Web Development

Lecture 11 – JavaScript 2
Window Object Model

JavaScript: The Definitive Guide by David Flanagan pub:O'Reilly

Object Oriented Software

The Browser is constructed as a collection of objects.

An object can have:

Properties	A set of variables which describe the state of the object.
Methods	A set of functions which can be carried out on the object.
Events	These are triggered by the object when some action takes place.

This collection of objects is known as the Windows Object Model (WOM).

Window

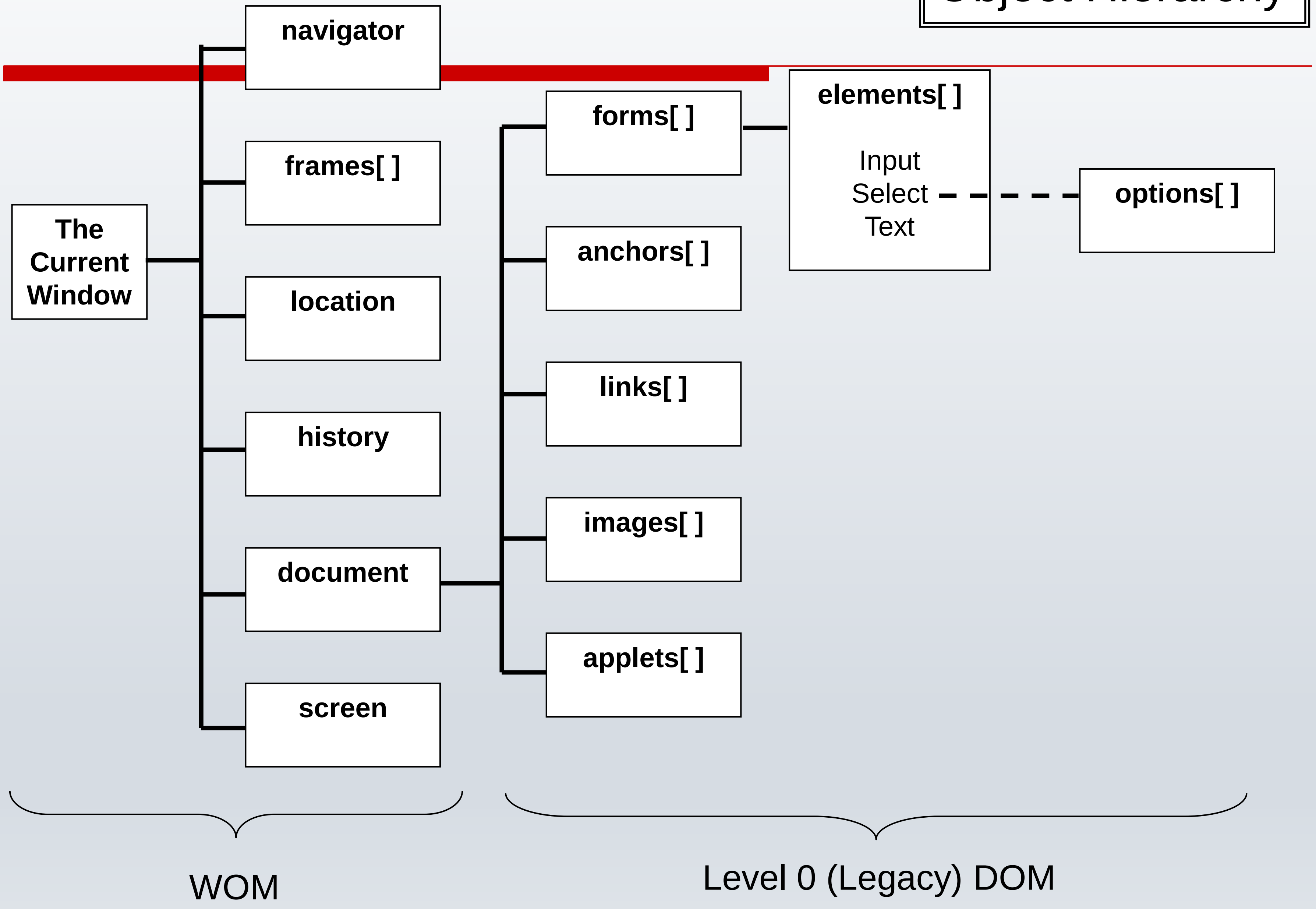
The Window object is at the top of the hierarchy in client-side programming.

