The behavioural layer (JavaScript)

Webpage Design

Web design layers

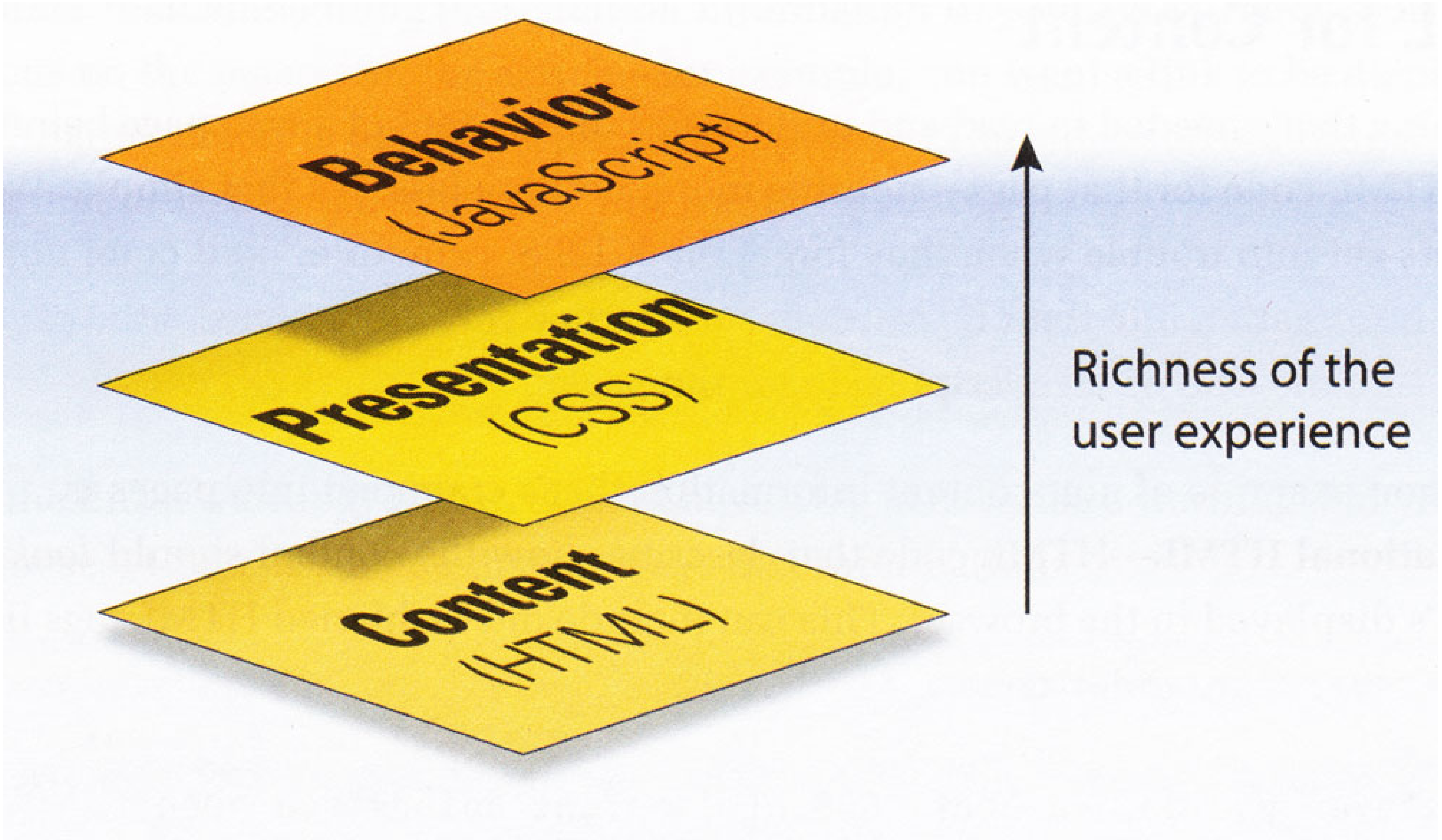


Figure 1.2. The three layers of the Web

Taken from: “Simply JavaScript” by Kevin Yank and Cameron Adams

JavaScript

- Originally developed by Netscape with Sun Microsystems is 1995
- Microsoft followed with Jscript
- JavaScript now standardised by ECMA*
- Got a bad name because of misuse by some designers
- Has now been rehabilitated and used with the DOM and other web standards (basis of Ajax**)

