



Java Script - 03. 브라우저객체

BOM 객체

1. BOM, Browser Object Model: 웹 브라우저와 관련된 객체의 집합
2. window, location, navigator, history, screen, document 객체
3. 협의의 브라우저 객체 모델은 document 객체문서는 제외
4. 객체 모델DOM, Document Object Model 에 포함

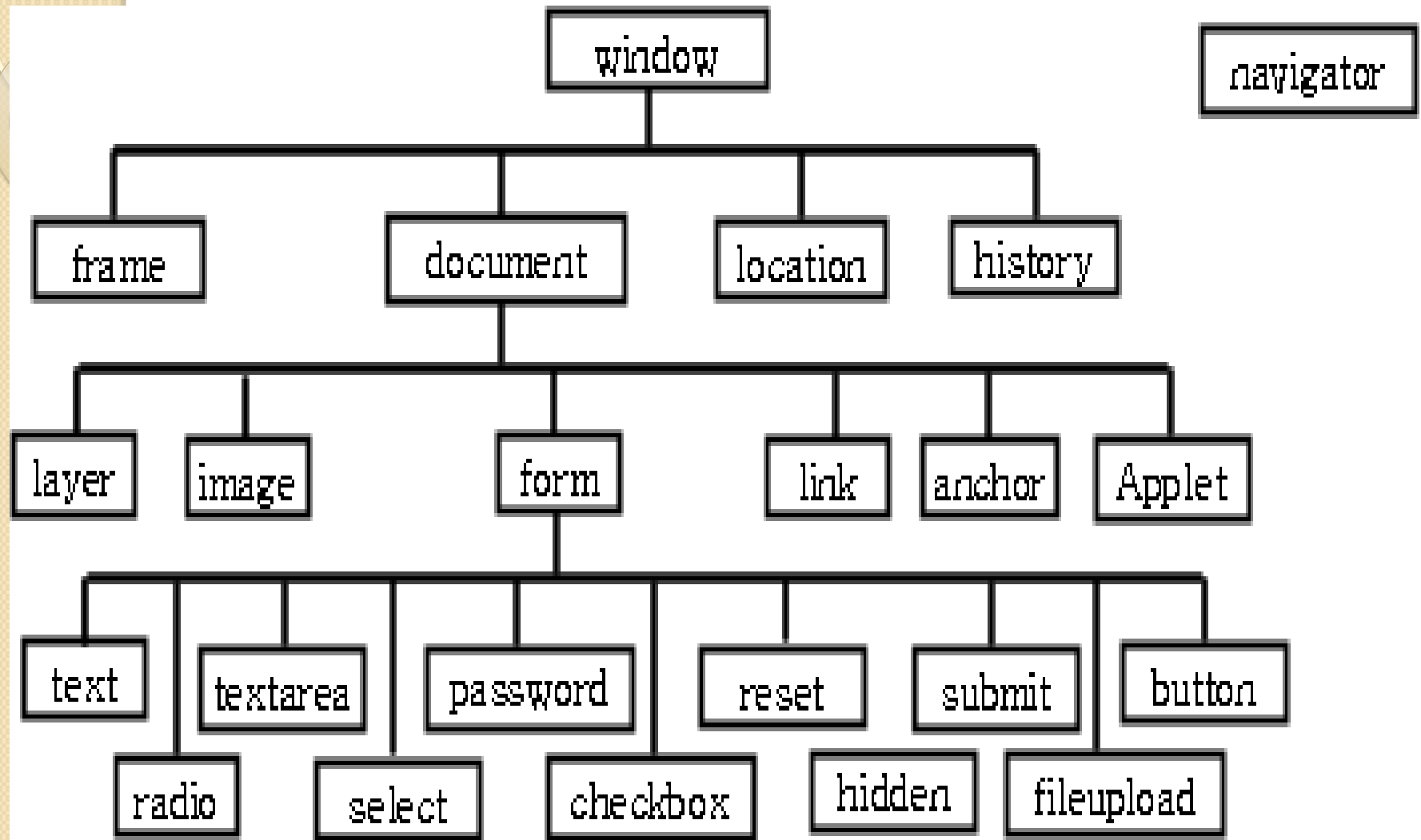
객체	설명
window	창에 대한 전반적인 모든 상황을 제어하는 최상위 객체입니다. 자바스크립트에서 사용되는 모든 객체는 window 객체의 하위에 존재합니다. 예를 들어 상태 바에 메시지를 출력하려면 window 객체로 접근해야 합니다.
document	문서에 대한 정보를 제어하는 객체입니다. 이미지나 폼을 구현할 경우 웹 페이지의 문서에 출력합니다. document 객체는 이미지를 출력하기 위한 image 객체, 입력양식을 구현하기 위한 form 객체 등 다양한 객체들을 하위 객체로 갖고 있습니다.
location	주소 정보를 제어하는 객체입니다.
history	웹 브라우저에 기록되어 있는 히스토리 정보를 제어하는 객체입니다.
navigator	웹 브라우저의 종류를 판별해 주는 객체입니다.

