



8

브라우저 객체 모델

모던 웹을 위한 Javascript jQuery 입문

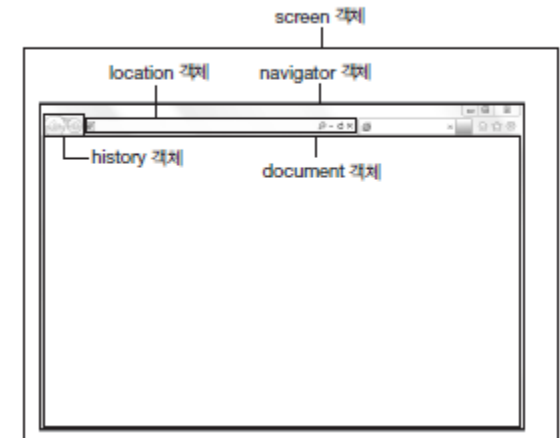
❖ 브라우저 객체 모델이란?

- BOM, Browser Object Model
- 웹 브라우저와 관련된 객체의 집합
 - window, location, navigator, history, screen, document 객체

그림 8-1 대표적인 브라우저 관련 객체



그림 8-2 브라우저 관련 객체



■ 협의의 브라우저 객체 모델

- document 객체문서는 제외
 - 객체 모델DOM, Document Object Model 에 포함



8.1 window 객체

❖ window 객체의 속성 메서드 출력

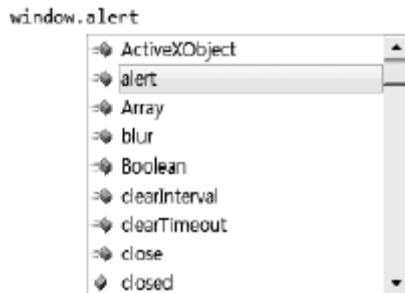
■ 코드 실행 시 에러 발생

- 파이어폭스 4나 오페라 브라우저 사용하면 결과 확인
- window 객체는 많은 속성과 메서드 가짐

코드 8-1 window 객체의 속성

```
<script>
// 출력합니다.
var output = '';
for (var key in window) {
    output += '●' + key + ': ' + window[key] + '\n';
}
alert(output);
</script>
```

그림 8-3 window 객체의 alert() 메서드



8.2 새로운 window 객체 생성

❖ window 객체

- window 객체의 open() 메서드
 - 네 개의 매개 변수 가짐
 - 입력해도 되고 입력하지 않아도 되는 매개 변수를 옵션
 - open() 메서드의 모든 매개 변수는 옵션

표 8-1 window 객체의 윈도우 생성 메서드

메서드 이름	설명
open(URL, name, features, replace)	새로운 window 객체를 생성합니다.

코드 8-2 window 객체의 open() 메서드

```
<script>  
    window.open();  
</script>
```

그림 8-4 새로운 윈도우 생성



8.2 새로운 window 객체 생성

❖ window 객체

- open() 메서드의 모든 옵션을 사용하는 예제

코드 8-3 open() 메서드의 옵션

```
<script>
    window.open('http://hanb.co.kr', 'child', 'width=600, height=300', true);
</script>
```

표 8-2 윈도우 형태 옵션

옵션 이름	설명	입력할 수 있는 값
height	새 윈도우의 높이	픽셀 값
width	새 윈도우의 너비	픽셀 값
location	주소 입력창의 유무	yes, no, 1, 0
menubar	메뉴의 유무	yes, no, 1, 0
resizable	화면 크기 조절 가능 여부	yes, no, 1, 0
status	상태 표시줄의 유무	yes, no, 1, 0
toolbar	상태 표시줄의 유무	yes, no, 1, 0



8.2 새로운 window 객체 생성

❖ window 객체

- open() 메서드의 모든 옵션을 사용하는 예제
 - 첫 번째 매개 변수 - 열고자 하는 HTML 페이지 URL
 - 두 번째 매개 변수 - 윈도우 간 통신을 하는데 사용하는 윈도우 이름
 - 세 번째 매개 변수 - 윈도우를 어떠한 모양으로 출력할지 지정하는 옵션
- 팝업창을 여는 것에서 끝나지 않고 윈도우 객체 리턴
 - 예제 코드 8-4]
 - 새로운 윈도우 객체에 접근해 속성과 메서드 사용 가능