* European Computer Manufacturers Association

**Asynchronous JavaScript And XML

JavaScript and me

- In the past, web designers could get away with just using HTML and CSS, leaving JavaScript to the “geeks”.
- However, most front-end developers are now expected to have at least a rudimentary understanding of JavaScript.
- Recently, many developers have taken to using JavaScript frameworks/libraries like jQuery to make life easier.

The nature of JavaScript

- JavaScript is a “client-side” scripting language.
- That means it runs in the browser on the clients own device (phone, tablet, laptop).
- Scripts will therefore only work if the device supports JavaScript and the user has it turned on.
- JavaScript should not be used for essential functions (such as navigation) unless there is a fall-back that works if JavaScript does not.

Where does it go?

- Just like CSS, scripts can be embedded in your HTML document or they can placed in an external file and linked to from the <head> section of the document.
- Where scripts will be used by many pages, it makes sense to store them in external files but if they are unique to a page, it may be better to include them in the HTML document, thereby saving a *http request* (more efficient).
- If scripts are added to the HTML file, they are usually best located just before the closing </body> tag because the browser will have completed the DOM (Document Object Model) at that point.

Adding Scripts

```
<script>
  JavaScript goes here
</script>
```

Embedded scripts are enclosed in a <script> tag.

```
<script src="script.js"></script>
```

Linked scripts also use the <script> tag but a src (source) attribute is added to point to the file. Usually, the file has a .js extension. Notice that this forms an empty element – there's nothing between the opening and closing tags.

Note: in HTML5, the type="text/javascript" attribute is not required but it is required for XHTML.

Hello world!

A simple message:

```
<script>
// This is a very simple JavaScript statement that prints "Hello World!"
document.write("<h1>Hello World!</h1>");
</script>
```

Example

The above example uses a single statement to write some markup to the document. The script is placed between opening and closing script tags.

The text in quotes (including the html tags) is written to the document by JavaScript using the document.write method.

Notice that the script includes a comment, which describes what the statement does. Single-line comments begin with a double slash "//".

Day of the week

Not so simple:

```
<script>
// Sometimes, doing simple things in JavaScript is quite difficult
// The following code gets the name of the day of the week and prints it
var d = new Date();
var n = d.getDay();
var weekday=new Array(7);
weekday[0]="Sunday";
weekday[1]="Monday";
weekday[2]="Tuesday";
weekday[3]="Wednesday";
weekday[4]="Thursday";
weekday[5]="Friday";
weekday[6]="Saturday";
document.write("<p>Today is " + weekday[n] + "</p>");
</script>
```

We need to build an array of day names because JavaScript (unlike PHP) only knows the day number and because arrays always begin with 0, the week runs from 0 to 6.

Example

Testing for conditions

```
<script>
// The following code prints something different on a Thursday
var d = new Date();
var n = d.getDay();
var weekday=new Array(7);
weekday[0]="Sunday";
weekday[1]="Monday";
weekday[2]="Tuesday";
weekday[3]="Wednesday";
weekday[4]="Thursday";
weekday[5]="Friday";
weekday[6]="Saturday";
if (weekday[n]==="Thursday"){
		document.write("<p>Hurrah! Today is " + weekday[n] + ", it's a Greenwich day</p>");
}else{
		document.write("<p>Today is " + weekday[n] + ", just an ordinary day</p>");
}
</script>
```