내장 객체

내장 객체는 자바 스크립트로 프로그램을 작성하는 동안 자주 사용되는 것들을 묶어서 미리 정의해 놓은 객체

내장 객체	설 명
Date	날짜와 시간을 처리하기 위한 객체입니다.
Array	배열을 만들어 주는 객체입니다.
String	문자열을 다루기 위한 객체입니다.
Math	수학 계산을 위한 객체입니다.
Screen	화면의 해상도나 색상, 화면의 크기에 관한 정보를 제공하는 객체입니다

브라우저 객체의 계층구조



브라우저 객체의 접근 방법

1. 웹 페이지의 최상위 객체인 window 객체의 하위 객체 중 문서를 제어하는 document 객체의 글자를 출력하는 write() 메소드를 접근하는 것이 원칙

```
window.document.write("출력할 글자");
```

2. 하지만, 모든 객체는 당연히 window 객체의 하위 객체이므로 window를 생략하여 사용함.

```
document.write("출력할 글자");
```

상태 표시바에 글자를 표시하는 defaultstatus/status 속성

1. 웹 브라우저의 상태바에 다양한 형태로 메시지를 제공하려면 window 객체의 status 속성을 사용해야 한다.
2. 문자열은 별도의 지시가 없는 `status` 상태 바에 나타나도록 할 문자열은 defaultstatus 속성에 대입한다

창 속성의 종류

창 속성	속성지정방법	설 명
toolbar	yes/no	도구모임을 보여줄지를 설정합니다.
location		URL를 표시하는 주소 표시줄을 보여줄지를 설정합니다.
status		상태 바를 보여줄지를 설정합니다.
menubar		메뉴 표시줄을 보여줄지를 설정합니다.
scrollbars		스크롤 바를 보여줄지를 설정합니다.
directories		디렉토리 바를 보여줄지를 설정합니다.
resizable		창의 크기 조절 가능하게 할지를 설정합니다.
height	픽셀 수	창의 높이를 픽셀 단위로 지정합니다.
width		창의 너비를 픽셀 단위로 지정합니다.

새로운 창을 열어주는 window.open() 메소드

1. open()은 새로운 창을 만들어 화면에 출력하는 기능을 담당하는 메소드
2. 연결할 URL : 새로 만들어진 창에 로드할 웹 페이지 주소나 HTML 파일을 지정.
3. 창 이름 : 새로 만들어줘 창의 이름을 지정합니다.
4. 창 속성 : 창의 모양이나 특징을 지정합니다.

- window.open("연결할 URL", "창 이름" [, 창의 속성"]);

