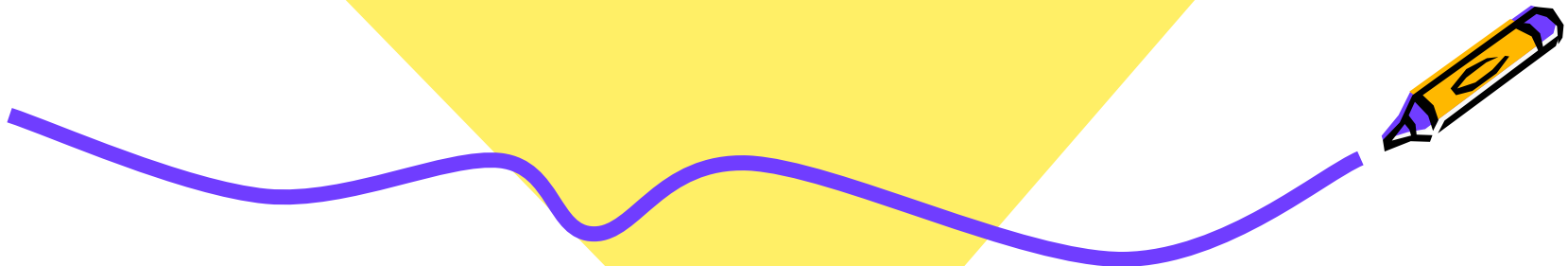




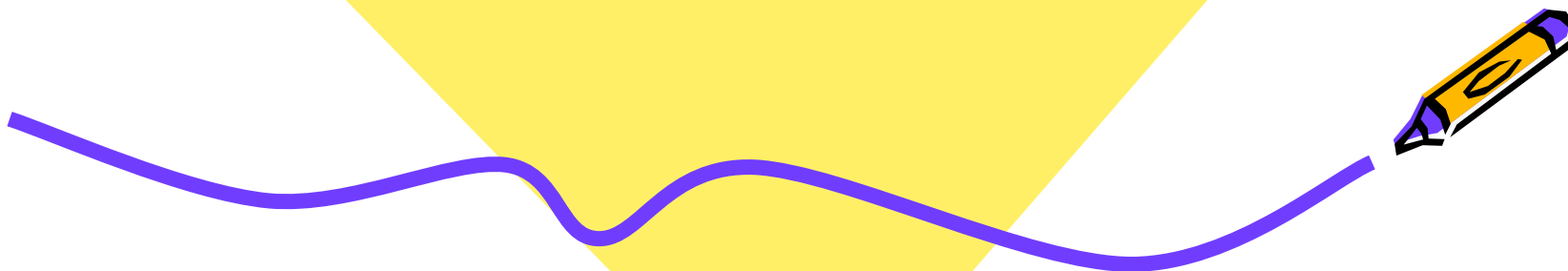
Developing with JavaScript Technologies

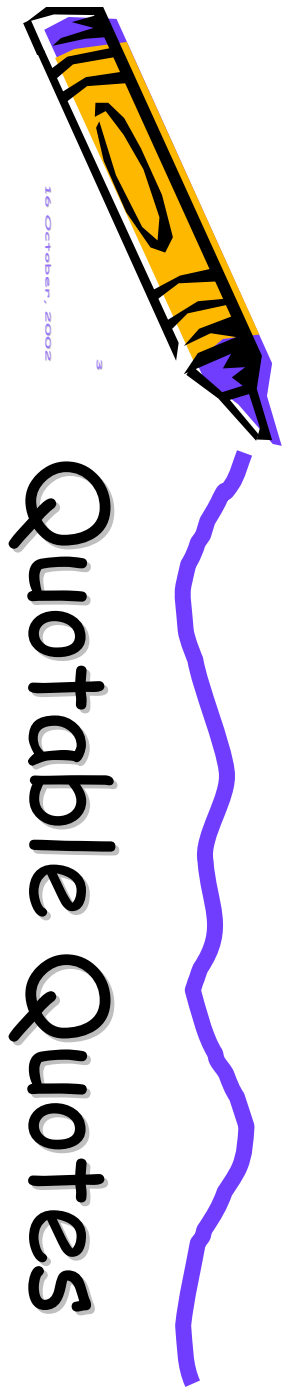


transentia

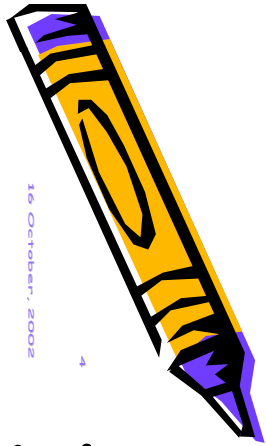


Introduction





- *The only excuse for making a useless thing is that one admires it intensely.*
— **Oscar Wilde**
- *"After a time of coding dHTML or JavaScript, you'll notice that the most difficult part of web development is knowing which browsers support which code."*
— Next handbooks' *Programming in JavaScript 2nd Ed.*
- *"As with playing the guitar, JavaScript is very easy to pick up and begin noodling around with, but difficult to master. Good technique can go a long way, but creativity and inventiveness (and willingness to take chances) can make the difference between code that barely runs now, and code that will continue to perform in the future."*
— http://www.oreillynet.com/pub/a/javascript/2001/05/17/why_javascript.html
- *"JavaScript, aka Mocha, aka LiveScript, aka JScript, aka ECMAScript, is one of the world's most popular programming languages."*
— <http://www.crockford.com/javascript/javascript.html>
- *"JavaScript is a compact, object-based scripting language for developing client and server Internet applications."*
— <http://developer.netscape.com/support/faqs/champions/javascript.html>
- *"What's the deal with JavaScript? Is it a programming language, a scripting language, or just a big mistake?"*
— <http://hotwired.lycos.com/packet/garfinkel/96/45/index2a.html>
- *"If it moves, script it",* http://msdn.microsoft.com/library/en-us/dnclinic/html/scripting061499.asp?frame=true#scripting06_topic8



In the Beginning...

"In the beginning, there was HTML.

And the hordes used HTML to produce Web pages.

And they looked at the Web pages, and saw that they were good.

But after a time, visitors to the Web sites grew restless and bored with the Web sites.

They clamored for more.

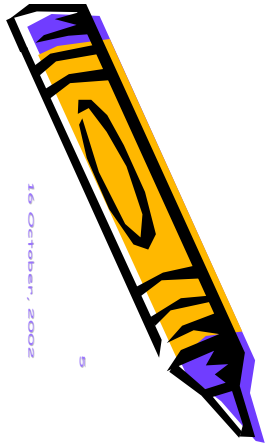
They wanted the page to interact with them.

They wanted dynamic content.

The hordes were also restless.

They wanted to be able to provide different content in different contexts.

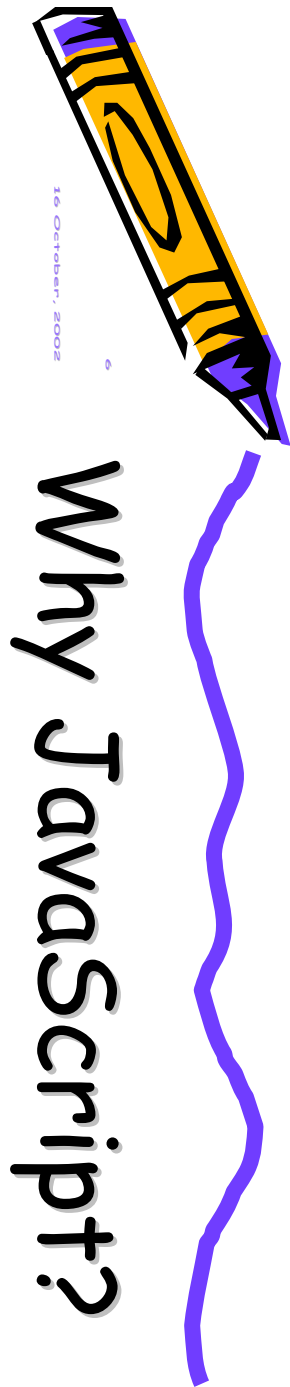
Thus, the notion of scripting was born."



A Trivial Example

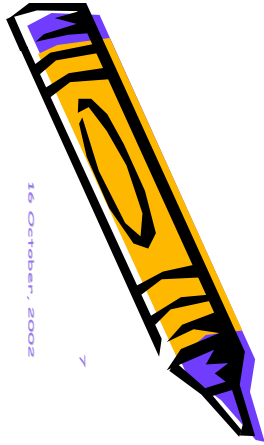
```
<html>
<head>
<title> Random Colour Button </title>
<script language="JavaScript">
<!--
    var colourArray =
        new Array ("red","cyan","green","yellow","purple");
    function rand0to4() {
        return Math.round(Math.random()*4);
    }
//-->
</script>
</head>
<body>
<input type="BUTTON" value="Click Me"
        onClick="document.bgColor=colourArray[rand0to4()];">
</body>
</html>
```





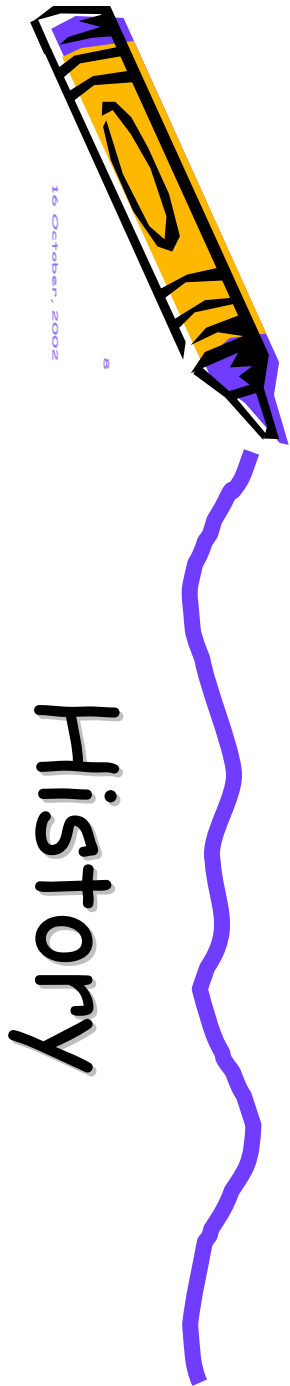
- According to http://www.oreillynet.com/pub/a/javascript/2001/05/17/why_javascript.html:
 - JavaScript may well have the widest install base and impact of any cross-platform scripting language.
 - JavaScript is easy to learn, but powerful, and rewards the developer who invests some time into exploring the finer points of its many features.
 - If you already know Java, Perl, C/C++, or other, similar languages, you're already familiar with many of the fundamentals of JavaScript. If you're not familiar with those languages, the fundamentals may be learned quickly (though it may take some getting used to if your only experience is with Python or Visual Basic).
 - JavaScript may be used in both procedural and object fashion, and at varying levels of abstraction and design discipline.
 - Tied to the browser by way of Document Object Models of varying quality and depth of implementation, the trend is nonetheless strongly toward cross-platform, cross-browser compatibility and standards compliance (though you will likely have to work a bit to figure out which 90 percent of the standard is supported in which browser, and jury-rig the rest).
 - It's only fair to say that if you avoid JavaScript, you will also avoid a great deal of testing for incompatibilities in dozens of browsers. But what developer can avoid some testing? And what language is free of platform-specific quirks?
 - Due to the increasing acceptance of JavaScript-enhanced web sites, more developers are using JavaScript and DHTML on their sites.
 - The next generation of client-side web applications will be written in JavaScript (and already is).

Uses



16 October, 2002

- Dynamic HTML
 - User interface components
 - Menuing systems
 - Image maps
- Simple calculations
- Image rollovers
- Cookie manipulation for state maintenance
- Form validation
- Glue
 - Link Java, Flash, media players, etc.
- "Un-dumbing" the dumb-client application
- OS Scripting
- Allied technologies: SVG, MathML, XML, SMIL, etc.
- Keep committee members in a job ☺
- The basis for Flash's ActionScript language
- Gaining favour as an embeddable scripting language



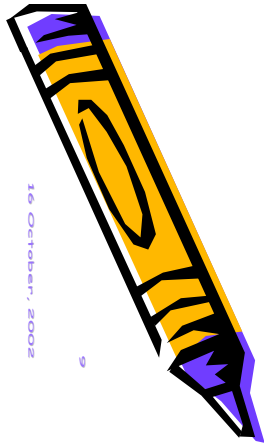
- Originally "LiveScript"
- JavaScript announced alongside Netscape 2.0, December 4, 1995
 - See <http://wp.netscape.com/newsref/pr/newsrelease67.html>
 - MicroSoft soon followed up with JScript
 - Thus are nightmares born...

"Go back and pull the original Netscape announcement from December 4, 1995," says Charles Fitzgerald, a program manager in Microsoft's Internet platform and tools division who seemed more than a little miffed when I asked him about JavaScript. "[Netscape] said it would be an open standard. That it would be fully documented. That it would be submitted to standards bodies - they mentioned IETF and W3C in the press release. [They said] they would provide sample source code. And they said that anybody who had a Java license would get a JavaScript license."

It turns out that Netscape didn't make good on any of these promises until the beginning of October 1996. I called up Eckart Walther, a product marketing manager for Netscape Navigator, and asked why. He said that little things like putting the JavaScript reference implementation up on Netscape's Web site simply fell through the cracks.

In the meantime, Microsoft had a product to ship - Internet Explorer 3.0 - which simply had to have JavaScript in order to be compatible with Navigator. So Microsoft wrote its own implementation of the language. They called it JScript."

— <http://hotwired.lycos.com/packet/garfinkel/96/45/index2a.html>



Version Control

- And you thought the HTML browser wars were bad...

Here's a table of Browsers versus the JavaScript versions they claim to support:

	IE PC 4.01	IE Mac 4.01	IE PC 4.01	IE PC 5.0	IE Mac 5.0	IE Mac 5.1	IE PC 5.5	IE PC 6	Netscape 4.7 6.0		Mozilla 0.9-1.0	Opera 6.03
JavaScript	X	X	X	X	X	X	X	X	X	X	X	X
JavaScript1.0	X	X	X	X	X	X	X	X	X	X	X	X
JavaScript1.1	X	X	X	X	X	X	X	X	X	X	X	X
JavaScript1.2	X	X	X	X	X	X	X	X	X	X	X	X
JavaScript1.3			X	X	X	X	X	X	X	X	X	X
JavaScript1.4					X	X				X	X	X
JavaScript1.5										X	X	
Jscript	X	X	X	X	X	X	X	X				
EcmaScript			X	X			X	X				X
exceptions supported?	no	no	yes	yes	yes	yes	yes	yes	no	yes	yes	yes

An "X" in row "Yyyyy" means code under `<script language="Yyyyy">` will run in the browser for that column. For example:

```

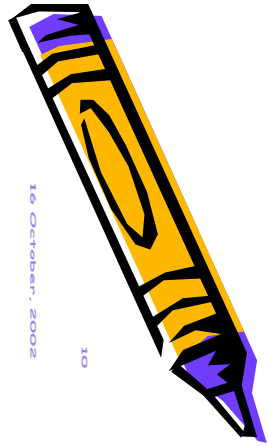
IE6/PC
-----
<script language="JavaScript 1.3">
    // runs
</script>

Netscape 6
-----
<script language="EcmaScript">
    // ignored
</script>

```

Yes, there are two radically different browsers both claiming to be IE4.01, one supports JavaScript 1.3 and EcmaScript and exceptions work perfectly, and the other not.

<http://javascript-reference.info>

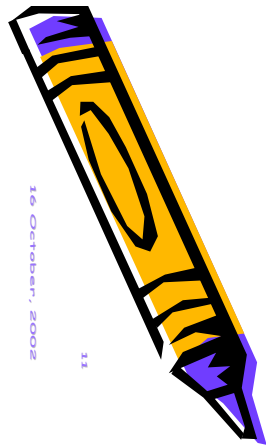


Version Control...

- Rapid development
 - A casualty of the browser wars...
 - Detrimental

JavaScript version	Navigator version
JavaScript 1.0	Navigator 2.0
JavaScript 1.1	Navigator 3.0
JavaScript 1.2	Navigator 4.0-4.05
JavaScript 1.3	Navigator 4.06-4.7x
JavaScript 1.4	
JavaScript 1.5	Navigator 6.0
	Mozilla (open source browser)

JScript version	IE version
JScript 1.0	IE 3.0
JScript 2.0	IIS 1.0
JScript 3.0	IE 4.0
JScript 4.0	Visual Studio 6.0
JScript 5.0	IE 5.0
JScript 5.5	IE 5.5
JScript 5.6	IE 6.0



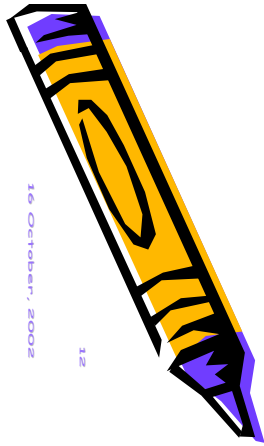
Version Control...

- Don't forget ECMAScript!
 - First edition: June 1997; second edition: August 1998; third edition: December 1999
 - An attempt to
 - Bring order to chaos
 - Give legitimacy to JScript
 - Payback Netscape ☺

JavaScript version	Relationship to ECMA version
JavaScript 1.1	ECMA-262, Edition 1 is based on JavaScript 1.1.
JavaScript 1.2	<p>ECMA-262 was not complete when JavaScript 1.2 was released. JavaScript 1.2 is not fully compatible with ECMA-262, Edition 1, for the following reasons:</p> <ul style="list-style-type: none">• Netscape developed additional features in JavaScript 1.2 that were not considered for ECMA-262.• ECMA-262 adds two new features: internationalization using Unicode, and uniform behavior across all platforms. Several features of JavaScript 1.2, such as the Date object, were platform-dependent and used platform-specific behavior.
JavaScript 1.3	<p>JavaScript 1.3 is fully compatible with ECMA-262, Edition 1.</p> <p>JavaScript 1.3 resolved the inconsistencies that JavaScript 1.2 had with ECMA-262, while keeping all the additional features of JavaScript 1.2 except == and !=, which were changed to conform with ECMA-262.</p>
JavaScript 1.4	<p>JavaScript 1.4 is fully compatible with ECMA-262, Edition 1.</p> <p>The third version of the ECMA specification was not finalized when JavaScript 1.4 was released.</p>
JavaScript 1.5	JavaScript 1.5 is fully compatible with ECMA-262, Edition 3.

- ECMAScript 4 is/will be a very different beast from what went before

Alternatives



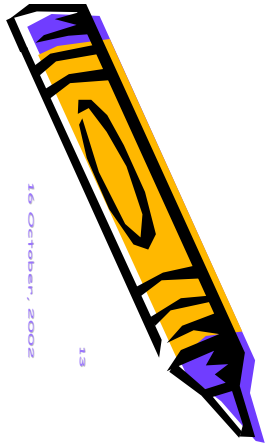
16 October, 2002

12

- Client-Side languages
 - VBScript
 - PerlScript
 - PythonScript
 - Java
- Server-Side languages
 - ASP
 - JSP
 - Perl
 - PHP
 - etc.
- Others
 - Flash

```
<HTML>
<HEAD>
<TITLE>PerlScript Hello World!</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
<h2>PerlScript Hello world!</h2>
<SCRIPT LANGUAGE="PerlScript">
$window->document->write('Hello world!');
</SCRIPT>
</BODY>
</HTML>
```

```
<html>
<body>
<%
SenderName = Request.Form("name")
SenderEmail = Request.Form("email")
Subject = "Enquiry from website"
Body = Request.Form("body")
Recipient = "bob@transentia.com.au"
smtp_server_address="mail.transentia.com.au"
Set myMail = Server.CreateObject("JMail.Message")
myMail.From = SenderEmail
myMail.Subject = Subject
myMail.AddRecipient Recipient
myMail.Body = SenderName & " asks: " & Body
myMail.Priority = 3
if not myMail.Send(smtp_server_address) then
    ' There was an error - print the error log
    Response.Write ("Error:<br>" & myMail.log)
end if
%>
Your email has been sent <br>
</body>
</html>
```



The Future

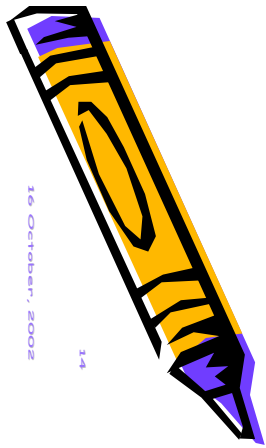
- ECMAScript 4
 - JScript.NET represents the first 'real' implementation
 - Looks a bit more like a 'proper' Java interpreter to me...
 - Hmm...so why not...don't ask!

```
class C
{
    var a:String;

    constructor function C(p:String) {this.a = "New "+p}
    constructor function make(p:String) {this.a = "Make "+p}
    static function obtain(p:String):C {return new C(p)}
}

var c:C = new C("one");
var d:C = C.C("two");
var e:C = C.make("three");
var f:C = C.obtain("four");

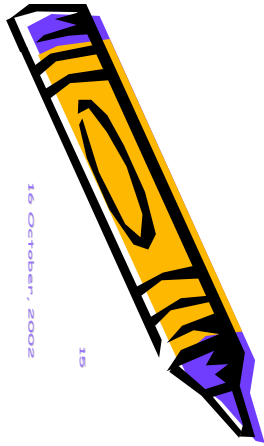
c.a;      // "New one"
d.a;      // "New two"
e.a;      // "Make three"
f.a;      // "New four"
```



Resources

- Books
 - Next Handbooks Programming in JavaScript
 - Professional JavaScript, Wrox Press
- URLs
 - <http://msdn.microsoft.com/scripting>
 - Web browser compatibility chart, <http://www.webreview.com/browsers/browsers.shtml>
 - Online quick reference, <http://javascript-reference.info>
 - Useful JavaScript, <http://members.ozemail.com.au/~dcrombie/javascript/index.html>
 - Danny Goodman's FAQ, <http://www.dannyg.com>
 - What is General Purpose scripting?, http://www.byte.com/documents/s=2473/byt1013632013554/0218_udell.html
 - IBM Sash Weblications, <http://sash.alphaworks.ibm.com/>
 - Dynamic HTML Central, <http://www.dhtmlcentral.com/>
 - <http://www.coolhomepages.com/>
 - <http://www.projectseven.com/>
 - <http://caucuscare.com/~roth/JAVASCRIPT/index.htm>

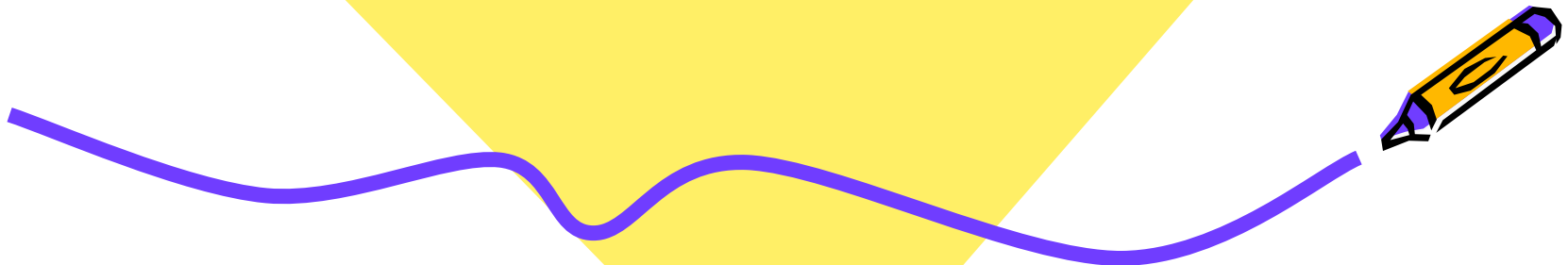
Resources...

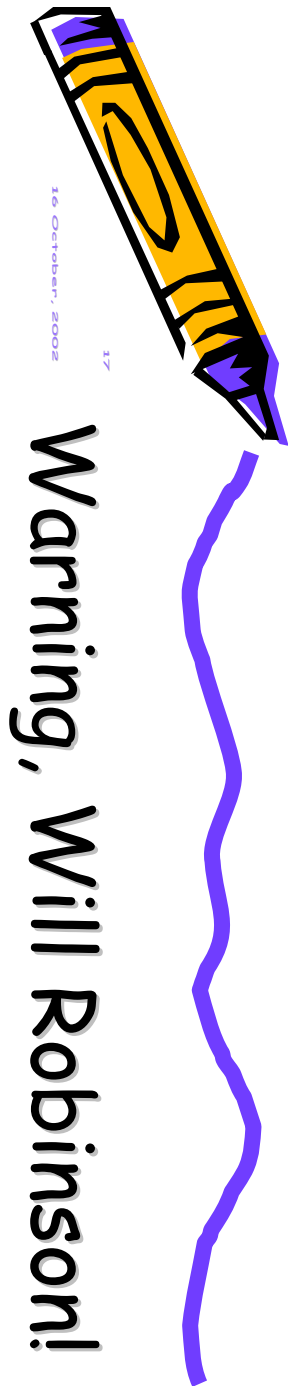


- Tools
 - JSLint (outdated now),
<http://www.crockford.com/javascript/lint.html>
 - JMyth Code Encryption Utility,
<http://www.geocities.com/SiliconValley/Vista/5233/jmyth.htm>
 - MicroSoft script debugger,
<http://msdn.microsoft.com/scripting/debugger/default.htm>
 - Netscape's form validation library,
<http://developer.netscape.com/library/examples/javascript/formval/overview.html>
 - Netscape's Visual JavaScript,
<http://wp.netscape.com/enterprise/vjs/>
 - Venkman JavaScript Debugger - Don't cross the streams,
<http://www.mozilla.org/projects/venkman/>
 - Macromedia Homesite,
<http://www.macromedia.com/software/homesite/>
 - AceHTML, <http://www.visicommedia.com/>
 - Foldertree visual tree view builder,
<http://www.treeview.net/FavoritesManagerASP/FolderTreeVisualBuilder.asp>
 - Web Engine Web Design and Animation Editor,
<http://www.virtualmechanics.com/products/engine/>
 - The JavaScript kit, <http://javascriptkit.com/>



The JavaScript Language: Basics





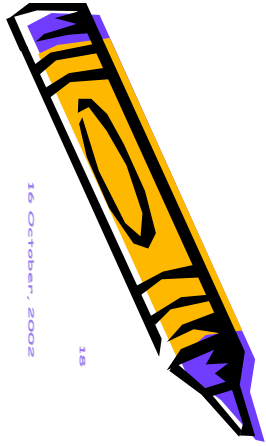
- There is *no such thing* as JavaScript!

- JavaScript1.{0,1,2,3,4,5}, ECMAScript{2,3,4}, JavaScript2.0, JScript...
- A given piece of source *will* behave differently when subjected to different language specs/interpreters/browsers, etc.
- An absurd state of affairs!
 - Casualty of the browser wars...
 - Irrevocably balkanized?

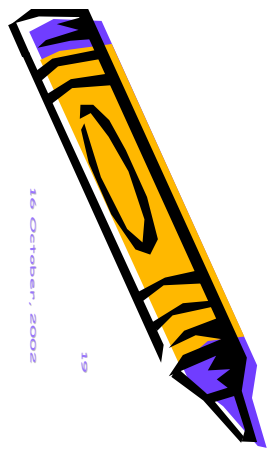


This script	Produces this output
<pre><SCRIPT> document.write("3" == 3); </SCRIPT></pre>	true
<pre><SCRIPT> document.write(("3"-0) == 3); </SCRIPT></pre>	true
<pre><SCRIPT LANGUAGE="JavaScript"> document.write("3" == 3); </SCRIPT></pre>	true
<pre><SCRIPT LANGUAGE="JavaScript1.1"> document.write("3" == 3); </SCRIPT></pre>	true
<pre><SCRIPT LANGUAGE="JavaScript1.2"> document.write("3" == 3); </SCRIPT></pre>	false
<pre><SCRIPT LANGUAGE="JavaScript1.2"> document.write(("3"-0) == 3); </SCRIPT></pre>	true
<pre><SCRIPT LANGUAGE="JavaScript1.2"> document.write(String(3) == "3"); </SCRIPT></pre>	true

Data Types



- 5 primitive types
 - number
 - 64-bit values: 10^{-324} ... 10^{308}
 - Always manipulated/stored in floating-point format
 - `isNaN()` and `isFinite()`
 - boolean
 - string
 - null
 - undefined
 - *"Unless you're doing something advanced, undefined is usually bad news..."*
- Complex types
 - Array
 - Object



Variables

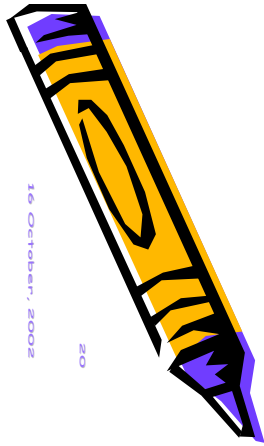
- No real surprises...
 - Case-sensitive
 - Normal rules for naming
 - May be initialised at declaration
- Typeless

```
theString = "Hello, World\n"  
var  
  b = true,  
  o = new Object(),  
  an_array = new Array(),  
  PI = 3.141;  
  
b = theString;  
PI = o;
```

- JS1.5 introduces constants

```
const PI = 3.141;
```
- Scope must be considered
 - Simple assignment gives an entity with global scope
 - Declaration using var keyword ensures local scope

Expressions



- Standard Java-style set of behaviours with some modifications
 - Semicolon at end of line optional if one-statement-per-line formatting followed

- Standards:

- Arithmetic

- + - * / % ++ -- -

- Comparison

- == !=

- > >= < <=

- Boolean

- && || !

- String

- + +=

- Bitwise

- & | ^ >> << ~ >>>

- Assignment

- = += -= *= /= &= |=, etc.

- Special

- (condition) ? trueVal : falseVal

- Mods:

- Arithmetic

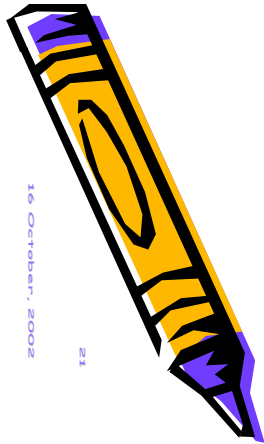
- unary +

- Comparison

- === !==

- Special

- void typeof instanceof



Expressions...

```
function countBitsInANumber()
{
    var u = ~0 // set u to all 1s
    var n = 0
    do
    {
        n ++
    } while ((u >>= 1) != 0);
    return (n)
}
```

```
a = b = c = 0;
```

```
C:\WINDOWS\System32\cmd.exe
C:\> cscript /nologo eq.js
float == str: true
float === str: false
C:\> _
```

```
var float = 1.0
var str = "1"
WScript.echo("float == str: " + (float == str))
WScript.echo("float === str: " + (float === str))
```

```
11101000 >> 2 -> 11111010
11101000 >>> 2 -> 00111010
```

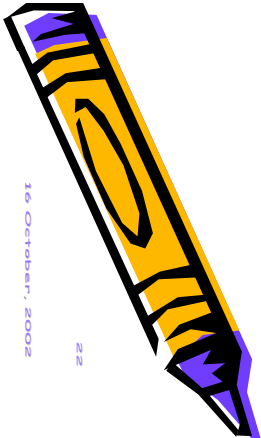
```
var float = 1.0
WScript.echo("float: " + float)
WScript.echo("+float: " + +float)
float = -1.0
WScript.echo("float: " + float)
WScript.echo("+float: " + +float)
```

```
C:\WINDOWS\System32\cmd.exe
C:\> cscript /nologo uplus.js
float: 1
+float: 1
float: -1
+float: -1
C:\>
```

```
{
var x = 99,
    y = 0;
if ((y != 0) && ((x / y) > 10))
    nothing gets done here...;
```

```
{
var x = 99;
var y = 1;
if ((x += y) == 100)
    something gets done ...;
```

Click here to submit



16 October, 2002 22

Regular Expressions

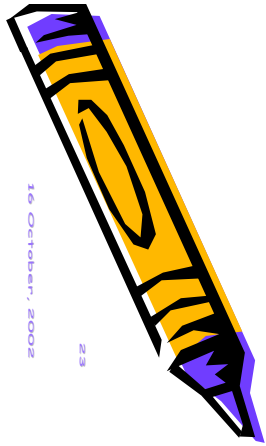
- Introduced in JavaScript1.2
 - Looks like PERL
 - Based around the RegExp Object

```
<HTML>
<BODY>
  <SCRIPT LANGUAGE='JavaScript'>
    var myRegExp = new RegExp();
    myRegExp.compile("^\\d{5}(?:-\\d{4})?$/i");
    var myZipCode = prompt("Enter a Zip code", "");
    alert((myRegExp.test(myZipCode) ? "Valid" : "invalid") + " ZIP");
  </SCRIPT>
</BODY>
</HTML>
```

```
<HTML>
<BODY>
  <SCRIPT LANGUAGE='JavaScript'>
    var myRegExp = /^\\d{5}(?:-\\d{4})?$/i;

    var myZipCode = prompt("Enter a Zip code", "");
    alert((myRegExp.test(myZipCode) ? "Valid" : "invalid") + " ZIP");
  </SCRIPT>
</BODY>
</HTML>
```

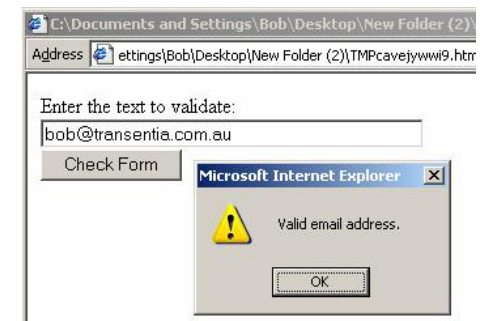


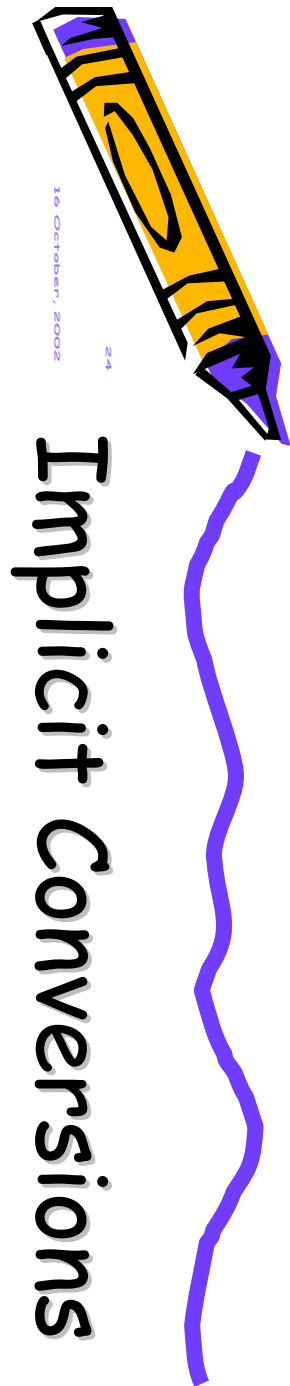


RegExp...

```
<HTML>
<BODY>
  <SCRIPT LANGUAGE='JavaScript'>
    var browserName = "Unknown";
    var browserCheckRegExp = /(Microsoft|Netscape|Opera)/;
    if (browserCheckRegExp.test(navigator.appName))
      browserName = RegExp.$1;
    alert(browserName);
  </SCRIPT>
</BODY>
</HTML>
```