It represents the browser window.

It defines a number of properties and methods that allow you to manipulate the browser window.

It also defines a property called 'document', which refers to the web page that is being displayed inside the window.

Object Hierarchy



How to refer to things on the legacy DOM

If you wanted to refer to the third textbox on a form that is on the web page, you would have written:

```
window.document.forms[0].elements[2]
```

You could end up writing things like this:

```
parent.frames[0].document.forms[0].elements[3].options[2].text
```

Note that the 'document' subtree represents the Document Object Model (DOM) of the HTML page that is being displayed inside the window.

Standard DOM

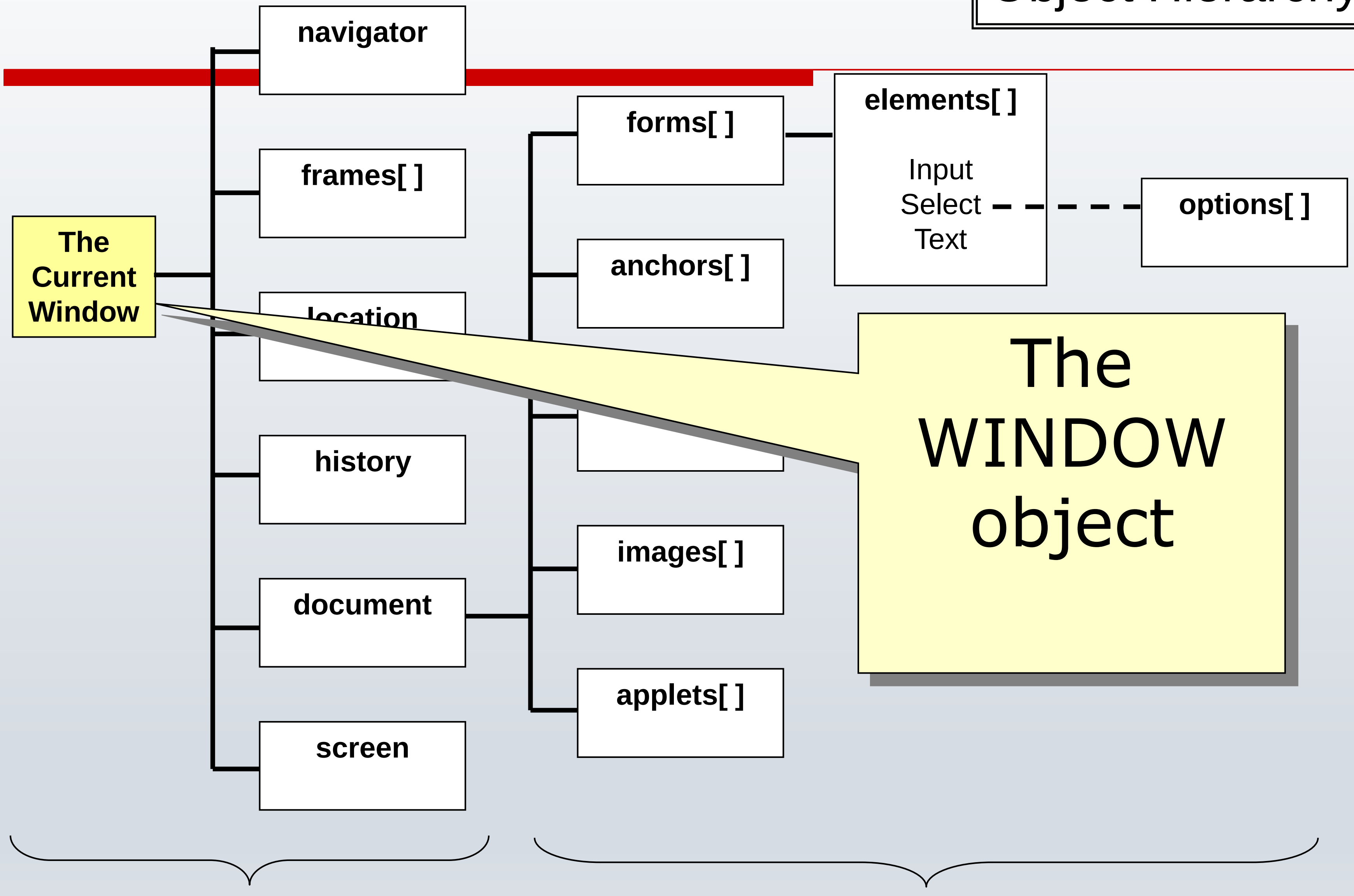
Note that the document objects shown were accepted as standard by all browsers.