창을 닫는 window.close() 메소드

1. 열린 창을 닫기 위해서는 window 객체의 close() 메소드를 사용하면 된다.
2. close() 메소드는 매개변수 없이 사용된다.
3. window.close();

setTimeout/closeTimeout 메소드

1. setTimeout는 일정 시간이 지난 후에 “호출함수”를 한번만 실행하는 메소드
 - 타이머ID = setTimeout("호출함수", "지연시간");
2. clearTimeout() 메소드는 setTimeout()로 설정된 내용을 정지하도록 하는 메소드
 - clearTimeout(타이머ID);

setInterval/closeInterval 메소드

1. setInterval() 메소드
 - setTimeout() 메소드와 마찬가지로 일정 시각 각격 후에 명령을 수행.
2. 차이점이 있다면 setTimeout() 메소드는 “호출함수”를 한번 실행하는 반면 setInterval() 메소드는 주어진 시간 간격동안 "호출함수"를 무한정으로 반복 실행.
3. 일정한 간격동안 지속적으로 작업이 일어나기 원할 경우 setInterval 함수를 사용.
타이머ID = setInterval("호출함수", "간격");

하위 창을 열어준 부모 창을 가르키는 opener 속성

opener 속성이 바로 새로 열릴 창에서 자기를 열어준 부모창을 가리키는 속성.

window.opener.속성

window.opener.메소드

프레임의 위치를 지정하는 frames 속성

1. 브라우저 창이 HTML의 <FRAME> 태그에 의해서 여러 개의 프레임으로 나뉜다.
2. 이들 프레임들 중에서 특정 위치의 프레임을 지정하기 위해서 frames 속성을 이용할수 있다.
3. frames 속성은 따로 선언되지 않아도 생성된다.
4. 생성된 프레임 개수만큼 배열 형태로 제공된다.

자식의 상위 계층에 지칭하는 parent 속성

1.frame 객체를 사용하여 프레임의 위치를 설정하기 위해서는 상위 계층부터 차례대로 접근해야 함.

2.왼쪽 프레임에서 상대방의 프레임을 제어하려면 브라우저 객체 계층 구조에 의해 최상위 계층임 window 객체를 지정함 후 프레임을 나눈 상위 계층을 선택하기 위해서 parent 속성으로 접근해야 함.

`window.parent.frames[첨자]`

<FRAME> 태그의 NAME 속성을 프레임 객체로 사용하기

<frame> 태그에서 <frame name="프레임이름">와 같은 방법으로 프레임의 이름을 지정.

프레임 태그의 name 속성으로 부여된 프레임의 이름을 직접 프레임 객체 변수 이름을 사용.

이 경우에는 저절로 생성되는 프레임 배열을 사용하지 않고 바로 “프레임이름”을 프레임 객체로 접근하여 사용할 수 있다.

자신의 상위 계층에 지칭하는 parent 속성

1. parent 속성은 현재 프레임을 기준으로 함 단계 위의 상위 계층을 접근함.
 - 프레임 여러 단계의 계층 구조로 구현한다면 parent로 여러 번 거슬러 올라가야 함.
 - 만일 top이라는 속성을 사용하면 최상위의 계층을 접근할수 있으므로 이를 사용하여 접근하면 간편하게 접근할 수 있다.

window.top.프레임객체이름

자기 자식이 속한 프레임을 지칭하는 self 속성

1. self 속성은 브라우저 창을 여러 개의 프레임으로 나눈 후에 각자 자식의 프레임을 지칭하고자 할 때 사용되는 속성
2. 브라우저 창을 왼쪽과 오른쪽으로 분할한 후, 각자 자식의 프레임에서만 작업을 수행할 때 self 속성으로 자식의 프레임을 제어할 수 있다

history 객체

1. 웹 브라우저를 띄우고 주소 입력란에 웹 페이지의 주소를 입력하여 여러 사이트들을 방문하였다면, 이미 한 번 방문했던 사이트는 브라우저의 도구모음에 있는 [뒤로], [앞으로] 단추로 쉽게 다시 찾아갈 수 있다.

2. 이는 접속했던 사이트의 URL 주소들을 기록하는 히스토리 리스트 덕분.

3. 이러한 편리한 기능을 제공하는 히스토리 리스트를 관리하는 객체가 바로 history 객체.

4. history 객체의 length 속성은 브라우저에 로드되었던 주소를 기억하고 있는 히스토리 리스트의 개수를 알린.

history.length

5. back() 메소드는 히스토리 리스트에 저장된 주소 중에서 현재 로드된 문서를 기준으로 바로 이전에 열었던 페이지로 이동하게 해주는 메소드

- forward() 메소드는 이전 페이지로 이동한 후 다시 앞으로 이동하기 위해 제공되는 메소드

. history.back();

. history.forward();

go() 메소드는 히스토리 리스트에 저장된 주소에서 몇 단계 뒤에 있는 페이지로 이동.

