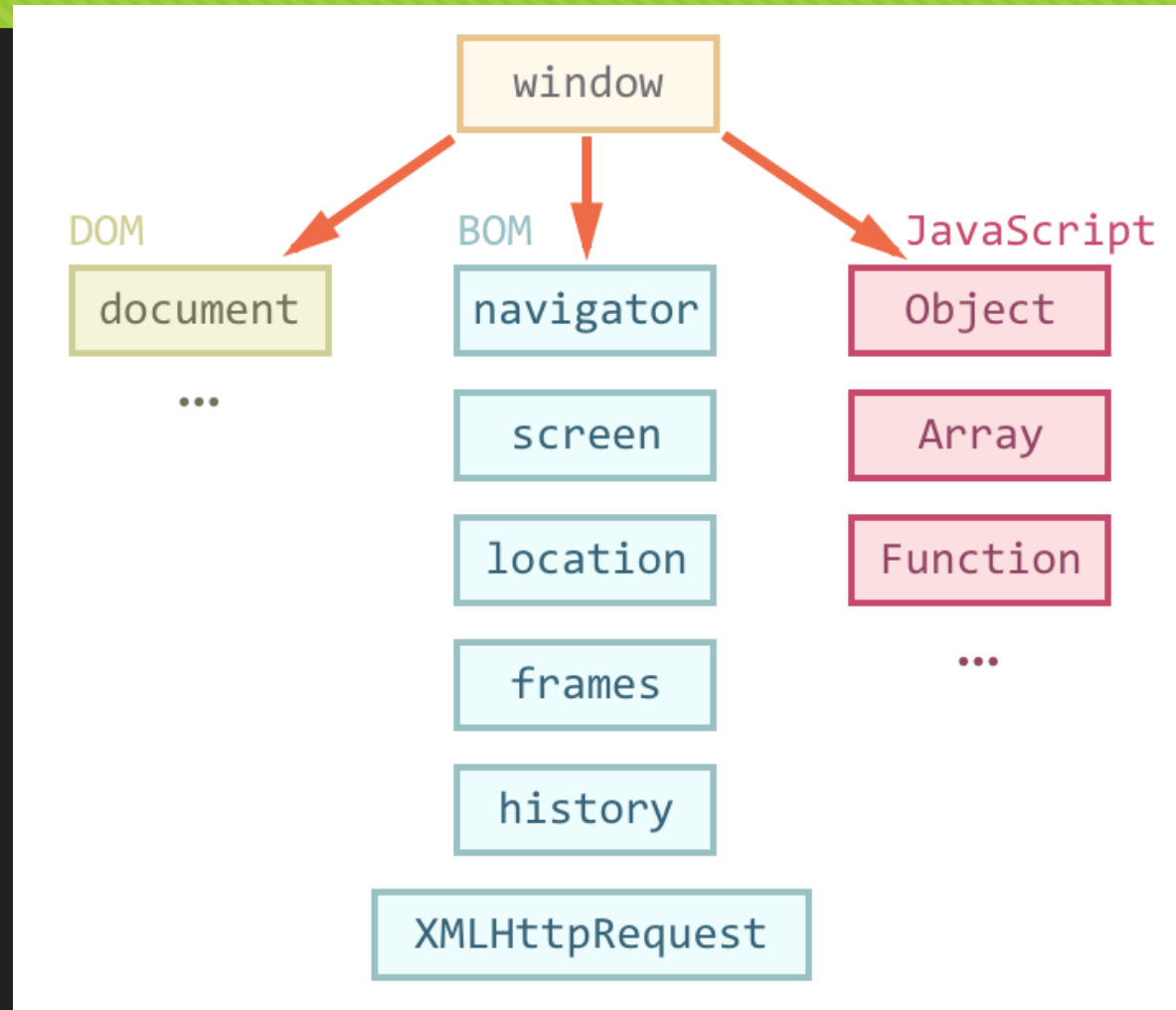


JavaScript

BOM(Browser Object Model) – 김근형 강사

BOM(Browser Object Model)

- BOM(Browser Object Model)



BOM(Browser Object Model)