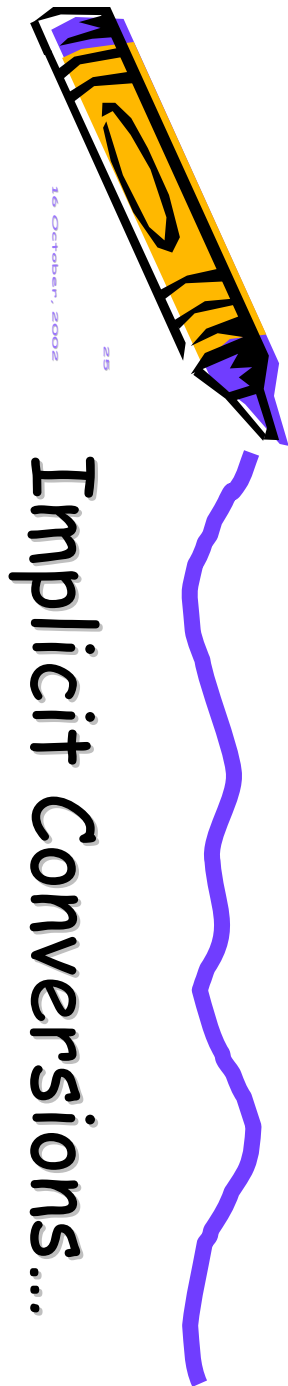
```
<html>
<head>
<script language="JavaScript">
function validEmail(email)
{
  var expr =
/^\\w(\\.?\\w)*@\\w(\\.?[-\\w])*\\.([a-z]{3}\\.([a-z]{2})?)|([a-z]{2}\\.([a-z]{2})?)$/i;
  return expr.test(email);
}
function val_onclick(email)
{
  alert((validEmail(email)? "Valid" : "Invalid") + " email address.");
}
</script>
</head>
<body>
<form name="frm">
  Enter the text to validate:<br>
  <input type="text" name="txt"><br>
  <input type="button" value="Check Form"
    onClick="val_onclick(document.frm.txt.value);">
</form>
</body>
</html>
```





- JavaScript automatically converts values when required
 - Most notable with conversion to string

Value:	String	Number	Used As:		
			Boolean	Object	Function
non-empty string	-	Numeric value of string, or error	true	String object	error
empty string	-	0	false	String object	error
0	"0"	-	false	Number object	error
NaN	"NaN"	-	true	Number object	error
Infinity	"Infinity"	-	true	Number object	error
Negative Infinity	"-Infinity"	-	true	Number object	error
any other number	string value of number	-	true	Number object	error
true	"true"	1	-	Boolean object	error
false	"false"	0	-	Boolean object	error
object or array	toString() result, or object type	valueOf() result, or error	valueOf() result, or true	-	error (unless Function obj)
null	"null"	0	false	-	error
undefined value	"undefined"	error	false	error	error
function	Complete function text	error	true	Function object	-



- Exemplar
 - Lots of lines like these...

```
// First operand is a string
document.write("<TR>")
document.write("<TH>string \"test\"</TH>")
document.write("<TD>")
document.write(s1+s2)
document.write("</TD><TD>")
document.write(s1+i)
document.write("</TD><TD>")
document.write(s1+r)
document.write("</TD><TD>")
document.write(s1+lt)
document.write("</TD><TD>")
document.write(s1+lf)
document.write("</TD><TD>")
document.write(s1+n)
document.write("</TD>")
document.write("</TR>")
```

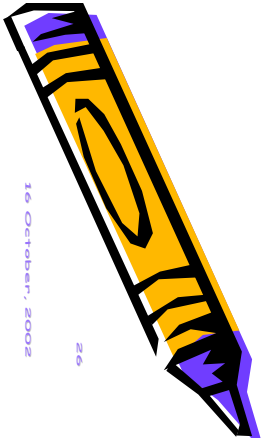
```
//First operand is a logical false
document.write("<TR>")
document.write("<TH>null</TH>")
document.write("<TD>")
document.write(n+s2)
document.write("</TD><TD>")
document.write(n+i)
document.write("</TD><TD>")
document.write(n+r)
document.write("</TD><TD>")
document.write(n+lt)
document.write("</TD><TD>")
document.write(n+lf)
document.write("</TD><TD>")
document.write(n+n)
document.write("</TD>")
document.write("</TR>")
```

Implicit conversion between types - Microsoft Internet Explorer

Address C:\Documents and Settings\Bob\Desktop\Mum's ex1.html File Edit View Favorites Tools

Implicit conversion between types

row + column	string "12.34"	integer 123	float .123	logical true	logical false	null
string "test"	test12.34	test123	test0.123	testtrue	testfalse	testnull
Integer 123	12312.34	246	123.123	124	123	123
Float .123	0.12312.34	123.123	0.246	1.123	0.123	0.123
logical true	true12.34	124	1.123	2	1	1
logical false	false12.34	123	0.123	1	0	0
null	null12.34	123	0.123	1	0	0

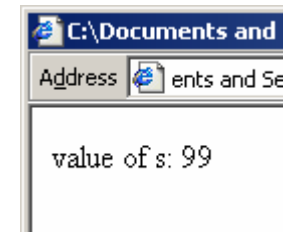


16 October, 2002

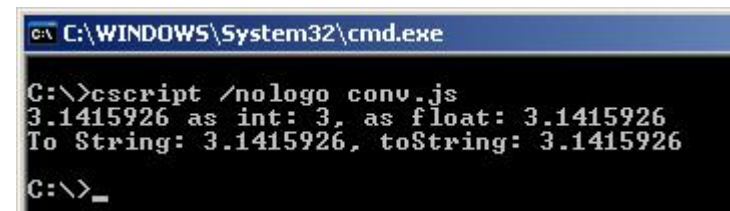
Explicit Conversions

- Functions: `eval()`, `parseInt()`, `parseFloat()`

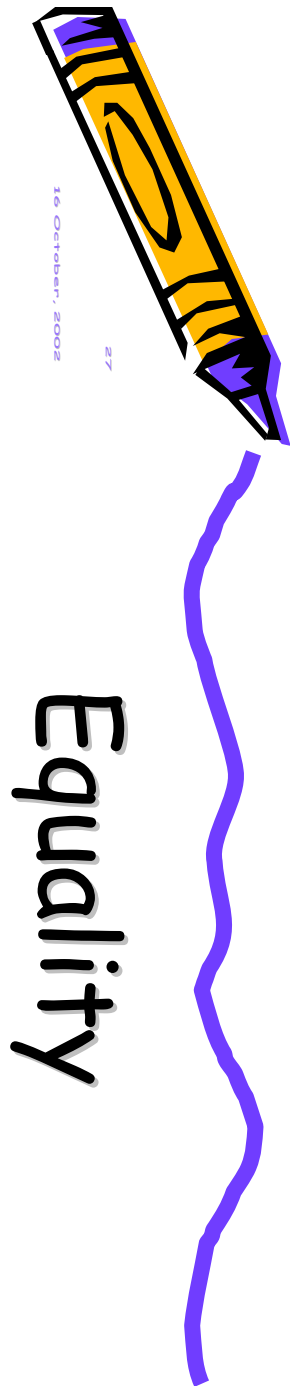
```
<html>
<body>
<script language="JavaScript">
  eval("var s = 99; document.write('value of s: ' + s);");
</script>
</html>
```



```
var val = "3.1415926"
var int = parseInt(val)
var float = parseFloat(val)
WScript.echo(val + " as int: " + int + ", as float: " + float)
WScript.echo("To String: " + val + ", toString: " + val.toString())
```

A screenshot of a Windows command prompt window. The title bar shows 'C:\WINDOWS\System32\cmd.exe'. The command prompt shows the following text:

```
C:\>cscript /nologo conv.js
3.1415926 as int: 3, as float: 3.1415926
To String: 3.1415926, toString: 3.1415926
C:\>_
```



- Pretty standard operators

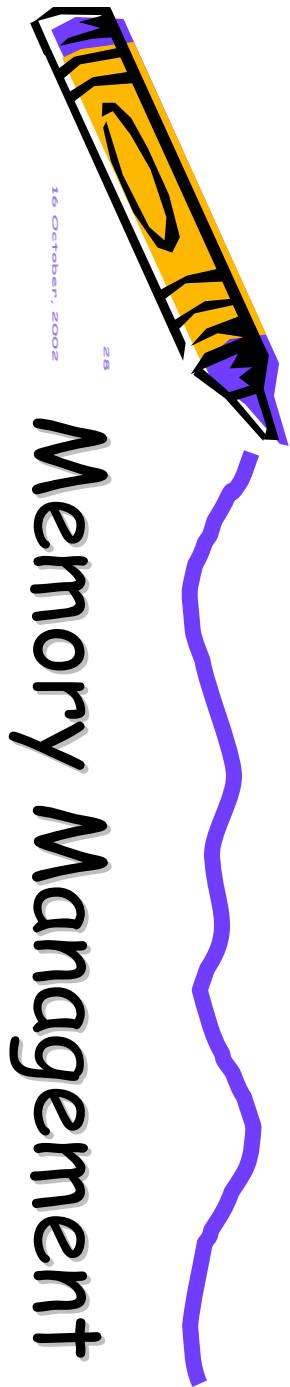
Operator	Description	Examples returning true ¹
Equal (==)	Returns true if the operands are equal. If the two operands are not of the same type, JavaScript attempts to convert the operands to an appropriate type for the comparison.	3 == var1 "3" == var1 3 == '3'
Not equal (!=)	Returns true if the operands are not equal. If the two operands are not of the same type, JavaScript attempts to convert the operands to an appropriate type for the comparison.	var1 != 4 var2 != "3"
Strict equal (===)	Returns true if the operands are equal and of the same type.	3 === var1
Strict not equal (!==)	Returns true if the operands are not equal and/or not of the same type.	var1 !== "3" 3 !== '3'
Greater than (>)	Returns true if the left operand is greater than the right operand.	var2 > var1
Greater than or equal (>=)	Returns true if the left operand is greater than or equal to the right operand.	var2 >= var1 var1 >= 3
Less than (<)	Returns true if the left operand is less than the right operand.	var1 < var2
Less than or equal (<=)	Returns true if the left operand is less than or equal to the right operand.	var1 <= var2 var2 <= 5

*Netscape
documentation*

¹ These examples assume that `var1` has been assigned the value 3 and `var2` has been assigned the value 4.

```
var float = 1.0
var str = "1"
WScript.echo(typeof(float));
WScript.echo(typeof(str));
```

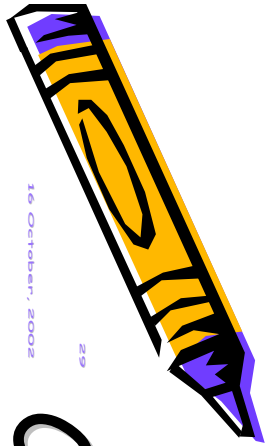
```
C:\WINDOWS\System32\cmd.exe
C:\> cscript /nologo typeof.js
number
string
C:\>
```



- Garbage collection
 - Removes need to keep track of memory (de)allocation
 - Humans are notoriously bad at this!
 - Objects created with the 'new' operator
 - Memory reused when variable goes out of scope/is no longer referenced
 - Delete operator can be used to explicitly mark something as "no longer needed"
 - Only for array elements or object properties

```
var o = new Object();  
o.s = "hello";  
WScript.echo("one: " + o.s);  
delete o.s;  
WScript.echo("two: " + o.s);
```

```
C:\WINDOWS\System32\cmd.exe  
  
C:\> cscript /nologo del.js  
one: hello  
two: undefined  
  
C:\>
```



16 October, 2002

Control Structures

- Basic flow of control
 - *mostly* as in C/Java

```
do
{
    statement(s);
} while (condition);
```

```
while (condition)
{
    statement(s);
}
```

```
TEST:
if (check (i))
{
    for (var j = 0; j < 10; j ++ )
    {
        if (j > i) break;
        if (a [i][j] == null)
            break TEST;
        // do something ...
    }
    // break TEST goes to HERE!
```

```
switch (expression)
{
    case v0:
        statement(s);
        break;

    case v1:
    case v2:
        statement(s);
        break;

    default:
        statement(s);
        break;
}
```

```
if (condition0)
{
    statement(s);
}

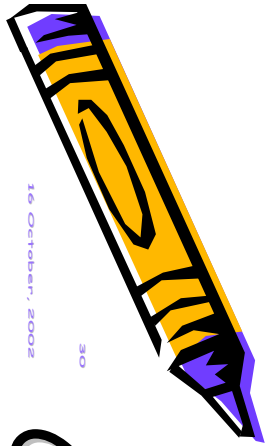
else if (condition1)
{
    statement(s);
}

else
{
    statement(s);
}
```

```
for (initial; condition; progress)
{
    statement(s);
}
```

```
for ( ; ; )
{
    statement(s);
    if (condition)
        break;
    statement(s);
}
```

```
for ( ; ; )
{
    ;
}
```



Control Structures...

- Object related

```
with (object)
{
    statement(s);
}
```

```
var a, x, y;
var r=10;
with (Math)
{
    a = PI * r * r;
    x = r * cos(PI);
    y = r * sin(PI/2);
}
```

```
for (var in object)
{
    statement(s);
}
```

```
// Create an object with some properties
var myObject = new Object();
myObject.name = "James";
myObject.age = "22";
myObject.phone = "555 1234";
```

```
// Enumerate (loop through)_all the properties in the object
for (prop in myObject)
{
    // This displays "The property 'name' is James", etc.
    window.alert("The property '" + prop + "' is " + myObject[prop]);
}
```



Functions

- Pascal-like syntax

```
function factorial(n)
{
  if ((n == 0) || (n == 1))
    return 1;
  else
    return(n * factorial(n-1));
}
```

- Arguments:

- Copied in by value
- Placed in an array
 - Thus, allows variable number

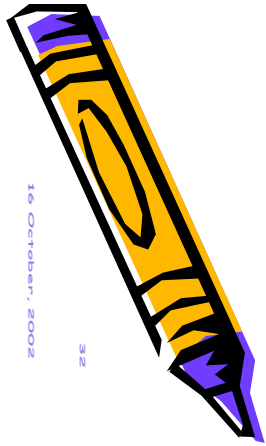
```
function myConcat(separator)
{
  result="" // initialize list
  for (var i=1; i<arguments.length; i++)
    result += arguments[i] + separator;
  return result
}
```

```
// returns "red, orange, blue, "
myConcat(", ", "red", "orange", "blue")
// returns "sage. basil. oregano. pepper. parsley. "
myConcat(".", "sage", "basil", "oregano", "pepper", "parsley")
```

```
C:\WINDOWS\System32\cmd.exe
C:\> cscript /nologo func.js
Val: 100
C:\>
```

```
function outer()
{
  function inner()
  {
    return 1;
  }
  return(inner() + 99);
}
WScript.echo("Val: " + outer());
```

Exceptions



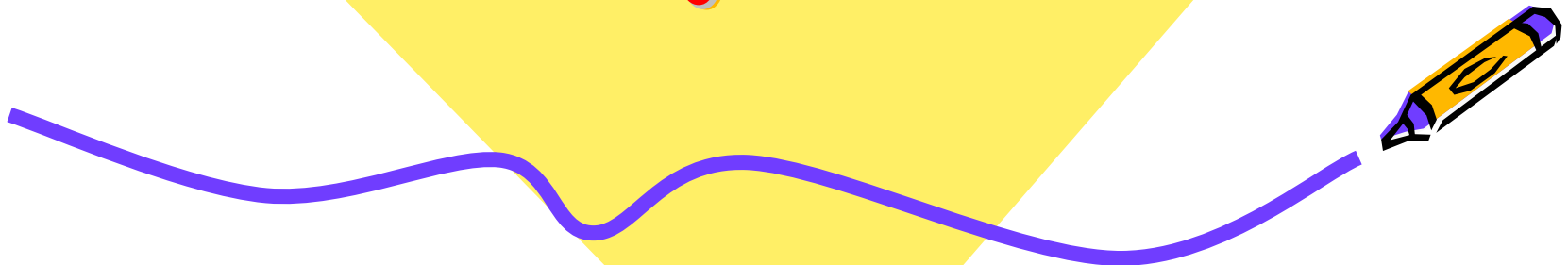
- Introduced in JScript5
 - See also the Error object

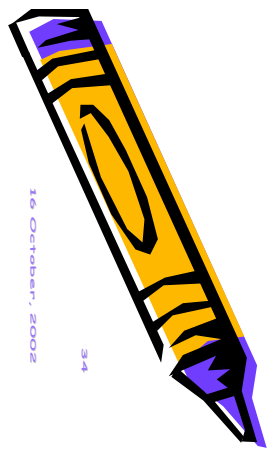
```
<HTML>
<HEAD>
<TITLE>Exception Handling Example</TITLE>
</HEAD>
<BODY>
<IMG SRC="image.gif">
<SCRIPT LANGUAGE="JScript">
<!--
try {
    for (image=0; image < document.images.length + 1; image++) {
        alert("The width of image " + image + " is " +
            document.images[image].width);
    }
}
catch(obj) {
    alert("We have experienced a JavaScript error");
}
finally {
    alert("Nothing more to do.");
}
// -->
</SCRIPT>
</BODY>
</HTML>
```





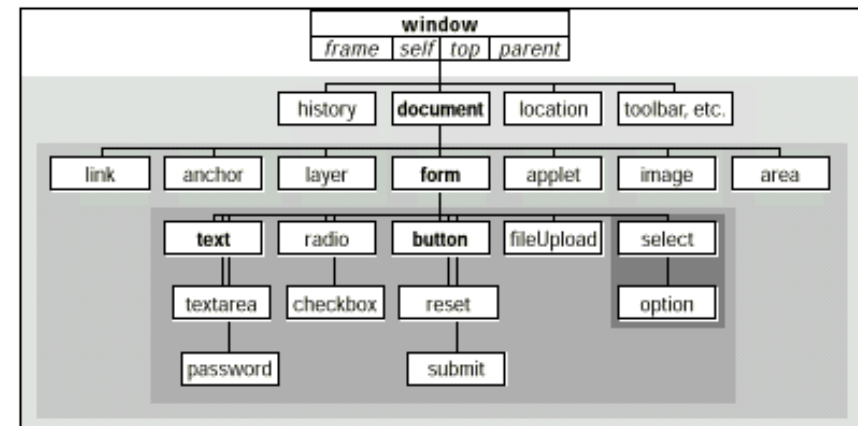
The JavaScript Language: Objects



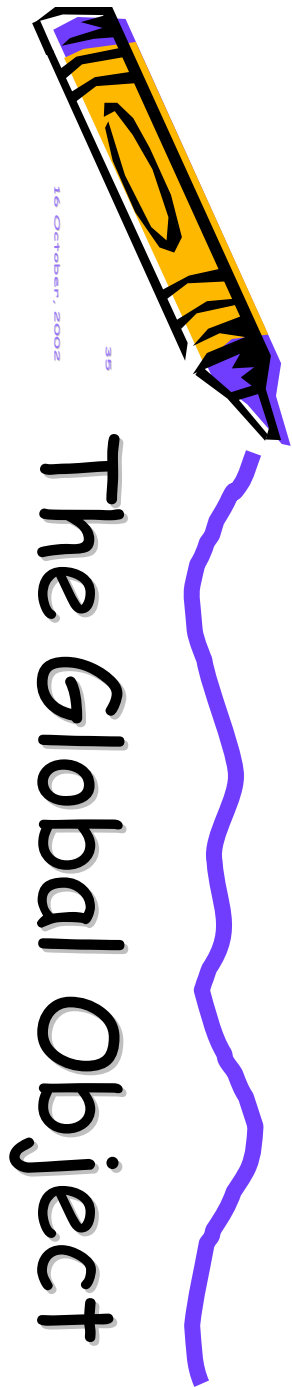


Preamble

- The standard Objects effectively constitute a function library arranged into meaningful groupings
 - Much of JavaScript is encapsulated into the standard Objects
- Various Objects are pre-supplied as singletons
- Other Objects require instantiation

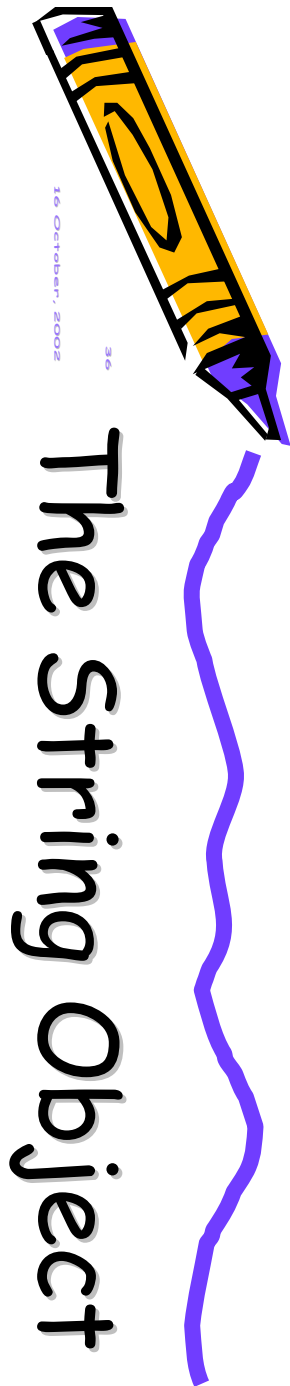


Netscape 4 object heirarchy



- (Partially) defined and provided by the environment
 - May be known by another term
 - E.g. browser's 'window' object

"The global object is not discussed in the JavaScript documentation because you do not use it directly. The methods and properties of the global object, which you do use, are discussed in the JavaScript documentation but are called top-level functions and properties."



- Fairly standard string operations
- Also has a number of methods for formatting HTML

- bold() to create boldface text
- link() to create a hyperlink
- Etc.

```

<BODY>
<SCRIPT language="JavaScript">
<!--
    var heading = prompt("Please enter a heading", "Test Heading");
    var colour = prompt("Please enter a colour", "darkred");
    document.write(heading.fontSize(7).fontcolor(colour).bold().toUpperCase());

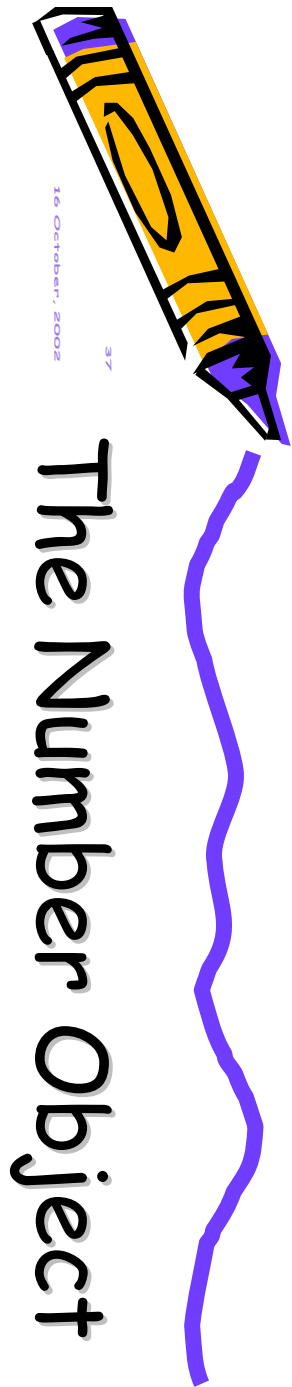
// -->
</SCRIPT>
</BODY>

```

Method	Description
anchor	Creates HTML named anchor
big, blink, bold, fixed, italics, small, strike, sub, sup	Creates HTML formatted string
charAt, charCodeAt	Returns the character or character code at the specified position in string
indexOf, lastIndexOf	Returns the position of specified substring in the string or last position of specified substring, respectively
link	Creates HTML hyperlink
concat	Combines the text of two strings and returns a new string
fromCharCode	Constructs a string from the specified sequence of ISO-Latin-1 codeset values
split	Splits a String object into an array of strings by separating the string into substrings
slice	Extracts a section of an string and returns a new string
substring, substr	Returns the specified subset of the string, either by specifying the start and end indexes or the start index and a length
match, replace, search	Used to work with regular expressions
toLowerCase, toUpperCase	Returns the string in all lowercase or all uppercase, respectively

*Netscape
documentation*



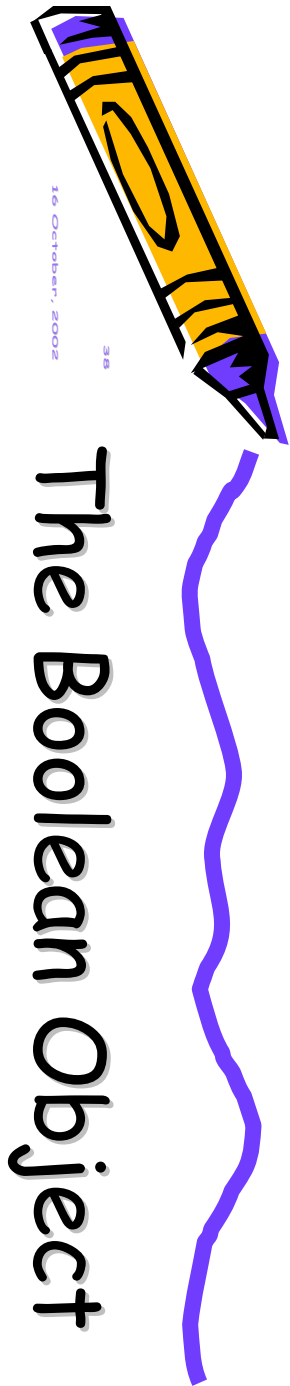


- Wraps the number primitive

Method	Description
MAX_VALUE	The largest representable number
MIN_VALUE	The smallest representable number
NaN	Special "not a number" value
NEGATIVE_INFINITY	Special infinite value; returned on overflow
POSITIVE_INFINITY	Special negative infinite value; returned on overflow

*Netscape
documentation*

"The Number Object represents the number primitive...Contains few methods or properties and is rarely used in web pages. All the useful mathematical functions are in the Math object."

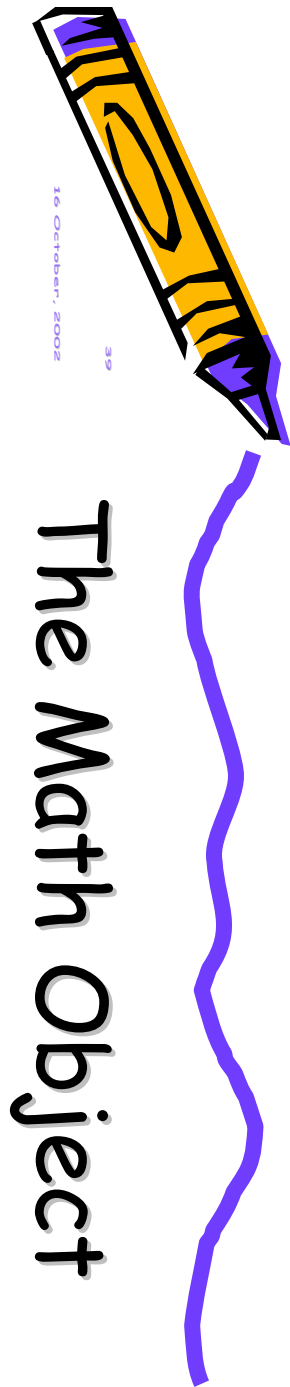


- A wrapper around the primitive Boolean data type

"The Boolean object...has even less functionality than the Number object, and it is extremely unlikely that we would ever actually use it."

"Do not confuse the primitive Boolean values true and false with the true and false values of the Boolean object. Any object whose value is not undefined or null, including a Boolean object whose value is false, evaluates to true when passed to a conditional statement."

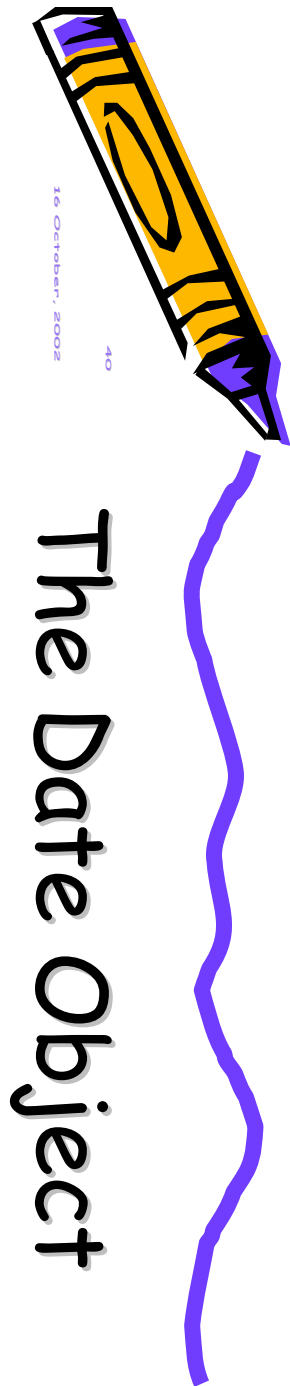
```
var b = new Boolean(false);  
if (b) // this condition evaluates to true  
{  
  ...  
}
```



- Predefined: never needs to be created
 - Library of useful functions

Method	Description
abs	Absolute value
sin, cos, tan	Standard trigonometric functions; argument in radians
acos, asin, atan	Inverse trigonometric functions; return values in radians
exp, log	Exponential and natural logarithm, base e
ceil	Returns least integer greater than or equal to argument
floor	Returns greatest integer less than or equal to argument
min, max	Returns greater or lesser (respectively) of two arguments
pow	Exponential; first argument is base, second is exponent
round	Rounds argument to nearest integer
sqrt	Square root

*Netscape
documentation*



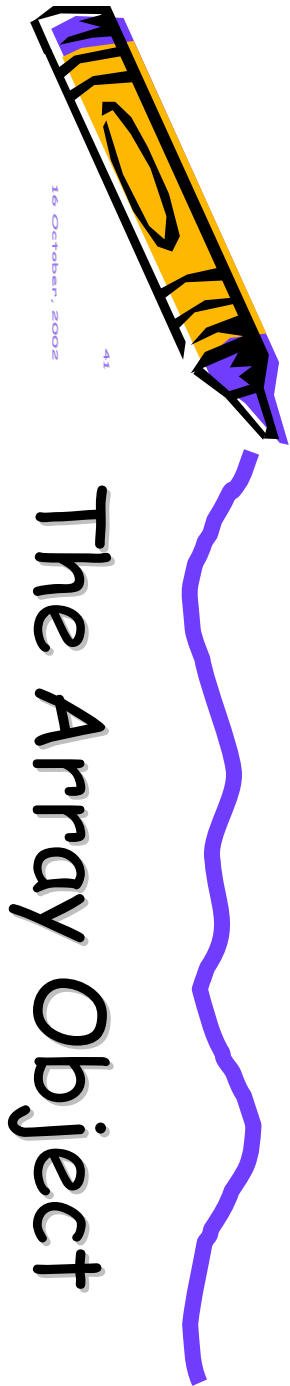
- Large number of methods for setting, getting, and manipulating dates

```
function JSClock()  
{  
  var time = new Date()  
  var hour = time.getHours()  
  var minute = time.getMinutes()  
  var second = time.getSeconds()  
  var temp = "" + ((hour > 12) ? hour - 12 : hour)  
  temp += ((minute < 10) ? ":0" : ":") + minute  
  temp += ((second < 10) ? ":0" : ":") + second  
  temp += (hour >= 12) ? " P.M." : " A.M."  
  return temp  
}
```

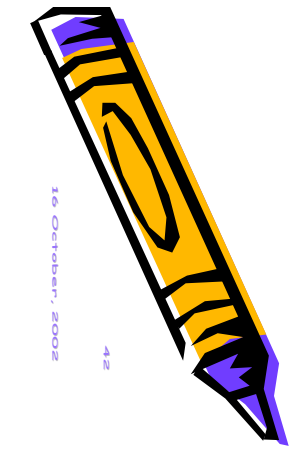
- Similar to Java's Date class
 - Many of the same methods
 - Broken in the same way, as well!
 - Store dates as the number of milliseconds since January 1, 1970, 00:00:00 (the epoch)
 - Range is +/-100,000,000 days relative to the epoch
 - Older versions (< 1.2) don't allow dates < the epoch
 - Just one of many 'scenes' in the nightmare

```
WScript.echo("Date: " +  
  new Date(2002, 1, 3666));
```

```
C:\WINDOWS\System32\cmd.exe  
C:\> cscript /nologo date.js  
Date: Tue Feb 14 00:00:00 UTC+1000 2012  
C:\> _
```

- JavaScript does not have an inbuilt array type
 - Use the predefined Array object and its methods to work with arrays
- Has methods for manipulating arrays in various ways:
 - `concat()`
 - `join()`
 - `pop()/push()`
 - `reverse()`
 - `shift()/unshift()`
 - `sort()`
 - Etc.
- Can grow dynamically
 - Can also truncate an array
- Can be sparse
- Indexed from zero
- Have a standard dynamic property for determining the current array length
- Can have array literals
- Can hold mixed contents
 - Since JavaScript is typeless



Arrays...

