Web Development

Lecture 10 – JavaScript

The Behaviour Layer

Most modern web pages not only look good (thanks to CSS), they are also quite interactive. They do things.

This is due to the web page containing one or more programs. They will be written in a scripting language such as JavaScript.

This is sometimes called the 'Behaviour Layer'.

JavaScript isn't only used in web pages. It is used to add functionality to PDF files and is also used in the Apple Mac operating system to animate the desktop.

History

It was developed by Brendan Eich at Netscape.

First called 'Mocha', then 'LiveScript', finally named 'JavaScript', although it had nothing to do with the programming language 'Java'.

First released as part of Netscape Navigator v2 in 1995.

Now developed by the European Computer Manufacturer's Association (ECMA). The standard form is known as ECMAscript.

Microsoft's Languages

Microsoft offers a competing language, VBScript, but it is only supported properly in Internet Explorer.

They also developed their own port of JavaScript called Jscript, which supports most of the standard JavaScript functions – sometimes in odd ways – and adds some of their own.

Both JavaScript and Jscript will adhere to the ECMA standard – it pays to be aware of which functions are part of the standard, and which are a particular manufacturer's 'add-ons'.

Client-Side Scripting

Remember JavaScript runs in the browser on the client machine.

This means that the way it works will be dependent on the way that the browser is configured.

Be aware that some people and companies will have disabled JavaScript due to security considerations.

In a well designed page, JavaScript should degrade gracefully. The page should remain usable – or at least tell the user that they need to enable JavaScript to get the full benefit.

Implementation Methods

Like CSS, JavaScript can either be embedded in an HTML page, or stored in an external file.

It can be placed anywhere in the webpage, but the custom is to place it in the <head> section.

You would use the <script> tags to indicate where the scripts are:

```
<head>
  <script type="text/javascript">
        #####This is where the scripts would go#####
  </script>
</head>
```

External Scripts

Storing scripts externally avoids this problem – and it means that they can be used by more than one web page.

```
<script   type="text/javascript"   src="myscript.js">
</script>
```

The myscript.js file will contain a set of JavaScript functions which you can call from within your webpage

Filesize vs Readability

Normally in programming, you use layout and indentation to improve readability.

The white spaces and comments are stripped out by the compiler before the program runs.

With JavaScript programs, they have to be sent over the internet before being run by the browser.

Some programmers keep a working copy of the script for debugging, and then take out the spaces before including them in the final website.

Statements

Separated by a line-break or a semi-colon

```
first statement
second statement
```

or

```
first statement; second statement;
```

or

```
first statement;
second statement;
```

Comments

Single line

```
// this is a comment
```

Multi-line

```
/*   this is also
     a comment
     in javascript  */
```

Variables

These are declared by using the reserved word var.

```
var MYVAR;
var myvar;
var MyVar;
```

JavaScript IS case-sensitive.

JavaScript is not strongly typed. You don't have to say what type of value the variable is going to hold.

```
var student_name = 'John', age = 18, course_name;
```

is a valid statement.

Data Types - Strings

You usually use single quotes to enclose strings, although you can use double quotes. Note the use of escape codes below.

```
var empty = ' ';
var full = 'This string contains some text';

var my_string = 'This quoted text is fine';
var my_string = "This quoted text is fine";
var my_string = 'This string\'s "quote" is escaped';
var my_string = "This string's \"quotes\" are escaped";
```

DataTypes – Numbers and Booleans

```
var my_age = 28;
var negNum = -12.234;
```

Booleans are true / false values

```
var bald = true;
var beard = false;
var bald = 1;
var beard = 0;
```

DataTypes - Arrays

Arrays allow you to group multiple values in a single variable. The values are numerically indexed starting at 0.

They can be declared in a few different ways:

```
var array_one = new Array( );
var array_two = [ ];
```

You can set up initial values when they are declared:

```
var friends = new Array( 'Kevin', 'Helen', 'Peter');
var names = [ 'Alan', 'Kate', 'Joe' ];
```

Or values can be assigned afterwards:

```
var people = new Array( );
people[0] = 'Robert';
people[1] = 'Sally';
people[2] = 'Sharon';
```

An array can contain a mixture of data types:

```
var student = ['John', 'Smith', 18, true ];
```

even other arrays:

```
var pets = new Array(2);
pets[0] = ['Fifi', 'Dog', 2, 'Poodle' ];
pets[1] = ['Harry', 'Cat', 3, 'Siamese'];
```

Operators

The standard operators are used:

Arithmetic: `+` `-` `*` `/`

Comparison: `==` `!=` `<` `>` `>=` `<=`

`&&` `||` `!`

The + sign is also used for concatenation:

```
var sentence = 'This is one part' + 'of a string';
```

The alert box

This will pop up a little box containing a message or a value of a variable, with an OK button.

Quite useful when debugging.

```
alert( 'display this message' );
```

Control Structures

The standard control structures are available:

Conditional

```
if ( age > 18 )
  {
    alert( 'You are an adult' );
  }
else
  {
    alert( 'You are not an adult' );
  }
```

Control Structures

while loop

```
var T = 10;
while ( T >= 0)
   {
     alert( T );
     T = T - 1;
   }
```

Control Structures

do-while loop

```
var T = 10;
   do {
        alert( T );
        T = T - 1;
       } while ( T >= 0)
```

Control Structures

for loop

```
for (var T = 10; T >= 0; T = T+1 )
  {
    alert( T );
  }
```

Control Structures

for – in loop

This can be used to loop through all of the members of an array:

```
var name_array = ['Kevin','Helen','Paul'];
for ( person in name_array)
  {
    alert( person );
  }
```

Functions

Reusable sections of code

They can return a value, although they don't have to

```
function addThese( a, b )
  {
    var answer;
    answer = a + b;
    return answer;
  }

myResult = addThese( 3, 8 );
otherResult = addThese('This is ', 'a string');
```

Objects, Properties and Methods

JavaScript is an object based language.