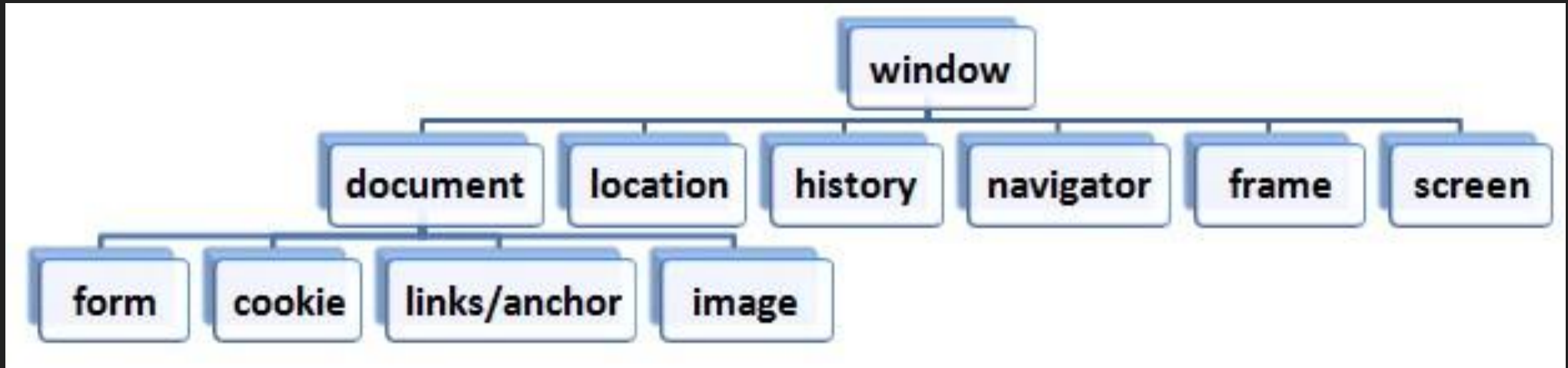
- BOM(Browser Object Model)
 - 자바스크립트가 브라우저와 소통하기 위한 모델
 - 공식 표준은 아니지만 모든 브라우저 대부분이 자바스크립트 상호작용에 있어 비슷한 메소드와 속성으로 동작하기에 이와 같은 메소드들을 통칭하여 BOM이라고 한다.
 - BOM(브라우저 객체 모델)은 웹 브라우저 창을 관리할 목적으로 제공되는 객체 모음을 대상으로 하는 모델로써, 자바 스크립트 등에서 이를 사용할 수 있다.
 - 브라우저 제작사마다 세부사항이 다소 다르게 구현되고 한정적이다.

BOM(Browser Object Model)

- BOM(Browser Object Model) 특징
 - 웹브라우저의 윈도우 객체를 대상으로 하는 윈도우 객체 모델의 일정으로 비표준적이거나 대부분의 웹브라우저는 Netscape3를 표준처럼 많이 따른다.
 - 이를DOM Level 0 이라고도 부르며 웹브라우저를 위한 객체 모델의 표준으로는DOM(문서 객체 모델)이 있다.
 - BOM의 역할은 웹 브라우저의 버튼, URL 주소 입력 창, 타이틀 바 등 웹브라우저 윈도우 및 웹페이지의 일부분을 제어할 수 있게끔 하는 윈도우 객체 모델이다.

BOM(Browser Object Model)

○ BOM(Browser Object Model) 종류



window

- window 오브젝트

- 모든 내장 객체의 상위 오브젝트
- 전역객체(글로벌 객체)라고도 하며 모든 객체를 다 포함하기 때문에 window는 생략이 가능하다.

```
// 위와 아래는 window를 스든 안쓰든 동작을 수행한다.  
window.alert('윈도우를 쓴 alert');  
alert('윈도우를 안쓴 alert');
```

- window 객체 아래를 보면 String, Boolean, Object, Number, Function, Array 같은 자료형이 다 들어있다.

```
// 함수가 아닌 외부에서 만든 변수는 모두 window 객체 안에 등록된다.  
var a = 'hello';  
console.log(window.a); // hello
```