- Arrays can be multidimensional
 - And ragged
- The Array Object can be extended
 - See later...

```
myArray = new Array("Hello", myVar, 3.14159)
```

```
stars = [  
    "Aldebaran",  
    "Alcor",  
    "Betelgeuse"  
]
```

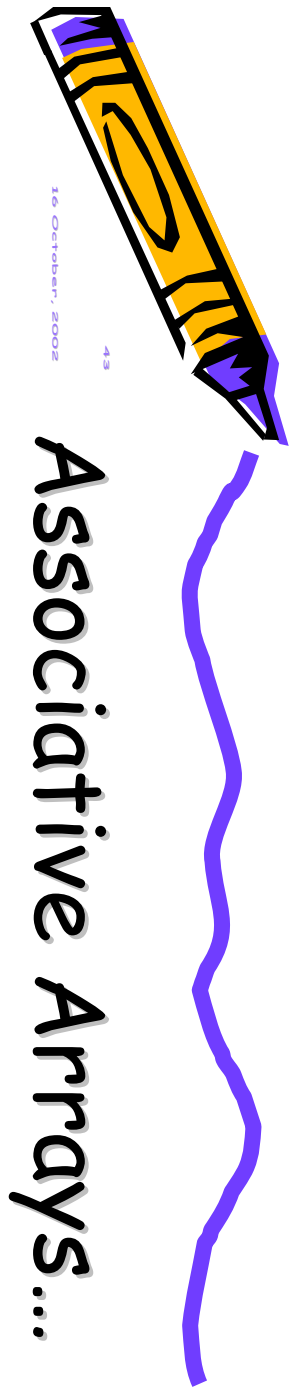
```
C:\WINDOWS\System32\cmd.exe  
C:\> cscript /nologo array.js  
a.length: 10  
ai: 0  
ai: 4  
ai: 9  
a[0]: zero  
a[1]: undefined  
a[2]: undefined  
a[3]: undefined  
a[4]: four  
a[5]: undefined  
a[6]: undefined  
a[7]: undefined  
a[8]: undefined  
a[9]: nine  
C:\> _
```

```
var a = new Array()  
a[0] = "zero"  
a[4] = "four"  
a[9] = "nine"  
WScript.echo("a.length: " + a.length)  
for (ai in a)  
    WScript.echo("ai: " + ai);  
for (var i = 0; i < a.length; i++)  
    WScript.echo("a[" + i + "]: " + a[i]);
```

```
var where_is_mytool="home/mytool/mytool.cgi";  
var mytool_array=where_is_mytool.split("/");  
WScript.echo(mytool_array[0]+" "+  
    mytool_array[1]+" "+mytool_array[2]);
```

```
var aSize = 4  
var a = new Array(aSize)  
for (var i=0; i < aSize; i++)  
{  
    a[i] = new Array(i)  
    for (var j=0; j < i; j++)  
    {  
        a[i][j] = "["+i+","+j+"]"  
    }  
}
```

```
C:\WINDOWS\System32\cmd.exe  
C:\> cscript /nologo mytool.js  
home mytool mytool.cgi  
C:\> _
```



- Indexed by string, not by number
 - Interesting object/array tie-up

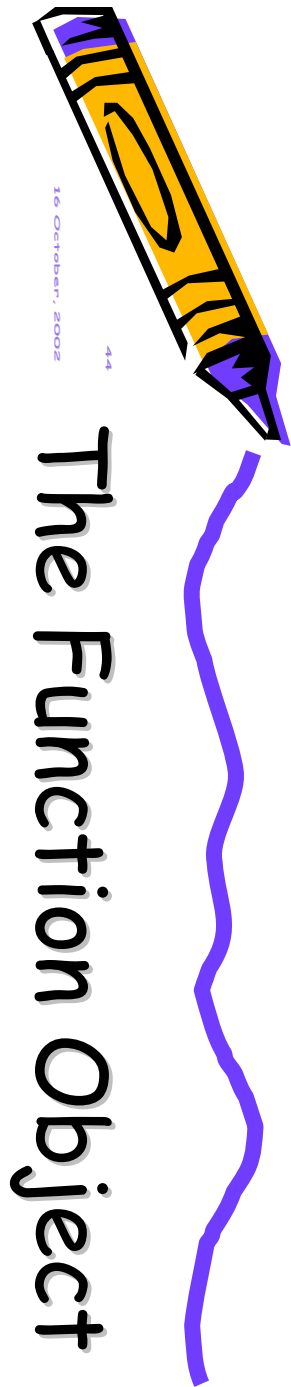
```
var myCar = new Array()
myCar["make"] = "Ford"
myCar["model"] = "Mustang"
myCar["year"] = 1967
for (var i in myCar)
    WScript.echo("myCar [" + i + "] = " + myCar[i]);
WScript.echo(myCar.make)
WScript.echo(myCar.model)
WScript.echo(myCar.year)
```

"Properties and arrays in JavaScript are intimately related; in fact, they are different interfaces to the same data structure"

```
C:\WINDOWS\System32\cmd.exe

C:\> cscript /nologo assoc.js
myCar [make] = Ford
myCar [model] = Mustang
myCar [year] = 1967
Ford
Mustang
1967

C:\> _
```



The Function Object

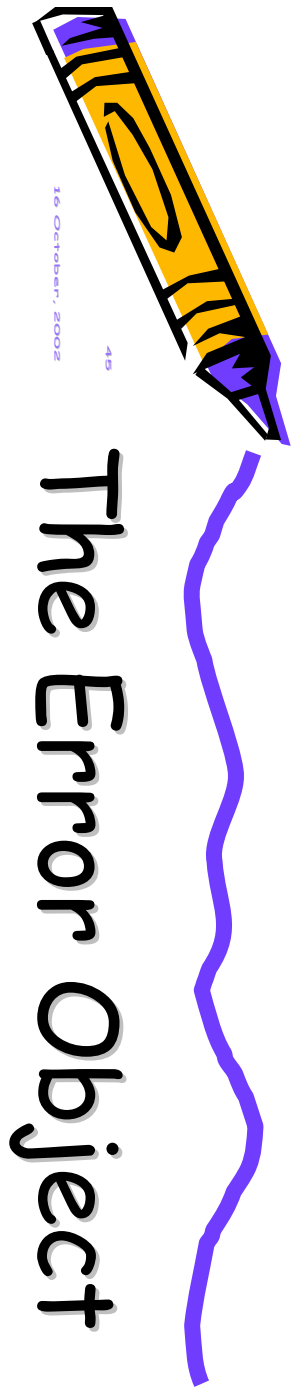
- A Metaprogramming facility
- Has a number of useful properties
 - arguments
 - callee, caller

```
C:\WINDOWS\System32\cmd.exe
C:\> cscript /nologo callee.js
Callee: function inner()
{
  WScript.echo("Callee: " + arguments.callee);
  return 1;
}
Value: 100
C:\> _
```

```
function outer()
{
  function inner()
  {
    WScript.echo("Callee: " + arguments.callee);
    return 1;
  }
  return(inner() + 99);
}
WScript.echo("Value: " + outer())
```

- Compiles a function from a string

```
var average =
  new Function("sum = 0.0; " +
    "for (var i = 0; i < arguments.length; i++) " +
    "sum += arguments[i]; return(sum/arguments.length);")
WScript.echo("Avg: " + average(3,7,5,9,6))
```



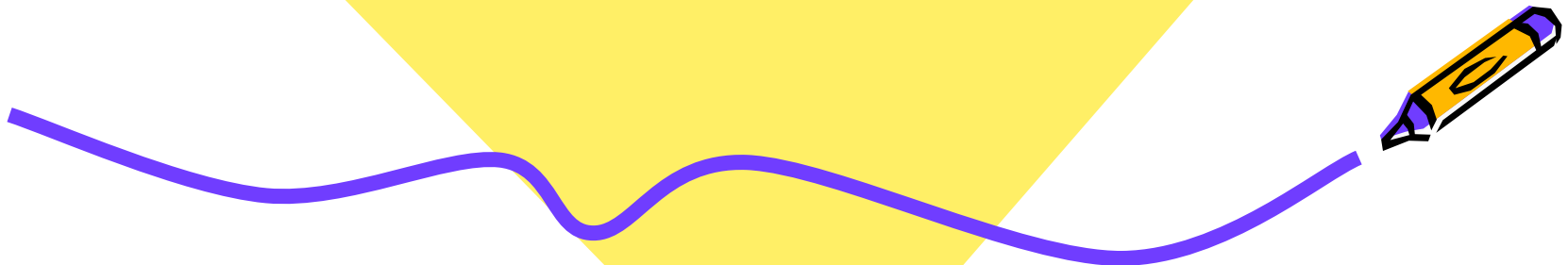
- Introduced in JScript5
 - Also in JavaScript1.5
- An instance of the Error object is created whenever a run-time error occurs
 - Two intrinsic properties used to describe the error:
 - description
 - number

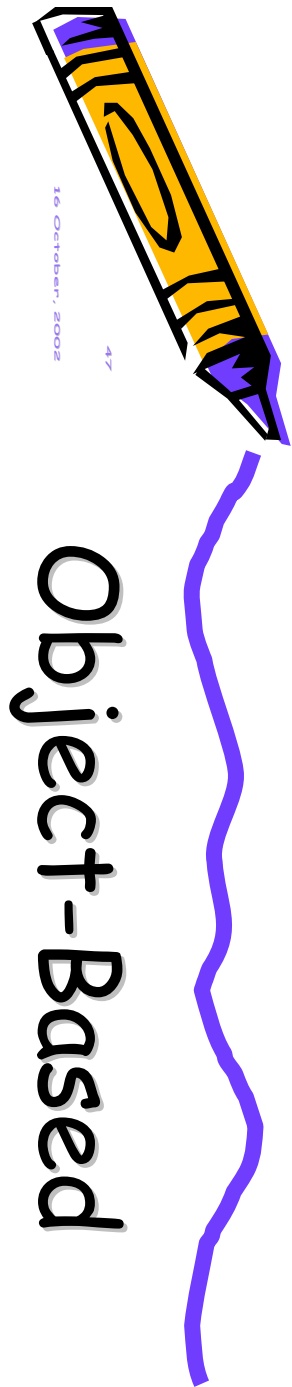
```
try
{
    x = y    // Cause an error.
}
catch(e)
{
    WScript.echo(e)
    WScript.echo(e.number & 0xFFFF)
    WScript.echo(e.description)
}
```

```
C:\WINDOWS\System32\cmd.exe
C:\> cscript /nologo unde.js
[object Error]
5009
'y' is undefined
C:\> _
```

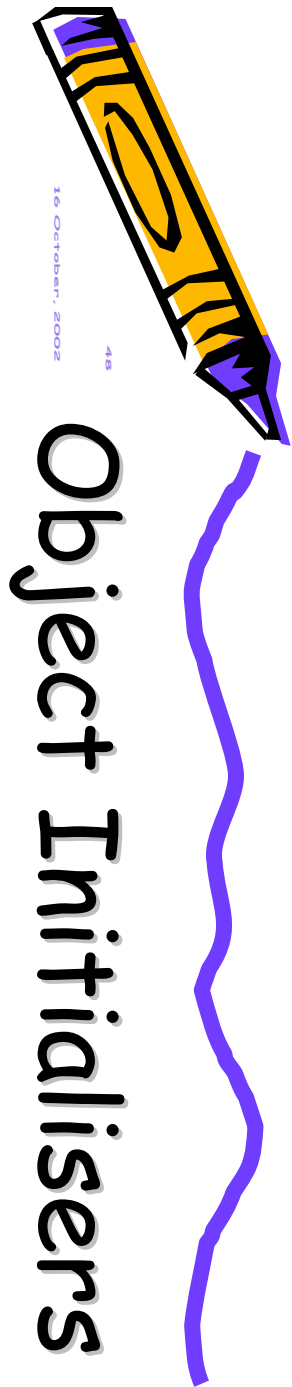


The JavaScript Language: OOP





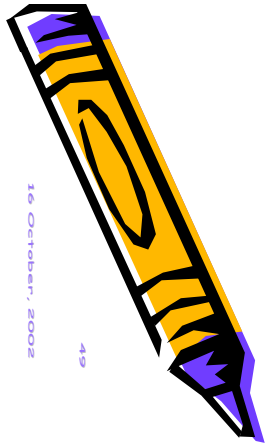
- Not Object-Oriented
 - According to Netscape:
 - *"JavaScript is designed on a simple object-based paradigm. An object is a construct with properties that are JavaScript variables or other objects. An object also has functions associated with it that are known as the object's methods."*
 - A strange, uncomfortable beast...



- Inline, literal Objects

```
myHonda = {color:"red",wheels:4,engine:{cylinders:4,size:2.2}}  
WScript.echo(myHonda.engine.cylinders)
```

```
C:\WINDOWS\System32\cmd.exe  
  
C:\> cscript /nologo honda.js  
4  
  
C:\> _
```

Constructors

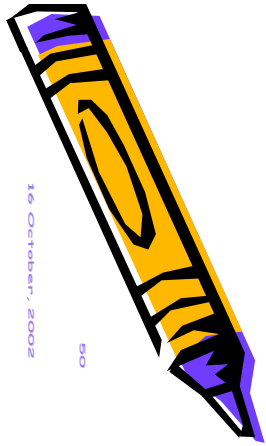
- Function to which the new operator is applied to create an instance of an object
 - Note the use of the this keyword
 - Refers to *"this instance"*

```
function Person(lastName, firstName){
  this.lastName = lastName;
  this.firstName = firstName;
}
var bobBrown = new Person("Brown", "Bob");
```

```
function Book(title, pages, price)
{
  var arity = 3;
  var authorArgs = arguments.length-arity;
  this.title = title;
  this.pages = pages;
  this.price = price;
  this.authors = new Array(authorArgs);
  for(i=0; i<authorArgs; i++)
    this.authors[i] = arguments[i+arity];
}
var bobsBook = new Book("Sleeping for Beginners", 731, 29.95, bobBrown);
```

```
function MyObject() { ... }
var object = new MyObject();
switch (object.constructor)
{
  case MyObject:
    WScript.echo("MyObject...");
    break;
  default:
    break;
}
```

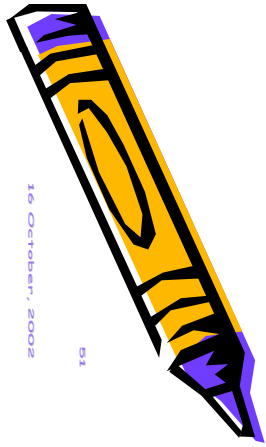
Properties



- Members of an object
- Object instances can be *dynamically* updated with new properties
 - Sounds hoopy, makes for *painful* debugging!
 - Typos, etc. have *strange* consequences...

```
var bobsBook = new Book("Sleeping for Beginners", 731, 29.95, bobBrown)
bobsBook.binding = "hardback"
```

Prototypes



- Object properties associated on a *per-instance* basis
- Every object has a prototype; all instances *share* the prototype

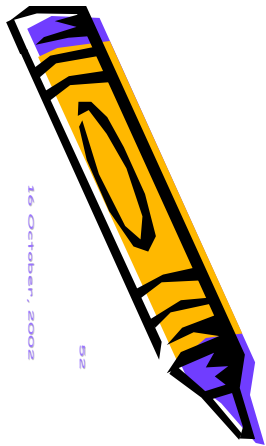
```
function Fred()
{
    this.thing = "world"
}

var f = new Fred()
var f1 = new Fred()
WScript.echo("Add to instance...");
f.something = "hello"
WScript.echo(" f:");
for (var i in f)
    WScript.echo("  " + i);
WScript.echo(" f1:");
for (var i in f1)
    WScript.echo("  " + i);

WScript.echo("Add to prototype...");
Fred.prototype.anotherThing = "something"
WScript.echo(" f:");
for (var i in f)
    WScript.echo("  " + i);
WScript.echo(" f1:");
for (var i in f1)
    WScript.echo("  " + i);
```

```
C:\WINDOWS\System32\cmd.exe
C:\> cscript /nologo y.js
Add to instance...
f:
  thing
  something
f1:
  thing
Add to prototype...
f:
  anotherThing
  thing
  something
f1:
  anotherThing
  thing
C:\> _
```

Prototypes...



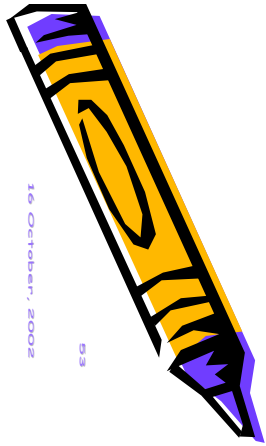
16 October, 2002

62

- Prototypes "give shape" to all instances
 - Provides functions and properties

```
function Car(){ }  
Car.prototype.speed = 0;  
Car.prototype.purchased = false;  
Car.prototype.accelerate = function( )  
{  
    this.speed++;  
}  
Car.prototype.decelerate = function( )  
{  
    this.speed--;  
}  
Car.prototype.getSpeed = function( )  
{  
    return this.speed;  
}  
  
var myCar = new Car();
```

Prototypes...

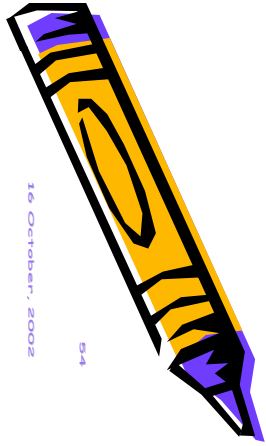


- Make it possible to augment standard libraries
 - Can provide new functionality
 - Can compensate for environmental 'strangeness'

```
// does this feature exist? Is it broken?
if(!Array.prototype.push||Array(6,6,6,6).push(4)!= 5)
  Array.prototype.push=function(items)
  {
    for(var x=0;x<arguments.length;x++)
      this[this.length]=arguments[x];
    return this.length;
  }
```

```
// augments the Date standard Object
Date.prototype.isLeapYear=function(intYear)
{
  return (intYear%400==0 ? true:
    (intYear%100==0 ? false:
      (intYear%4==0 ? true:false )));
}
```

Inheritance

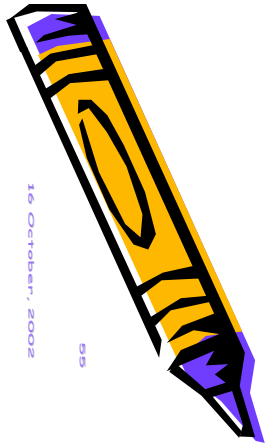


- Reflects the 'is-a' relationship
- Actually a strange, codified form of containment...
- Simplest (= most commonly used) technique is to inherit from an instance

```
function Employee(name, dept)
{
  this.name = name || "";
  this.dept = dept || "general";
}

function WorkerBee(projs)
{
  this.projects = projs || [];
}
WorkerBee.prototype = new Employee;

function Engineer(mach)
{
  this.dept = "engineering";
  this.machine = mach || "";
}
Engineer.prototype = new WorkerBee;
```



Inheritance...

- Simple instance inheritance fails when a class contains private members

- Inherit from constructor

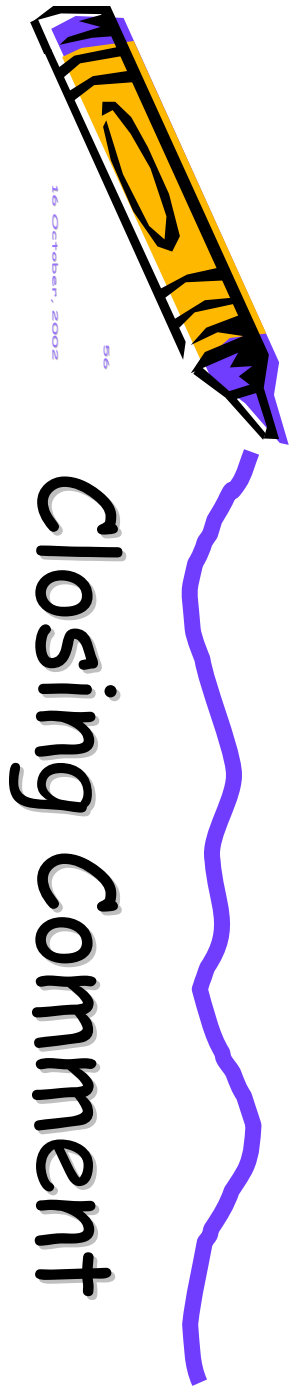
```
function Car()
{
    var speed = 0
    this.accelerate = function() { speed ++ }
    this.decelerate = function() { speed -- }
    this.getSpeed = function() { return speed }
}
```

```
function Mitsubishi()
{
    Car.call(this);
    this.make = "Mitsubishi"
}
```

```
function Magna()
{
    Mitsubishi.call(this)
    this.door = 4
    this.year = 2001
}
```

```
var m = new Magna();
for (var i in m)
    WScript.echo("Property: " + i + ", value: " + m[i])
```

```
C:\WINDOWS\System32\cmd.exe
C:\> cscript /nologo magna.js
Property: accelerate, value: function() { speed ++ }
Property: decelerate, value: function() { speed -- }
Property: getSpeed, value: function() { return speed }
Property: make, value: Mitsubishi
Property: door, value: 4
Property: year, value: 2001
C:\> _
```

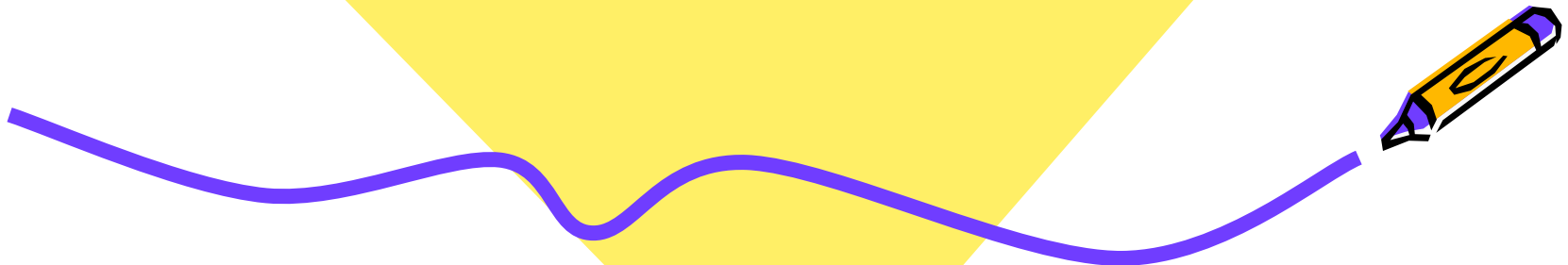


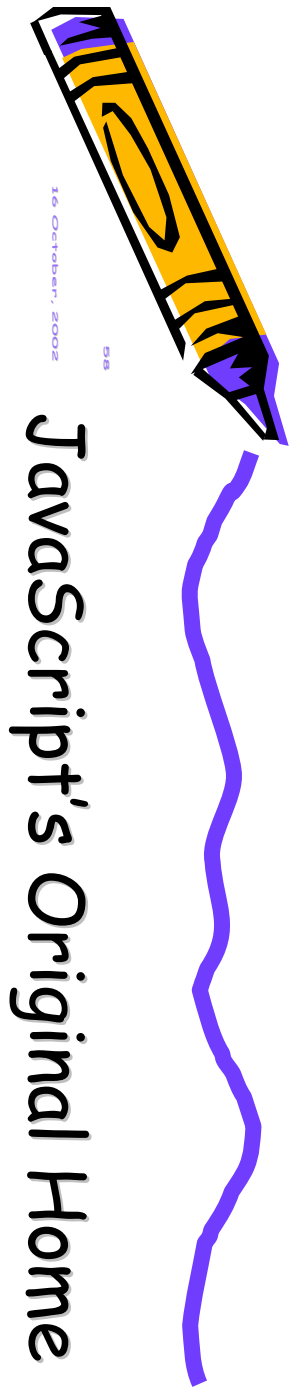
- From *Professional JavaScript 2nd Ed.*:

"In conclusion, we can say that although JavaScript is not a true OO language, it has, or can be made to act as though it has most of the features that identify an OO language. JavaScript is often referred to as an object-based language, because of the features it is lacking."

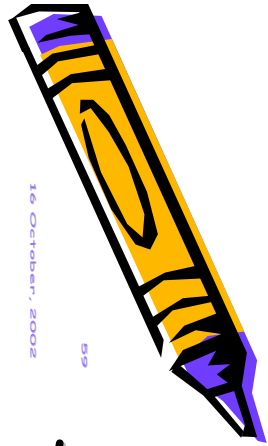


JavaScript and the Browser Environment





- JavaScript was originally conceived as a browser-specific technology
 - Still the major use, but branching out
 - Shell scripts
 - Server-side functionality
 - IBM's Sash 'weblications' system
 - Etc.

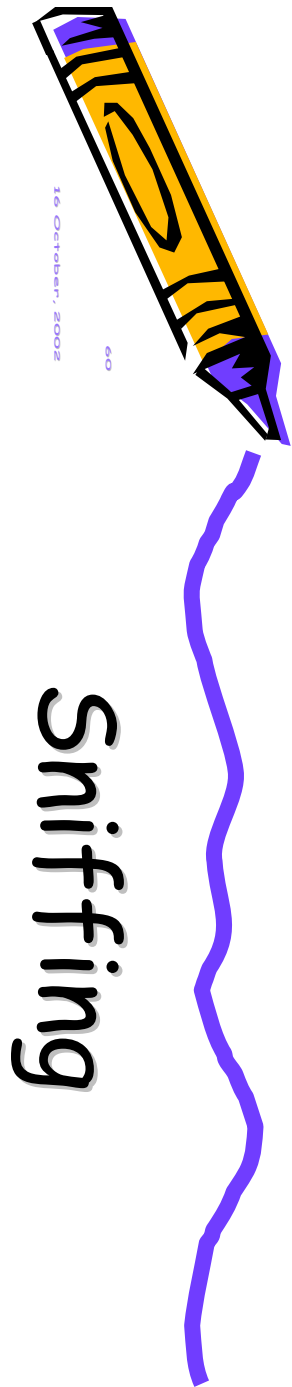


The Browser Environment

- Special features
 - Special Objects
 - The navigator object
 - The window object
 - Constrained container

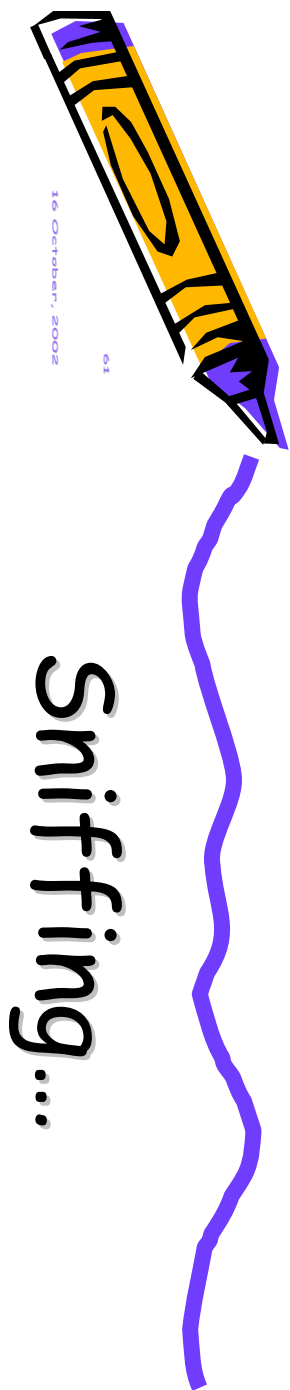
"Inside a host environment, JavaScript can be connected to the objects of its environment to provide programmatic control over them. ...client-side extensions allow an application to place elements on an HTML form and respond to user events such as mouse clicks, form input, and page navigation."

*Netscape
documentation*



- So many browser versions...what's a poor developer to do?
- Two approaches
 - Browser sniffing code
 - Look for various properties in the 'navigator' browser-supplied object
 - Object sniffing code
 - Explicitly look for required features

'A common practice on the web when coding cross browser javascript is to scatter "if" statements all throughout your scripts checking for which browser is being used.'



- Browser Sniffing

- Extensively covered at

- http://www.mozilla.org/docs/web-developer/sniffer/browser_type_oo.html

```
var agt=navigator.userAgent.toLowerCase();

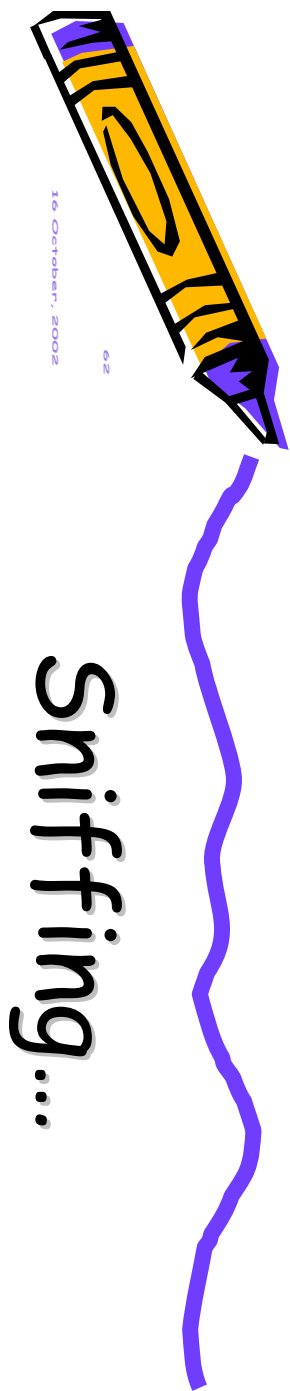
this.major = parseInt(navigator.appVersion);
this.minor = parseFloat(navigator.appVersion);

this.nav = ((agt.indexOf('mozilla')!=-1) && (agt.indexOf('spoofer')==-1)
  && (agt.indexOf('compatible') == -1) && (agt.indexOf('opera')==-1)
  && (agt.indexOf('webtv')==-1) && (agt.indexOf('hotjava')==-1));

...
if (this.nav2 || this.ie3) this.js = 1.0;
else if (this.nav3) this.js = 1.1;

...
this.win1b = ((agt.indexOf("win1b")!=-1) || (agt.indexOf("1bbit")!=-1)
  || (agt.indexOf("windows 3.1")!=-1) ||
  (agt.indexOf("windows 1b-bit")!=-1) );
```

'One thing to be aware of: when checking the "navigator.appVersion" property, you'll notice on the Windows side that Internet Explorer 3.0 says it's Navigator 2.0, and thus the "navigator.appVersion" property returns a "2." On the Macintosh, Internet Explorer 3.0 says it's Navigator 3.0, and "navigator.appVersion" will return a "3".'



- Object Sniffing

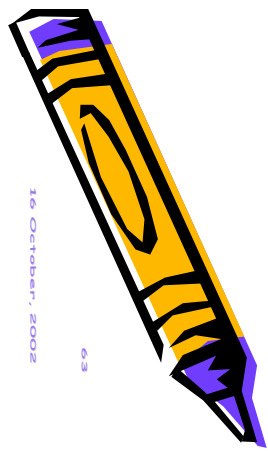
- Simpler and more to the point
 - Gets (more?) fiddly, though...

- Covered at

<http://developer.netscape.com/evangelism/tools/practical-browser-sniffing/ua-test.html>

```
<script>
if (document.images){
//do rollover effect
}
</script>
```

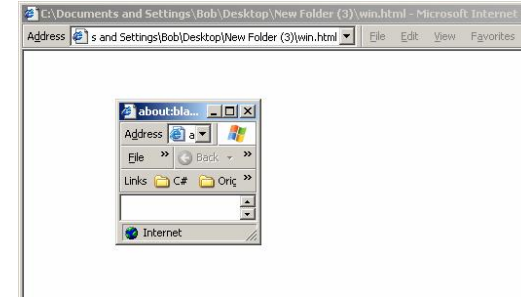
"For checking for support of a particular feature, capabilities detection is the way to go. However, some methods or objects may be supported by both browsers, but have slightly different semantics. When you need to distinguish between how one browser or another works with a particular feature, you need to use browser detection."



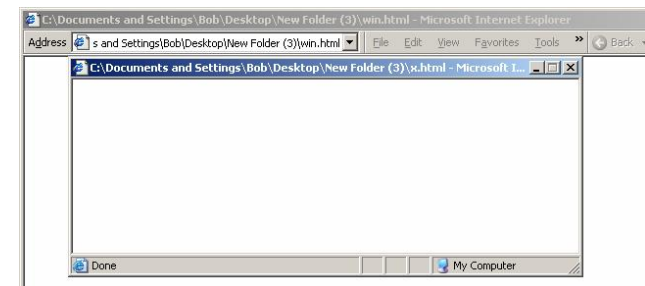
Windows

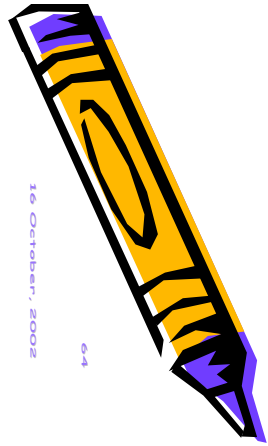
- The window object provides useful functionality

```
<html>
<head>
<script language="JavaScript">
var win = window.open()
win.resizeTo(150,150)
win.moveTo(100,100)
</script>
</head>
</html>
```



```
<html>
<head>
<script language="JavaScript">
var win = window.open("x.html",
    "ANewWindow",
    "left=50,top=50,toolbar=no,status=yes")
</script>
</head>
</html>
```

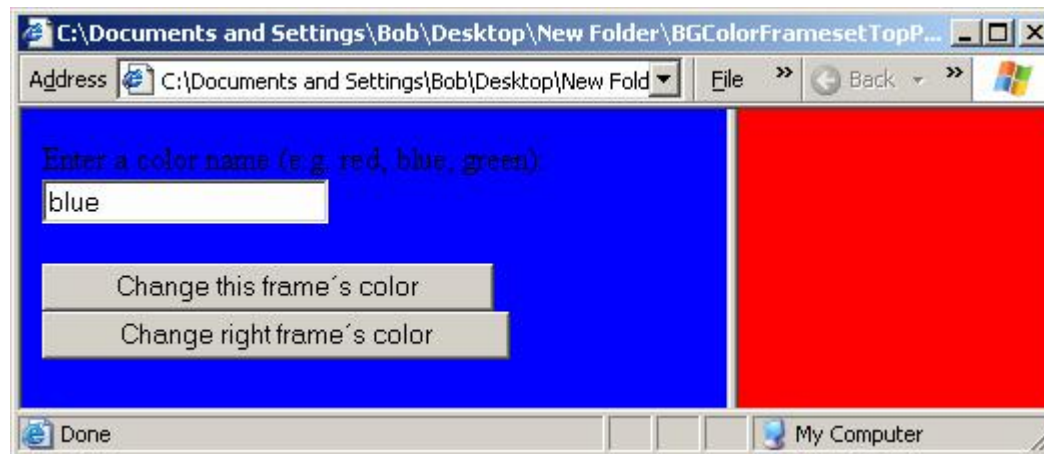


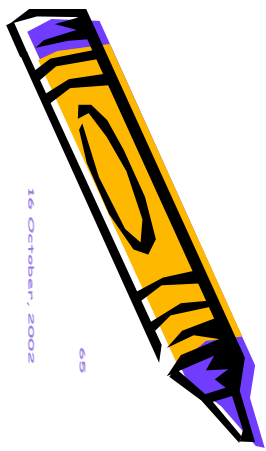


Frames

- Navigation/addressing between frames is possible
 - Again via the window object
 - And the associated parent/frames objects

```
<HTML>
<FRAMESET cols='50%,*'>
  <FRAME SRC='BGColorLeftFrame.htm' NAME='LeftFrame'></FRAME>
  <FRAME SRC='about:blank' NAME='RightFrame'></FRAME>
</FRAMESET>
</HTML>
```





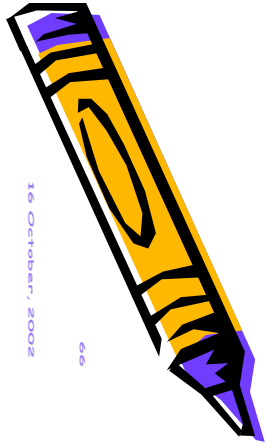
Frames...

```
<HTML>
<HEAD>
  <SCRIPT LANGUAGE='JavaScript'>
    function cmdChangeLeft_onclick() {
      var newBGColor = document.frmColorChanger.txtBGColor.value
      window.document.bgColor = newBGColor;
    }

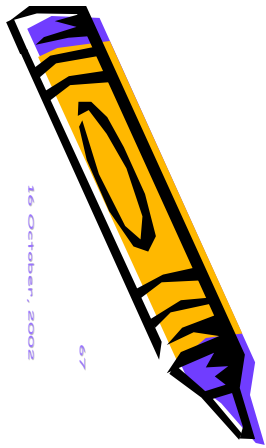
    function cmdChangeRight_onclick() {
      var windowObject = window.parent.frames["RightFrame"];
      var newBGColor = document.frmColorChanger.txtBGColor.value;
      windowObject.document.bgColor = newBGColor;
    }
  </SCRIPT>
</HEAD>
<BODY>
  <FORM ACTION='' METHOD=POST NAME='frmColorChanger'>
    <P>
      Enter a color name (e.g. red, blue, green):
      <INPUT TYPE='text' NAME='txtBGColor'></INPUT>
    </P>

    <INPUT TYPE='button' VALUE='Change this frame's color'
      NAME='cmdChangeLeftFrame'
      onClick='cmdChangeLeft_onclick()'></INPUT>
    <INPUT TYPE='button' VALUE='Change right frame's color'
      NAME='cmdChangeRightFrame'
      onClick='cmdChangeRight_onclick()'></INPUT>
  </FORM>
</BODY>
</HTML>
```

Form Handling



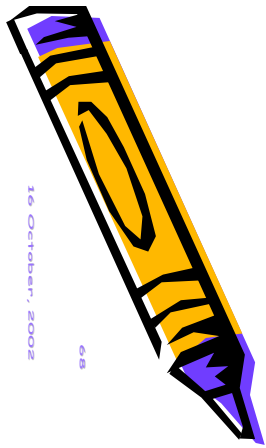
- A major use for JavaScript...
- Intercept the form submission process to allow simple validation of data entered by a user
 - Can save bandwidth, provide more 'immediate' feel from a web-based application
 - No more round-trips to a server simply to find missing fields, etc.
 - Probably still need to do validation at the server side *as well*...
 - Structure/contents checking division...
- Netscape form validation info/library
 - <http://developer.netscape.com/docs/examples/javascript/formval/overview.html>



Forms...