Arrays and Functions are stored as objects.

An object will have an associated set of Properties (values) and Methods (functions).

There are many built-in objects which can make it easier to write programs.

Use of methods

```
num = 1.764538;
num2 = Math.round(num);      // Object called Math
                             // which has a method
                             // called round

days = ['Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat'];

now = new Date( );           // Object called now
daynum = now.getDay( );      // which has a method
day = days[ daynum ];        // called getDay
```

Event Handling

Whenever you use the mouse or keyboard, you cause events.

onchange
onclick
onerror
onfocus
onkeydown
onkeyup
onload
onmousedown
onmousemove
onmouseout
onmouseover
onmouseup
onsubmit

Event Handling

You can use events to trigger the execution of scripts.

```
<img
        src = "blueArrow.gif"
        name = "arrow"
        onmouseover = "document.arrow.src = 'redArrow.gif ' "
        onmouseout = "document.arrow.src = 'blueArrow.gif ' "
/>
```

when the mouse moves over the image,it will load a different colour picture into the img element.

NOTE - Mixing up event handlers with HTML tags in this way is now considered bad practice – like mixing up style information with HTML tags.

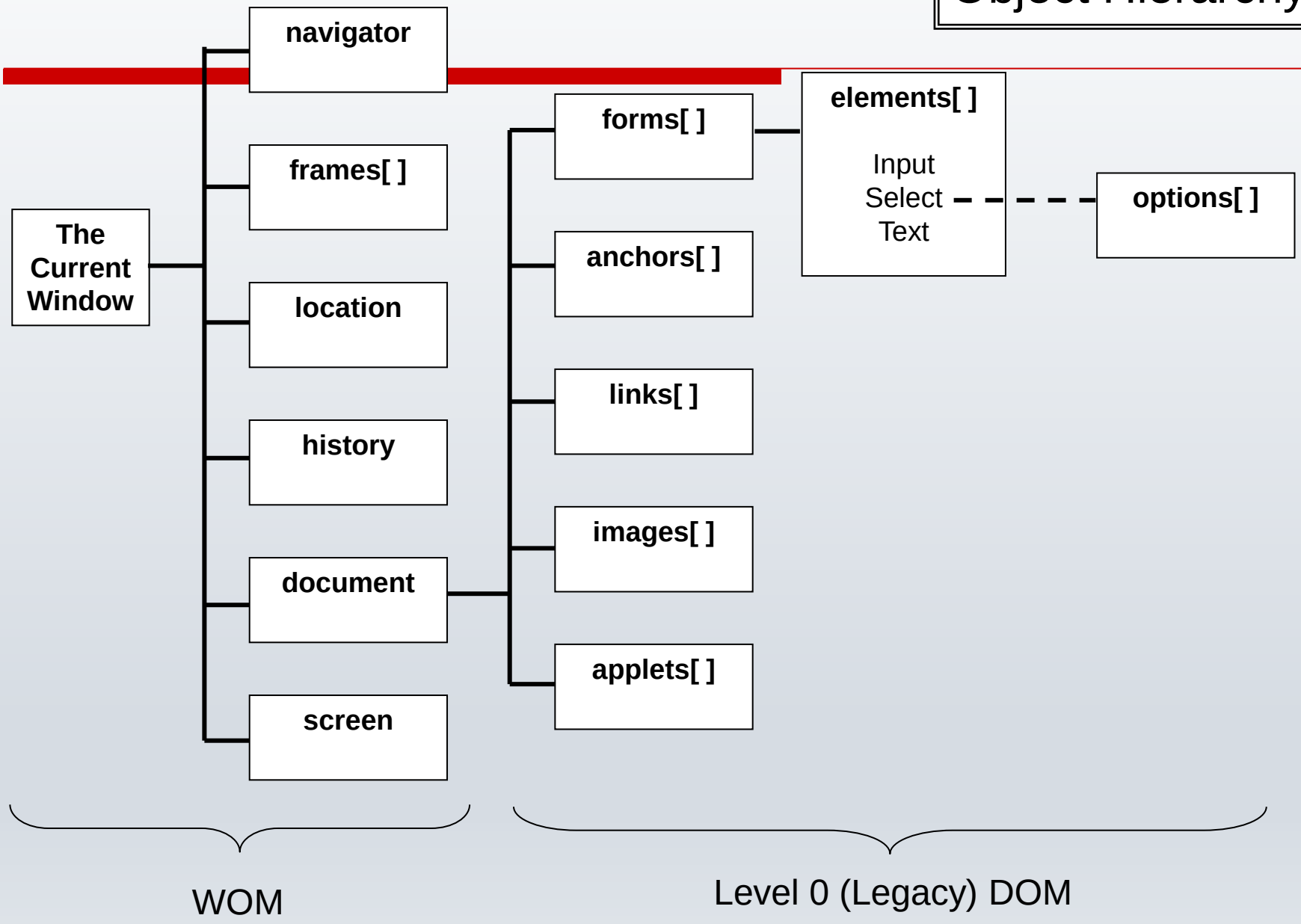
The Browser Object

The browser is considered to be an object as well. It has a list of properties which you can access.

For example, the browser window has a status bar into which you can place a value.

```
<a href="http://www.google.com"
   onmouseover="window.status='Go to google'; return true;"
   onmouseout="window.status='  '; return true;"
>
Link to Google
</a>
```

Object Hierarchy



How to refer to things

If you want to refer to the third textbox on the first form on a web page, you would write:

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

You can end up writing things like this:

window.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.