- window 오브젝트의 모든 내용을 브라우저가 다 지원하는 건 아니므로 쓸 때 반드시 확인하고 쓸 것

window

○ window 프로퍼티/객체

이름	설명	이름	설명
status	브라우저의 상태바에 문자열을 출력하는 경우에 사용	closed	창이 닫혀 있는 상태
defaultStatus	브라우저의 상태바에 초기 문자열을 설정	locationbar	location 바
length	창안의 프레임 수	menubar	창 메뉴 바
name	창 이름	innerHeight	창 표시 영역의 높이(픽셀), 익스플로러 지원되지 않음
self	현재 창 자신, window와 같음	innerWidth	창 표시 영역의 너비(픽셀), 익스플로러 지원되지 않음
window	현재 창 자신, self와 같음	outerHeight	창 바깥쪽 둘레의 높이, 익스플로러 지원되지 않음
parent	프레임에서 현재프레임의 상위프레임	outerWidth	창 바깥쪽 둘레의 너비, 익스플로러 지원되지 않음
top	현재프레임의 최상위프레임	pageXOffset	현재 나타나는 페이지의 X위치, 익스플로러 지원되지 않음
opener	open()으로 열린 창에서 볼 때 자기를 연 창	pageYOffset	현재 나타나는 페이지의 Y위치, 익스플로러 지원되지 않음
document	document 오브젝트	personalbar	창의 퍼스널 바
frames	창안의 모든 프레임에 대한 배열정보	scrollbar	창의 스크롤 바
history	history 오브젝트 및 배열	statusbar	창의 상태 바
location	location 오브젝트	toolbar	창의 툴 바

window

○ window 메서드

이름	설명
alert()	경고용 대화상자를 보여줌
clearTimeout()	setTimeout 메소드를 정지
confirm()	확인, 취소를 선택할 수 있는 대화상자를 보여줌
open()	새로운 창을 오픈
prompt()	입력창이 있는 대화상자를 보여줌
setTimeout()	일정 간격으로 함수를 호출하여 수행, millisecond 단위로 지정
eval()	문자열을 숫자로 바꿈
toString()	오브젝트를 문자열로 바꿈
blur()	focus를 이동
focus()	focus를 줌
scroll()	창을 스크롤 함
valueOf()	오브젝트 값을 반환
back()	한 단계 전 URL(이전화면)로 돌아감. 익스플로러 지원 안함
find()	창안에 지정된 문자열이 있는지 확인, 있다면 true 없으면 false. 익스플로러 지원 안함
forward()	한 단계 뒤의 URL(다음화면)로 이동. 익스플로러 지원 안함
home()	초기화 홈페이지로 이동. 익스플로러 지원 안함

window

○ window 메서드

이름	설명
moveby()	창을 상대적인 좌표로 이동. 수평방향과 수직방향의 이동량을 픽셀로 지정
moveto()	창을 절대적인 좌표로 이동. 창의 왼쪽 상단 모서리를 기준으로 픽셀을 지정
resizeby()	창의 크기를 상대적인 좌표로 재설정. 밑변의 모서리를 기준으로 수평방향, 수직방향을 픽셀로 지정
resizeto()	창의 크기를 절대적인 좌표로 재설정. 창 크기를 픽셀로 지정
scrollby()	창을 상대적인 좌표로 스크롤. 창의 표시영역의 수평방향과 수직방향에 대해 픽셀로 지정
scrollto()	창을 절대적인 좌표를 스크롤. 창의 왼쪽 상단 모서리를 기준으로 픽셀로 지정
stop()	불러오기를 중지. 익스플로러는 지원 안함
captureEvents()	모든 타입의 이벤트를 판단
setInterval()	일정시간마다 지정된 처리를 반복
clearInterval()	setInterval 메소드의 정지
handleEvent()	이벤트 취급자를 정함
print()	화면에 있는 내용을 프린터로 출력
releaseEvent()	다른 계층의 이벤트로 이벤트를 넘김
routeEvent()	판단한 이벤트와 같은 계층의 이벤트
toSource()	오브젝트 값을 문자열로 반환

window

○ window 창의 특성

특성명	값	설명
directories	yes no	익스플로러 연결도구모음, 익스플로러 전용
location	yes no	주소입력란
menubar	yes no	메뉴표시줄
scrollbars	yes no	스크롤바
status	yes no	상태표시줄
toolbar	yes no	도구모음
copyhistory	yes no	히스토리정보를 복사
resizable	yes no	창 크기 조절 가능여부
width	픽셀	창의 너비
height	픽셀	창의 높이

window

- window.console

- 브라우저의 디버깅 콘솔에 대한 액세스를 제공하는 콘솔 객체에 대한 참조를 반환한다.
- Console 객체는 아무 전역 객체에서나 접근할 수 있다.