- Relies on the ability to navigate through a page's contents
 - Using the Document Object Model
 - More about this later...

```
<html>
<head>
  <title>A Simple Form</title>
</head>
<body>
  <form id="form1" name="form1" onSubmit="return validateForm(this)">
    <p>Name:
      <input size="32" name="fullname"><br>
      Email Address:
      <input size="32" name="email"><br>
      Date Available:
      <input size="16" name="available"><i>(mm/dd/yyyy)</i><br>
      Years of Experience:
      <input size="4" name="yearsexperience"></p>
    <p>
      <input type="submit" value="Submit" name="submit"></p>
  </form>
</body>
</html>
```



Forms...

```
function allDigits(str)
{
    return invalidCharSet(str,"0123456789");
}

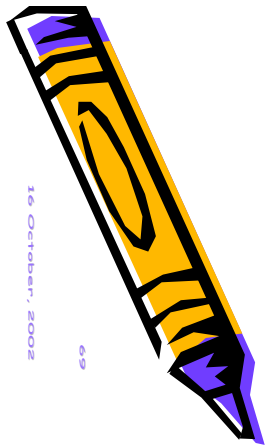
function invalidCharSet(str,charset)
{
    var result = true;
    for (var i=0;i<str.length;i++)
        if (charset.indexOf(str.substr(i,1))<0)
            { result = false; break; }
    return result;
}

function validNum(formField,fieldLabel,required)
{
    var result = true;
    if (required && !validRequired(formField,fieldLabel))
        result = false;
    if (result)
    {
        if (!allDigits(formField.value))
        {
            alert('Please enter a number for the "' +
                fieldLabel +'" field.');
```

```
        formField.focus();
        result = false;
        }
    }
    return result;
}

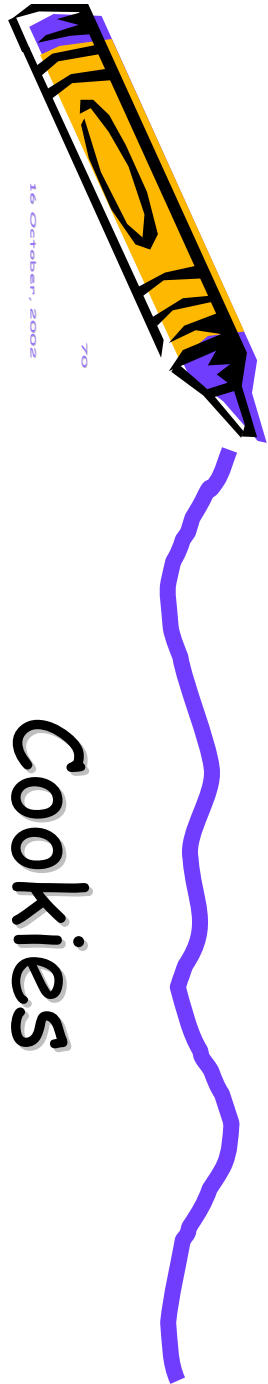
... (lots more)

function validateForm(theForm)
{
    if (!validRequired(theForm.fullname,"Name"))
        return false;
    if (!validEmail(theForm.email,"Email Address",true))
        return false;
    if (!validDate(theForm.available,"Date Available",true))
        return false;
    if (!validNum(theForm.yearsexperience,"Years Experience",true))
        return false;
    return true;
}
```



Forms...

- Issues
 - Duplication of code
 - Within HTML/JavaScript page
 - Across browser/server processing loci
 - Gives maintenance problems, etc.



- State management mechanism
 - Store & retrieve information on the client
 - When a browser contacts a Web server, the server may send "state information" (a cookie) in the HTTP header to be stored on the client
 - Can also store values submitted to the server by the user, such as user name, preferences, etc.
 - The cookie is asserted into the HTTP header to the server as a part of follow-up requests (for restoring state, security information, client preferences, etc.)
 - Cookie use is optional for the user
 - May disable cookies in the browser or block them in proxies, etc.



Cookies...

- Restrictions
 - A web browser can only hold up to 300 cookies with only 20 from any one web server and each cookie being not more than 4Kb in length
 - Prevents denial of service problems
- Accessed via the document.cookie property
 - Strange object: assigning to it simply *appends* a new value

```
document.cookie = "name=Fred";  
document.cookie="name=Joe;age=30;  
expires=Fri, 22-Aug-2002 00:00:00 GMT";
```



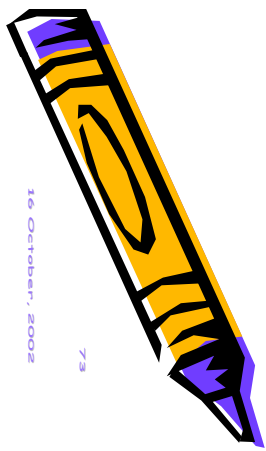
Cookies...

- Convenience functions

```
function getCookie(name) {
    var start = document.cookie.indexOf(name+"=");
    var len = start+name.length+1;
    if ((!start) && (name != document.cookie.substring(0,name.length)))
        return null;
    if (start == -1)
        return null;
    var end = document.cookie.indexOf(";",len);
    if (end == -1)
        end = document.cookie.length;
    return unescape(document.cookie.substring(len,end));
}

function setCookie(name,value,expires,path,domain,secure) {
    document.cookie = name + "=" +escape(value) +
        ( (expires) ? ";expires=" + expires.toGMTString() : "") +
        ( (path) ? ";path=" + path : "") +
        ( (domain) ? ";domain=" + domain : "") +
        ( (secure) ? ";secure" : "");
}

function deleteCookie(name, path, domain) {
    if (getCookie(name)) {
        document.cookie = name + "=" +
            ((path) ? "; path=" + path : "") +
            ((domain) ? "; domain=" + domain : "") +
            "; expires=Thu, 01-Jan-70 00:00:01 GMT";
    }
}
```

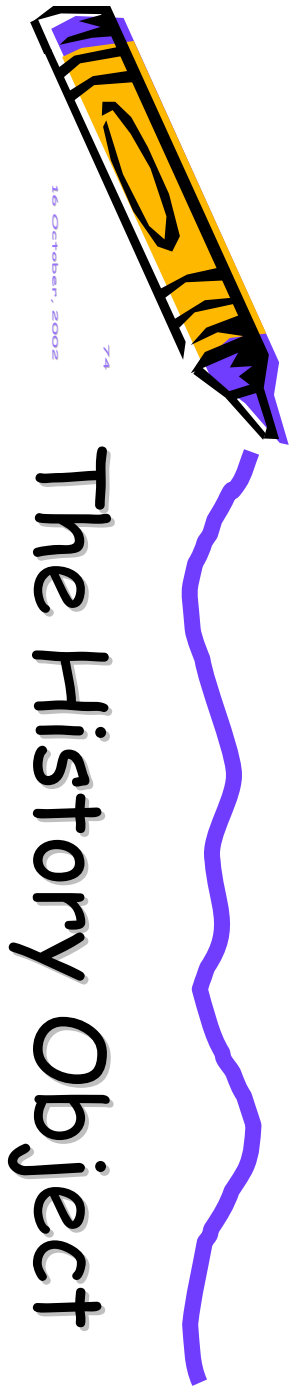
16 October, 2002

73

Cookies...

```
<SCRIPT LANGUAGE="JavaScript">
<!--
var now = new Date();
// cookie expires in one year (actually, 365 days)
// 365 days in a year
// 24 hours in a day
// 60 minutes in an hour
// 60 seconds in a minute
// 1000 milliseconds in a second
now.setTime(now.getTime() + 365 * 24 * 60 * 60 * 1000);
var visits = getCookie("counter");
// if the cookie wasn't found, this is your first visit
if (!visits) {
    visits = 1; // the value for the new cookie
    document.write("By the way, this is your first time here.");
} else {
    // increment the counter
    visits = parseInt(visits) + 1;
    document.write("By the way, you have been here " + visits + " times.");
}
// set the new cookie
setCookie("counter", visits, now);

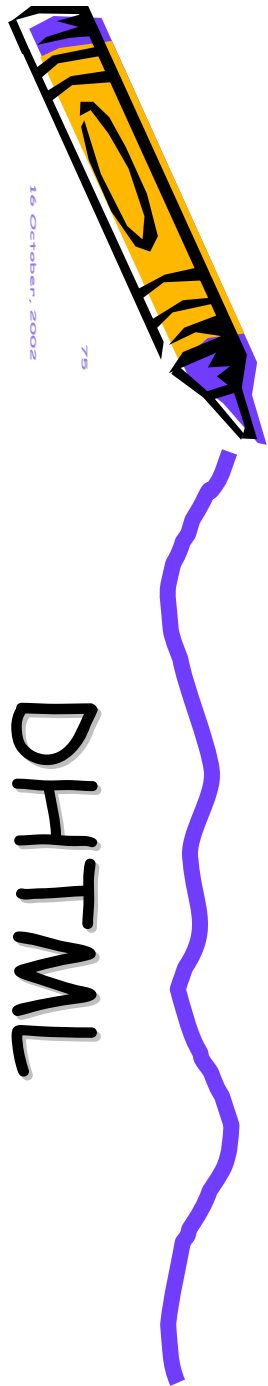
// -->
</SCRIPT>
```



- Deals with the browser's history list

```
<p>  
  <a href="javascript:window.history.go(-2)">click here</a>  
  to go back two pages.  
</p>
```

- IE and NS differ in advanced use
 - In IE the URLs cannot be extracted or otherwise modified by a script
 - In NS, access to `history.item()` is a protected operation
 - Relies on signed scripts and `UniversalBrowserRead` privilege



- Dynamic HTML

- For version 4+ (really ++) browsers

- Delivers four key innovations that allow Web authors to create truly dynamic pages:

- Access to all page elements

- Instant page update

- Full event model

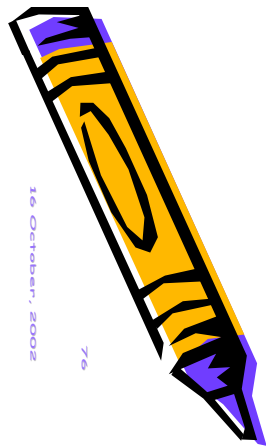
- Changing the text on the page

- Really a combination of technologies

"It is difficult to completely separate DOM, CSS and JavaScript when we are looking at Dynamic HTML. They are so interrelated that to do any meaningful work requires that we use a little of each. Significant amounts of dynamism can be accomplished with very simple techniques."

"Shockingly (we say mockingly) Netscape and Internet Explorer interpret DHTML commands individually and differently."

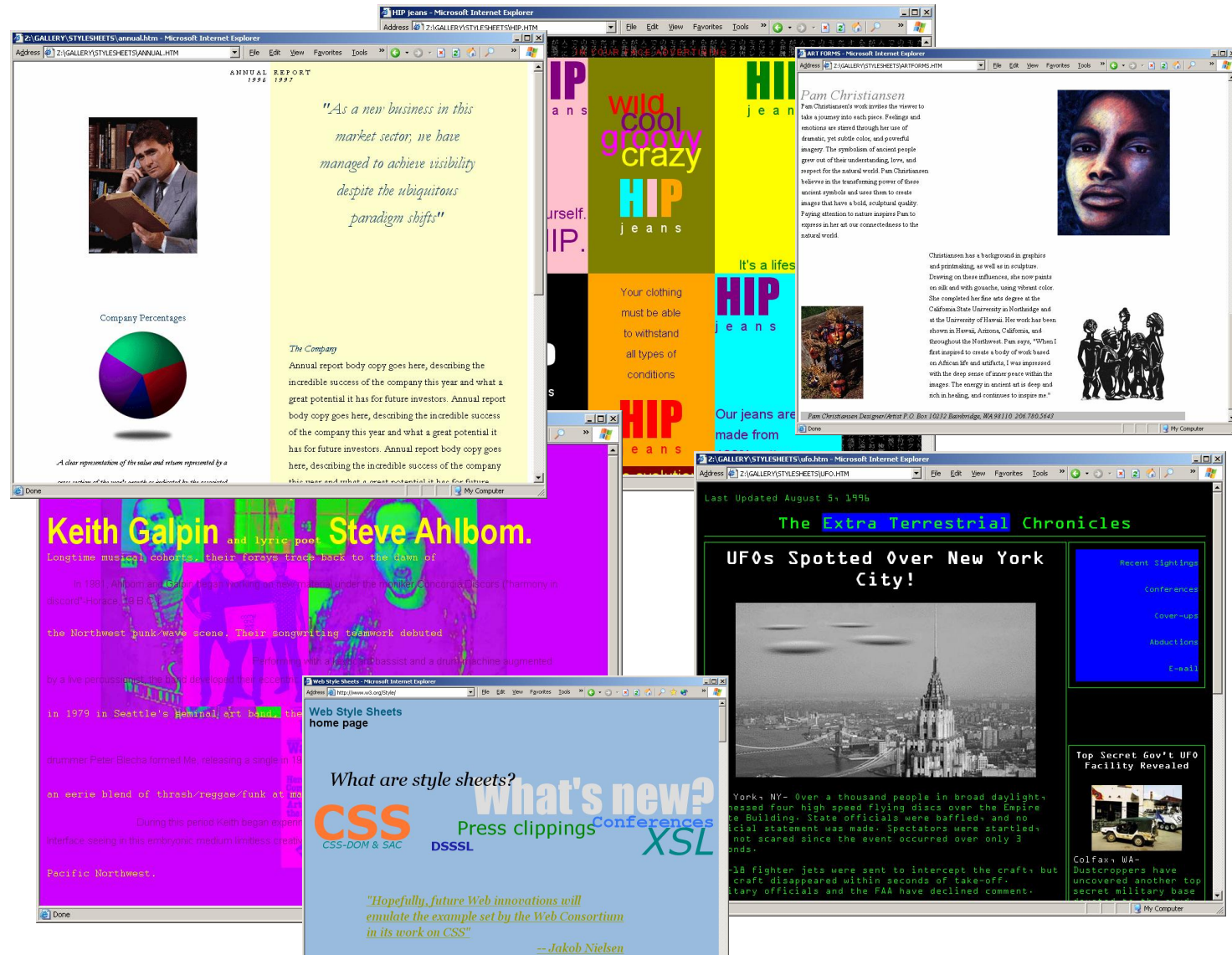
"Go light on DHTML, heavy on valuable content."

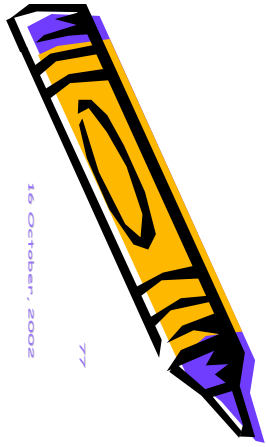


16 October, 2002

Interlude: CSS

- W3C ordained style mechanism for presentation of HTML



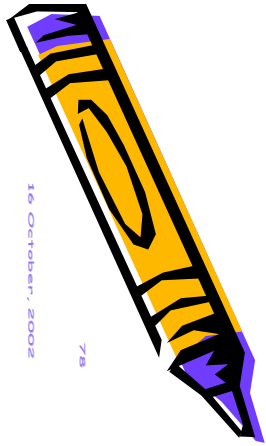


16 October, 2002

77

Interlude: CSS...

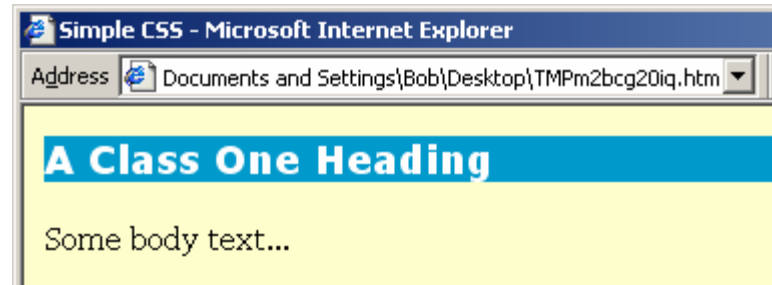
- Stylesheet technology for HTML
 - Simple, lookup-table technology
 - "...make *this* look like *this*..."
 - First steps to separation of presentation and processing markup
 - 3 levels of CSS
 - CSS1: simple visual styles
 - CSS2: media-specific formatting
 - Printing/aural/etc.
 - CSS3: (in progress) modularisation
 - So that implementors can catch up...
 - Well supported
 - Most browsers
 - Many development tools
 - Styles can be defined per-site, per-file or per-tag
 - Style definitions build additively ("cascade")
 - Also override
 - » Local definition wins

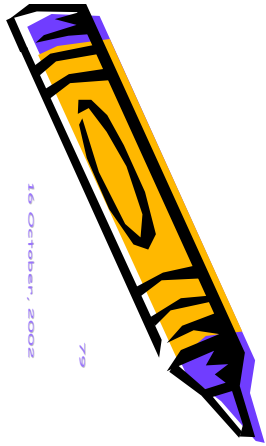


Interlude: CSS...

```
<html>
<head>
<title>Simple CSS</title>
<style type="text/css">
<!--
.h1 { font: bold 14pt Verdana, Arial, Helvetica, sans-serif;
      text-transform: capitalize; color: white;
      background: #0099CC}
body { font-family: Georgia, "Times New Roman", Times, serif;
      background: #FFFFCC}

-->
</style>
</head>
<body>
<p class="h1">a class one heading</p>
<p>Some body text...</p>
</body>
</html>
```



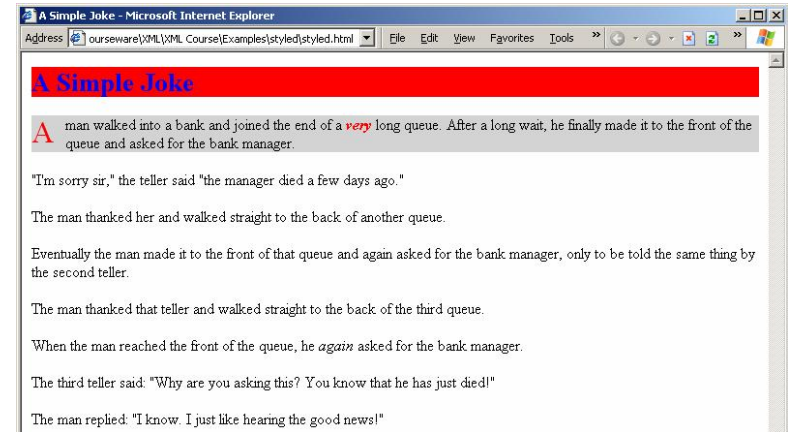


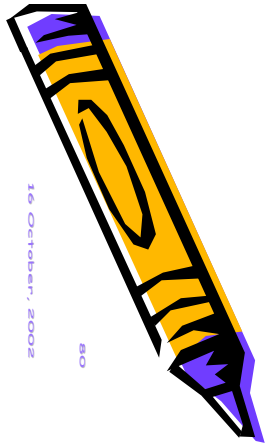
Interlude: CSS...

- Style elements can 'cascade'
 - Hence the name
 - Allows fine-grain/coarse control

```
<html>
<head>
  <LINK REL="STYLESHEET" TYPE="text/css"
        HREF="mystyles.css">
  <style>
    h1 { background: red }
    strong { font-style: italic }
    .dropcap { float: left; color: red;
               font-size: x-large;
               width: 1em; height: 1em; }
    .first { background: lightGrey }
  </style>
  <title>A Simple Joke</title>
</head>
<body>
  <h1 style="color: blue">A Simple Joke</h1>
  <p class="first"><span class="dropcap">A</span>
man walked into a bank and joined the end of a <strong>very</strong> long queue.
After a long wait, he finally made it to the front of the queue
and asked for the bank manager.</p>
<p>"I'm sorry sir," the teller said "the manager died a few days ago."</p>
<p>The man thanked her and walked straight to the back of another queue.</p>
<p>Eventually the man made it to the front of that queue and again asked
    for the bank manager, only to be told the same thing by the second teller.</p>
<p>The man thanked that teller and walked straight to the back of the third queue.</p>
<p>When the man reached the front of the queue, he <em>again</em> asked for
    the bank manager.</p>
<p>The third teller said: "Why are you asking this? You know that he has just died!"</p>
<p>The man replied: "I know. I
    just like hearing the good news!"</p>
</body>
</html>
```

```
<!-- file mystyles.css -->
h1 { font-family: Braggadocio; font-size: 20pt }
strong { color: red }
```



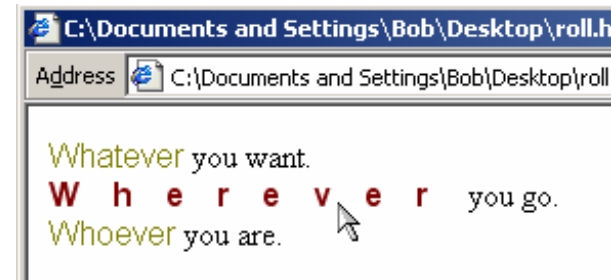


Interlude: CSS...

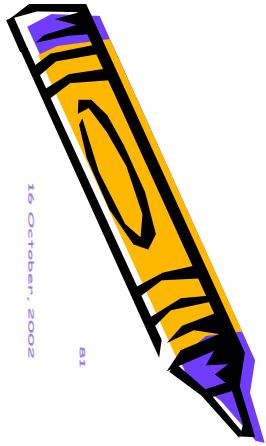
- Pseudo elements
 - Useful for 'packaging' features

```
<html>
<head>
<style>
.roll {
  font-family:Arial;
  font-size:12pt;
  text-decoration:none;
  color:olive;
}

A:hover {
  font-weight:bold;
  letter-spacing:2ex;
  color:maroon;
}
</style>
</head>
<body>
<A href="whatever.html" class="roll">
  Whatever </A>you want.<BR>
<A href="wherever.html" class="roll">
  Wherever </A>you go.<BR>
<A href="whoever.html" class="roll">
  Whoever </A>you are.<BR>
</body>
</html>
```



```
a:link {color: green;}
a:visited {color: olive;}
a:link:hover {background: yellow;}
a:visited:hover {background: cyan;}
```

Interlude: CSS...

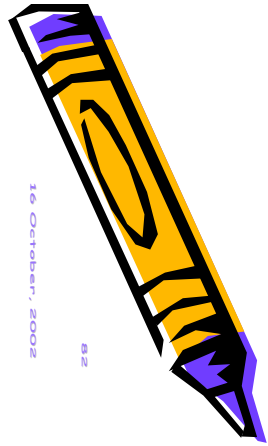
- Behavioural Extensions to CSS
 - W3C work in progress
 - Makes a lot of sense

```
<style>
.Rollover
{
  border : thin solid blue;
  onmouseover :
    "this.src = this.getAttribute('oversrc');
    this.style.borderColor = 'red';
    statusText.data = this.getAttribute('status');"
  onmouseout :
    "this.src = this.getAttribute('outsrc');
    this.style.borderColor = 'blue';
    statusText.data = '';"
}
</style>
```

...

```

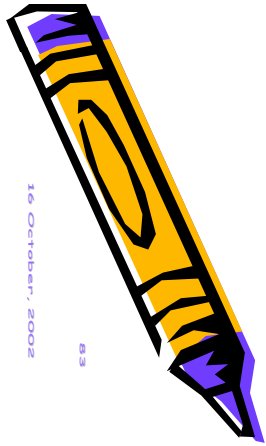

```



Interlude: DOM...

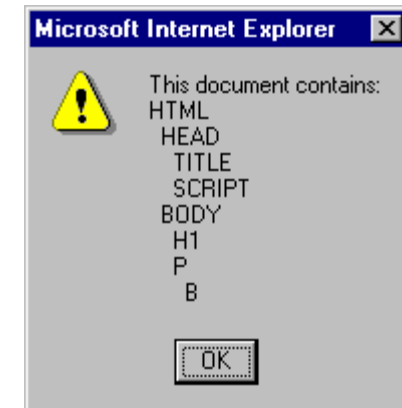
- Document Object Model
 - A description of how the objects that make up a page are connected together to form a whole

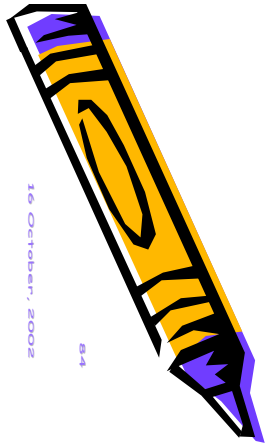
"Both Internet Explorer and Navigator have a Document Object Model, which provides access to Web page elements through a scripting language such as JavaScript. The problem is that Navigator and IE have different object models."
- A document is a tree
 - Can be walked and manipulated accordingly
- At the moment there are three models of the DOM:
 - the DOM according to Microsoft,
 - the DOM according to Netscape,
 - the DOM according to W3C (several levels 0,1 and 2 which includes events).
 - The Microsoft DOM is by far the most complete, almost everything in the DOM is accessible. The W3C DOM is more conservative, and the Netscape DOM trails in third place.



Interlude: DOM...

```
<HTML>
<HEAD><TITLE>Elements: Hierarchy</TITLE>
<SCRIPT LANGUAGE="JScript">
function showHierarchy() {
    var depth = 0;
    var msg = document.all(0).tagName;
    for (i=1; i<document.all.length; i++) {
        if (document.all(i-1).contains(document.all(i))==true) {
            depth = depth + 1;
        } else {
            var elParent = document.all(i-1).parentElement;
            for ( ; depth>0; depth--) {
                if (elParent.contains(document.all(i))==true)
                    break;
                elParent = elParent.parentElement;
            }
        }
        msg = msg + "\n";
        for (j=1; j<=depth; j++)
            msg = msg + "  ";
        msg = msg + document.all(i).tagName;
    }
    alert("This document contains:\n" + msg);
}
</SCRIPT>
</HEAD>
<BODY onload="showHierarchy()">
<H1>Welcome!</H1>
<P>This document is <B>very</B> short.
</BODY>
</HTML>
```



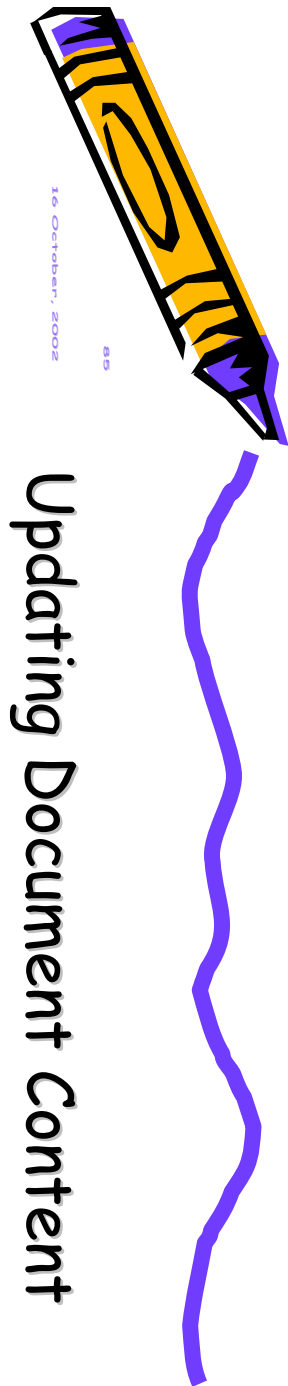


16 October, 2002

Interlude: W3C DOM

- Updating the DHTML idea
- The DOM (Document Object Model of IE5/NS6) promises to standardize the scripting interface across browsers/platforms
- Many changes ahead
 - http://developer.netscape.com/views/ource/goodman_cross/goodman_cross.htm

"I'm happy to report that as I stare into the dark tunnel of incompatibility, I see a pinhole of light that isn't another train racing toward me: the Document Object Model (DOM) specification from the World Wide Web Consortium (W3C)"

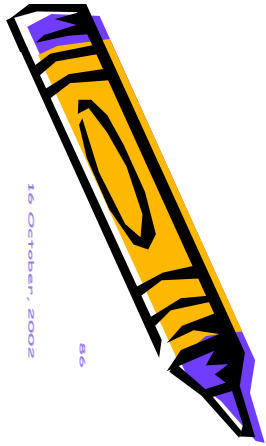


- One of the earliest ways of using JavaScript

```
<html>
<body>
Dynamically generated HTML using document.write()...<br>
<table border='1'>
  <tr>
    <th>Property</th>
    <th>Value</th>
  </tr>
  <script language='JavaScript'>
    for(myEnum in navigator)
    {
      document.write("<tr>");
      document.write("<td>");
      document.write(myEnum);
      document.write("</td>");
      document.write("<td>");
      document.write(navigator[myEnum]);
      document.write("</td>");
      document.write("</tr>");
    }
  </script>
</table>
</body>
</html>
```

Dynamically generated HTML using document.write()...

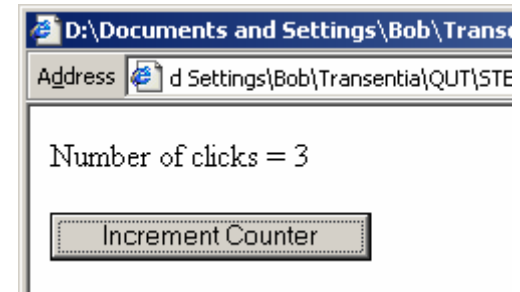
Property	Value
appCodeName	Mozilla
appName	Microsoft Internet Explorer
appMinorVersion	;Q316059;q319182;Q321232;
cpuClass	x86
platform	Win32
plugins	undefined
opsProfile	undefined
userProfile	undefined
systemLanguage	en-au
userLanguage	en-au
appVersion	4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET CLR 1.0.3705)
userAgent	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET CLR 1.0.3705)
onLine	true
cookieEnabled	true
mimeTypes	undefined



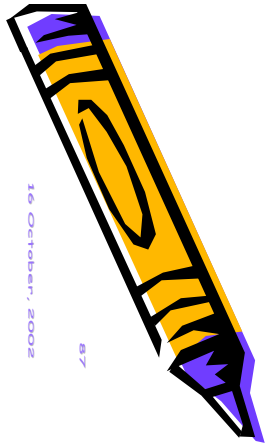
inner*

- innerHTML, innerText properties
 - innerText strips any contained tags, innerHTML preserves them
- Supported in modern browsers
 - Has been in IE for a while, introduced in NS6 by popular demand
 - Controversial

```
<html>
<body>
<div id="counter">Number of clicks = 0</div>
<form>
  <input type="button" value="Increment Counter" onClick="updateMessage()">
</form>
<script language="JavaScript">
  var hits = 0;
  function updateMessage()
  {
    document.getElementById("counter").innerHTML =
      "Number of clicks = " + ++hits;
  }
</script>
</body>
</html>
```



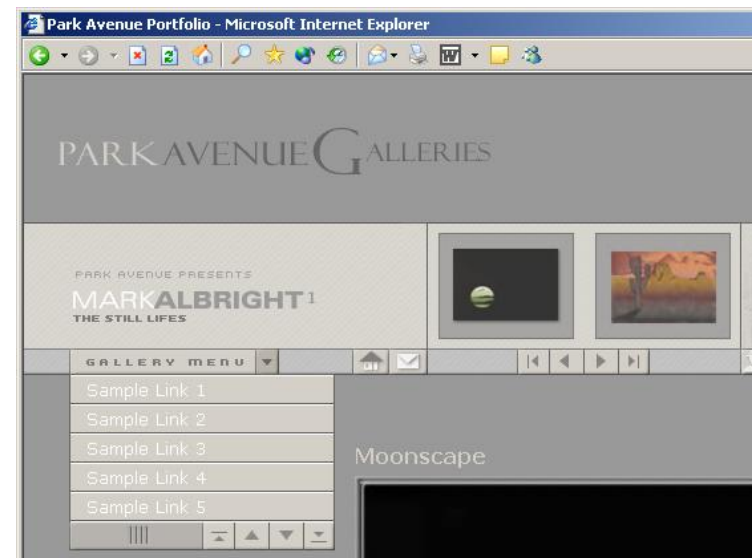
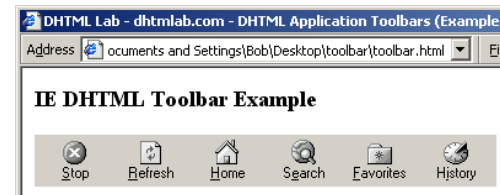
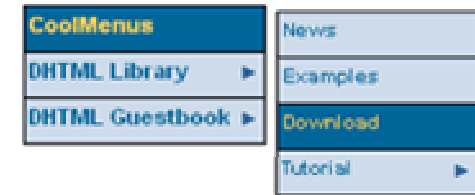
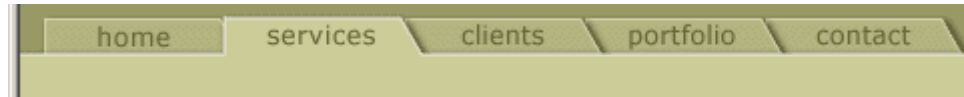
Widgets

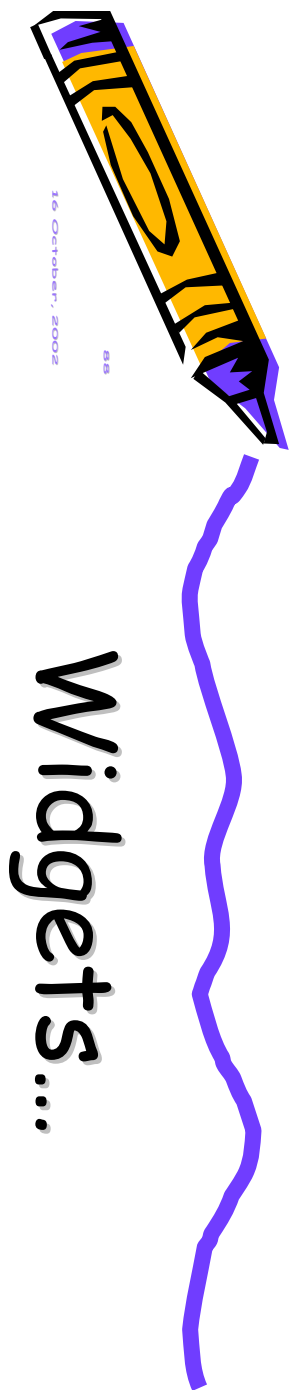


16 October, 2002

87

- Major DHTML use: provide rich GUI widget sets
 - Trees, menus, collapsible lists, etc.