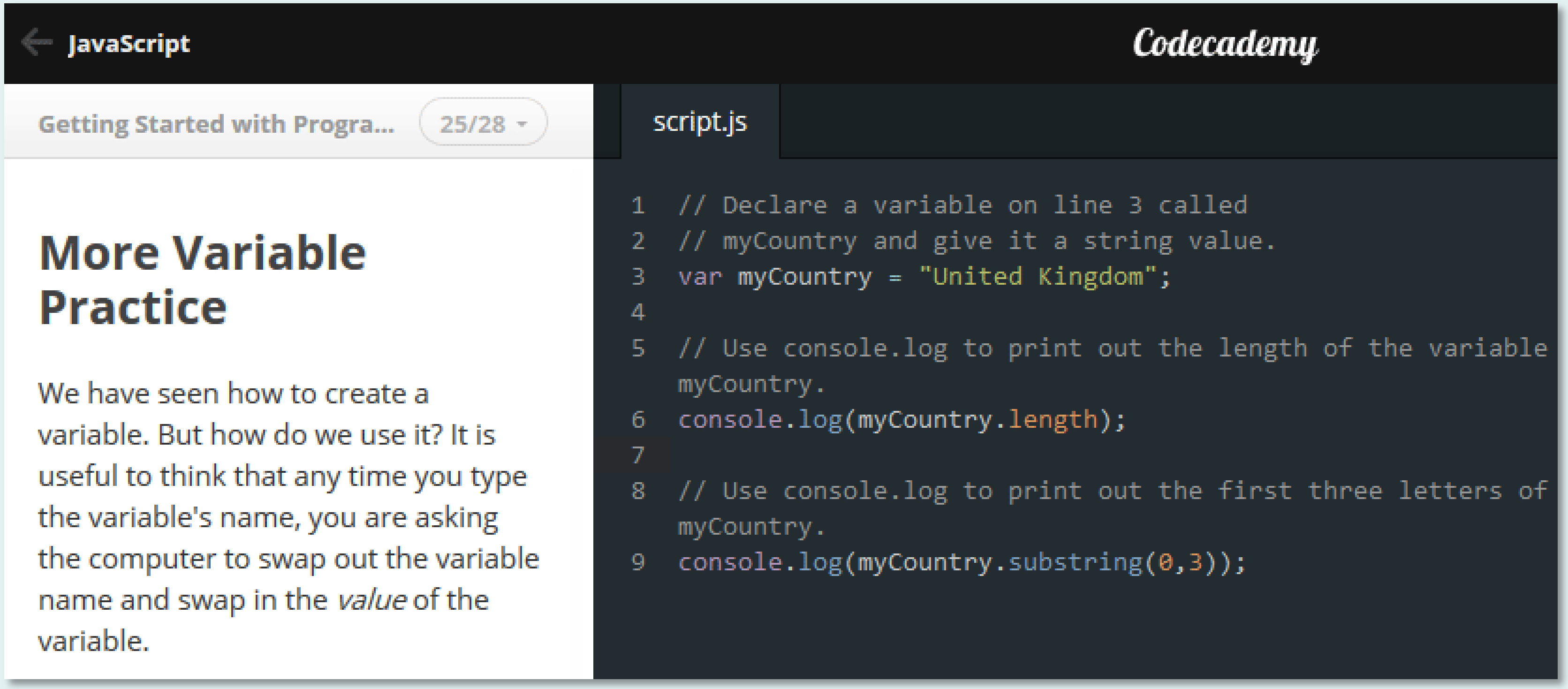
This script uses an *if else* test to decide what message should be printed. If today is Thursday, a special message is printed otherwise (else) print a standard message. Using this technique, web pages can be dynamic, with content that changes depending on time.

Example

JavaScript is power

OK, I LIKE THIS JAVASCRIPT, HOW CAN I LEARN MORE?

How can I learn JavaScript?



More Variable Practice

We have seen how to create a variable. But how do we use it? It is useful to think that any time you type the variable's name, you are asking the computer to swap out the variable name and swap in the *value* of the variable.