메서드명	설명
Console.error()	웹 콘솔에 에러 메시지를 출력한다
Console.group()	Web Console log 에 새로운 인라인 그룹을 만든다.
Console.groupEnd()	Web Console log 의 기존 인라인 그룹을 나운다.
Console.warn()	경고 메시지를 출력한다. 추가 매개변수와 함께 문자열 치환을 사용할 수 있다.
Console.log()	일반 메시지를 출력한다. 추가 매개변수와 함께 문자열 치환을 사용할 수 있다.
Console.time()	주어진 이름의 타이머를 실행한다. 하나의 페이지에서는 최대 10,000개의 타이머를 동시에 실행할 수 있다.
Console.timeEnd()	지정한 타이머를 멈추고, 소요시간을 출력한다.
Console.trace()	스택 추적을 출력한다.

window

○ window.console

```
console.time('aaa');

console.error('이것은 에러를 출력하는 문구입니다.');
```

```
console.error('%d 열에 에러가 발생했습니다.', 3);

console.log("아웃 레벨");
console.group();
console.log("Level 2");
console.group();
console.log("Level 3");
console.warn("More of level 3");
console.groupEnd();
console.log("Back to level 2");
console.groupEnd();
console.log("아웃레벨 복귀");

console.timeEnd('aaa');
```

```
✖ ▶ 이것은 에러를 출력하는 문구입니다.
✖ ▶ 3 열에 에러가 발생했습니다.

아웃 레벨
▼ console.group
  Level 2
    ▼ console.group
      Level 3
        ⚠ ▶ More of level 3
      Back to level 2
    아웃레벨 복귀
aaa: 2.752197265625ms
```

```
function foo() {
  function bar() {
    console.trace();
  }
  bar();
}

foo();
```

```
▼ console.trace
bar @ VM456:3
foo @ VM456:5
(anonymous) @ VM456:8
```

window

○ window.frames / window.length

프로퍼티명	설명
window.frames	윈도우 내에 iframe을 접근할 수 있는 객체를 제공한다.
window.length	윈도우 내에 iframe의 개수를 알아낸다.

<p>버튼을 누르면 해당 iframe의 페이지가 바뀝니다.</p>

```
<button onclick="myFunction()">Try it</button>  
<br><br>
```

```
<iframe src="http://59.23.150.58/index.php"></iframe>  
<iframe src="http://59.23.150.58/index.php"></iframe>
```

```
<script type="text/javascript">  
  function myFunction() {  
    window.frames[0].location = "http://59.23.150.58/30stair/";  
  }  
  
  console.log(window.length);  
</script>
```

window

○ Window.localStorage / Window.sessionStorage

프로퍼티명	설명
Window.localStorage	사용자 로컬의 Storage객체에 접근하게 한다.
Window.sessionStorage	session Storage 객체를 접근 하게 한다.

- localStorage는 sessionStorage와 비슷하다.
- 유일한 차이점은 localStorage에 저장된 데이터는 만료 기간이 없지만, sessionStorage에 저장된 데이터는 세션이 끝나면(브라우저가 종료되면) 지워진다.
- localStorage또는 sessionStorage에 저장될 데이터는 프로토콜 페이지에 명시되어 있다.
- 모든 key와 value는 항상 string으로 저장된다.