- Example: making a toolbar like IE's

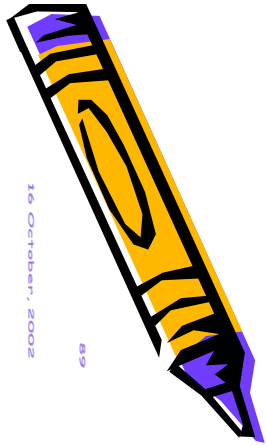
```
<html>
<head>
<title>DHTML Application Toolbars Example</title>
<style>
.but{border:1px buttonface solid;font-family:MS Sans Serif;font-size:8pt;}
</style>
<script language="JavaScript1.2" type="text/javascript">
allBUTTONs = toolbar.children;
maxWidth = 0;
for (i=0;i<allBUTTONs.length;i++) {
    tSpan = allBUTTONs(i);
    tSpan.onselectstart = function(){return false}
    tSpan.img = tSpan.children(0);
    tSpan.oversrc = tSpan.innerText + "0n.gif";
    tSpan.outsrc = tSpan.innerText + "0ff.gif";

    maxWidth = Math.max(maxWidth,tSpan.offsetWidth);

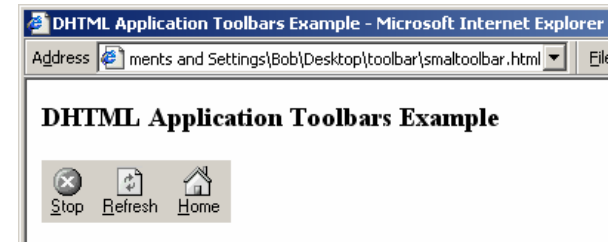
    tSpan.onmouseover = function(){
        this.style.border = "1px buttonhighlight outset";
        this.img.src = this.oversrc;
    }
    tSpan.onmouseout = function(){
        this.style.border = "1px buttonface solid";
        this.img.src = this.outsrc;
    }
    tSpan.onmousedown = function(){ this.style.border = "1px buttonhighlight inset"; }
    tSpan.onmouseup = function(){ this.style.border = "1px buttonhighlight outset"; }
}
for (i=0;i<allBUTTONs.length;i++) {
    tSpan = allBUTTONs(i);
    tSpan.style.pixelWidth = maxWidth;
}
</script>
</head>
...
```

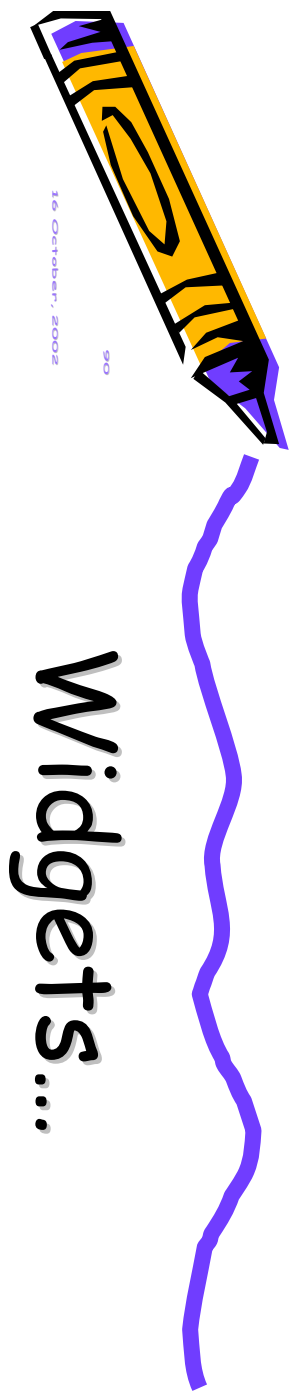
<http://www.webreference.com/dhtml/column29/>

Widgets...



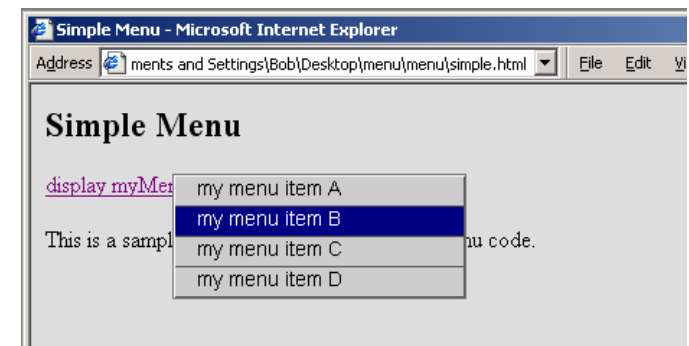
```
...
<body bgcolor="white" text="black">
<h3>DHTML Application Toolbars Example</h3>
<div id=toolbar NOWRAP style="background-color:buttonface;width:20;padding:2px">
  <button class=but accesskey="s"
    onClick="clickHandler('Stop');">
    <br><u>S</u>top
  </button>
  <button class=but accesskey="r"
    onClick="clickHandler('Refresh');">
    <br><u>R</u>efresh
  </button>
  <button class=but accesskey="h"
    onClick="clickHandler('Home');">
    <br><u>H</u>ome
  </button>
</div>
<!-- OMIT THE NEXT DIV. INCLUDED FOR DEMONSTRATION PURPOSES ONLY -->
<br>
<div id=disp NOWRAP style="font-family:Arial,sans-serif;font-size:10pt;padding-
left:20;width:20">&nbsp;&nbsp;&nbsp;</div>
<script language="JavaScript1.2" type="text/javascript">
function clickHandler(but){
  disp.innerHTML=but + " was clicked!";
  window.focus();
}
</script>
</body>
</html>
```

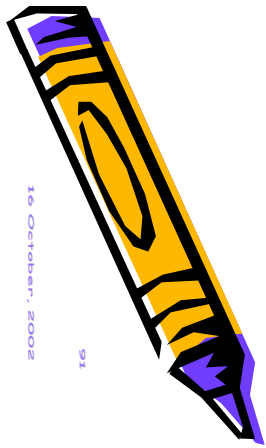




- Netscape's menu.js freeware

```
<html>
<head>
<title>Simple Menu</title>
<script language="JavaScript1.2" src="menu.js"></script>
<script language="JavaScript1.2">
<!--
function onLoad() {
    window.myMenu = new Menu();
    myMenu.addMenuItem("my menu item A");
    myMenu.addMenuItem("my menu item B");
    myMenu.addMenuItem("my menu item C");
    myMenu.addMenuItem("my menu item D");
    myMenu.writeMenus();
}
//-->
</script>
</head>
<body bgcolor="#dddddd">
<h2>Simple Menu</h2>
<a href="javascript:window.showMenu(window.myMenu);"
    onMouseOver="window.showMenu(window.myMenu);">display myMenu</a>
<p>
This is a sample page that contains simplified menu code.
<script language="JavaScript1.2">
<!--
//For IE
if (document.all) {
    onLoad();
}
//-->
</script>
</body>
</html>
```





16 October, 2002

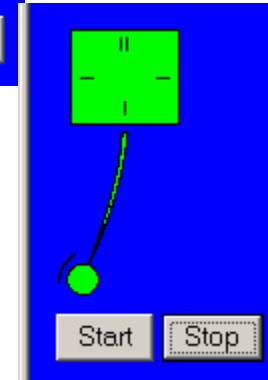
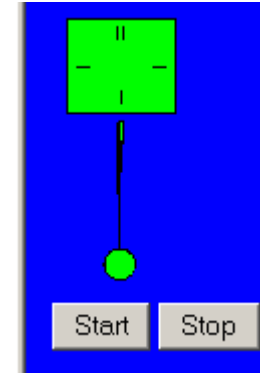
Animation

- Roll-over buttons, image displays, builds, etc.

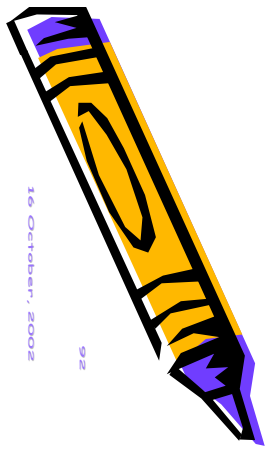
```
<html>
<body>
<table>
<tr>
<td>

<script language="javascript">
var clockImg = new Array(4)
clockImg[0] = new Image()
clockImg[0].src="clock_l.gif"
clockImg[1] = new Image()
clockImg[1].src="clock_c.gif"
clockImg[2] = new Image()
clockImg[2].src="clock_r.gif"
clockImg[3] = new Image()
clockImg[3].src="clock_c.gif"
var i=0
var timerID=null
function u()
{
i = (i + 1) % 4
document.t.src=clockImg[i].src
timerID = setTimeout("u()", 333)
}
</script>
</td>
</tr>
...
```

```
...
<tr>
<td>
<form>
<input type="button" value=" Start "
onClick="clearTimeout(timerID); u();">
<input type="button" value=" Stop "
onClick="clearTimeout(timerID);">
</form>
</td>
</tr>
</table>
</body>
</html>
```

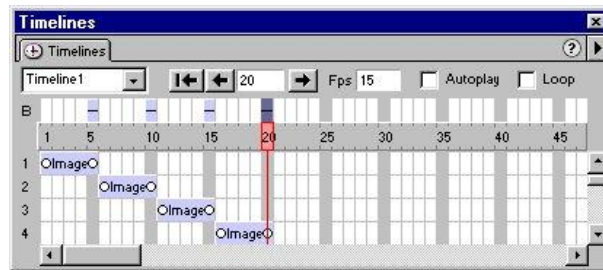


<http://schrenk.com/js/exp3.htm>

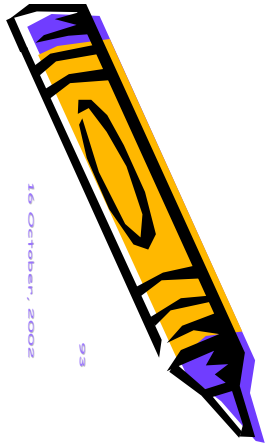


Animation...

- Dreamweaver provides a feature it calls 'Timelines'
 - Uses JavaScript to move images/layers around the page



"Allow you to add a sense of timing to your Web pages through the use of a series of complex JavaScripts. There are many different uses for timelines on Web pages. Timelines can be used to automate the movement of layers, control the timing of image changeovers and other objects, as well as control the start of a behavior. Timelines allow you to add a whole new dimension onto your Web page. You get to take a step past user interaction and provide pure eye candy to your page visitors."



Animation...

- DHTML Asteroids

- <http://msdn.microsoft.com/library/en-us/dndude/html/asteroids3.asp>

```
...
function tick() {
    doTick();
    window.setTimeout("tick();", tickTimeout, "JavaScript");
}

function doTick() {
    moveRocks();
}

function doKeysLater() {
    window.setTimeout("doKeys(" + window.event.key
}

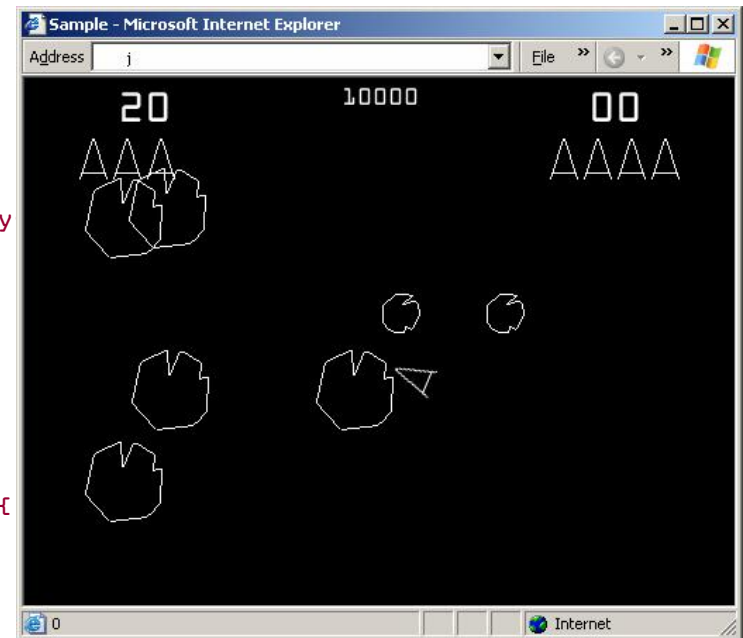
function moveRocks() {
    var i, r;

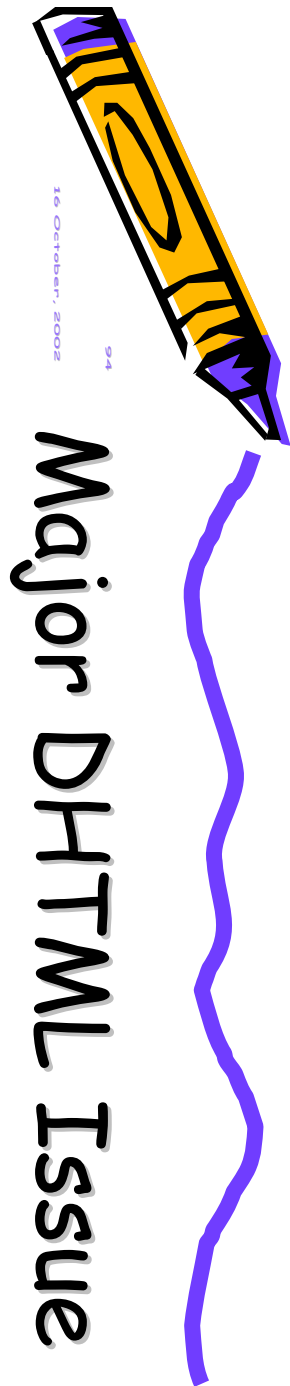
    roundCount++;

    checkShipDies();

    if ((Ship.dx != Ship.dy ) || (Ship.dx != 0)) {
        moveRock(Ship);
    }

    if (activeLasers > 0) {
        if (eLaser1.style.visibility == "visible") {
            if (!checkLaserHits(eLaser1)) {
                eLaser1.round++;
                if (eLaser1.round > maxLaserRange) {
                    eLaser1.style.visibility = "hidden";
                    activeLasers--;
                } else {
                    ...
```





- Compatibility

- Or, rather, lack of it...

"Compatibility issues are the heart of the matter when creating DHTML pages. Whether that means authoring pages that are compatible with older browsers or merely compatible with both Navigator 4.0 and later and Microsoft Internet Explorer 4.0 and later, DHTML coders face daunting challenges."

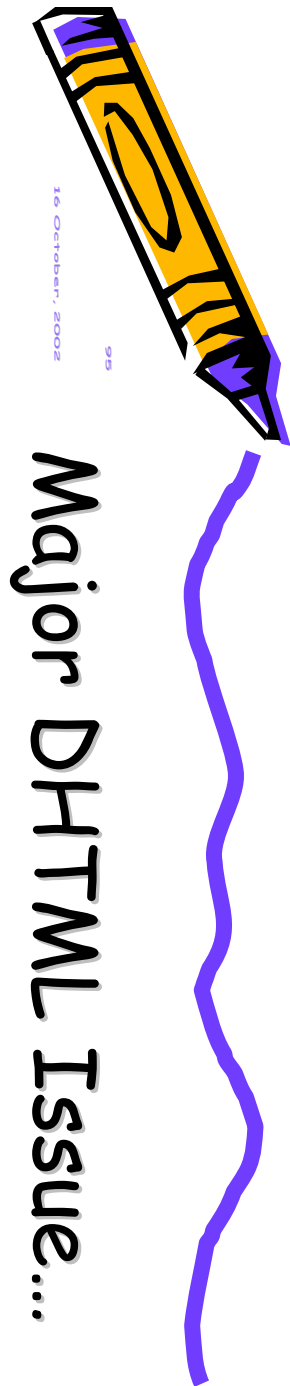
- Well reviewed at

- <http://builder.cnet.com/webbuilding/pages/Authoring/Dhtml>

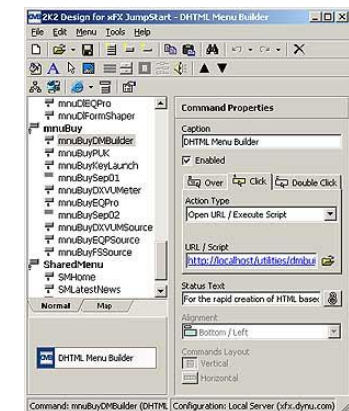
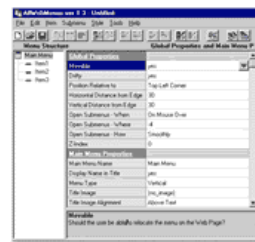
- Many tips, rules of thumb, etc.

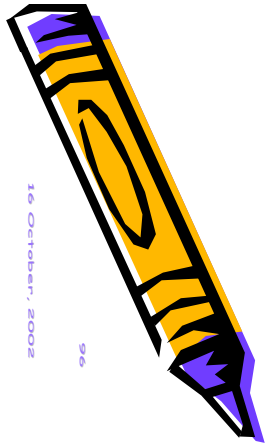
"Test until it hurts"

Visual DHTML: <http://developer.netscape.com/docs/examples/dynhtml/visual/index.html>



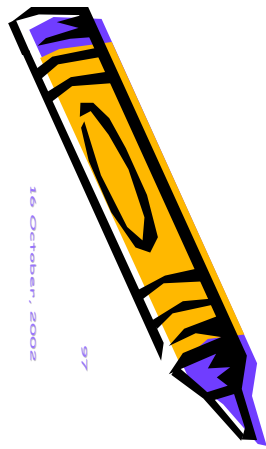
- Avoid all the problems, use tools
 - There are a myriad of menu makers, for instance
 - <http://www.xtreeme.com/sitexpert/>
 - <http://popup.jscentral.com/>
 - <http://software.xfx.net/utilities/dm-builder/>
 - <http://www.compusolvetechology.co.uk/Products/WebNavigator/About.asp>
 - <http://www.likno.com/>





Events

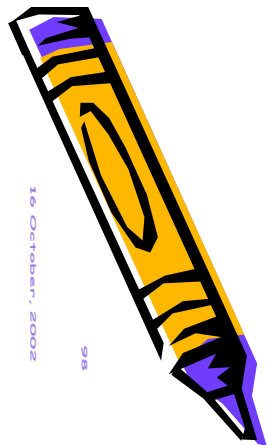
- Where DOM gives access to document content and CSS provides a mechanism for changing visual representation, Events allow for interactivity
- Events are actions that occur in the Browser, usually from something the user does
 - Button click/input text field change/mouse activity/etc.
 - Most things in IE generate events but only a few things in Netscape do
- Event *objects* are created by the browser when an event occurs
 - Object can be analysed
- *Events* trigger *Event Handlers*
 - Specifies which JavaScript code to execute for the event
 - The handler is actually called by the event object



Events...

- Browsers differ
 - IE: window.event globally available object
 - NS: Event object created and propagated
- Event object contains a number of fields that can be examined by a handler
 - Fields differ, of course!

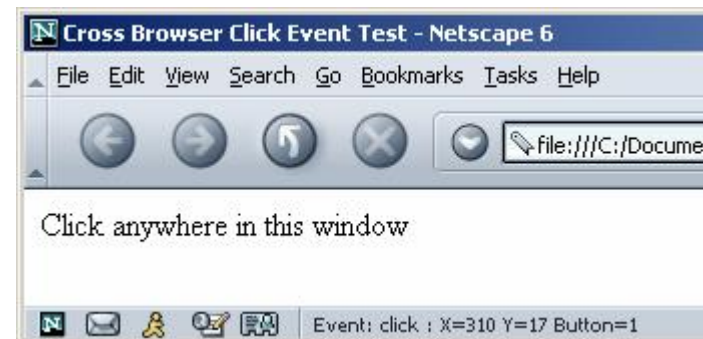
Navigator		Description	Internet Explorer	
Property	Value		Value	Property
modifiers	Event object properties	Specifies which modifier keys, if any, were pressed when the event occurred.	Boolean	altKey ctrlKey shiftKey
pageX	Integer	Specifies the horizontal coordinate of the mouse cursor at the time of event, with respect to the target object's page (the browser window).	Integer	clientX
pageY	Integer	Specifies the vertical coordinate of the mouse cursor at the time of event, with respect to the target object's page (the browser window).	Integer	clientY
screenX	Integer	Specifies the horizontal coordinate of the mouse cursor at the time of event, with respect to the client's screen.	Integer	screenX
screenY	Integer	Specifies the vertical coordinate of the mouse cursor at the time of event, with respect to the client's screen.	Integer	screenY
target	Object	A reference of the object that fired the event.	Object	srcElement
type	String	Specifies the event type (e.g., "click", "mousedown", "keypress").	String	type
which	Integer	Specifies the mouse button or keyboard key code. Note that some code values differ with the browser.	Integer	button keyCode



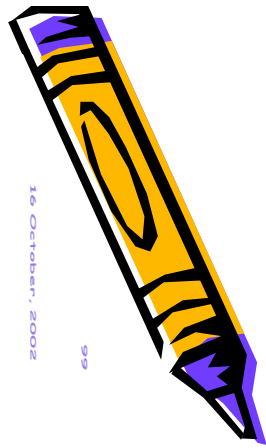
Events...

- Snooping an event

```
<html>
<head>
<title>Cross Browser Click Event Test</title>
<script language="JavaScript">
  <!--
    function statusClicks(e) {
      var theEvent = window.Event ? e : window.event;
      var theX = window.Event ? theEvent.pageX : theEvent.clientX;
      var theY = window.Event ? theEvent.pageY : theEvent.clientY;
      var theButton = window.Event ? theEvent.which : theEvent.button;
      window.status = "Event: " + theEvent.type +
        " : X=" + theX +
        " Y=" + theY +
        " Button=" + theButton;
    }
    document.onclick = statusClicks;
  //-->
</script>
</head>
<body>
<p>Click anywhere in this window</p>
</body>
</html>
```



Note the button number reported...

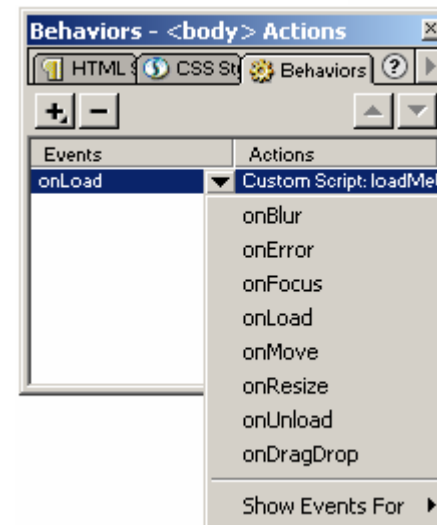
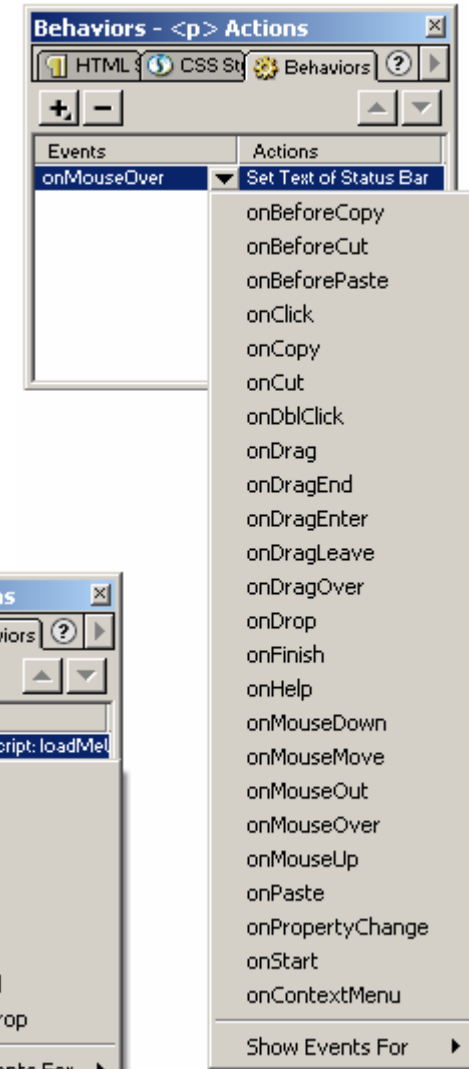


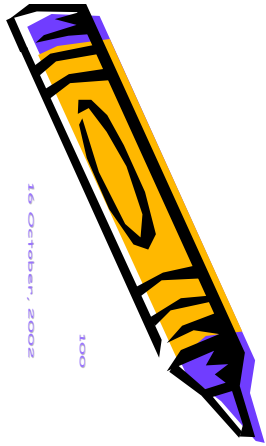
Event Handlers

- Typically placed within the HTML tag for the objects on which the event acts

```
<INPUT TYPE="Button" VALUE="Hello"  
onClick="alert('Hi there');">
```

- Many different things may cause an event
 - Evolves with browser/version

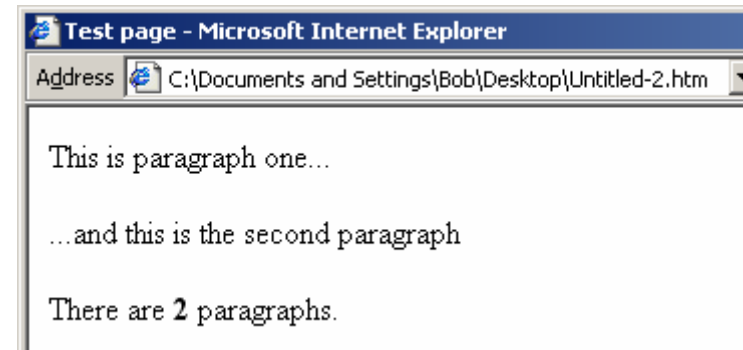


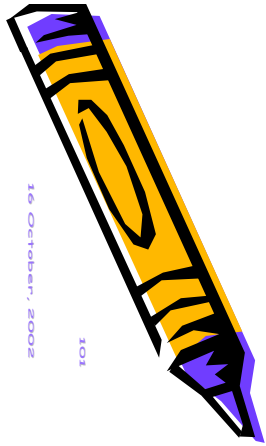


Event Handlers...

- Simple Example

```
<html>
<head>
<title>Test page</title>
<script language="JavaScript" type="text/javascript">
  function loadMeUp()
  {
    var nParas = document.getElementsByTagName("p").length;
    document.getElementById("counter").innerHTML =
      "There are <b>" + nParas + "</b> paragraphs.";
  }
</script>
</head>
<body onLoad="loadMeUp();">
<p>This is paragraph one...</p>
<p>...and this is the second paragraph</p>
<div id="counter"></div>
</body>
</html>
```



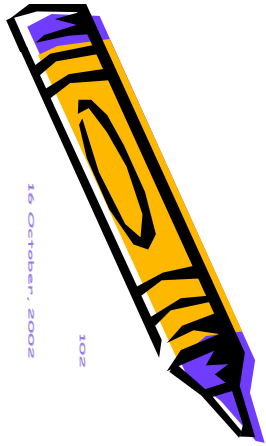


Events & 'this'

- The `this` reference adds much convenience

```
<SCRIPT LANGUAGE="JavaScript">
function process1() {
    document.forms[0].elements[0].value = "thank you";
}
</SCRIPT>
<FORM>
    <INPUT TYPE="text" NAME="myField"
        onChange="process1()">
</FORM>
```

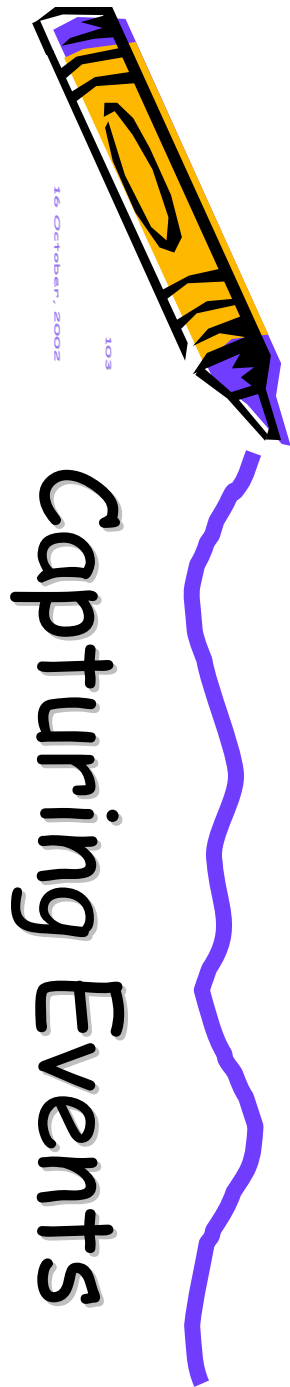
```
<SCRIPT LANGUAGE="JavaScript">
function process2(callingElement) {
    callingElement.value = "thank you";
}
</SCRIPT>
<FORM>
    <INPUT TYPE="text" NAME="myField"
        onChange="process2(this)">
</FORM>
```



Events & 'this' ...

```
<html>
<head>
<title>Form Test Page</title>
</head>
<body>
<form name="theForm">
Select a car:
<select name="cars"
    onChange="window.location=this[this.selectedIndex].value">
    <option selected value="jag.html">Jaguar</option>
    <option value="porche.html">Porche</option>
    <option value="rolls.html">Rolls-Royce</option>
</select>
</form>
</body>
</html>
```

```
<html>
<head>
<title>When is a link not a link?</title>
<style type="text/css">
.active { color: red; cursor: pointer; }
.inactive { color: gray; cursor: pointer; }
</style>
<script language="JavaScript" type="text/javascript">
function someFunction() {
    ...
}
</script>
</head>
<body>
<span class="inactive"
    onclick="someFunction();"
    onMouseOver="this.className='active';"
    onMouseOut="this.className='inactive';">
    This is not a link, but is clickable.
</span>
</body>
</html>
```



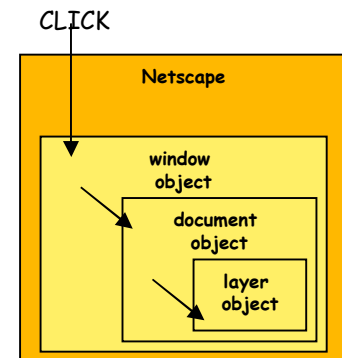
- Used by the older Netscape browsers (< NS6)
 - Events propagate from window to target
 - Usually document/window/layer
 - Any intermediate interposed object can capture the event if it wants/needs to
 - Flexible routing possible
 - Complex!

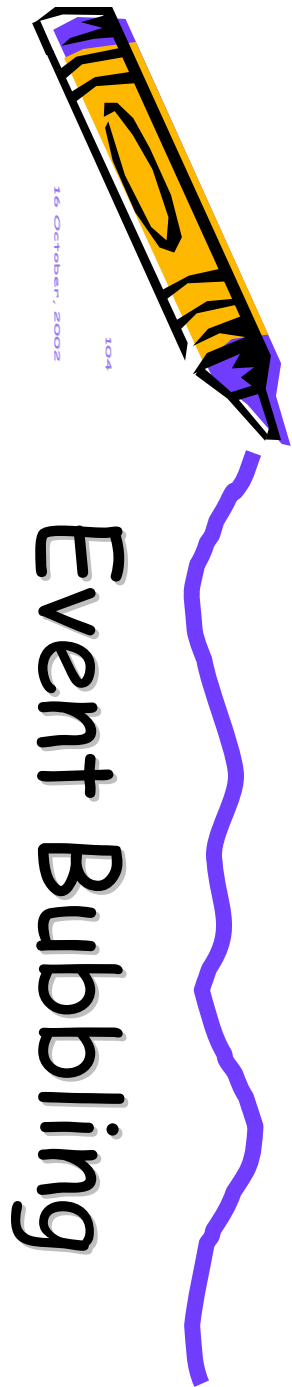
```
function myHandler(anEvent)
{
  ...
  if (someOtherObjectShouldHandleThisEvent)
    someOtherObject.handleEvent(anEvent);

  if (eventWasNotFullyHandledByMyHandler)
    document.routeEvent(anEvent);
}

document.mouseover = myHandler;
document.mouseout = myHandler;

document.captureEvents(Event.MOUSEOVER | Event.MOUSEOUT);
```

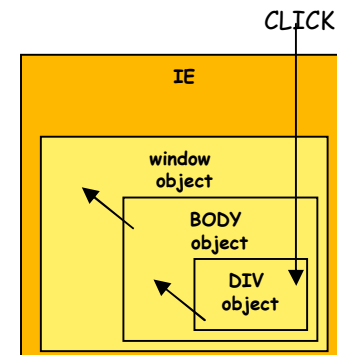


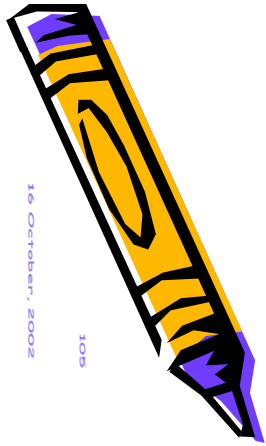


- IE's mechanism
 - Events may bubble 'up' the document's object containment hierarchy
 - Quite OO-style

```
function myHandler()  
{  
  ...  
  window.event.cancelBubble = true;  
  return false;  
}
```

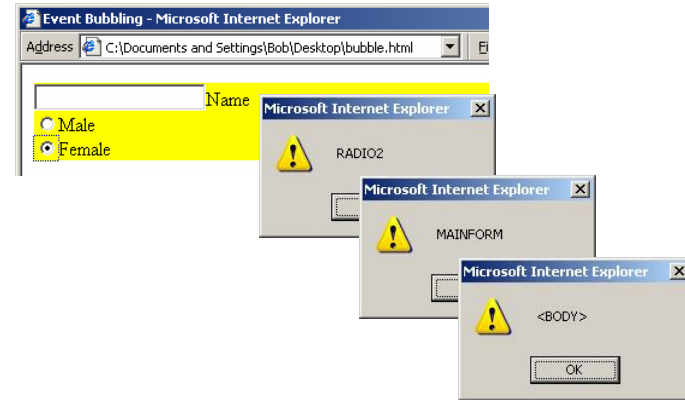
```
myDiv.onClick = myHandler;
```



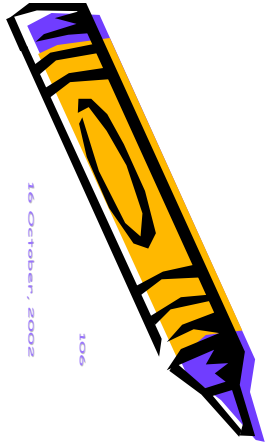


Event Bubbling...