When the dust had settled after the browser wars, all the browser manufacturers had more or less agreed to implement this object model for their html documents.

Collectively they are known as Level 0 DOM or the legacy DOM. This made DHTML pages possible.

With the advent of XHTML, the W3C is putting forward a more advanced DOM for web pages – it is more structured and tree-like, and can be used for all XML-type data, not just XHTML web pages.

Object Hierarchy



WOM

Level 0 (Legacy) DOM

Non-Standard WOM

Like the DOM, the JavaScript language is standardised, and all modern browsers support it.

Unfortunately, the Window Object Model is not standardised.

A JavaScript function that runs on one browser may not run on another – it may not even run on a previous version of the same browser.

A web developer needs to be aware of the differences between the various browsers.

http://www.quirksmode.org/dom/w3c_cssom.html
http://webdevout.net/browser_support.php

Window properties – Most browsers (inc IE 9)

outerWidth	{ Overall size of the
outerHeight	{ browser window
screenX	{ Position of the browser
screenY	{ window on the desktop
innerWidth	{ Size of the document
innerHeight	{ window (exc menu bar)
pageXOffset	{ Position of the document
pageYOffset	{ relative to the window

Window properties – Internet Explorer 6, 7, 8

Not supported	{ Overall size of the
Not supported	{ browser window
screenLeft	{ Position of the browser
screenTop	{ window on the desktop
document.body.clientWidth	{ Size of the document
document.body.clientHeight	{ window (exc menu bar)
document.body.scrollLeft	{ Position of the document
document.body.scrollRight	{ relative to the window

Window Methods

alert( )	Display a dialog box
close( )	Close a window
confirm( )	Display a yes or no dialog box
getComputedStyle( )	Get CSS styles that belong to this document
moveBy( )	Move window by relative amount
moveTo( )	Move window to an absolute position
open( )	Creates and opens a new window

Window Methods 2

print( )	Prints the document
prompt( )	Asks for text input with a dialog box
resizeBy( )	Resizes the window by a specified amount
resizeTo( )	Resizes the window to a specified size
scrollBy( )	Scrolls the window by a specified amount
scrollTo( )	Scrolls the window to a specified position.