window

○ Window.localStorage / Window.sessionStorage 메서드

메서드명	설명
key(n)	n번째 있는 key 값을 가져온다
length	현재 들어있는 데이터의 개수를 출력한다.
getItem(keyname)	key 이름에 해당하는 데이터를 가져온다
setItem(keyname, value)	keyname에 해당하는 value값을 저장한다.
removeItem(keyname)	keyname에 해당하는 값을 지운다
clear()	storage 안의 모든 key를 지운다

window

- Window.localStorage / Window.sessionStorage

```
// localStorage를 이용한 데이터 저장
localStorage.setItem('myCat', 'Tom');
// 아이템 읽기
var cat = localStorage.getItem('myCat');
console.log(cat); // Tom
// localStorage 아이템 삭제
localStorage.removeItem('myCat');
var cat = localStorage.getItem('myCat');
console.log(cat); // null
// localStorage 아이템 전체 삭제
localStorage.clear();

// sessionStorage에 데이터 저장
sessionStorage.setItem('myDoc', 'foo');
// 아이템 읽기
var doc = sessionStorage.getItem('myDoc');
console.log(doc); // foo
// sessionStorage 아이템 삭제
sessionStorage.removeItem('myDoc');
var doc = sessionStorage.getItem('myDoc');
console.log(doc); // null
// sessionStorage 아이템 전체 삭제
sessionStorage.clear();
```


window

- open / close / opener
 - 윈도우를 열고 닫을 때 쓰는 메서드 및 프로퍼티

메서드 / 프로퍼티	설명
window.open(URL, name [, specs] [, replace])	명시된 리소스를 명시된 이름으로 브라우징 컨텍스트(윈도우, <iframe> 또는 탭)에 로드한다
window.opener	현재 윈도우(window)를 오픈한 윈도우의 레퍼런스(reference)를 반환합니다.
Window.close()	윈도우 창을 닫는다.
window.stop()	윈도우 로딩 시 윈도우를 멈춘다.

window

○ open / close / opener

옵션	설명
URL	웹 문서 URL, 이 인자를 생략하거나 빈 문자열이면 about:blank라는 특별한 빈 페이지 URL이 사용된다.
name	<p>창의 이름을 가리키는 문자열. 이 이름이 이미 사용되고 있고 접근도 허용된다면 해당 창을 사용한다. 그렇지 않다면 지정한 이름을 사용하여 새로운 창을 생성한다. 이 인자가 생략되면 "_blank"가 사용되고 이름 붙지 않은 새 창을 연다.</p> <ul style="list-style-type: none">- _blank : 새 창에 열린다.- _parent : 부모 프레임에 열린다.- _self : 현재 페이지를 대체한다.- _top : 로드된 프레임셋을 대체한다.- name(임의의 이름) : 새 창이 열리고 창의 이름을 지정한다. 동일한 이름에 다시 open() 을 하면 기존의 열린 창의 내용이 바뀐다. 다른 이름을 사용하면 또다른 새 창이 열린다.
replace	<p>히스토리 목록에 새 항목을 만들지 현재 항목을 대체할지 지정한다.</p> <ul style="list-style-type: none">- true : 현재 히스토리를 대체한다.- false : 히스토리에 새 항목을 만든다.

window

○ open / close / opener

옵션	설명
specs	<p>선택적인 값으로 창의 크기, 스크롤여부, 리사이즈 가능 등의 속성을 지정한다. 사용가능한 값을 다음과 같다.</p> <ul style="list-style-type: none">- channelmode=yes no 1 0 : 전체화면으로 창이 열린다. IE에서만 동작한다.- fullscreen=yes no 1 0 : 전체 화면 모드. IE에서만 동작한다.- height=pixels : 창의 높이를 지정한다.(height=600)- width=pixels : 창의 너비를 지정한다.(width=500)- left=pixels : 창의 화면 왼쪽에서의 위치를 지정한다. 음수는 사용할 수 없다.- top=pixels : 창의 화면 위쪽에서의 위치를 지정한다. 음수는 사용할 수 없다.- location=yes no 1 0 : 주소 표시줄 사용여부를 지정한다. Opera에서만 동작한다.- menubar=yes no 1 0 : 메뉴바 사용여부를 지정한다.- resizable=yes no 1 0 : 창의 리사이즈 가능 여부를 지정한다. IE에서만 동작한다.- scrollbars=yes no 1 0 : 스크롤바 사용여부를 지정한다. IE, Firefox, Opera에서 동작한다.- status=yes no 1 0 : 상태바를 보여줄지 지정한다.- titlebar=yes no 1 0 : 타이틀바를 보여줄지 지정한다. 호출 응용 프로그램이 HTML 응용 프로그램이거나 신뢰할 수 있는 대화 상자가 아니면 무시된다.- toolbar=yes no 1 0 : 툴바를 보여줄지 지정한다. IE, Firefox에서 동작한다.