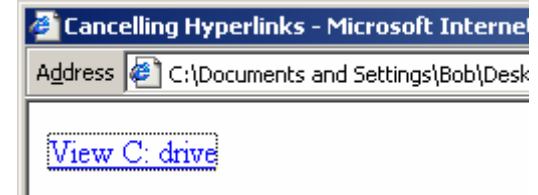
```
<HTML>
<HEAD>
<TITLE>Event Bubbling</TITLE>
</HEAD>
<BODY onclick="alert('<BODY>');">
<FORM NAME="MAINFORM" style="background-color:yellow"
      onclick="alert('MAINFORM');">
<INPUT TYPE="TEXT" NAME="TEXT1"
        onclick="alert('TEXT1');event.cancelBubble=true;">Name<BR>
<INPUT TYPE="RADIO" NAME="R1" VALUE="RADIO1"
        onclick="alert('RADIO1');">Male
<BR>
<INPUT TYPE="RADIO" NAME="R1" VALUE="RADIO2"
        onclick="alert('RADIO2');">Female
</FORM>
</BODY>
</HTML>
```



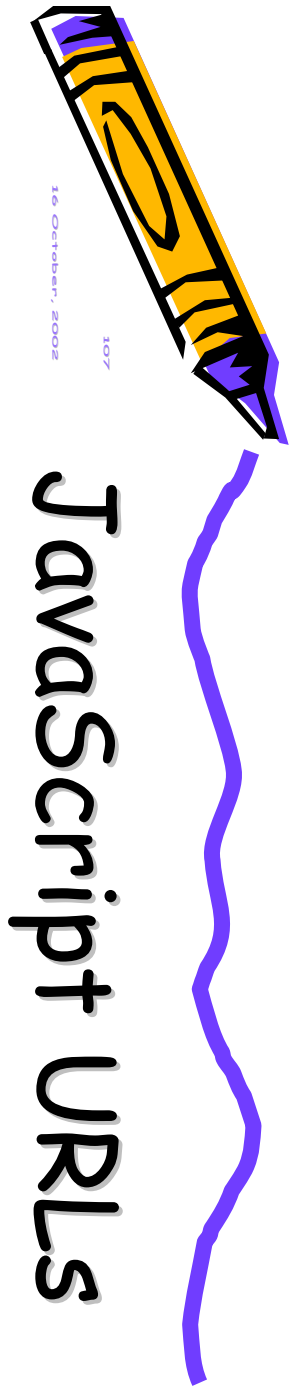
- Occasionally useful



Cancelling Events



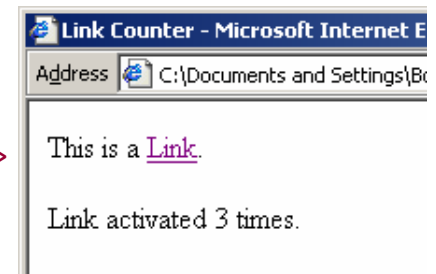
```
<HTML>
<HEAD>
<TITLE>Cancelling Hyperlinks</TITLE>
</HEAD>
<BODY>
<A HREF="c:\\" onclick="return false;">View C: drive</A>
</BODY>
</HTML>
```

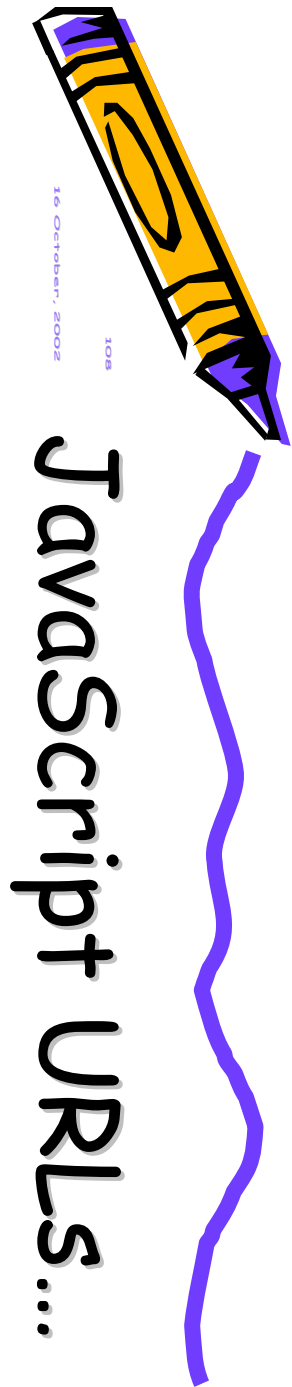


- Use URLs of type javascript: to execute JavaScript statements instead of loading a document
 - Can interface with browser's features
 - Can simplify/reduce event handling effort

```
<A HREF="javascript:history.go(0)">Reload Now</A>
```

```
<html>
<head>
<title>Link Counter</title>
<script language="JavaScript" type="text/javascript">
var numAsDone = 0;
function doA() {
    document.getElementById("message").innerHTML =
        "Link activated " + (++ numAsDone) + " times.";
}
</script>
</head>
<body bgcolor="#FFFFFF" text="#000000">
<p>This is a <A HREF="javascript:doA()">Link</A>.</p>
<div id="message"></div>
</body>
</html>
```





- Javascript:void(0)

- Very common
- The void operator discards the value of its operand and returns an undefined value
- Consider:

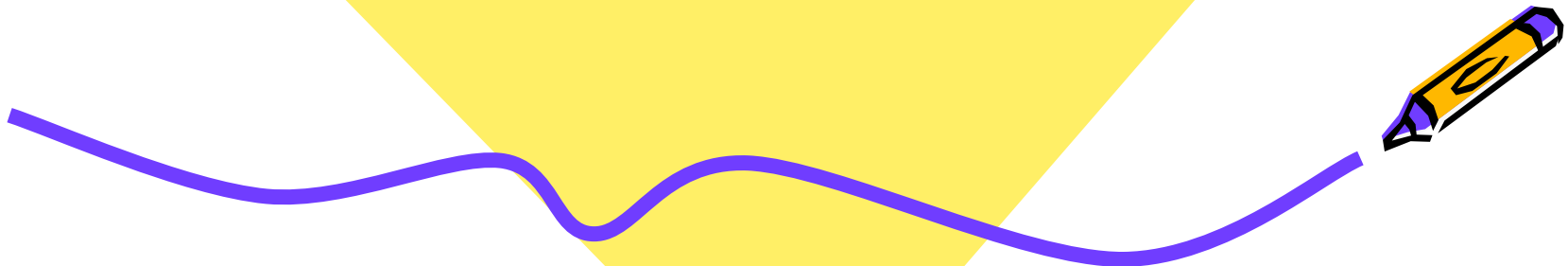
```
<a href="javascript:
  window.open('about:blank','win','width=100,height=100')">
  Click Me
</a>
```

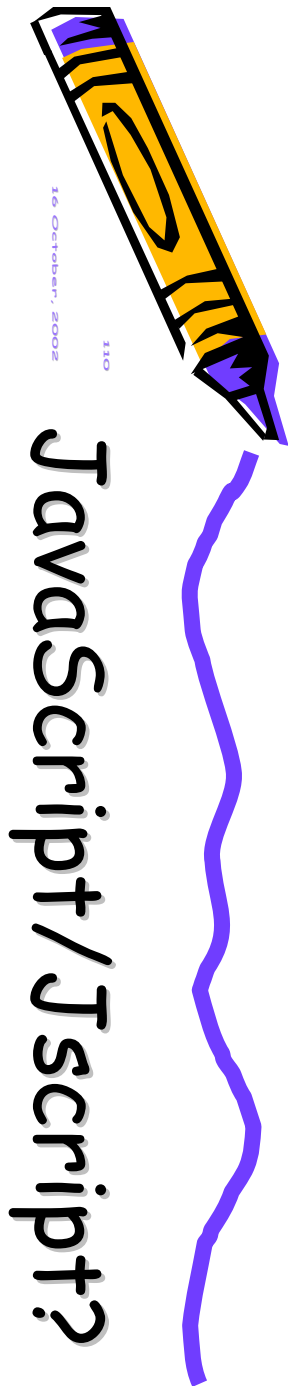
- the returned window reference will cause the browser to display the returned result (normally '[object Window]') in the current window, thus overwriting the original contents. Using void stops this and allows the browser to retain the original window contents:

```
<a href="javascript:
  void(window.open('about:blank','win','width=100,height=100'))">
  Click Me
</a>
```



JavaScript Goodies in IE





- From MSDN:

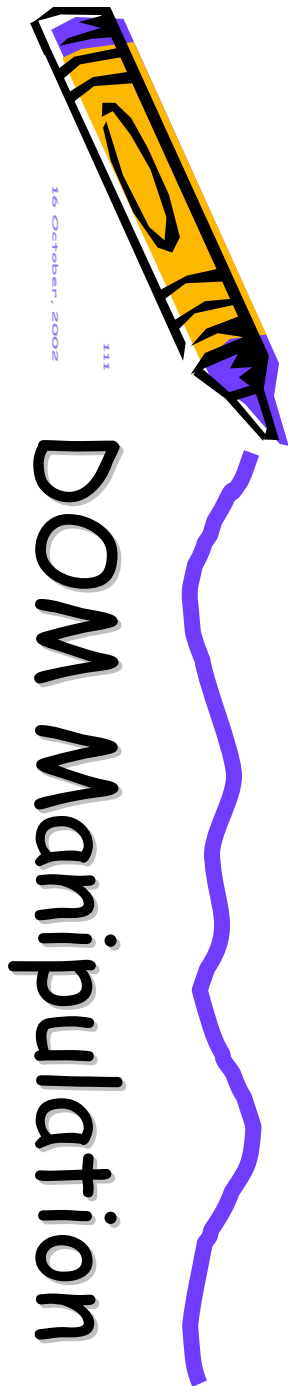
JScript

JScript is Microsoft's implementation of an ECMA-compliant scripting language (like JavaScript) that is targeted specifically to the Internet. Like VBScript, JScript is implemented as a fast, portable interpreter for use in Web browsers and applications that use ActiveX controls, Java applets, and OLE Automation servers. JScript is not Java and has nothing to do with Java. It is closer in syntax to C or C++. If you are a C or C++ developer, you will probably find JScript to be a very easy scripting language to learn (I know I did).

JScript? JavaScript? ECMAScript? What's the Deal Here?

You may be confused by the term ECMAScript You're in good company. Here's the deal. ECMA (European Computer Manufacturers Association) is a European-based association for standardizing information and communications systems. The standard recently approved, known as ECMA-262, is based on joint submissions from Microsoft and Netscape. JScript 3.0 is Microsoft's implementation of the new ECMA-262 scripting language. JavaScript is a scripting language written by Netscape that preceded the ECMA standard. Basically, when talking about JScript or JavaScript, we are talking about implementations of the same standard scripting language, ECMA—the implementations are just marketed by different companies.

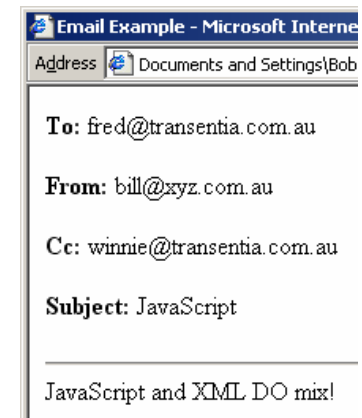
<http://msdn.microsoft.com/library/en-us/dnscript/html/allabout.asp>

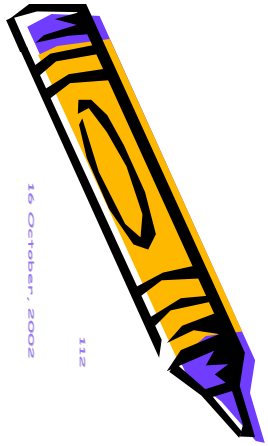


- Using JavaScript to work with data in an external XML file

```
<html>
<head>
<script language="JavaScript"
    for=window event=onload>
    loadDoc();
</script>
<script language="JavaScript">
    var xmlDoc = new ActiveXObject("microsoft.xmlDOM");
    xmlDoc.load("email.xml");
    function loadDoc() {
        if (xmlDoc.readyState == "4")
            start();
        else
            window.setTimeout("loadDoc()", 4000);
    }
    function start() {
        todata.innerText =
            xmlDoc.selectSingleNode("EMAIL/T0").text;
        fromdata.innerText =
            xmlDoc.selectSingleNode("EMAIL/FROM").text;
        ccdata.innerText =
            xmlDoc.selectSingleNode("EMAIL/CC").text;
        subjectdata.innerText =
            xmlDoc.selectSingleNode("EMAIL/SUBJECT").text;
        bodydata.innerText =
            xmlDoc.selectSingleNode("EMAIL/BODY").text;
    }
</script>
...
```

```
<?xml version="1.0"?>
<EMAIL>
  <T0>fred@transentia.com.au</T0>
  <FROM>bill@xyz.com.au</FROM>
  <CC>winnie@transentia.com.au</CC>
  <SUBJECT>JavaScript</SUBJECT>
  <BODY>JavaScript and XML DO mix!</BODY>
</EMAIL>
```



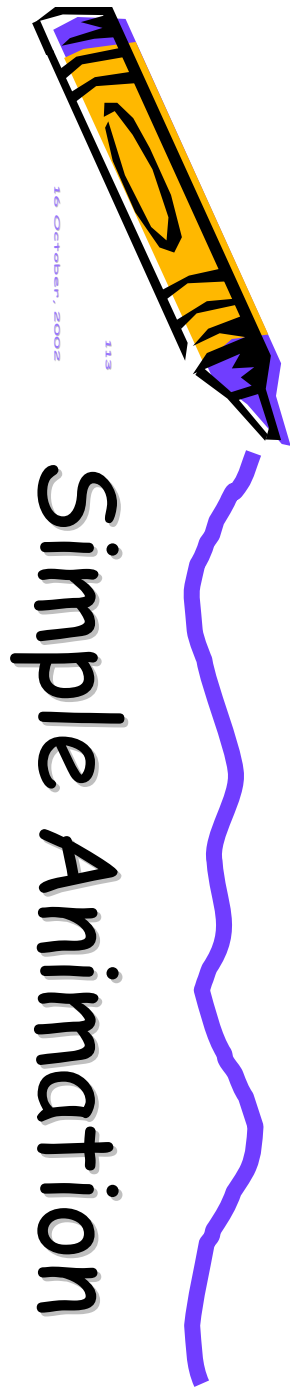


16 October, 2002

312

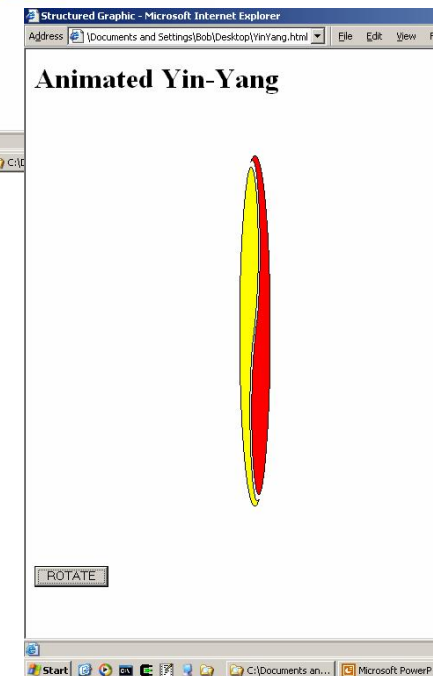
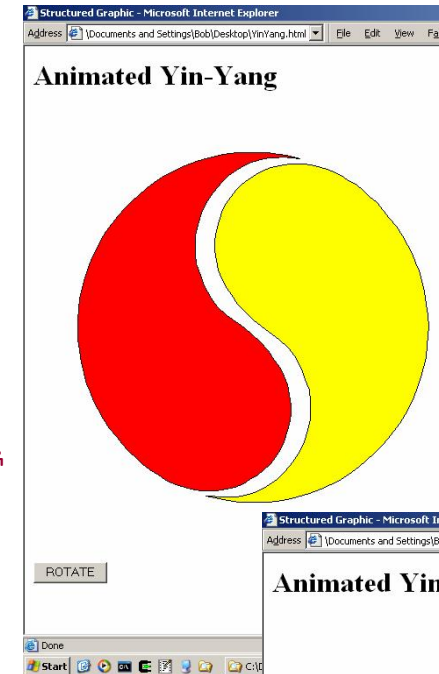
DOM Manipulation...

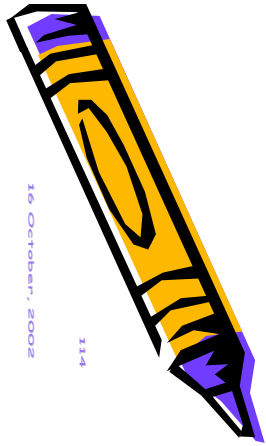
```
...  
<title>Email Example</title>  
</head>  
<body>  
<div id="to" style="font-weight:bold;font-size:16"> To: <span  
id="todata" style="font-weight:normal"></span>  
</div>  
<br>  
<div id="from" style="font-weight:bold;font-size:16"> From: <span  
id="fromdata" style="font-weight:normal"></span>  
</div>  
<br>  
<div id="cc" style="font-weight:bold;font-size:16"> Cc: <span  
id="ccdata" style="font-weight:normal"></span>  
</div>  
<br>  
<div id="from" style="font-weight:bold;font-size:16"> Subject: <span  
id="subjectdata" style="font-weight:normal"></span>  
</div>  
<br>  
<hr>  
<span id="bodydata" style="font-weight:normal"></span>  
<p>  
</body>  
</html>
```

- JavaScript and VML can interact nicely

```
<html>
<head>
<title>Structured Graphic</title>
<SCRIPT LANGUAGE="JavaScript">
<!--
function Rotate(degVar){
    star.rotate(0,degVar,0);
    window.setTimeout("Rotate(5)","040","JavaScript");
}
-->
</script>
</head>
<body>
<H1>Animated Yin-Yang</H1>
<p>
<object
ID="star"
CLASSID="CLSID:369303C2-D7AC-11D0-89D5-00A0C90833E6"
STYLE="height: 500; width:500; zindex:10">
<PARAM NAME="Line0001" VALUE="SetLineColor(0,0,0)">
<PARAM NAME="Line0002" VALUE="SetFillColor(255,0,0)">
<PARAM NAME="Line0003" VALUE="SetLineStyle(6)">
<PARAM NAME="Line0004" VALUE="Polygon(102,8,...,24,38,17,27)">
<PARAM NAME="Line0005" VALUE="SetFillColor(255,255,0)">
<PARAM NAME="Line0006" VALUE="Polygon(102,-9,...,-18,-26)">
</object>
<p>
<INPUT TYPE=button ID=rotate VALUE="ROTATE" onclick="Rotate(0)">
</body>
</html>
```

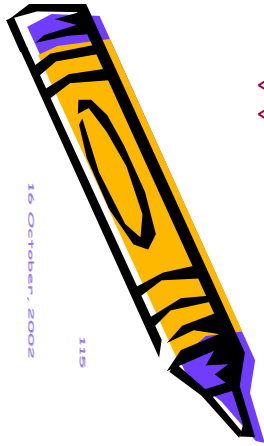




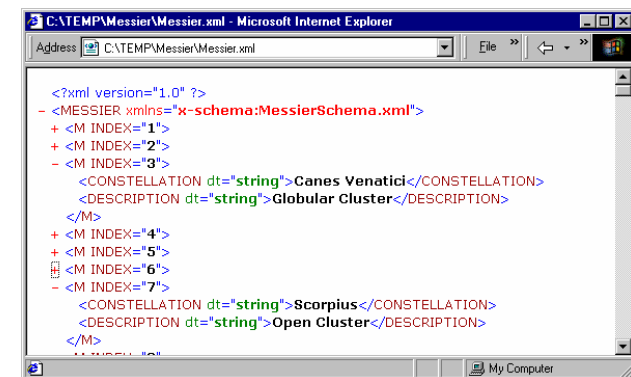
IE's DSO

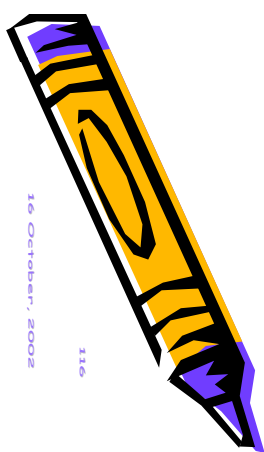
- Data Source Objects
 - Bind XML data to a control on a page
 - Much more streamlined than walking a DOM tree or applying XSL/CSS to a document
 - Can work with data en-masse or with individual nodes
 - Can also work with an XML data island
 - Can also bind tables, remote datasources, JDBC datasources, etc.

IT'S DSO...



```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<HTML>
  <HEAD>
    <TITLE>The Messier Objects</TITLE>
    <SCRIPT FOR="window" EVENT="onload">
      var xmlDso = xmlDso.XMLDocument;
      xmlDso.load("Messier.xml");
      countMess.innerHTML = " " +
        xmlDso.documentElement.childNodes.length + " ";
    </SCRIPT>
  </HEAD>
  <BODY BACKGROUND="M81.gif">
    <OBJECT WIDTH="0" HEIGHT="0" ID="xmlDso"
      CLASSID="clsid:550dda30-0541-11d2-9ca9-00b0b0ec3d39">
    </OBJECT>
    <H1 STYLE="color:blue">The <SPAN ID="countMess"></SPAN> Messier Objects</H1>
    <TABLE WIDTH="100%">
      <TR>
        <TD>
          <TABLE BORDER="1" WIDTH=75% DATASRC=#xmlDso>
            <TR>
              <TD BGCOLOR="lightYellow" STYLE="font-size:18; font-weight:bold">
                M<SPAN DATAFLD="INDEX"></SPAN>
              </TD>
            </TR>
            <TR>
              <TD>
                <TABLE BORDER="1" WIDTH=100%>
                  <THEAD ALIGN="left" BGCOLOR="silver">
                    <TH>Constellation</TH>
                    <TH>Description</TH>
                  </THEAD>
                  <TR ALIGN="left" BGCOLOR="lightBlue">
                    <TD><DIV DATAFLD="CONSTELLATION"></DIV>
                    <TD><DIV DATAFLD="DESCRIPTION"></TD>
                  </TR>
                </TABLE>
              </TD>
            </TR>
          </TABLE>
        </TD>
      </TR>
    </TABLE>
  </BODY>
</HTML>
```





IE'S DSO...

```
<html>
<head>
<title>Dynamically Binding a table</title>
<script for=cboField event=onchange>
    cDSO = tblComposers.dataSrc; // remember the DSO
    tblComposers.dataSrc = ""; // unbind the table
    spanField.dataFld = this.options(this.selectedIndex).value;
    tblComposers.dataSrc = cDSO; // rebind the table
</script>
</head>
<body>
<object classid="clsid:333C7BC4-460F-11D0-BC04-0080C7055A83"
    id="tdcComposers" height=0 width=0>
    <param name="DataURL" value="composer.csv">
    <param name="UseHeader" value="True">
    <param name="TextQualifier" value="' '>
</object>
<h2>Dynamically Binding a table</h2>
<table border="1" id="tblComposers" datasrc="#tdcComposers">
    <thead>
    <tr style="font-weight:bold">
        <td>Last Name</td>
        <td>
            <select id="cboField">
                <option value="compsr_first" SELECTED>First Name</option>
                <option value="compsr_last">Last Name</option>
                <option value="compsr_birth">Date of Birth</option>
                <option value="compsr_death">Date of Death</option>
                <option value="origin">Origin</option>
            </select>
        </td>
    </tr>
    </thead>
    <tr>
        <td><span datafld="compsr_Last"></span></td>
        <td><span id="spanField" datafld="compsr_first"></span></td>
    </tr>
</table>
</body>
</html>
```

Dynamically Binding a table - Microsoft Internet Explorer

Address C:\Documents and Settings\Bob\Desktop\dbdyntab

Dynamically Binding a table

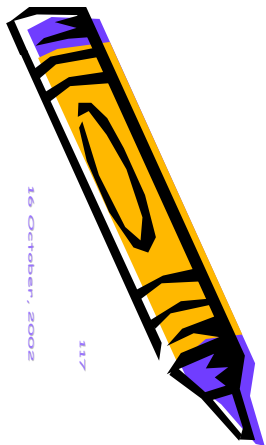
Last Name	Origin
Berlioz	France
Moussorgsky	Russia
Liszt	France
Vivaldi	Italy
Bach	Germany
Beethoven	Germany
Mozart	Austria
Haydn	Germany
Debussy	France

Dynamically Binding a table - Microsoft Internet Explorer

Address C:\Documents and Settings\Bob\Desktop\dbdyntab

Dynamically Binding a table

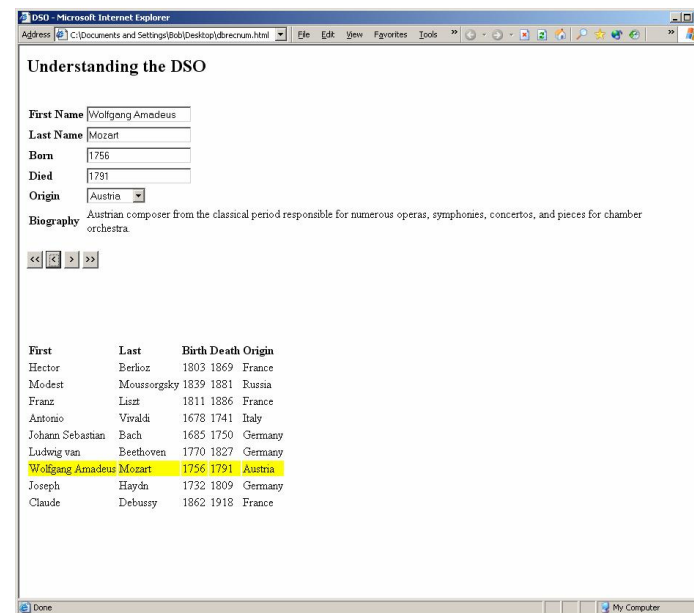
Last Name	First Name
Berlioz	Hector
Moussorgsky	Modest
Liszt	Franz
Vivaldi	Antonio
Bach	Johann Sebastian
Beethoven	Ludwig van
Mozart	Wolfgang Amadeus
Haydn	Joseph
Debussy	Claude



IE's DSO...

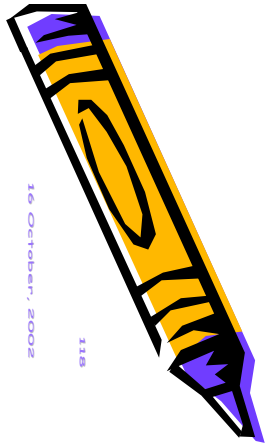
- This example shows JavaScript controlling a cursor that walks through the associated data

```
<html>
<head><title>DSO</title>
<style>
    .ColumnLabel {font-weight:bold}
</style>
<script language="javascript">
function HandleRowEnter() {
    tblComposers.rows[tdcComposers.recordset.AbsolutePosition].style.backgroundColor = 'yellow';
}
</script>
<script for=tblComposers event=onreadystatechange>
    if (this.readyState == 'complete') {
        this.rows(tdcComposers.recordset.
            AbsolutePosition).style.
            backgroundColor = 'yellow';
        tdcComposers.onrowenter = HandleRowEnter;
    }
</script>
<script for=tdcComposers event=ondatasetcomplete>
    scrNavBtns.DSO = this;
</script>
<script for=tdcComposers event=onrowexit>
    tblComposers.rows[tdcComposers.recordset.
        AbsolutePosition].style.
        backgroundColor = '';
</script>
<script for=trTemplate event=onclick>
    tdcComposers.recordset.
        AbsolutePosition = this.recordNumber;
    window.event.cancelBubble = true;
</script>
</head>
```



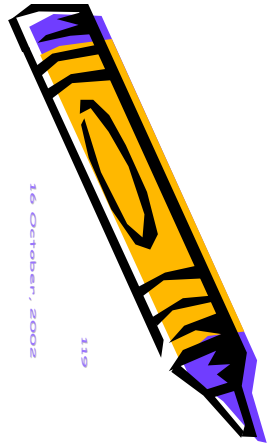
<http://msdn.microsoft.com/workshop/samples/author/databind/dbrecnum.htm>

IE'S DSO...



```
...
<body>
<h2>Understanding the DSO</h2>
<object classid="clsid:333C7BC4-460F-11D0-BC04-0080C7055A83"
  id=tdcComposers height=0 width=0>
  <param name="DataURL" value="composer.csv">
  <param name="UseHeader" value="True">
  <param name="TextQualifier" value="' '>
</object>
<p>
<table>
<tr><td class=ColumnLabel>First Name</td>
  <td><input id=txtFirst type=TEXT datasrc=#tdcComposers datafld=compsr_first></td></tr>
<tr><td class=ColumnLabel>Last Name</td>
  <td><input type=TEXT datasrc=#tdcComposers datafld=compsr_last></td></tr>
<tr><td class=ColumnLabel>Born</td>
  <td><input type=TEXT datasrc=#tdcComposers datafld=compsr_birth></td></tr>
<tr><td class=ColumnLabel>Died</td>
  <td><input type=TEXT datasrc=#tdcComposers datafld=compsr_death></td></tr>
<tr><td class=ColumnLabel>Origin</td>
  <td><select datasrc=#tdcComposers datafld=origin>
    <option value=Austria>Austria
    <option value=France>France
    <option value=Germany>Germany
    <option value=Italy>Italy
    <option value=Russia>Russia
  </select>
  </td></tr>
<tr><td class=ColumnLabel>Biography</td>
  <td><div dataformatas=HTML
    datasrc=#tdcComposers
    datafld=bio></div>
  </td></tr>
</table>
...
<p>
<object id=scrNavBtns type="text/x-scriptlet">
  <param name=URL value="navbtns.htm">
</object>

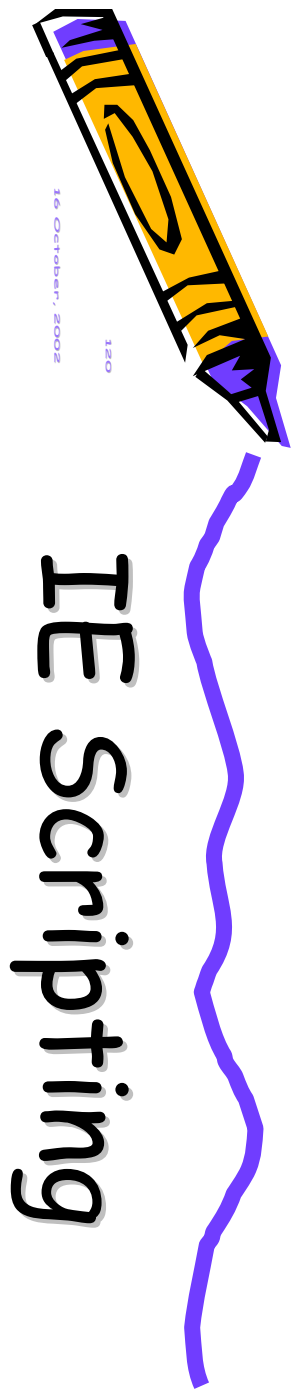
<table id=tblComposers datasrc=#tdcComposers>
<thead><tr style="font-weight:bold">
  <td>First</td><td>Last</td><td>Birth</td><td>Death</td>
  <td>Origin</td>
</tr></thead>
<tbody>
<tr id=trTemplate>
  <td><span datafld="compsr_first"></span></td>
  <td><span datafld="compsr_last"></span></td>
  <td><span datafld="compsr_birth"></span></td>
  <td><span datafld="compsr_death"></span></td>
  <td><span datafld="origin"></span></td>
</tr>
</tbody>
</table>
</body>
</html>
```



IE Data Islands

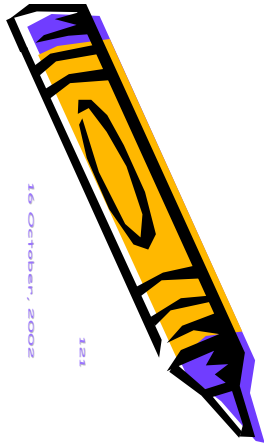
- XML embedded within a page
 - May break the usual separation of data and display
 - Sometimes data is self-contained...
 - Often very useful...
 - IE5-specific
 - Island can be used as a DSO

```
<HTML>
  <HEAD>
    <TITLE>Data Island</TITLE>
  </HEAD>
  <BODY>
    <XML ID="email">
      <EMAIL ENCRYPTED="128" PRIORITY="HIGH">
        <TO>king@asert.com.au</TO>
        <FROM>wizard@asert.com.au</FROM>
        <CC>queen@asert.com.au</CC>
        <SUBJECT>My XML Data Island</SUBJECT>
        <BODY>Isn't life GRAND!</BODY>
      </EMAIL>
    </XML>
    <!-- More Stuff Here -->
  </BODY>
</HTML>
```

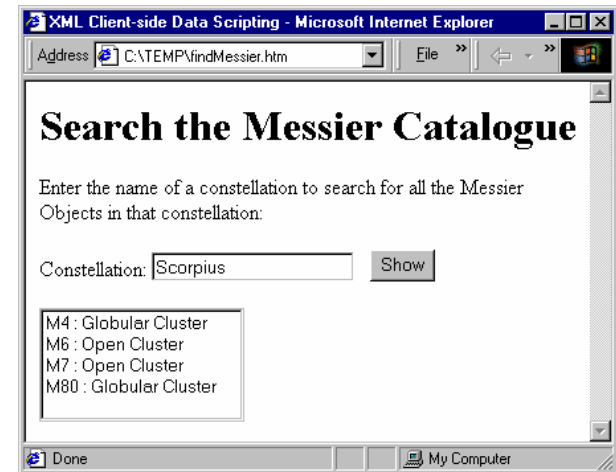


- Using a script language (PERL, JavaScript, VBScript, etc.) to perform processing
- The following example shows:
 - JavaScript
 - XML Data Island linked to an external XML file
 - navigating the HTML Document Model
 - To access the text box and list, etc.