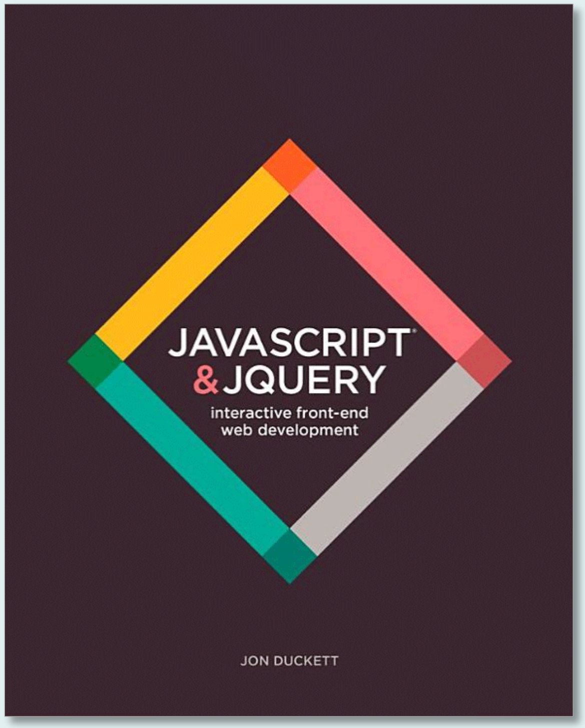
```
// Declare a variable on line 3 called
// myCountry and give it a string value.
var myCountry = "United Kingdom";

// Use console.log to print out the length of the variable
myCountry.
console.log(myCountry.length);

// Use console.log to print out the first three letters of
myCountry.
console.log(myCountry.substring(0,3));
```

www.codecademy.com/tracks/javascript

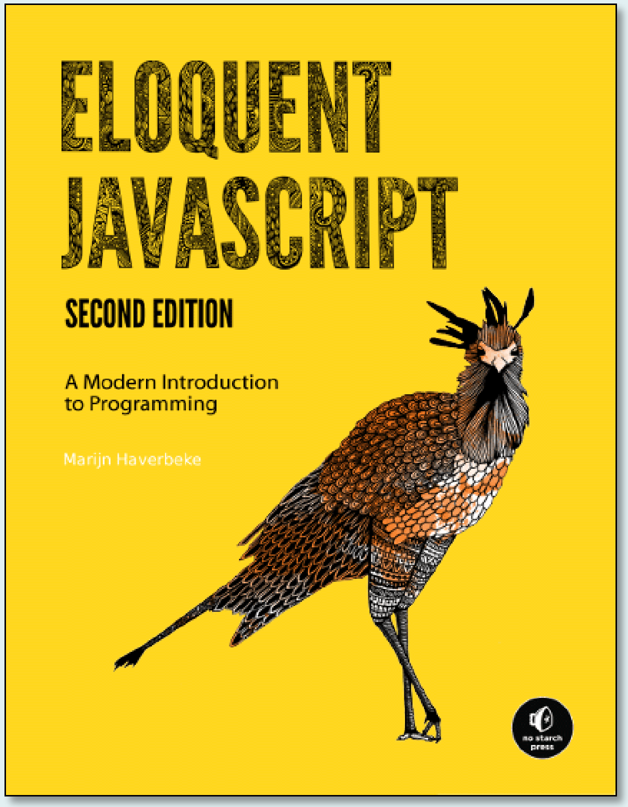
Any good beginner books?



If you're a complete beginner and you've never done any programming before, this book is written specially for you. It assumes nothing and is extremely clear. 3 cheers for Jon Duckett!

http://javascriptbook.com/

What other resources are there?



The second edition of Eloquent Javascript by Marijn Haverbeke is released on 11th December 2014 and is totally recommended if you want to get to grips with JavaScript but an online version of the book is also available for free. Check it out.

http://eloquentjavascript.net/

DOM

WHAT'S THIS DOCUMENT OBJECT MODEL THING?