Window Methods 3 - Timers

You can schedule a function to run after a specified number of milliseconds.

```
setTimeout( functionname, delay );
```

You can also schedule a function to run repeatedly after an interval (in milliseconds).

```
setInterval( functionname, interval );
```

There are two related functions which cancel the timers:

```
clearTimeout( );
clearInterval( );
```

Window Events

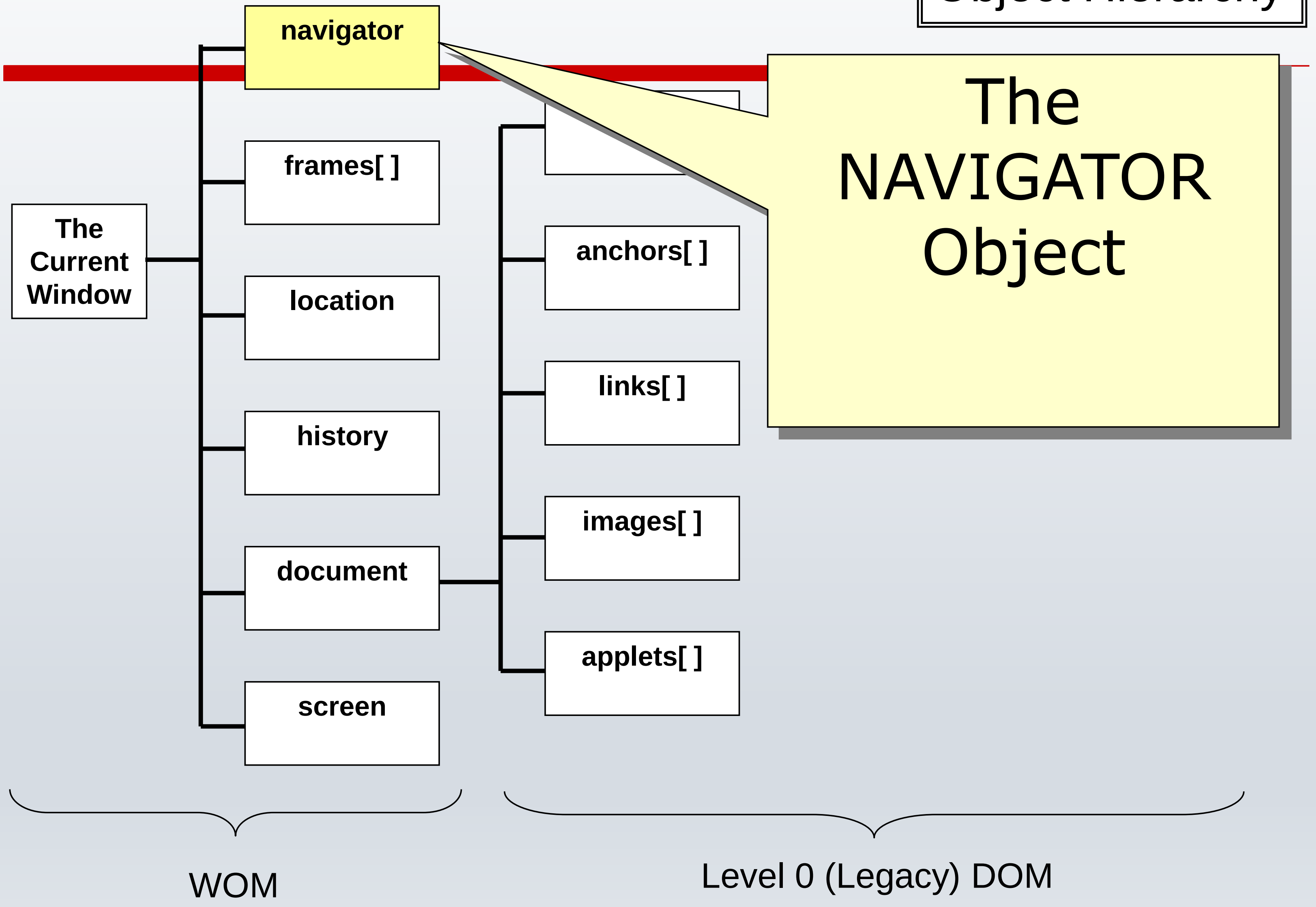
JavaScript code is executed once, when the page is loaded into the browser.

If you want your webpage to respond to the user, you need to provide Event Handler functions, which are triggered when an event is detected.