IE Scripting...



```
<HTML>
<HEAD>
<TITLE>XML Client-side Data Scripting</TITLE>
<XML ID="dsoMessierCatalog" SRC="Messier.xml"></XML>
<SCRIPT LANGUAGE="JavaScript">
function showAccordingTo() {
    strCriteria =
        document.forms[0].txtCriteria.value.toLowerCase();
    objListBox = document.forms[0].selListBox;
    objListBox.options.length = 0;
    messierRS = dsoMessierCatalog.recordset;
    messierRS.MoveFirst();
    while (!messierRS.EOF)
    {
        StrConstell = ' ' + messierRS("CONSTELLATION");
        StrConstell = StrConstell.toLowerCase();
        if (StrConstell.indexOf(strCriteria) >= 0)
        {
            objListBox.options.length += 1;
            objListBox.options[objListBox.options.length - 1].text =
                "M" + messierRS("INDEX") + " : " + messierRS("DESCRIPTION");
        }
        messierRS.MoveNext();
    }
};
</SCRIPT>
</HEAD>
<BODY>
<H1>Search the Messier Catalogue</H1>
Enter the name of a constellation to search for all the Messier
Objects in that constellation:
<FORM>
    Constellation: <INPUT TYPE="TEXT" NAME="txtCriteria"> &nbsp;    
    <INPUT TYPE="BUTTON" VALUE="Show" ONCLICK="showAccordingTo()"><P>
    <SELECT NAME="selListBox" SIZE="5">
        <OPTION>Enter a Constellation name...
    </SELECT>
</FORM>
</BODY>
</HTML>
```



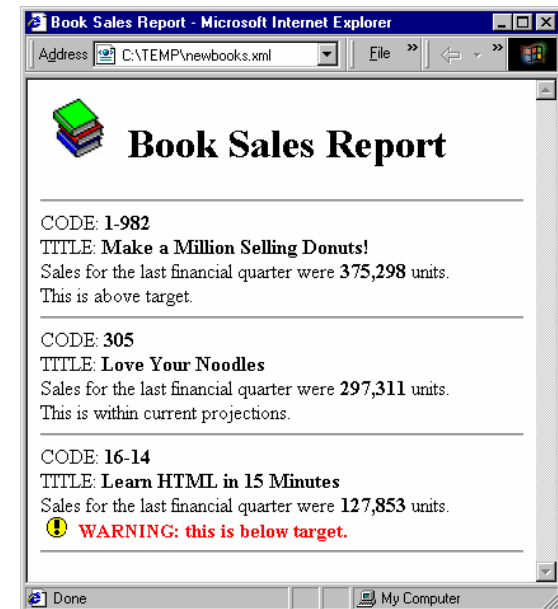
16 October, 2002

122

A Better Way

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
<xsl:template match="/">
  <HTML>
  <HEAD>
    <TITLE>Book Sales Report</TITLE>
    <STYLE TYPE="text/css">
      .under {font-size:11pt; color:red;
              font-weight:bold}
    </STYLE>
  </HEAD>
  <BODY>
    <H1><IMG SRC="books.gif"
              ALIGN="BOTTOM" HSPACE="10" />
      Book Sales Report</H1>
    <HR />
    <xsl:for-each select="//ITEM" order-by="-SALES">
      <xsl:apply-templates select="CODE" />
      <xsl:apply-templates select="TITLE" />
      <xsl:apply-templates select="SALES" />
      <HR></HR>
    </xsl:for-each>
  </BODY>
</HTML>
</xsl:template>
<xsl:template match="*">
  <xsl:node-name />: <B><xsl:value-of /></B><BR />
</xsl:template>
<xsl:template match="SALES">
  Sales for the last financial quarter were<B>
  <xsl:eval>formatNumber(this.text, "#,###,##0")</xsl:eval>
  </B>units.<BR />
  <xsl:choose>
    <xsl:when match="*[. > $gt$ 350000]">
      <SPAN>This is above target.</SPAN><BR />
    </xsl:when>
    <xsl:when match="*[. $lt$ 150000]">
      <SPAN CLASS="under">
        <IMG SRC="under.gif" ALIGN="BOTTOM" HSPACE="5" />
        WARNING: this is below target.</SPAN><BR />
      </xsl:when>
    <xsl:otherwise>
      This is within current projections.<BR />
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>
</xsl:stylesheet>
```

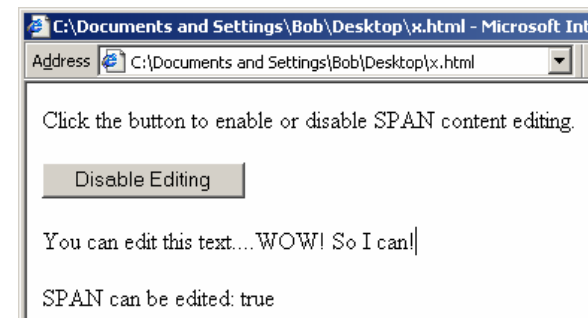
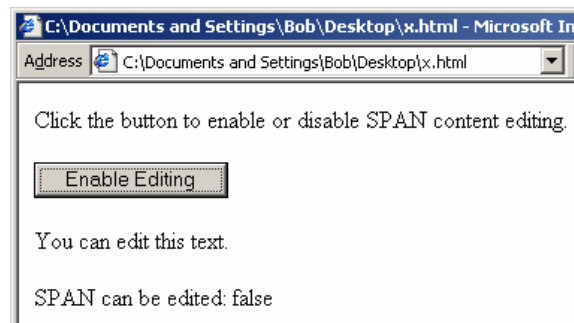
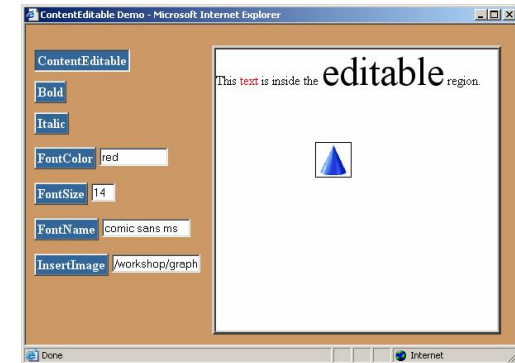
```
<?xml version="1.0"?>
<?xml-stylesheet
  type="text/xsl" href="newbooks.xsl"?>
<BOOKLIST>
  <ITEM>
    <CODE>1-982</CODE>
    <TITLE>Make a Million Selling Donuts!</TITLE>
    <SALES>375298</SALES>
  </ITEM>
  <ITEM>
    <CODE>305</CODE>
    <TITLE>Love Your Noodles</TITLE>
    <SALES>297311</SALES>
  </ITEM>
  <ITEM>
    <CODE>16-14</CODE>
    <TITLE>Learn HTML in 15 Minutes</TITLE>
    <SALES>127853</SALES>
  </ITEM>
</BOOKLIST>
```

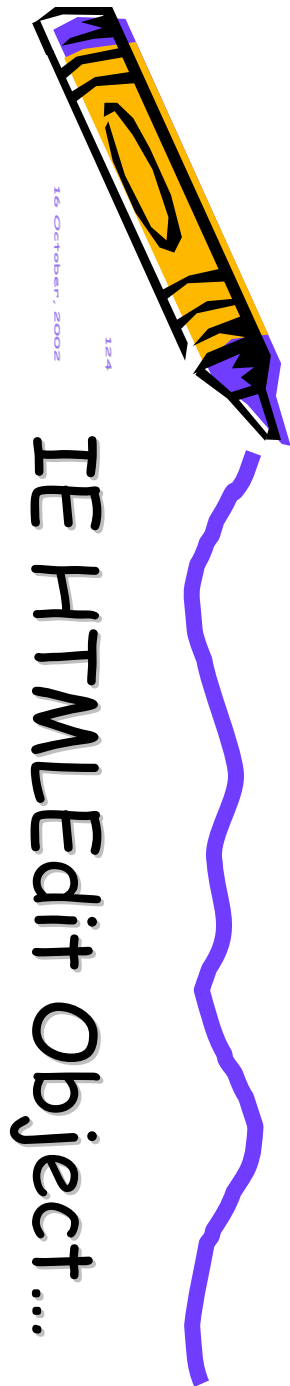


IE HTMLEdit Object

- Built in to IE; extensively controllable via JavaScript

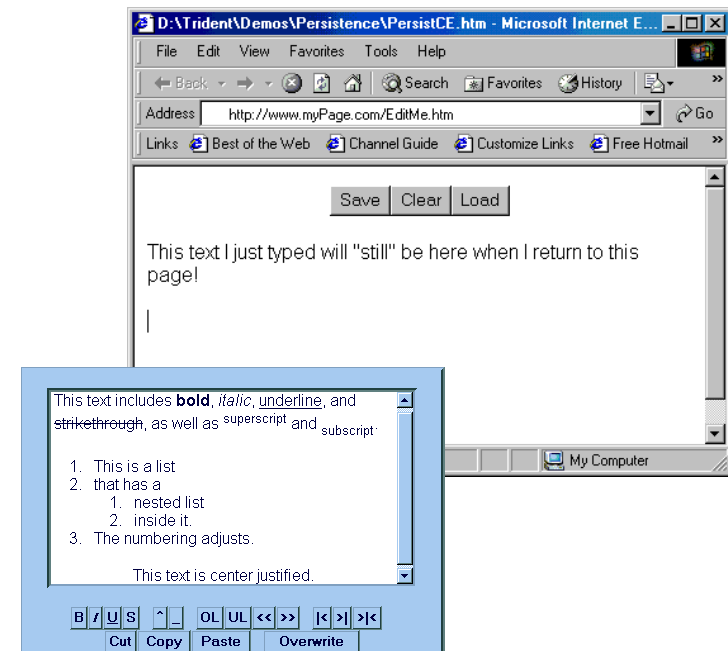
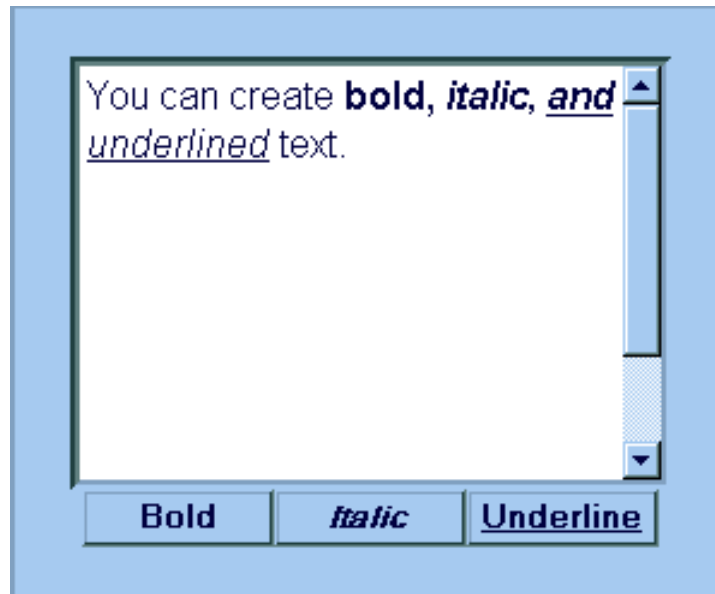
```
<HEAD>
<SCRIPT>
function chgSpan() {
    currentState = oSpan.isContentEditable;
    newState = !currentState;
    oSpan.contentEditable = newState;
    oCurrentValue.innerText = newState;
    newState==false ? oBtn.innerText="Enable Editing" :
        oBtn.innerText="Disable Editing"
}
</SCRIPT>
</HEAD>
<BODY onload="oCurrentValue.innerText = oSpan.isContentEditable;">
<P>Click the button to enable or disable SPAN content editing.</P>
<P>
<BUTTON ID="oBtn" onclick="chgSpan()">Enable Editing</BUTTON>
</P>
<P><SPAN ID="oSpan">You can edit this text.</SPAN></P>
SPAN can be edited: <SPAN ID="oCurrentValue"></SPAN>
</BODY>
```

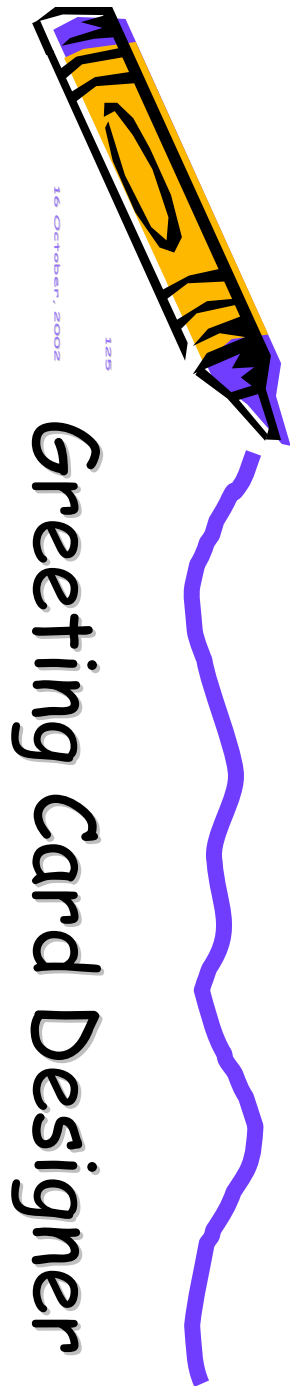




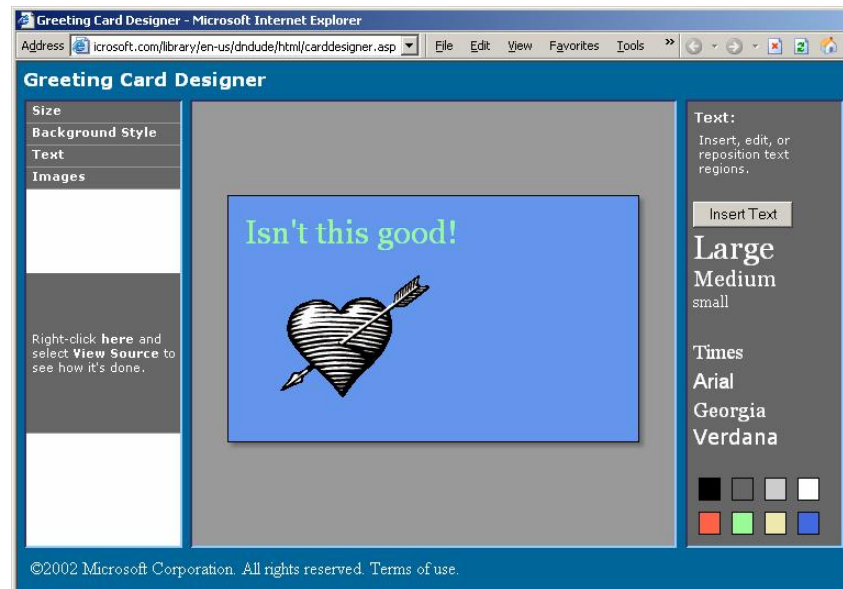
- Can be used very flexibly

```
<div align=center>
<div unselectable="on" align=center
style="height:300; width:400; background-color:powderblue; border:outset powderblue">
  <br>
  <div id=foo contenteditable align=left
    style="height:250; width:350;background-color:white; font-face:Arial; padding:3;
    border:inset powderblue; scrollbar-base-color:powderblue; overflow:auto;">
  </div>
  <br>
  <button unselectable="0n" onclick='document.execCommand("Bold");foo.focus();'
    style="width:80; background-color:powderblue; border-color:powderblue">
    <B>Bold</B></button>
  <button unselectable="0n" onclick='document.execCommand("Italic");foo.focus();'
    style="width:80; background-color:powderblue; border-color:powderblue">
    <B><I>Italic</I></B></button>
  <button unselectable="0n" onclick='document.execCommand("Underline");foo.focus();'
    style="width:80; background-color:powderblue; border-color:powderblue;">
    <B><U>Underline</U></B></button>
</div>
</div>
```

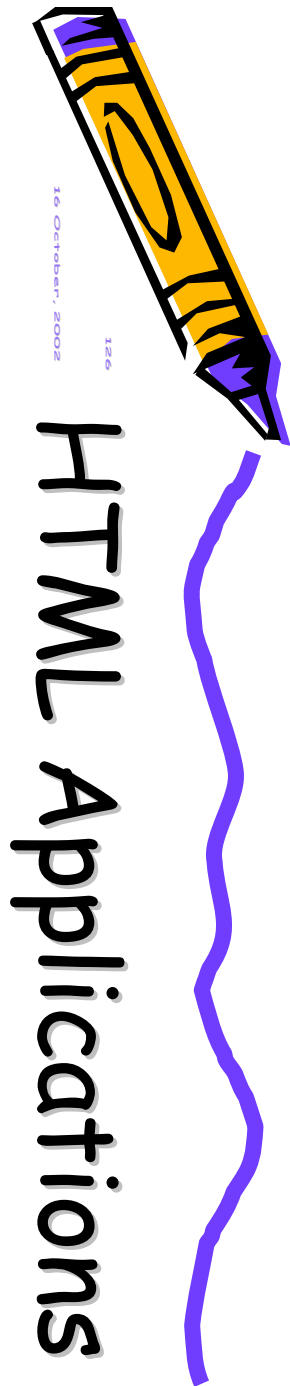




- A Microsoft Demo
 - <http://msdn.microsoft.com/library/en-us/dndude/html/carddesigner.asp>
 - Controlled by JavaScript, of course!

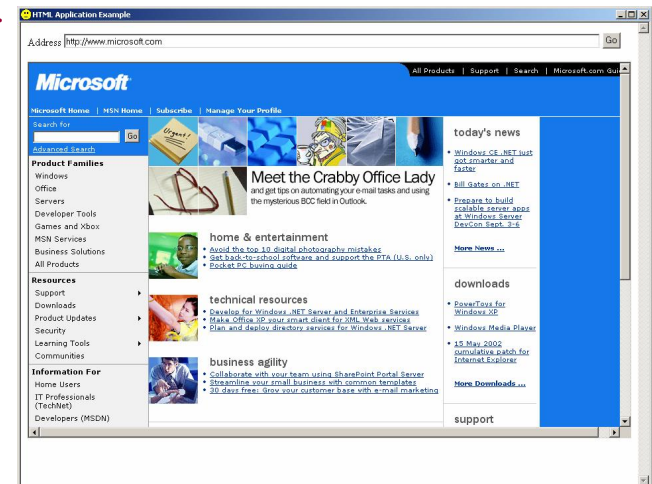


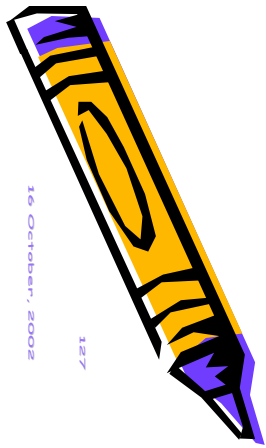
```
<!-- The fnChangeFontColor function sets the text color for control selected -->
<!-- text regions. -->
function fnChangeFontColor () {
    if (document.selection.type == "Control") {
        var oControlRange = document.selection.createRange();
        for (i = 0; i < oControlRange.length; i++)
            if (oControlRange.item(i).tagName != "IMG")
                oControlRange.item(i).style.color=event.srcElement.style.backgroundColor;
    }
}
```



- An HTML page that lets an author control window look and feel, and how it behaves in the system

```
<html>
<head>
  <TITLE>HTML Application Example</TITLE>
  <HTA:APPLICATION ID="HTAEx"
    APPLICATIONNAME="HTAEx"
    ICON="e.ico"
    WINDOWSTATE="normal">
</head>
<body>
<span id=AddressBar style="overflow: none">
<span id=AddText>Address</span>
<input type=text value="http://www.microsoft.com" id=TheAddress
  style="width: expression(document.body.clientWidth -
    AddText.offsetWidth - AddGo.offsetWidth - 45)">
<input type=button value="Go" id=AddGo onclick="navigate()"><br>
<span>
<br />
<iframe src="http://www.microsoft.com" id=TheFrame
  style="width: 100%; height: 85%"></iframe>
<script language=JavaScript>
function navigate() {
  document.all.TheFrame.src = TheAddress.value;
}
function clickShortcut() {
  if (window.event.keyCode == 13) {
    navigate()
  }
}
TheAddress.onkeypress = clickShortcut;
</script>
</body>
</html>
```





HTML+Time

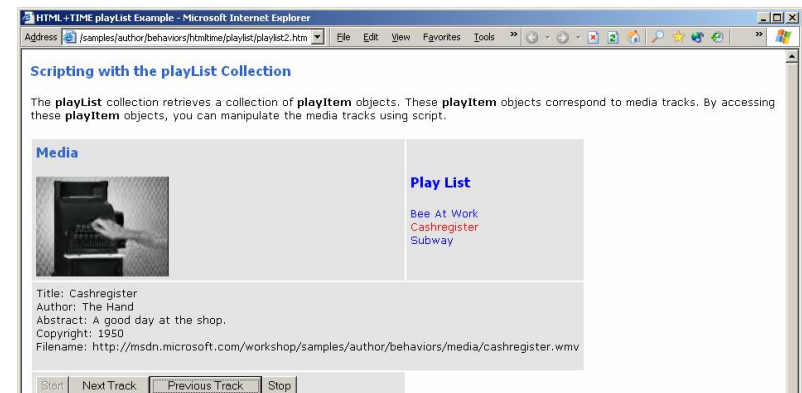
- Provides a declarative means for adding multimedia and interactivity to Web pages
 - Based on W3C's SMIL 2.0 standard

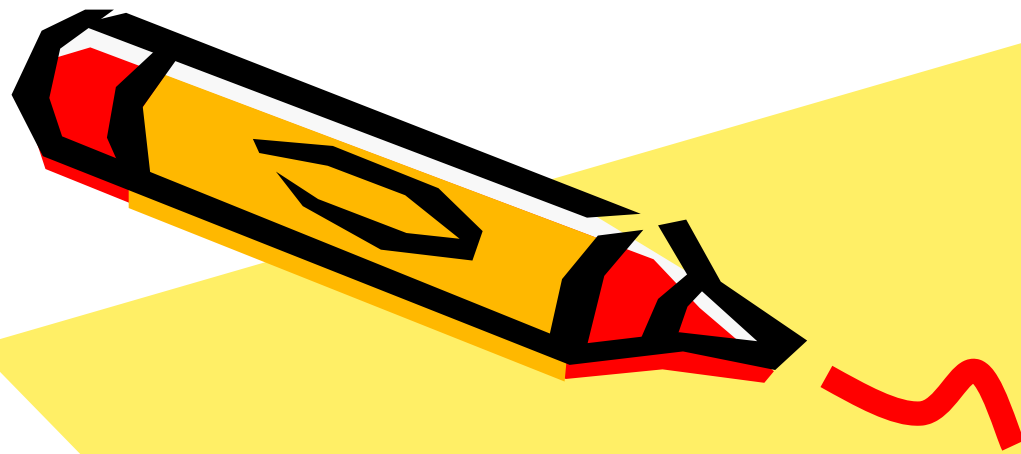
```
<STYLE>
  .time{ behavior: url(#default#time2);}
  .trackDisplay {cursor:hand;}
</STYLE>

<SCRIPT LANGUAGE="JScript">
...
<TABLE>
<TR>
<TD WIDTH="380px" HEIGHT="130px">
<H2>Media</H2>

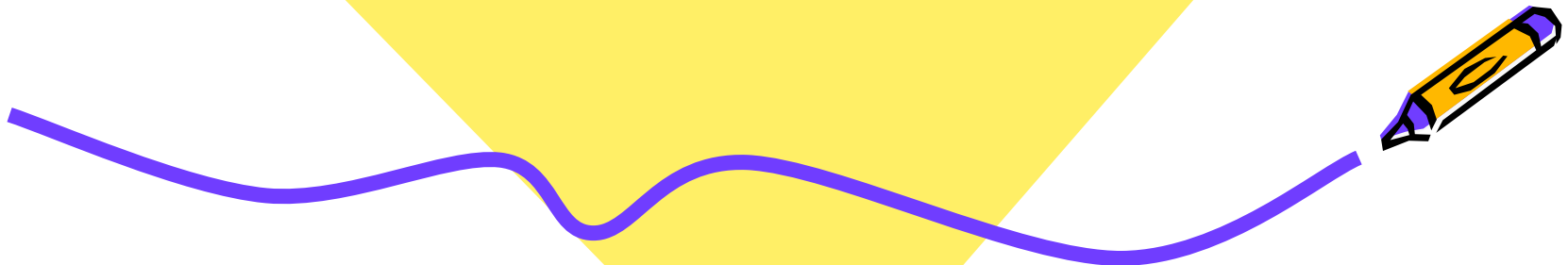
<t:MEDIA ID="oMedia" SRC="/workshop/samples/author/behaviors/media/short_3tracks.asx"
BEGIN="indefinite"
onend="updateBtns();" timeAction="visibility" ontrackchange="updateFields();" />

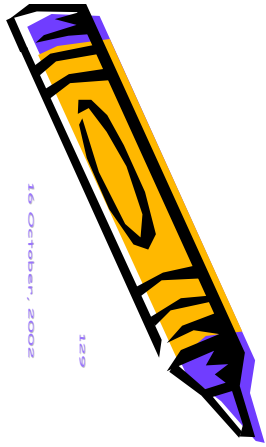
...
<BUTTON ID="btnStart"
  onclick="oMedia.beginElement();fnCreatePlayList();
        updateBtns();updateFields();">
  Start
</BUTTON>
<BUTTON ID="btnNext"
  onclick="oMedia.playlist.nextTrack();">
  Next Track
</BUTTON>
...
</TD>
</TR>
</TABLE>
...
```





JavaScript Goodies in NS

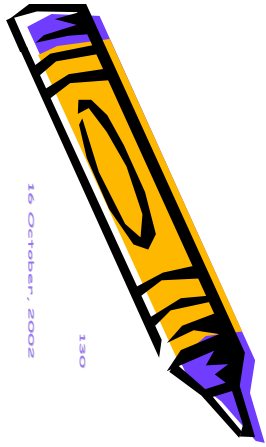




NS Features

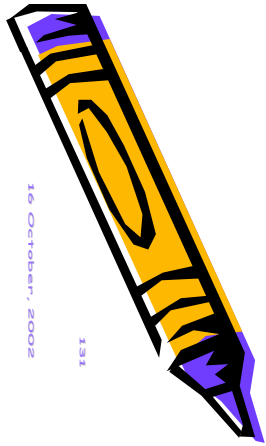
- Netscape (and Gecko in particular) isn't standing still any more than IE is
 - Takes a much more W3C-oriented focus
 - (funny how being killed in the market often causes a standards epiphany...)

"Netscape 6 has more in common with MSIE 5.x than with Netscape 4.x, but of course there are some big differences. Proprietary add-on features of each program aside, the basic differences lie mostly in the fact that Internet Explorer has built-in redundancy. Think of Internet Explorer as being a kind, old schoolteacher who lets you make plenty of mistakes and be late with homework. In contrast, Netscape 6 is more like a drill sergeant at boot camp!"



XBL

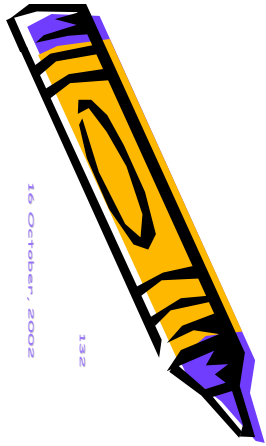
- XML Binding Language
 - Removes the need for much JavaScript
 - A declarative technology
 - *MUCH* better
 - A markup language for describing bindings that can be attached to elements in other documents
 - Can contain event handlers that are registered on the bound element, an implementation of new methods and properties that become accessible from the bound element, and anonymous content that is inserted underneath the bound element



16 October, 2002

XBL...

```
<html>
<head>
<link rel="stylesheet" type="text/css" href="xbl_menus.css">
</head>
<body>
<menus>
  <menu-group>
    <header>A header</header>
    <items>
      <item><a href="Url1.html">item 1</a></item>
      <item><a href="Url2.html">item 2</a></item>
    </items>
  </menu-group>
  <menu-group>
    <header>XBL</header>
    <items>
      <item>Questions</item>
      <item>Syntax</item>
      <item>Sample Code</item>
    </items>
  </menu-group>
</menus>
<br />
These drop down menus were written in XBL!
</body>
</html>
```



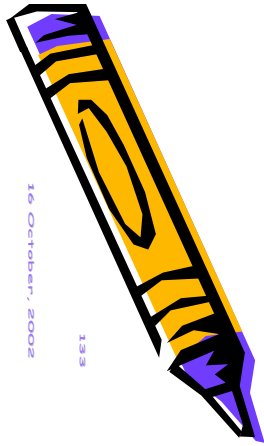
16 October, 2002

1312

XBL...

```
body { font-family: arial; }
menus { -moz-binding: url(doritos.xml#menus); }
menu-group {
    -moz-binding: url(doritos.xml#menu-group);
    display: block; background-color: transparent;
}
header {
    -moz-binding: url(doritos.xml#header);
    cursor: default;
    background-color: #ffd;
    border: 2px; border-color: black;
    color: black;
    font-weight: bold; font-size: 12pt;
}
items {
    -moz-binding: url(doritos.xml#items);
    position: absolute;
    left: -3000px;
    background-color: #ffd;
    border: 2px solid black;
    padding: 5px;
    color: black;
    font-family: arial; font-size: 10pt;
}
item { -moz-binding: url(doritos.xml#item); }
a { color: blue; }
a:visited { -moz-opacity: 50%; }
a:hover { background-color: #996; }
```

XBL...



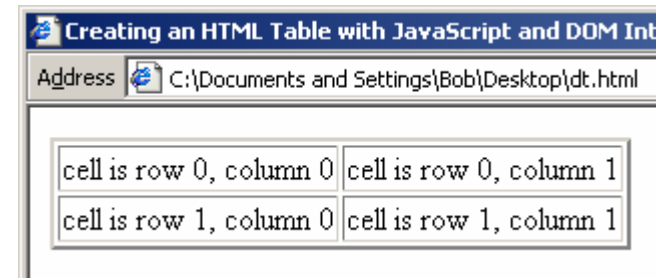
```
<?xml version="1.0" ?>
<bindings id="drop-down-menus"
  xmlns="http://www.mozilla.org/xbl"
  xmlns:html="http://www.w3.org/1999/xhtml"
  xmlns:xul="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
  <binding id="menus">
    <content>
      <xul:box orient="horizontal">
        <children/>
      </xul:box>
    </content>
  </binding>
  <binding id="menu-group">
    <content>
      <children/>
    </content>
    <handlers>
      <handler event="mouseover">
        menu=this.childNodes.item(3);
        menu.style.left="auto";
        header=this.childNodes.item(1);
        header.style.backgroundColor="#9966";
      </handler>
      <handler event="mouseout">
        ...
      </handler>
    </handlers>
  </binding>
  <binding id="header">
    <content>
      <children/>
    </content>
  </binding>
  ...
  ...
  <binding id="items">
    ...
  </binding>
  <binding id="item">
    <content>
      <children/>
      <html:br/>
    </content>
    <implementation>
      <constructor>
        this.style.cursor="default";
      </constructor>
    </implementation>
    <handlers>
      <handler event="mouseover">
        this.style.backgroundColor="#9966";
      </handler>
      <handler event="mouseout">
        this.style.backgroundColor="transparent";
      </handler>
    </handlers>
  </binding>
</bindings>
```

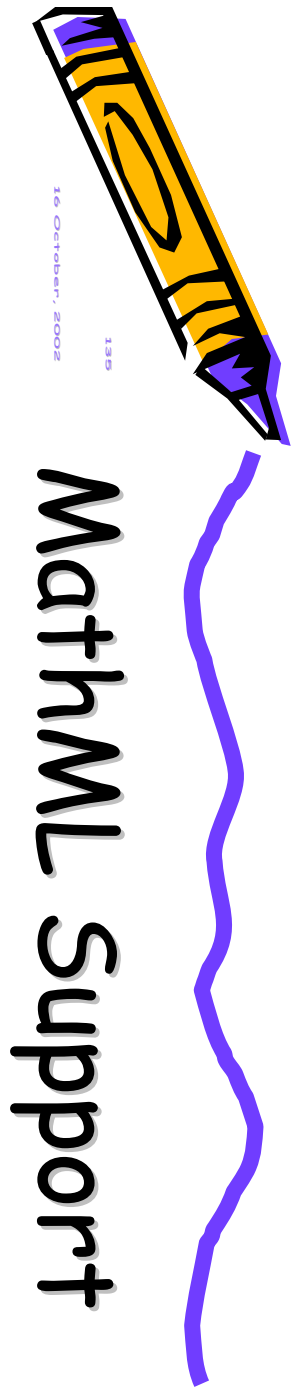
• Netscape example

```

<html>
<head>
<title>Creating an HTML Table with JavaScript and DOM Interfaces</title>
<script>
  function start() {
    var mybody=document.getElementsByTagName("body").item(0);
    mytable = document.createElement("TABLE");
    mytablebody = document.createElement("TBODY");
    for(j=0;j<2;j++) {
      mycurrent_row=document.createElement("TR");
      for(i=0;i<2;i++) {
        mycurrent_cell=document.createElement("TD");
        currenttext=
          document.createTextNode("cell is row "+j+", column "+i);
        mycurrent_cell.appendChild(currenttext);
        mycurrent_row.appendChild(mycurrent_cell);
      }
      mytablebody.appendChild(mycurrent_row);
    }
    mytable.appendChild(mytablebody);
    mybody.appendChild(mytable);
    mytable.setAttribute("border","2");
  }
</script>
</head>
<body onload="start()">
</body>
</html>

```



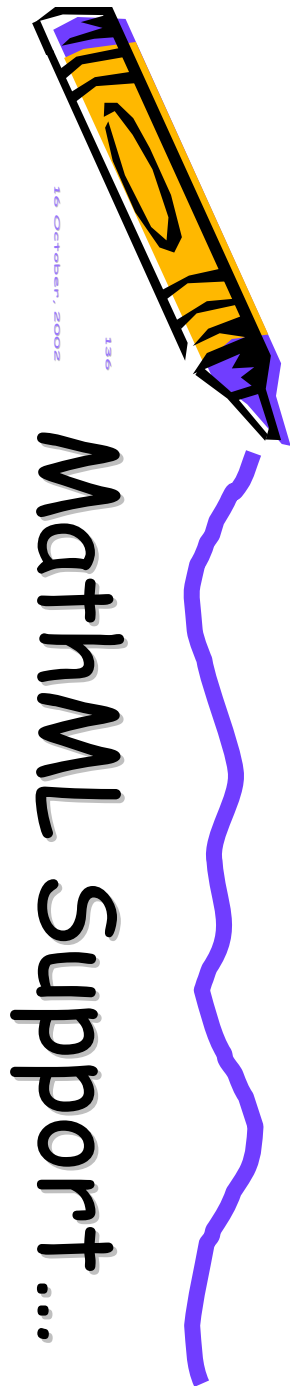


- MathML is an XML vocabulary for maths
 - Typesetting and more...