window

○ 길이 관련된 프로퍼티

프로퍼티	설명
window.innerWidth	브라우저 내용의 넓이
window.innerHeight	브라우저 내용의 높이
window.outerWidth	브라우저 전체 넓이
window.outerHeight	브라우저 전체 높이
window.pageXOffset	현재 나타나는 페이지의 x 좌표
window.pageYOffset	현재 나타나는 페이지의 y 좌표
window.scrollX	스크롤 x축의 위치 반환
window.scrollY	스크롤 y축의 위치 반환
window.screenX	창의 x좌표 반환
window.screenY	창의 y좌표 반환

window

○ 입력 관련 메서드

메서드	설명
<code>window.alert([message]);</code>	확인 버튼과 선택적 메시지를 가진 경고 대화 상자를 띄운다.
<code>window.confirm(message);</code>	옵션인 메시지와 확인과 취소 버튼으로 구성된 모달창을 화면에 보여준다.
<code>window.prompt(message, default);</code>	사용자가 텍스트를 입력할 수 있도록 안내하는 선택적 메시지를 갖고 있는 대화 상자를 띄운다.

○ base64 인코딩 디코딩

메서드	설명
<code>window.atob(base64str)</code>	base64타입의 문자열을 디코딩한다
<code>window.btoa(str)</code>	일반 문자열을 base64타입으로 인코딩한다

window

○ 입력 관련 메서드

메서드	설명
window.focus()	해당 객체가 활성화 되었을 시 이벤트
window.blur()	해당 객체가 비활성화 되었을 시 이벤트

○ Interval, timeout

메서드	설명
window.setInterval()	일정 간격으로 지속적으로 실행문을 실행시킬 때 사용한다.
window.clearInterval(interval)	setInterval 설정을 취소한다.
window.setTimeout()	일정 간격으로 한 번만 실행문을 실행시킬 때 사용한다.
window.clearTimeout(timeout)	setTimeout 설정을 취소한다.

window

○ 기타 이동 및 사이징 처리 메서드

메서드	설명
window.moveTo(x, y)	x, y 좌표로 이동한다
window.moveBy(x, y)	현재 좌표에서 x, y로 이동한다
window.resizeTo(x, y)	x, y 만큼 윈도우 크기를 조절한다
window.resizeBy(x, y)	현재 크기에서 x, y 만큼 윈도우 크기를 조정한다.
window.scrollTo(x, y)	웹 좌측 상단이 x, y 좌표로 스크롤을 이동시킨다.
window.scrollBy(x, y)	현재 웹 위치에서 x, y 좌표로 스크롤을 이동시킨다.

screen

- screen 객체
 - screen 객체는 사용자의 모니터 정보를 제공하는 객체이다.
 - 모니터의 너비가 높이 또는 컬러 표현 bit를 반환한다.

메소드	내용
screen.width	화면의 너비값을 반환한다.
screen.height	화면의 높이값을 반환한다.
screen.availWidth	작업 표시줄을 제외한 화면의 너비값을 반환한다.
screen.availHeight	작업 표시줄을 제외한 화면의 높이값을 반환한다.
screen.colorDepth	한 색상을 표시하는데 필요한 bit 값을 반환한다.
screen.pixelDepth	화면의 픽셀 깊이를 반환한다.(최신 컴퓨터는 색상 수와 픽셀 수가 동일하다)

screen

○ screen 객체

```
document.write( "<p>screen.width : " + screen.width + "</p>" );  
document.write( "<p>screen.height : " + screen.height + "</p>" );  
document.write( "<p>screen.availWidth : " + screen.availWidth + "</p>" );  
document.write( "<p>screen.availHeight : " + screen.availHeight + "</p>" );  
document.write( "<p>screen.colorDepth : " + screen.colorDepth + "</p>" );  
document.write( "<p>screen.pixelDepth : " + screen.pixelDepth + "</p>" );
```

screen.width : 1920

screen.height : 1080

screen.availWidth : 1920

screen.availHeight : 1040

screen.colorDepth : 24

screen.pixelDepth : 24

location

○ location 객체

- location 객체는 브라우저의 위치를 의미 하는게 아니라 url과 관련된 인터넷 주소를 의미한다.
- 즉 인터넷 상에서의 위치가 어디인지를 결정한다. 또한 새로고침 역시 제공하고 있다.