코드 8-4 open() 메서드의 리턴값

```
<script>
// 변수를 선언합니다.
var child = window.open('', '', 'width=300, height=200');

// 출력합니다.
child.document.write('<h1>From Parent Window</h1>');
</script>
```



8.3 window 객체의 타이머 메서드

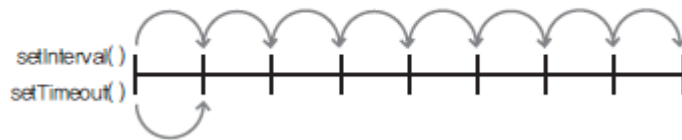
❖ window 객체의 타이머 메서드

- 특정한 시간에 특정한 함수를 실행할 수 있게 하는 메서드
 - setTimeout() 메서드 - 특정한 시간 후에 한 번 함수 실행
 - setInterval() 메서드 - 특정한 시간마다 함수 실행 (메모리 차지)

표 8-3 window 객체의 타이머 메서드

메서드 이름	설명
setTimeout(function, millisecond)	일정 시간 후에 함수를 한번 실행합니다.
setInterval(function, millisecond)	일정 시간마다 함수를 반복해서 실행합니다.
clearTimeout(id)	일정 시간 후에 함수를 한번 실행하는 것을 중지합니다.
clearInterval(id)	일정 시간마다 함수를 반복하는 것을 중단합니다.

그림 8-5 setTimeout() 메서드와 setInterval() 메서드



8.3 window 객체의 타이머 메서드

❖ 예제 코드 8-5]

- 경고창 출력하고 3초 후에 자동으로 윈도우 종료

코드 8-5 타이머 메서드

```
<script>
  // 윈도우가 로드될 때
  window.onload = function () {
    alert('3초후 이 페이지는 종료됩니다.');
```

// 3초 후에 함수를 실행합니다.

```
    window.setTimeout(function () {
      window.close();
    }, 3000);
  };
</script>
```



8.3 window 객체의 타이머 메서드

❖ 타이머 멈추기

- clearTimeout() 메서드와 clearInterval() 메서드 사용
 - setTimeout() 메서드와 setInterval() 메서드 사용하면 타이머 아이디 리턴
 - 타이머 아이디를 매개 변수에 넣어주면 타이머 정지 가능
- 예제 코드 8-6]
 - 10초 동안 화면에 현재 시간 출력하고 중지
 - 타이머 관련 메서드는 함수 형태로 많이 사용

코드 8-6 타이머 정지 메서드

```
<script>
  // 윈도우가 로드될 때
  window.onload = function () {
    // 1초마다 함수를 실행합니다.
    var intervalID = setInterval(function () {
      document.body.innerHTML += '<p>' + new Date() + '</p>';
    }, 1000);

    // 10초 후 함수를 실행합니다.
    setTimeout(function () {
      // 타이머를 종료합니다.
      clearInterval(intervalID);
    }, 10000);
  }
</script>
```



❖ window 객체의 형태와 메서드 제공

- ○○By() 형태의 메서드
 - 현재 윈도우를 기준으로 상대적으로 속성 변화
- ○○To() 형태의 메서드
 - 절대적인 기준으로 속성 변화

표 8-4 window 객체의 메서드

메서드 이름	설명
moveBy(x, y)	윈도우의 위치를 상대적으로 이동합니다.
moveTo(x, y)	윈도우의 위치를 절대적으로 이동합니다.
resizeBy(x, y)	윈도우의 크기를 상대적으로 지정합니다.
resizeTo(x, y)	윈도우의 크기를 절대적으로 지정합니다.
scrollBy(x, y)	윈도우 스크롤의 위치를 상대적으로 이동합니다.
scrollTo(x, y)	윈도우 스크롤의 위치를 절대적으로 이동합니다.
focus()	윈도우에 초점을 맞춥니다.
blur()	윈도우에 초점을 제거합니다.
close()	윈도우를 닫습니다.