The most common events are:

onclick	buttons, <a>, <area>
onmousedown, onmouseup	all document elements
onmouseover, onmouseout	all document elements
onchange	<input>, <select>, <text>
onload	<body>

Object Hierarchy



Navigator Object

This contains information about the web browser as a whole.

Supported by all browsers, but historically the name refers to the old Netscape Navigator browser.

IE also supports the name 'clientInformation' for the same object.

Unfortunately, other browsers have not adopted this more sensible name.

Navigator Properties

It contains a set of properties which include the following:

appName	the name of the browser
appVersion	the version
userAgent	name and version details that are transmitted to the webserver
appCodeName	the code name of the browser.
platform	the hardware platform on which the browser is running.

The exercises include a function to display this information

Function to display Navigator properties

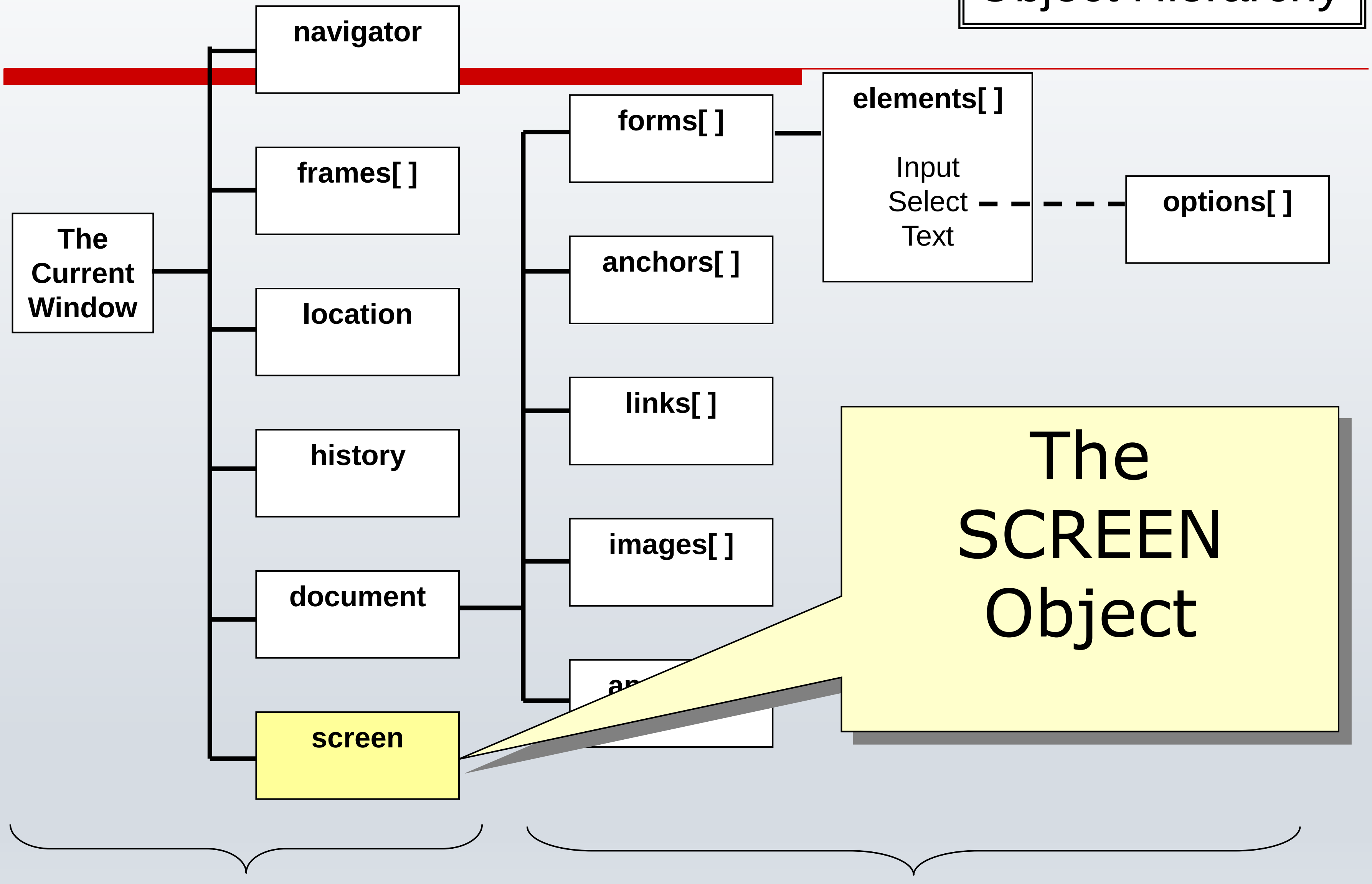
```
<html>
<head>
<script>
 function displayinfo()
  {
   var info = "BROWSER INFORMATION \n";

   for (var propname in navigator)
   {
   info = info + propname + ": " + navigator[propname] + "\n";
   }

   alert(info);
  }
</script>

<body>
  <button onclick="displayinfo()">Show Info</button>
</body>
</html>
```