메소드	내용
location.href	주소 영역에 참조 주소를 설정하거나 URL을 반환한다.
location.hash	URL에 해시값(#에 명시된 값)을 반환한다. http://iwantadmin.tistory.com#top
location.hostname	URL에 호스트 이름을 설정하거나 반환한다.
location.host	URL에 호스트 이름과 포트 번호를 가져온다.
location.port	URL에 포트 번호를 반환한다.
location.protocol	URL에 프로토콜을 반환한다.
location.search	URL에 쿼리(요청 값)을 반환한다. http://www.foo.com?id=Foo&pw=123123
location.reload()	새로 고침이 일어난다.

history

- history 객체
 - 브라우저 내의 기록에 접근할 수 있는 객체
 - 이전에 내가 접근했던 웹 페이지에 대한 기록들을 전부 담고 있으며 해당 페이지로의 이동도 제공한다.
 - history 객체는 아래와 같은 기능을 담고 있다.

메소드	설명
back()	현재 페이지에서 이전에 왔던 페이지를 호출한다.
forward()	현재 페이지에서 다음에 왔던 페이지를 호출한다.
go(number)	number 만큼 이전 페이지로 이동하거나 다음 페이지로 이동한다. 양수일 경우 해당 숫자 만큼 다음 페이지 이동을 하며 음수일 경우 해당 숫자 만큼 이전 페이지로 이동을 한다. 1과 -1은 forward()와 back()과 동일한 역할을 한다.

history

- history 객체
 - SPA(Single Page Application) 방식으로 웹 페이지가 나오면서 세션 내 주소 history를 임의대로 변경해야 하는 일이 발생
 - html5에서는 이러한 점을 고려하여 pushState와 replaceState를 마련함
 - 페이지에 대한 상태를 추가하거나 변경하여 페이지가 마치 이동한 듯한 효과를 주게 만든다.

history

- history 객체

- pushState와 replaceState 사용 방법은 아래와 같다.

```
history.pushState/history.replaceState(data, title, [, url]);
```

- 첫번째 데이터는 저장될 데이터이며 history.state에서 확인이 가능하다.
 - 두번째 데이터는 해당 방문 기록 세션의 제목을 지정한다.
 - 세번째 데이터는 url이며 해당 화면에 고유한 URL을 지정할 경우 사용한다.

history

○ history 프로퍼티/메서드/이벤트

프로퍼티 명	설명
state	history 세션에 저장된 데이터를 출력한다.

메서드 명	설명
pushState(data, title, [, url])	현재 페이지의 상태에 대한 항목을 추가한다.
replaceState(data, title, [, url])	현재 페이지에 대한 상태를 변경한다.

프로퍼티 명	설명
popstate	세션 기록 항목을 브라우저에 로드할 때 발생

navigator

○ navigator 객체

- navigator 객체는 클라이언트가 사용하는 브라우저 정보와 운영체제의 정보를 제공하는 객체이다.

프로퍼티명	내용
navigator.appCodeName	브라우저 코드명을 반환한다.
navigator.appName	브라우저 이름을 반환한다.
navigator.appVersion	브라우저 버전 정보를 반환한다.
navigator.cookieEnabled	쿠키를 브라우저에서 사용 가능한지 여부를 반환한다.
navigator.geolocation	사용자의 위치를 찾는 데 사용할 수 있는 Geolocation 객체를 반환한다
navigator.language	브라우저 사용 언어를 반환한다.
navigator.onLine	브라우저가 온라인 상태인지 확인한다.
navigator.product	브라우저 사용 엔진 이름을 반환한다.
navigator.platform	브라우저를 실행하는 운영체제를 반환한다.
navigator.userAgent	브라우저와 운영체제 종합 정보를 반환한다.