- 1.go() 메소드는 다음에 매개변수로 양의 정수값을 지정하면 이전에 로드되었던 페이지를 여러 단계 거슬러 올라감.
- 2.history.go(3)이라고 지정하면 3단계 앞의 페이지를 로드함.
- 3.반대로 다음 페이지로 이동하고 싶을 때에는 go() 메소드는 다음에 매개변수 음의 정수값을 지정함.
- 4.history.go(-2)라고 지정하면 2단계 다음 페이지로 이동함

주소 입력란의 주소 정보를 제어하는 location 객체

1. 원하는 웹사이트를 방문하기 위해서는 브라우저의 주소 입력란에 주소를 입력하여야한다.

이번 장에서는 주소 입력란에 입력된 주소에 대한 정보를 상세히 알려주는 location 객체를 살펴보자

현재 문서의 URL 주소를 알려주거나 설정하는 href 속성

1. location 객체의 href 속성은 현재 문서의 URL 주소를 알려준다.
2. 이 속성은 원하는 웹 페이지로 이동할 경우에도 사용.
3. href 속성에 URL 주소를 설정하면 원하는 사이트가 자동적으로 접속된다.
4. href 속성에 지정하는 주소는 이중따옴표로 묶임 문자열형태로 제공.
5. 또한 원하는 사이트로 이동할 경우에는 반드시 프로토콜("http://") 을 명시.
location.href="http://원하는 사이트의 주소"

replace() 메소드

1. location 객체의 replace() 메소드는 현재 문서의 URL 주소를 갱신.
2. replace() 메소드의 매개변수에 갱신을 원하는 웹 페이지를 주소를 지정.
- 3.지정한 URL 주소가 현재 문서의 URL 주소로 설정되면서 원하는 사이트가 자동적으로 로드됨.
- 4.URL 주소는 이중따옴표로 묶인 문자열형태로 "http://"을 명시해야 함.
5. replace() 메소드는 이점 페이지의 주소를 기록하지 않으므로 [뒤로] 단추를 사용하여 이전 페이지를 되돌릴 수 없음.
 - location.replace("http://원하는 사이트의 주소");

문서를 다시 읽어 들이는 reload() 메소드

1. 웹 브라우저의 웹 문서의 내용이 변경되었을 경우 이 내용이 나타내기 위한 [새로고침] 단추와 동일한 역할을 함.
2. 현재 로드되어 있는 웹 페이지를 새로 갱신하여 불러오게 해줌.
`location.reload();`

브라우저의 정보를 제공하는 Navigator 객체

1. Navigator 객체는 현재 사용하고 있는 웹 브라우저에 관한 정보를 제공.
주로 브라우저의 종류나 버전을 파악하는데 많이 사용

속성	설명
appName	웹 브라우저의 종류를 알려줍니다. 익스플로러에서는 [Microsoft Internet Explorer]라고 표시하고 넷스케이프에서는 [Navigator]라고 출력합니다.
appCodeName	웹 브라우저의 코드 이름을 알려줍니다. 익스플로러에서는 [Mozilla]나 [MSIE]로 표시하고 넷스케이프에서는 [Mozilla]라고 표시합니다.
platform	사용자의 시스템 환경을 알려줍니다.
userAgent	웹 브라우저의 종류와 버전을 알려줍니다.
appVersion	웹 브라우저의 버전을 알려줍니다. 익스플로러인 경우는 OS의 종류만 표시해 주고 넷스케이프인 경우에는 번과 국가, OS 종류, 국제 버전 등을 알려줍니다.

javaEnabled() 메소드

1. navigator 객체의 javaEnabled() 메소드는 자바를 지원하는지를 알려줌.
2. 자바를 지원하는 경우에는 true를 지원하지 않으면 false를 되돌려줌.

```
navigator.javaEnabled( );
```