The Document Object Model

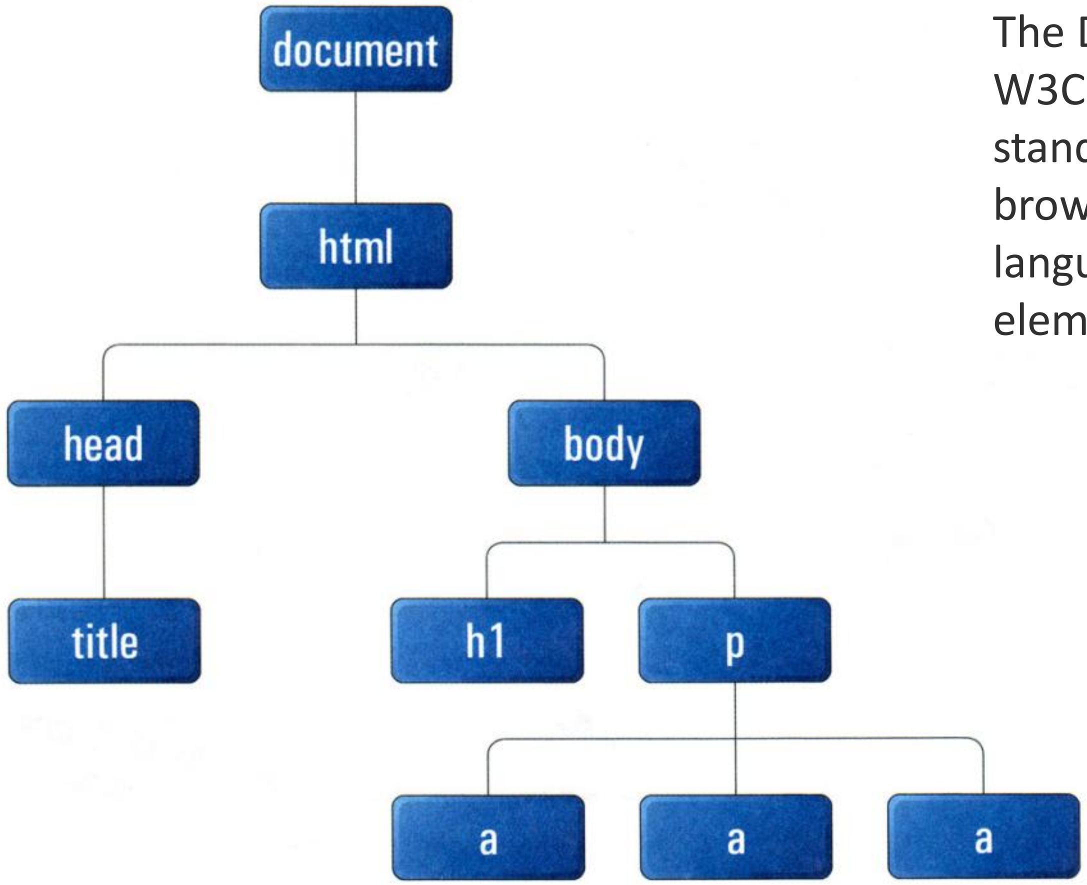


Figure 3.2. The DOM tree, including the document node

The DOM is controlled by W3C (who else!) and is a standard method for browsers and scripting languages to access page elements.