navigator

○ navigator 객체

```
<div id="demo"></div>

<script type="text/javascript">
    var txt = "";
    txt += "<p>Browser CodeName: " + navigator.appCodeName + "</p>";
    txt += "<p>Browser Name: " + navigator.appName + "</p>";
    txt += "<p>Browser Version: " + navigator.appVersion + "</p>";
    txt += "<p>Cookies Enabled: " + navigator.cookieEnabled + "</p>";
    txt += "<p>Browser Language: " + navigator.language + "</p>";
    txt += "<p>Browser Online: " + navigator.onLine + "</p>";
    txt += "<p>Browser Product: " + navigator.product + "</p>";
    txt += "<p>Platform: " + navigator.platform + "</p>";
    txt += "<p>User-agent header: " + navigator.userAgent + "</p>";

    document.getElementById("demo").innerHTML = txt;
</script>
```

Browser CodeName: Mozilla

Browser Name: Netscape

Browser Version: 5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.79 Safari/537.36

Cookies Enabled: true

Browser Language: ko-KR

Browser Online: true

Browser Product: Gecko

Platform: Win32

User-agent header: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.79 Safari/537.36

Geolocation

○ Geolocation

- Geolocation 객체는 사용자가 자신의 위치를 웹 애플리케이션에 제공한다.
- 사생활 보호를 위해 사용자에게 위치 정보를 접근할 수 있는 권한을 요청한다.
- Geolocation API는 Navigator.geolocation 개체를 통해 게시된다.

프로퍼티명	내용
coordinates	장치의 위치 및 고도를 반환한다
position	지정된 시간에 관련 장치의 위치를 반환한다
positionError	Geolocation 장치를 사용할 때 오류가 발생하는 이유를 반환한다
positionOptions	Geolocation.getCurrentPosition() 및 Geolocation.watchPosition()의 매개 변수로 전달할 옵션 속성이 포함된 개체를 설명한다

Geolocation

○ Geolocation

메서드명	내용
clearWatch()	이전에 Geolocation.watchPosition()를 사용하여 설치한 위치/오류 모니터링 핸들러 등록 취소
getCurrentPosition()	장치의 현재 위치 반환
watchPosition()	처리기 등록을 취소하는 데 사용할 수 있는 watch ID 값을 Geolocation.clearWatch() 메서드에 전달하여 반환

Geolocation

○ Coordinates Properties

프로퍼티명	내용
coordinates.latitude	위치의 위도를 10진수 단위로 반환
coordinates.longitude	위치 경도를 10진수 단위로 반환
coordinates.altitude	해수면을 기준으로 위치의 고도를 미터 단위로 반환
coordinates.accuracy	위도 및 경도 속성의 정확도(미터)
coordinates.altitudeAccuracy	고도 속성의 정확도(미터)
coordinates.heading	장치가 이동하는 방향을 반환한다. 도 단위로 지정되는 이 값은 장치가 북쪽 방향에서 얼마나 떨어져 있는지를 나타낸다. 0도는 북쪽을 나타내며, 방향은 시계 방향으로 결정된다 (동쪽은 90도, 서쪽은 270도). 속도가 0이면 헤딩은 NaN이다. 장치가 머리글 정보를 제공할 수 없는 경우 이 값은 null임
coordinates.speed	장치의 속도(초당 미터)를 반환한다. 이 값은 null일 수 있다
position.coords	현재 위치를 정의하는 좌표 객체를 반환한다
position.timestamp	위치가 검색된 시간을 나타내는 DOMTimeStamp 반환

Geolocation

○ Geolocation Example

```
<button onclick="getLocation()">Try It</button>

<p id="demo"></p>

<script type="text/javascript">
    var x = document.getElementById("demo");

    function getLocation() {
        if (navigator.geolocation) {
            navigator.geolocation.getCurrentPosition(showPosition);
        } else {
            x.innerHTML = "Geolocation is not supported by this browser.";
        }
    }

    function showPosition(position) {
        x.innerHTML = "Latitude: " + position.coords.latitude +
            "<br>Longitude: " + position.coords.longitude;
    }
</script>
```