8.4 window 객체의 기본 메서드

❖ 상대적/절대적 변경예제

- 새로운 window 객체 생성한 후 왼쪽 상단을 0, 0으로 위치
- setInterval() 함수 사용
 - 1초마다 왼쪽으로 10픽셀, 아래로 10픽셀씩 이동

코드 8-7 상대 이동과 절대 이동의 차이

```
<script>
// 변수를 선언합니다.
var child = window.open('', '', 'width=300, height=200');

child.moveTo(0, 0);

// 1초마다 함수를 실행합니다.
setInterval(function () {
    child.moveBy(10, 10);
}, 1000);
</script>
```



❖ screen 객체

- 웹 브라우저의 화면이 아니라 운영체제 화면의 속성을 가지는 객체
- 모든 브라우저가 공통적으로 가지는 screen 객체 속성

표 8-5 screen 객체의 속성

속성 이름	설명
width	화면의 너비
height	화면의 높이
availWidth	실제 화면에서 사용 가능한 너비
availHeight	실제 화면에서 사용 가능한 높이
colorDepth	사용 가능한 색상 수
pixelDepth	한 픽셀당 비트 수

코드 8-8 screen 객체의 속성

```
<script>
    // 출력합니다.
    var output = '';
    for (var key in screen) {
        output += '●' + key + ': ' + screen[key] + '\n';
    }
    alert(output);
</script>
```



❖ screen 객체 활용 예제

- 팝업창 생성하고 전체화면으로 만듦
- 2초마다 크기를 변환하고 이동해 점점 가운데로 수렴

코드 8-9 브라우저의 크기를 화면에 일치하는 방법

```
<script>
    // 변수를 선언합니다.
    var child = window.open('', '', 'width=300, height=200');
    var width = screen.width;
    var height = screen.height;

    child.moveTo(0, 0);
    child.resizeTo(width, height);

    // 2 초마다 함수를 실행합니다.
    setInterval(function () {
        child.resizeBy(-20, -20);
        child.moveBy(10, 10);
    }, 2000);
</script>
```



8.6 location 객체

❖ location 객체

- 브라우저의 주소 표시줄과 관련된 객체
- location 객체는 프로토콜의 종류, 호스트 이름, 문서 위치 등의 정보
- 객체 출력 예제

코드 8-10 location 객체의 속성

```
<script>
    // 출력합니다.
    var output = '';
    for (var key in location) {
        output += '●' + key + ': ' + location[key] + '\n';
    }
    alert(output);
</script>
```



❖ location 객체

■ 속성과 메서드

표 8-6 location 객체의 속성

속성 이름	설명	예
href	문서의 URL 주소	
host	호스트 이름과 포트 번호	localhost:30763
hostname	호스트 이름	localhost
port	포트 번호	30763
pathname	디렉토리 경로	/Projects/Location.htm
hash	앵커 이름(#~)	#beta
search	요청 매개 변수	?param=10
protocol	프로토콜 종류	http:

표 8-7 location 객체의 메서드

속성 이름	설명
assign(link)	현재 위치를 이동합니다.
reload()	새로고침합니다.
replace(link)	현재 위치를 이동합니다.



❖ location 객체 - 페이지 이동에 많이 쓰임

■ 페이지 이동 방법

- replace() 메서드는 뒤로 가기 버튼 사용할 수 없음
- location = 'http://hanb.co.kr';
- location.href = 'http://hanb.co.kr';
- location.assign('http://hanb.co.kr');
- location.replace('http://hanb.co.kr');

■ 페이지 새로 고침 방법

- location = location;
- location.href = location.href;
- location.assign(location);
- location.replace(location);
- location.reload();



8.7 navigator 객체

❖ navigator 객체

- 웹 페이지 실행하고 있는 브라우저에 대한 정보

표 8-8 navigator 객체의 속성

속성 이름	설명
appCodeName	브라우저의 코드명
appName	브라우저의 이름
appVersion	브라우저의 버전
platform	사용중인 운영체제의 시스템 환경
userAgent	브라우저의 전체적인 정보

코드 8-11 navigator 객체의 속성

```
<script>
// 출력합니다.
var output = '';
for (var key in navigator) {
    output += '●' + key + ': ' + navigator[key] + '\n';
}
alert(output);
</script>
```