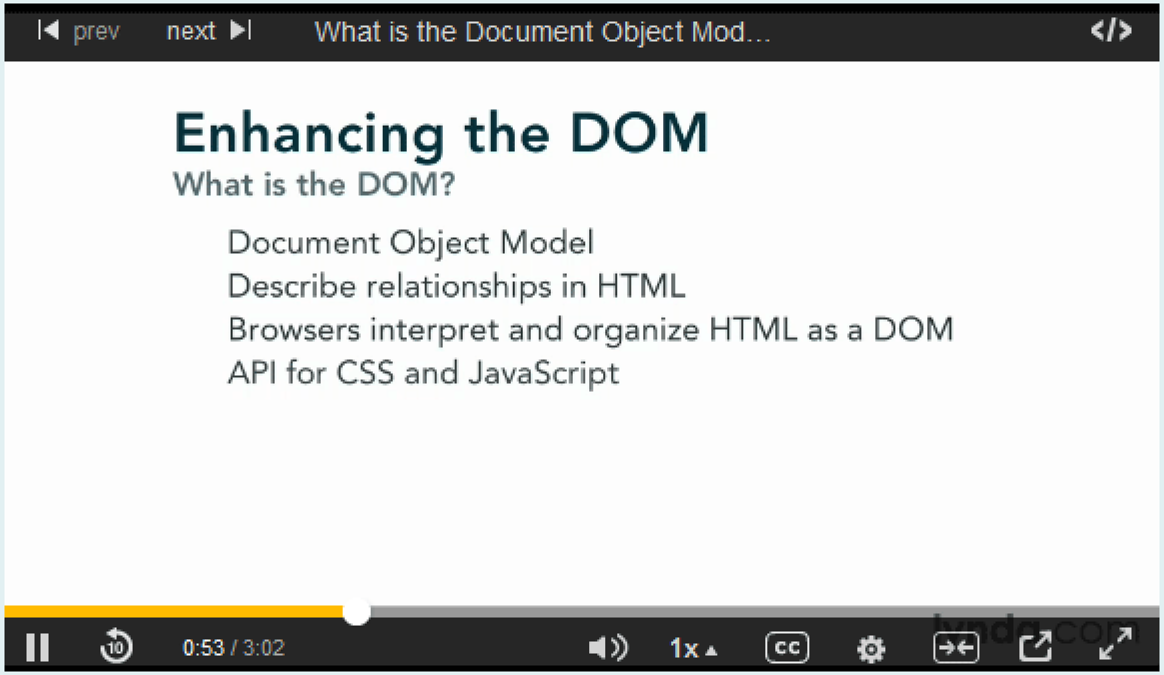
Taken from: “Simply JavaScript” by Kevin Yank and Cameron Adams

Which version of the DOM?

- Just like HTML and CSS, the DOM has evolved through a number of versions.
- The DOM Level 1 was standardised in 1997 by the W3C.
- The most recent version is DOM4, which is part of the HTML5 collection of technologies and is developed jointly by W3C and WHATWG.
- Modern browsers support all the features of DOM 3 but little of DOM4 at the moment.
- DOM4 reached *last call working draft* status on 10th July 2014.

http://www.w3.org/TR/dom/

Is it a map of relationships?



Enhancing the DOM

What is the DOM?

Document Object Model
Describe relationships in HTML
Browsers interpret and organize HTML as a DOM
API for CSS and JavaScript

This short video at Lynda.com is a good description of the DOM. Some of the terms used will be familiar to you by now; parents, siblings, children…

What is the DOM?

Values, variables and operators

THE BUILDING BLOCKS OF JAVASCRIPT

Values

Numbers	56
Strings	"letters and words"
Boolean	true *or* false

Variables

```
var myNumber = 36;
```

You can think of variables as being named boxes into which data can be placed. This could be simple data like a number or a string of text, or it could be an array containing many values.

```
var myNumber = 36;
myNumber / 3 === 12;
true
```

Variable names can be used in place of the value and this is what makes programs dynamic.

Comparison operators

==	is equal	6+3 == "9"
!=	is not equal to	6-3 != 4
===	is identical to	6*3 === 18
>	Is greater than	9 > 2
>=	Is greater than or equal to	6 >= 3
<	Is less than	2 < 3
<=	Is less than or equal to	4 <= 4

All of the above will return the boolean value *true*.