Object Hierarchy



WOM

Level 0 (Legacy) DOM

Screen object

This contains information about the size of the users display screen and the number of colours it can display.

It has the following properties:

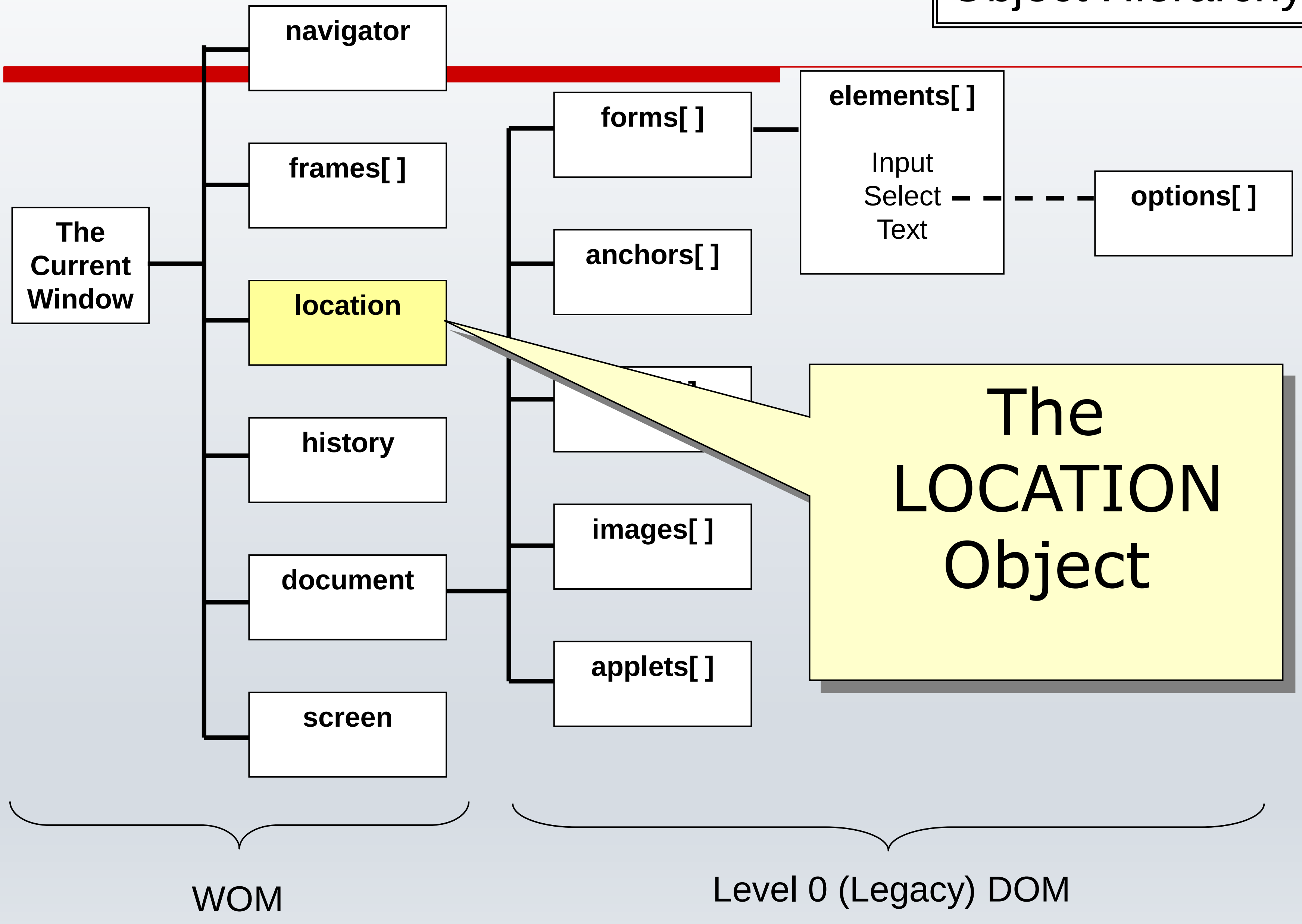
height
width
colorDepth
availHeight
availWidth