그림 8-6 navigator 객체의 속성



8.8 window 객체의 onload 이벤트 속성

❖ window 객체의 onload 속성

- window 객체의 onload 속성 사용
- 이벤트 속성
 - 문서 객체의 속성 중 on으로 시작하는 속성
 - 함수를 할당해야
- 예제 코드 8-12]
 - window 객체가 로드 완료되고 자동으로 할당한 함수 실행

코드 8-12 window 객체의 onload 속성

```
<script>  
    window.onload = function () {  
  
        };  
</script>
```



8.8 window 객체의 onload 이벤트 속성

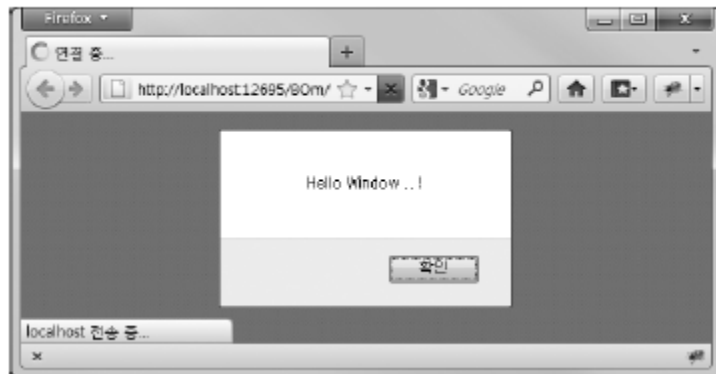
❖ HTML 실행 순서

- 웹 브라우저는 위에서 아래로 태그 차례대로 실행 후 화면 출력
- script 태그를 body 태그 전에 실행

코드 8-13 HTML 페이지의 실행 순서 (1)

```
<!DOCTYPE html>
<html>
<head>
  <script>
    alert('Hello Window .. !');
  </script>
</head>
<body>
  <h1>Hello Window..!</h1>
  <h1>Hello Window..!</h1>
</body>
</html>
```

그림 8-7 head 태그 안에 script 태그를 놓은 경우



8.8 window 객체의 onload 이벤트 속성

❖ HTML 실행 순서

- script 태그가 body 태그 중간에 있는 경우 실행

코드 8-14 HTML 페이지의 실행 순서 (2)

```
<!DOCTYPE html>
<html>
<head>

</head>
<body>
  <h1>Hello Window..!</h1>
  <script>
    alert('Hello Window .. !');
  </script>
  <h1>Hello Window..!</h1>
</body>
</html>
```

그림 8-8 body 태그 내에 script 태그를 놓은 경우



8.8 window 객체의 onload 이벤트 속성

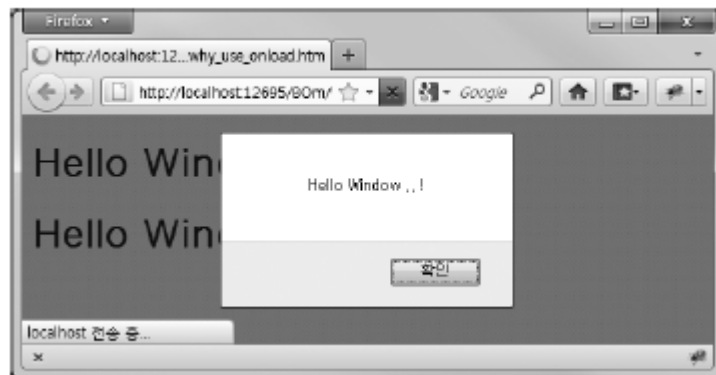
❖ HTML 실행 순서

- window 객체의 onload 이벤트 속성 사용

코드 8-15 window 객체의 onload 이벤트를 사용한 문서 객체 로딩 확인

```
<!DOCTYPE html>
<html>
<head>
  <script>
    window.onload = function () {
      alert('Hello Window .. !');
    };
  </script>
</head>
<body>
  <h1>Hello Window..!</h1>
  <h1>Hello Window..!</h1>
</body>
</html>
```

그림 8-9 window.onload를 사용할 경우





Thank You !

모던 웹을 위한 Javascript jQuery 입문