Operators

+	plus	6+3 = 9
-	minus	6-3 = 3
*	multiply	6*3 = 18
/	divide	6/3 = 2
%	modulo	6%3 = 0

Assigning elements to variables

```
<body>

var body = document.getElementsByTagName("body")[0];
```

Example (body background-color)

```
<div id= "box">some content</div>

var box = document.getElementById("box");
```

Example (#box background-color)

The DOM provides JavaScript with a number of methods for accessing elements within the HTML document such as *getElementByTagName* and *getElementById*.

Changing CSS values

One of the most powerful things we (designers) can do with JavaScript is to change the value of CSS properties. We can change the value of a single property or we can change the value of multiple properties.

A single value:

```
function changeText(){
document.getElementById("box").style.fontSize = "150%";
}
```

Example (#box text-size)

Multiple values:

```
function changeStuff(){
var box = document.getElementById("box");
box.style.backgroundColor = "#EFAE4C";
box.style.border = "10px solid #798492";
box.style.paddingLeft = "100px";
}
```

Example (#box *various*)

Other simple things

Hiding and revealing:

```
function displayNone(){
document.getElementById("box").style.display = "none";
}

function displayBlock(){
document.getElementById("box").style.display = "block";
}
```

Example

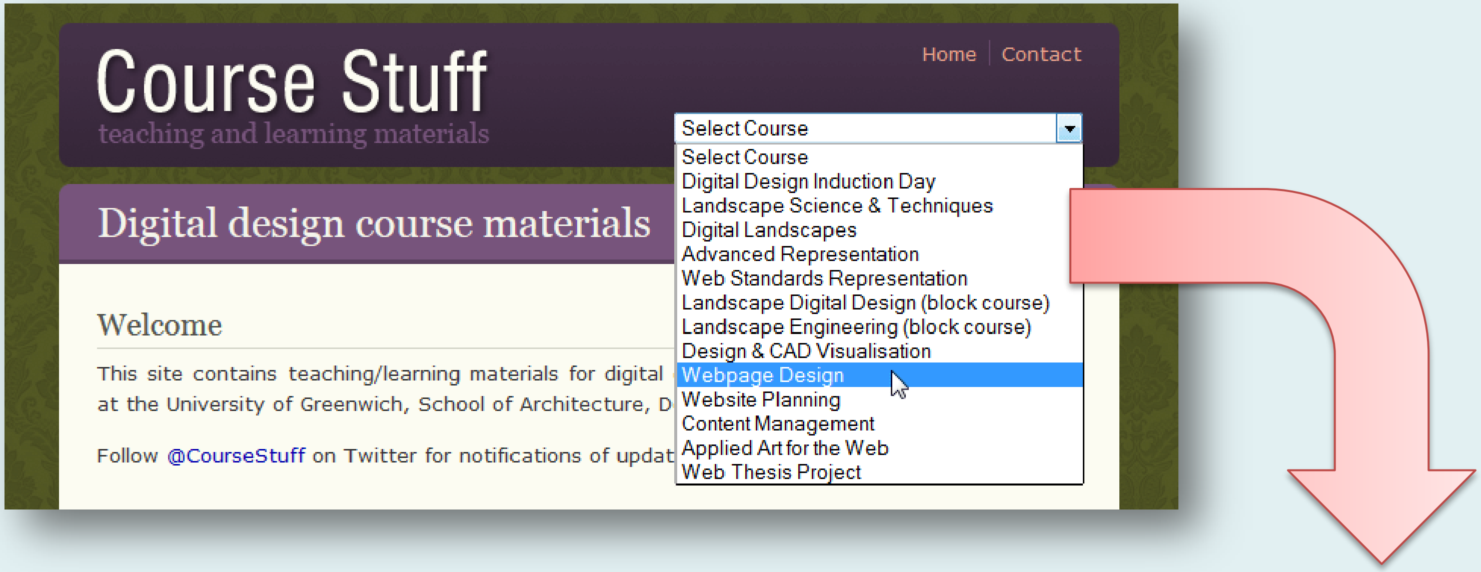
Opening a new window:

```
function newWindow()
{
window.open("window.html", "exampleWin", "width=650,height=300")
}
```