Displaying screen properties

```
<html>
<head>
<script>
  function displayinfo()
  {
    document.write("Height: " + screen.height + "  ");
    document.write("Width: " + screen.width + "  ");
    document.write("Colour Depth: " + screen.colorDepth + "  ");
    document.write("Available Height: " + screen.availHeight + "  ");
    document.write("Available Width: " + screen.availWidth + "  ");
  }
</script>

<body>
  <button onclick="displayinfo()">Show Info</button>
</body>
</html>
```

Object Hierarchy



Location Object

This contains information about the url of the page that is being displayed in the browser.

http://www.oreilly.com:8080/catalog/search.html?q=Javascript&m=10#results

It has the following properties:

hash	#results
host	www.oreilly.com:8080
hostname	www.oreilly.com
href	The entire url as shown above
pathname	/catalog/search.html
port	8080
protocol	http
search	?q=Javascript&m=10

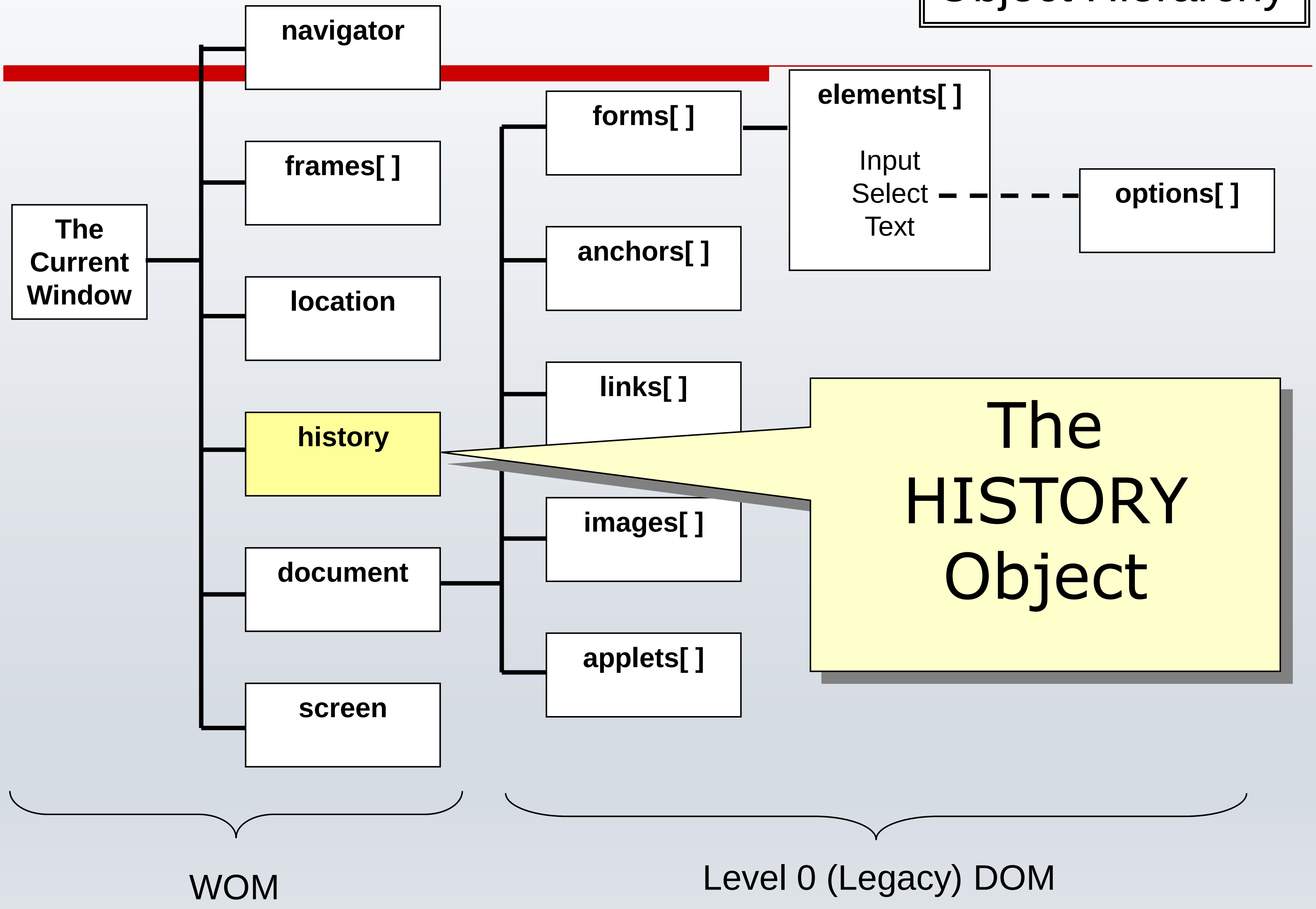
Location Object Methods

The location object also has two methods associated with it:

location.reload( )	reloads the current document
location.replace(url)	which loads the new document specified.

Note that this method does not place an new entry in the history – so using the back button would take you back to the page before last

Object Hierarchy



History Object

This contains a list of the urls that you have visited.

For security reasons, the History object no longer allows scripted access to the list, so there are no useful properties.

There are three methods that you can use:

history.back( )	Go to previous webpage
history.forward( )	Go to next webpage
history.go(number)	Go to specified webpage

Feature Testing

```
if (element.addEventListener)     //test for W3C method
  {
    element.addEventListener("keydown", handler, false);
  }
else if (element.attachEvent)      //test for IE method
  {
    element.attachEvent("onkeydown", handler);
  }
else                             // fall back on universal method
  {
    element.onkeydown = handler;
  }
```

The need for this sort of 'browser testing' is less important now because all manufacturers are attempting to make their browsers adhere to the W3C standards.

DOCTYPE

The use of the Document Type Declaration at the top of the page will make most modern browsers switch into a standards compliant mode.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
                      "http://www.w3.org/TR/html4/strict.dtd">
```

This specifies a Document Type Definition file, which is used to validate the html code in your page.

For modern versions of HTML (starting with HTML5), this has been replaced by:

```
<!DOCTYPE html>
```

For a more detailed discussion, see: http://hsivonen.iki.fi/doctype

Also, there are many function libraries, such as JQuery, which provide a standard way of handing the different DOMs of the various browsers.