Example

A Jump Menu

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



Course Stuff

teaching and learning materials

Home | Contact

Digital design course materials

Welcome

This site contains teaching/learning materials for digital
at the University of Greenwich, School of Architecture, D

Follow @CourseStuff on Twitter for notifications of updat

Select Course
Select Course
Digital Design Induction Day
Landscape Science & Techniques
Digital Landscapes
Advanced Representation
Web Standards Representation
Landscape Digital Design (block course)
Landscape Engineering (block course)
Design & CAD Visualisation
Webpage Design
Website Planning
Content Management
Applied Art for the Web
Web Thesis Project

```
<form name="jumpmenu" id="jumpmenu" action="">
<select name="jump" onchange="MM_jumpMenu('parent',this,0)">
<option selected="selected" value="#">Select Course</option>
<option value="/ENVT1053/">Landscape Science &amp; Techniques</option>
<option value="/ENVT1008/">Digital Landscapes</option>
<option value="/ENVT1010/">Advanced Representation</option>
<option value="/ENVT1016/">Landscape Digital Design</option>
<option value="/ENVT1023/">CAD &amp; Visualisation</option>
<option value="/DESI1046/">Webpage Design</option>
<option value="/DESI1047/">Website Planning</option>
<option value="/DESI1054/">Net Art/Criticism</option>
</select>
</form>
```

A Jump Menu

jump.js

```
function MM_jumpMenu(targ,selObj,restore){ //v3.0
  eval(targ+".location='"+s                          ].value+"'");
  if (restore) selObj.selecte
}

function MM_findObj(n, c
  var p,i,x;  if(!d) d=docume                        ent.frames.length) {
    d=parent.frames[n.subs                           ng(0,p);}
  if(!(x=d[n])&&d.all) x=d.a                         ;i++) x=d.forms[i][n];
  for(i=0;!x&&d.layers&&i<                           j(n,d.layers[i].document);
  if(!x && d.getElementByI                           ;
}

function MM_jumpMenu(
  var selObj = MM_findObj                            enu(targ,selObj,restore);
}
```

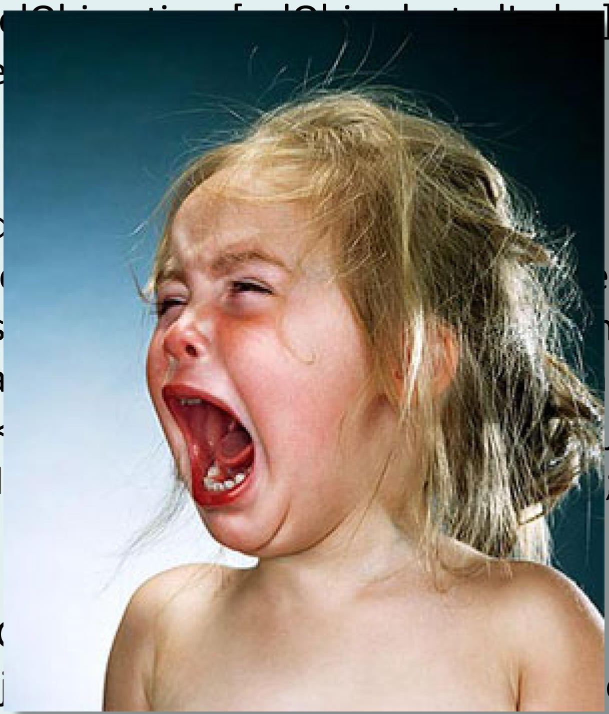

Typical reaction to complex javascript – it's not for designers, it's for coders.

Photo by Jill Greenberg

Start simple

- Don't be put off by complicated JavaScript.
- Start with the basics and work up from there.
- It takes time and some frustration, but programming can be very rewarding and will give you power over your web pages.

```
function endSlideshow(){
$("#slideshow").toggleClass('end');
     return false;
}
```