```
<math>
  <mrow>
    <mi>y</mi>
    <mo>=</mo>
    <mi></mi>
    <mfrac href="http://www.w3.org/" xml:link="simple">
      <mn>1</mn>
      <msqrt>
        <mrow>
          <msup>
            <mi>x</mi>
            <mn>2</mn>
          </msup>
          <mo>+</mo>
          <mn>1</mn>
        </mrow>
      </msqrt>
    </mfrac>
  </mrow>
</math>
```

$$y = \frac{1}{\sqrt{x^2 + 1}}$$

This example also shows the use of an XLink



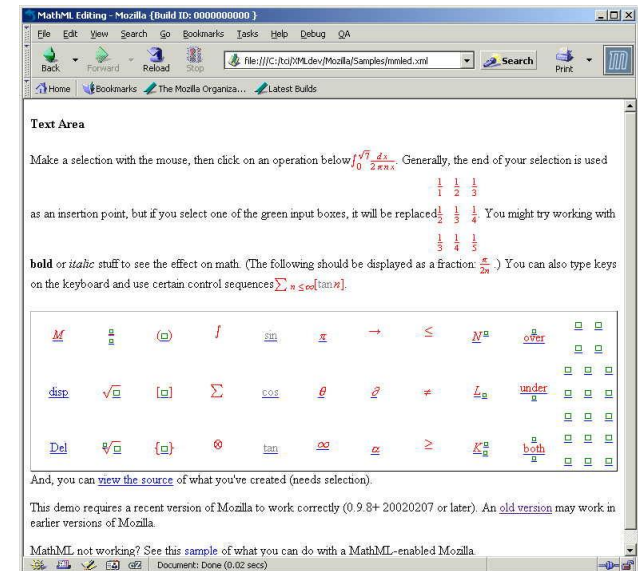
- A Prototype JavaScripted MathML editor is being worked on

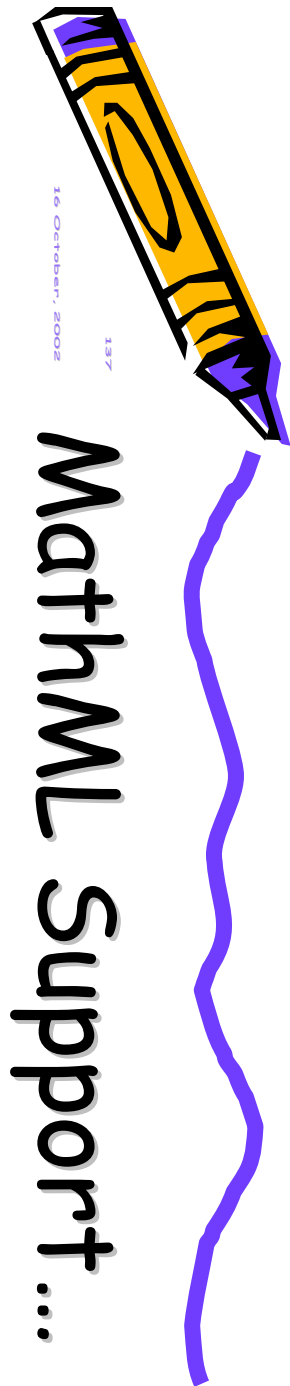
- <http://www.newmexico.mackichan.com/MathML/mmled.xml>

- Support for output, input and evaluation; will eventually allow

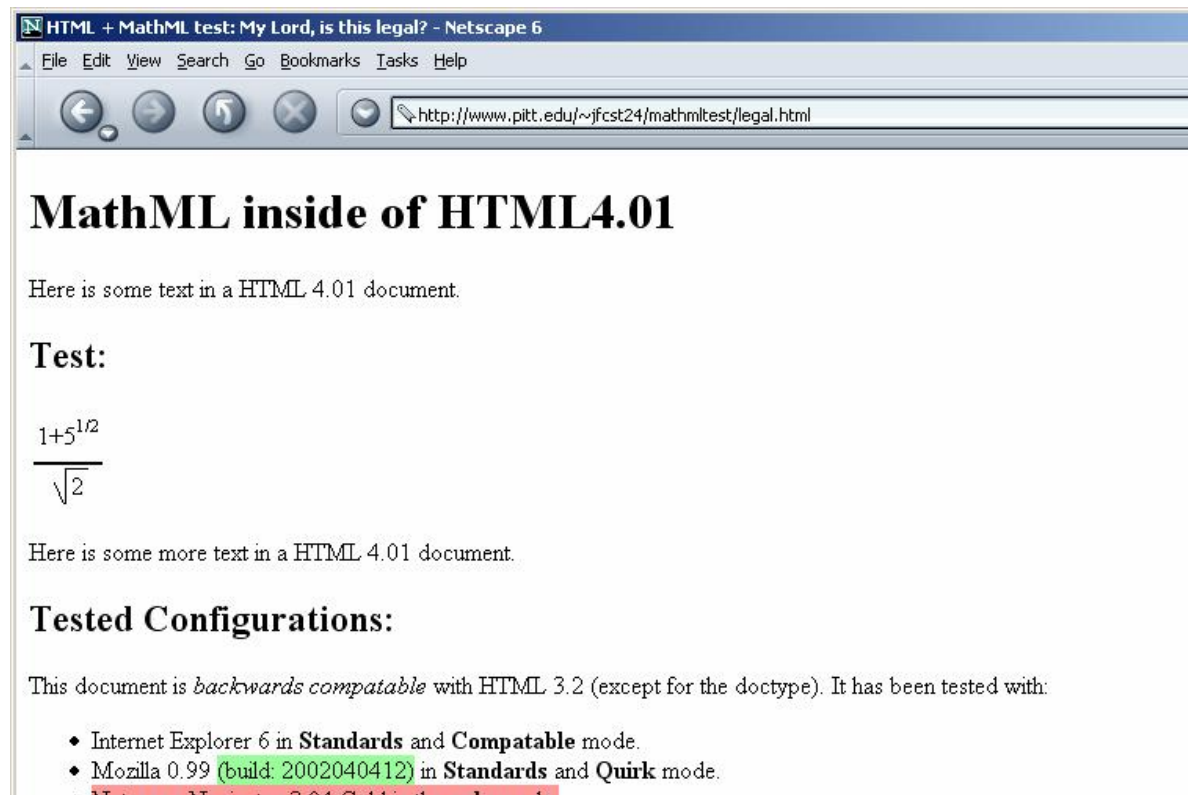
```
<form>
  <object type="text/mathml">
    ...
  </object>
</form>
```

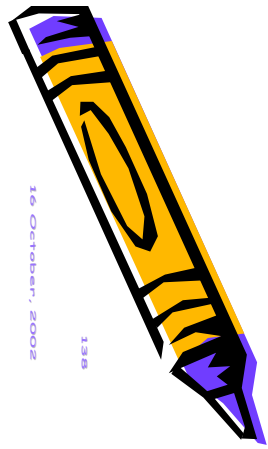
```
<a href="javascript:insertmatrix(2,2);"
  title="enter 2 by 2 matrix (CTRL 2)">
<mml:math>
  <mml:mtable>
    <mml:mtr>
      <mml:td> <mml:mi tempinput="true">&#x25A1;</mml:mi> </mml:td>
      <mml:td> <mml:mi tempinput="true">&#x25A1;</mml:mi> </mml:td>
    </mml:mtr>
    <mml:mtr>
      <mml:td> <mml:mi tempinput="true">&#x25A1;</mml:mi> </mml:td>
      <mml:td> <mml:mi tempinput="true">&#x25A1;</mml:mi> </mml:td>
    </mml:mtr>
  </mml:mtable>
</mml:math>
</a>
```





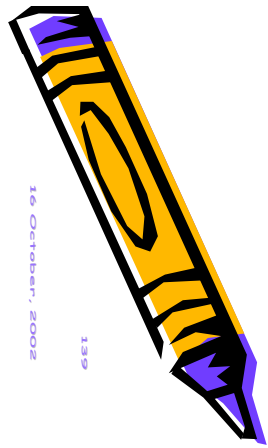
- JavaScript MathML processor/renderer
 - <http://www.pitt.edu/~jfcst24/mathmltest/legal.html>
 - Also works with IE





LiveConnect

- Allows JavaScript to interact with Java
- Provides two-way communication:
 - JavaScript to Java
 - Based around a number of fundamental Objects:
 - `JavaArray`, `JavaClass`, `JavaObject`, `JavaPackage`
 - Java to JavaScript
 - Based around two fundamental Objects:
 - `netscape.javascript.JSObject`
 - `netscape.javascript.JSException`



LiveConnect...

- JavaScript-side:

```
x = java.lang.reflect.Array.newInstance(java.lang.Integer, 10);

var myString = new Packages.java.lang.String("Hello world");

function getClass(javaClassName) {
    try {
        var theClass = java.lang.Class.forName(javaClassName);
    } catch (e) {
        return ("The Java-side exception is " + e);
    }
    return theClass
}
```

- Java-side:

```
import netscape.javascript.*;

public class JavaDog
{
    public String dogBreed;
    public String dogColor;
    public String dogSex;

    public JavaDog(JSObject jsDog)
    {
        this.dogBreed =
            (String)jsDog.getMember("breed");
        this.dogColor =
            (String)jsDog.getMember("color");
        this.dogSex =
            (String)jsDog.getMember("sex");
    }
}
```

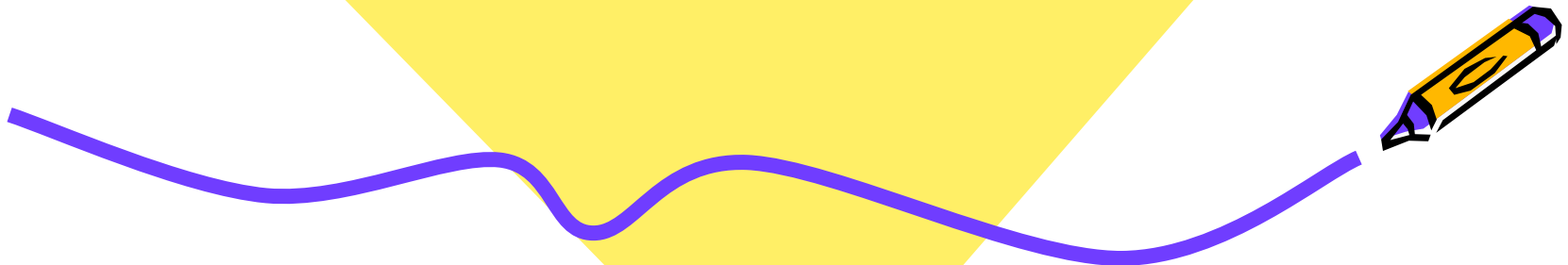
```
function JSDog(breed,color,sex) {
    this.breed = breed
    this.color = color
    this.sex = sex
}
```

```
var gabby =
    new JSDog("lab","chocolate",
              "female")
```

```
var javaDog =
    new Packages.JavaDog(gabby)
```

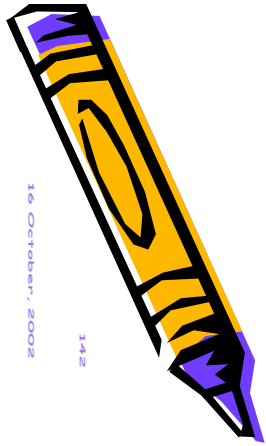


JavaScript and the Server Environment





- JavaScript isn't just about working with the browser, it also has a place on the server side as well
 - Especially in the Microsoft world

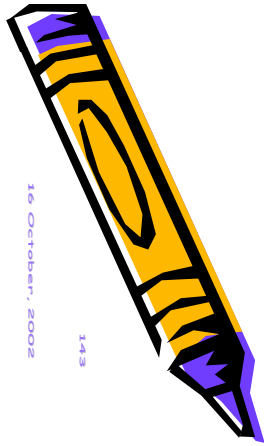


WebServices!

- Apache Tomcat permits JavaScript-based WebServices

```
<?xml version="1.0"?>

<isd:service
  xmlns:isd="http://xml.apache.org/xml-soap/deployment"
  id="urn:calculator">
  <isd:provider type="script" scope="Application"
    methods="plus minus">
    <isd:script language="javascript">
      function plus(x, y) {
        return x + y;
      }
      function minus (x, y) {
        return x - y;
      }
    </isd:script>
  </isd:provider>
</isd:service>
```



XML

- A fairly natural fit
 - JavaScript can access the XML DOM

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
```

```
<html>
```

```
<head>
```

```
<title>NamedNodeMap Processing</title>
```

```
<xml id="CD" src="CD.xml"></xml>
```

```
<script for=window event=onload>
```

```
<!--
```

```
div.innerHTML = "";
```

```
var xmlDoc = document.all("CD").XMLDocument;
```

```
var root = xmlDoc.documentElement
```

```
var attrs = root.attributes;
```

```
div.innerHTML += "<strong>Title: </strong>" +  
    attrs.getItem("title").text + "<br>";
```

```
div.innerHTML += "<strong>Artist: </strong>" +  
    attrs.getItem("artist").text + "<br>";
```

```
div.innerHTML += "<strong>Date: </strong>" +  
    attrs.getItem("date").text + "<br>";
```

```
div.innerHTML += "<strong>Number of tracks: </strong>" +  
    attrs.getItem("ntracks").text + "<br>";
```

```
-->
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<p>
```

```
<DIV ID="div"></DIV>
```

```
</p>
```

```
</body>
```

```
</html>
```

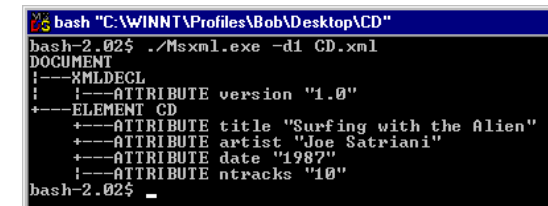
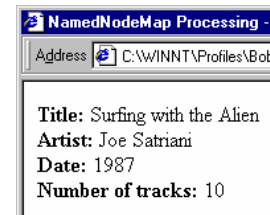
```
<?xml version="1.0"?>
```

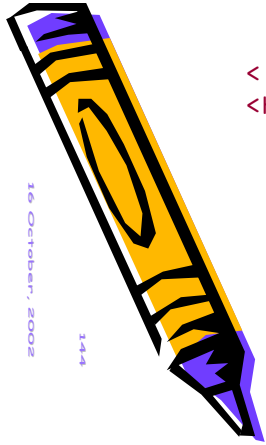
```
<CD title="Surfing with the Alien"
```

```
artist="Joe Satriani"
```

```
date="1987"
```

```
ntracks="10" />
```

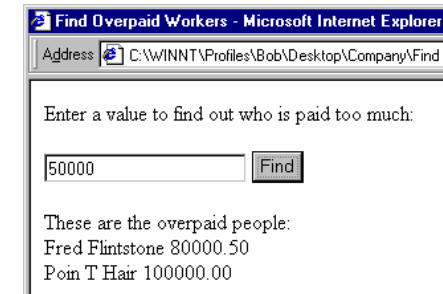


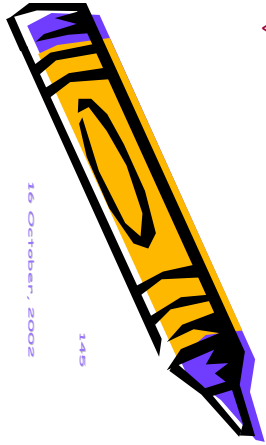


XML...

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
<html>
  <head>
    <title>Find Overpaid Workers</title>
    <xml id="Company" src="Company.xml"></xml>
    <script>
      <!--
      function find()
      {
        overPaid.innerHTML = "";
        var xmlDoc = document.all ("Company").XMLDocument;
        var root = xmlDoc.documentElement;
        var kids = root.getElementsByTagName ("emp");
        var value = inputText.value * 1.0;
        for (var i = 0; i < kids.length; i ++)
        {
          var node = kids.item(i);
          var salaries = node.getElementsByTagName ("salary");
          var salaryNode = salaries.item(0);
          var salaryValue = salaryNode.childNodes.item(0).nodeValue * 1.0;
          if (salaryValue >= value)
            overPaid.innerHTML += node.text + "<br>";
        }
      }
      -->
    </script>
  </head>
  <body>
    <p>
      Enter a value to find out who is paid too much:
    </p>
    <p>
      <input type="text" name="inputText" size="20">
      <input type="button" name="findButton"
        value="Find" onClick="find()">
    </p>
    <p>
      These are the overpaid people: <DIV ID="overPaid"></DIV>
    </p>
  </body>
</html>
```

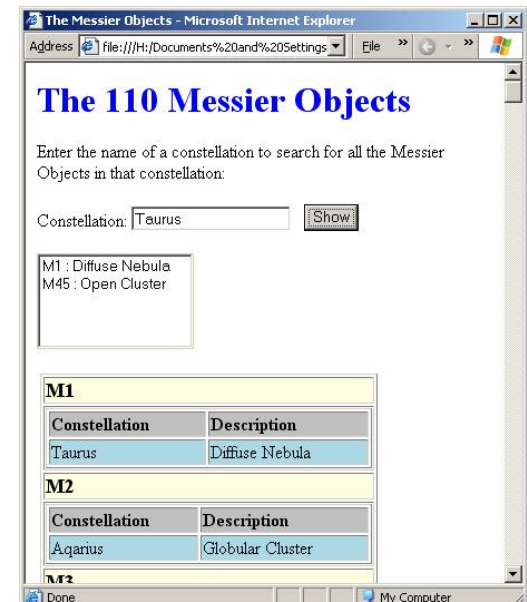
```
<?xml version="1.0"?>
<company>
  <dept code="software">
    <emp role="manager">
      <name>
        <first>Fred</first>
        <last>Flintstone</last>
      </name>
      <salary>80000.50</salary>
    </emp>
    ...
  </dept>
</company>
```

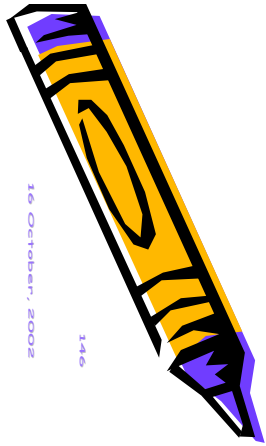




XML...

```
<html>
<head>
  <title>The Messier Objects</title>
  <script for="window" event="onload">
    <!--
    var xmlDso = xmldso.XMLDocument;
    countMess.innerHTML = " " + xmlDso.documentElement.childNodes.length + " ";
    -->
  </script>
  <script language="JavaScript" src="showAccordingTo.js">
  </script>
</head>
<body>
  <xml src="Messier.xml" id="xmldso"></xml>
  <h1 style="color:blue">The <span id="countMess"></span> Messier Objects</h1>
  Enter the name of a constellation to search for all the Messier Objects in that constellation:
  <form>
    <p>
      Constellation:
      <input type="TEXT" name="txtCriteria">
      &nbsp;
      <input type="BUTTON" value="Show" ONCLICK="showAccordingTo(document.xmlDso)">
    </p>
    <select name="selListBox" size="5"><option>Enter a Constellation name...</select>
  </form>
  <table width="100%">
    <tr>
      <td>
        <table border="1" width=75% datasrc=#xmldso>
          <tr>
            <td bgcolor="lightYellow" style="font-size:1.2em; font-weight:bold">
              M<span datafld="INDEX"></span>
            </td>
          </tr>
          <tr>
            <td>
              <table border="1" width=100%>
                <thead align="left" bgcolor="silver">
                  <th>Constellation</th>
                  <th>Description</th>
                </thead>
                <tr align="left" bgcolor="lightBlue">
                  <td><div datafld="CONSTELLATION"></div></td>
                  <td><div datafld="DESCRIPTION"></div></td>
                </tr>
              </table>
            </td>
          </tr>
        </table>
      </td>
    </tr>
  </table>
</body>
</html>
```

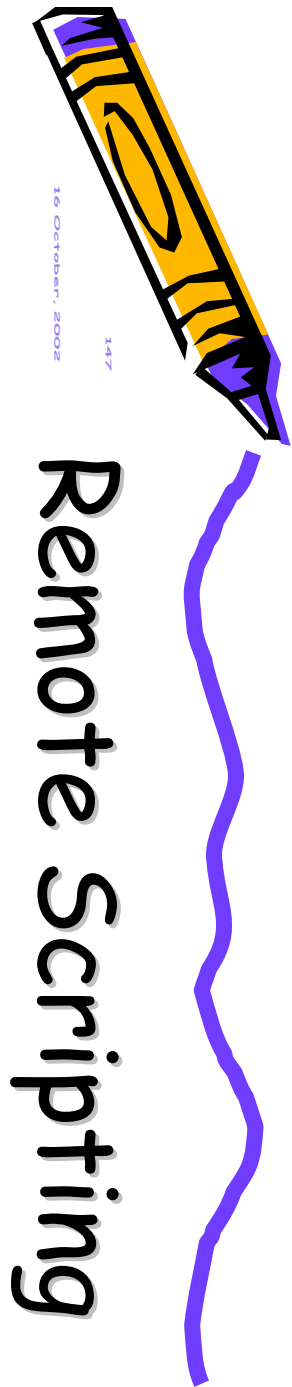




XML...

```
function showAccordingTo(doc,dsoM0)
{
    var form = doc.forms [0];
    var strCriteria = form.txtCriteria.value.toLowerCase();
    var objListBox = form.selListBox;
    objListBox.options.length = 0;
    var messierRS = dsoM0.recordset;
    messierRS.MoveFirst();
    while (!messierRS.EOF)
    {
        var StrConstell = (" " + messierRS("CONSTELLATION")).
                        toLowerCase();
        if (StrConstell.indexOf(strCriteria) >= 0)
            objListBox.options[objListBox.options.length++].text =
                "M" + messierRS("INDEX") + " : " +
                messierRS("DESCRIPTION");
        messierRS.MoveNext();
    }
}
```

```
<?xml version="1.0"?>
<MESSIER>
  <M INDEX="1">
    <CONSTELLATION>Taurus</CONSTELLATION>
    <DESCRIPTION>Diffuse Nebula</DESCRIPTION>
  </M>
  <M INDEX="2">
    <CONSTELLATION>Aqarius</CONSTELLATION>
    <DESCRIPTION>Globular Cluster</DESCRIPTION>
  </M>
  <M INDEX="3">
    <CONSTELLATION>Canes Venatici</CONSTELLATION>
    <DESCRIPTION>Globular Cluster</DESCRIPTION>
  </M>
  <M INDEX="4">
    <CONSTELLATION>Scorpius</CONSTELLATION>
    <DESCRIPTION>Globular Cluster</DESCRIPTION>
  </M>
  ...
</MESSIER>
```



- Available in both Netscape Navigator (4.x+) and Internet Explorer (4.x+)
- Allows interactive Web applications in which the browser can call scripts on the server without reloading the Web page
 - Makes it possible to validate data while the user is still filling out a form, without having to continually reload a complete page
 - Fine-grained interactions

"Remote scripting is a great technique for creating a more desktop-like feel for Web browser-based applications."

Remote Scripting...

16 October, 2002

148

```
<html>
<head>
<title> Remote Scripting Example </title>
<script language="JavaScript" src="_ScriptLibrary/RS.HTM"></script>

<script language="JavaScript">

function reverseString()
{
var strTest = prompt("Enter String To Reverse:");
var objRS = RSGetASPObject("rev.asp");
var objResult = objRS.revStr(strTest);

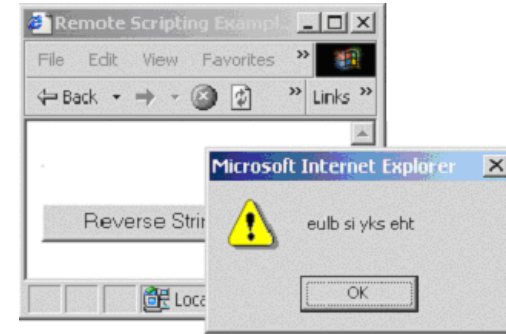
alert(objResult.return_value);
}

</script>
</head>
<body>

<script language="JavaScript">
RSEnableRemoteScripting("/_ScriptLibrary");
</script>

<form name="frmTest">
<input type="button"
      onClick="reverseString()"
      value="Reverse String >>">
</form>

</body>
</html>
```



```
<!--#INCLUDE FILE="_ScriptLibrary/RS.ASP"-->
<% RSDispatch %>

<script language="JavaScript" runat="server">

var public_description = new MainMethod();

function MainMethod()
{
this.revStr = Function('strString','return
reverse(strString)');
}

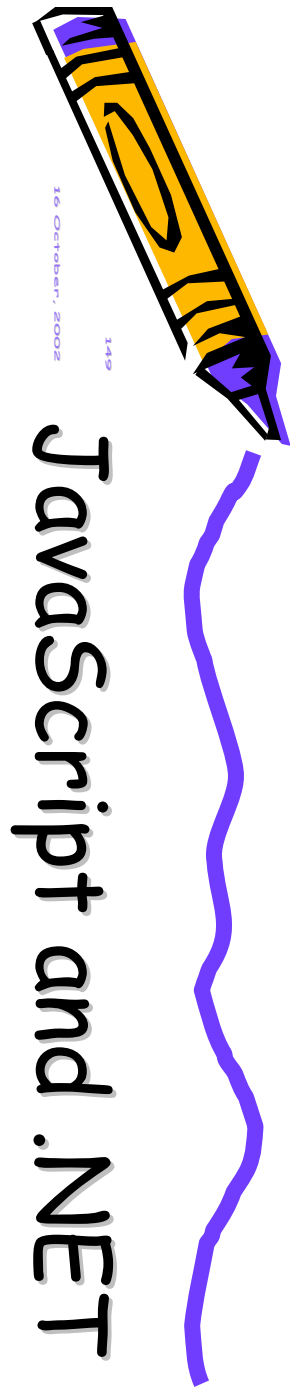
</script>

<script language="VBScript" runat="server">

function reverse(strString)
reverse = strReverse(strString)
end function

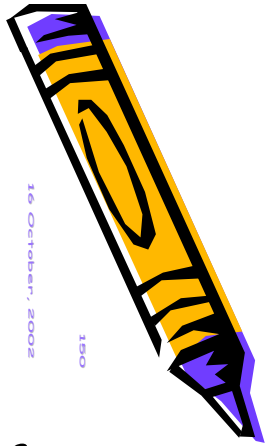
</script>
```

<http://www.devarticles.com/content.php?articleId=99&page=1>



- Script for the .NET Framework
 - First implementation of JavaScript2.0/ ECMAScript4
 - Very different language
 - Probably a good thing...
 - Compiled, not interpreted

```
package jsnetSample {  
    class jsnetClass {  
        public static var staticString : String;  
        public function methodOne(param1 : String) : int {  
            var i : int = 5;  
            Console.WriteLine("Hello {0}\n", param1);  
            return i;  
        }  
        public function methodA(param1 : int) : String {  
            return "this is a string";  
        }  
        private function privateMethod() {  
            //...  
        }  
        static {  
            sampleString = "This is a static initializer";  
        }  
    }  
}
```



JavaScript and .NET...

```
enum FormatFlags : byte {  
    ToUpperCase = 1,  
    ToLowerCase = 2,  
    TrimLeft    = 4,  
    TrimRight   = 8,  
    UriEncode   = 16.  
}
```

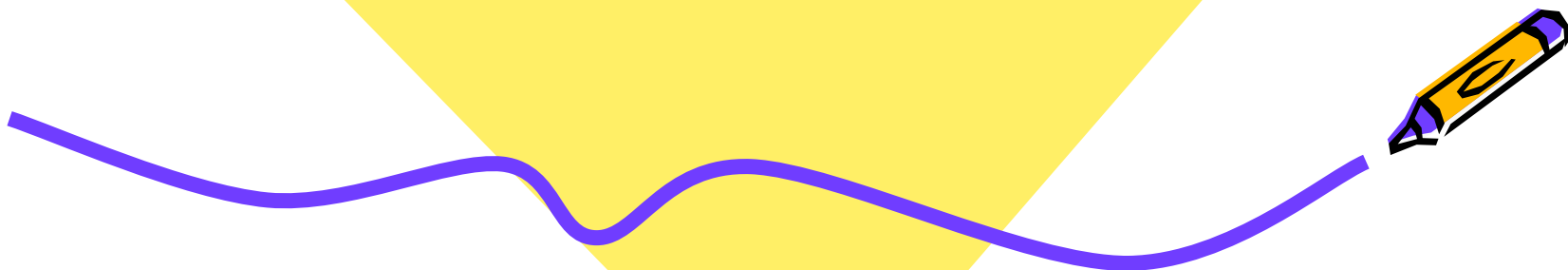
```
function Format(s : String, flags : FormatFlags) : String {  
    var ret : String = s;  
    if(flags & FormatFlags.ToUpperCase)  
        ret = ret.toUpperCase();  
    if(flags & FormatFlags.ToLowerCase)  
        ret = ret.toLowerCase();  
    if(flags & FormatFlags.TrimLeft)  
        ret = ret.replace(/^\s+/g, "");  
    if(flags & FormatFlags.TrimRight)  
        ret = ret.replace(/\s+$/g, "");  
    if(flags & FormatFlags.UriEncode)  
        ret = encodeURIComponent(ret);  
    return ret;  
}
```

```
const daysInWeek : int = 7;  
const favoriteDay : String = "Friday";  
const maxDaysInMonth : int = 31,  
      maxMonthsInYear : int = 12
```

```
<%@ language=jscript %>  
<html>  
<body>  
<script runat=server>  
class Circle {  
    function Circle (radius : double) {  
        this.r = radius;  
    }  
    var r : double;  
    static var pi : double = Math.PI;  
    function area () : double {  
        return Circle.pi * this.r * this.r;  
    }  
}  
</script>  
<%  
    var ACircle : Circle = new Circle(2);  
    Response.Write("Area of the circle is " + ACircle.area() + "<br />\n");  
%>  
</body>  
</html>
```

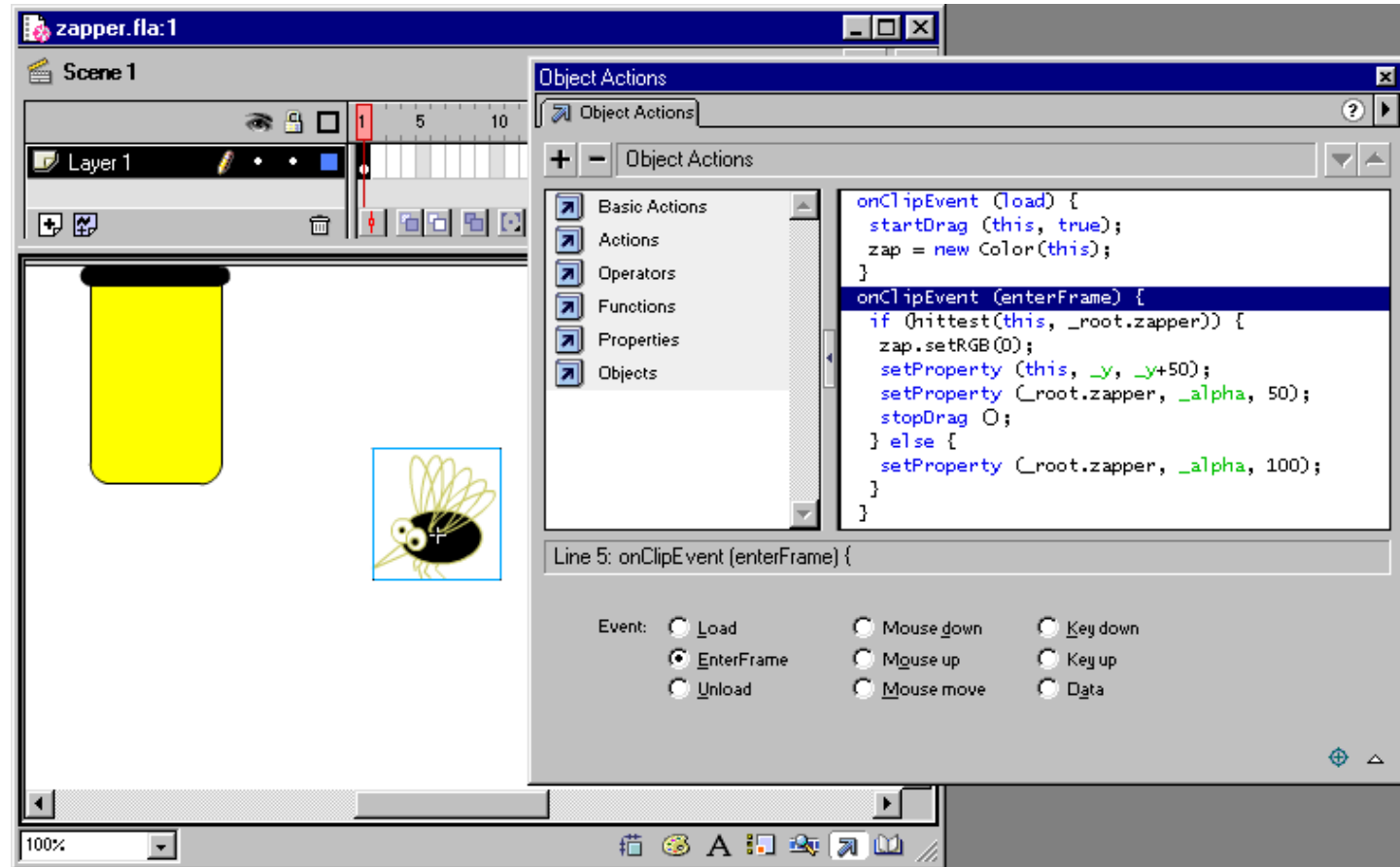


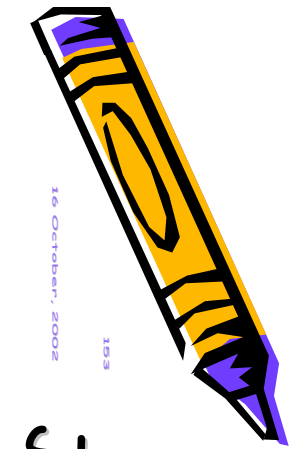
Miscelleny



- The scripting language underlying Macromedia's Flash

ActionScript





JavaScript Shell Scripting

- A Useful facility
 - A feature of the Windows Scripting Host

```
var e= new Enumerator(GetObject("winmgmts:").
                        InstancesOf("Win32_process"))

for (;!e.atEnd();e.moveNext())
{
    var process = e.item();
    WScript.echo (process.Name + "\t" +
                  process.processid)
}
```

```
C:\WINDOWS\System32\cmd.exe

C:\WMI> cscript /nologo ShowMemory.js
Total Physical Memory (kb): 261612

C:\WMI>
```

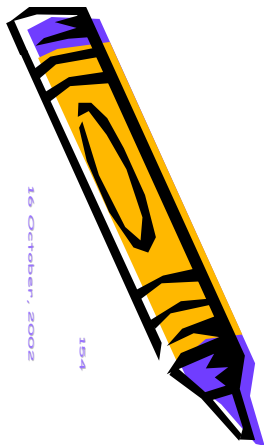
```
C:\WINDOWS\System32\cmd.exe

csrss.exe      508
winlogon.exe   532
services.exe   576
lsass.exe      588
svchost.exe    764
svchost.exe    816
svchost.exe    924
svchost.exe    984
spoolsv.exe    1108
mdm.exe        1236
NAVAPSUC.EXE   1296
nsvsc32.exe    1348
squid.exe      1448
svchost.exe    1464
ups.exe        1488
dnsserver.exe  1644
dnsserver.exe  1664
dnsserver.exe  1684
pingr.exe      1716
unlinkd.exe    1736
explorer.exe   1864
qtask.exe      1308
fpdisp4.exe    824
fppdis1.exe    1952
rundll32.exe   892
NAVAPW32.EXE   1976
Directed.exe   1984
ctfmon.exe     1904
CLICKER.EXE    1628
Proxomitron.exe 472
taskmgr.exe    216
OUTLOOK.EXE    1576
agentsvr.exe   592
explorer.exe   3432
wmplayer.exe   2812
IEXPLORE.EXE   3740
cmd.exe        2720
LEMMY.EXE      3016
cmd.exe        3528
wmiprvse.exe   3728
LEMMY.EXE      3220
POWERPNT.EXE   2528
cscript.exe    1012

C:\WMI>
```

```
var e= new Enumerator(GetObject("winmgmts:").
                        InstancesOf("Win32_LogicalMemoryConfiguration"));
WScript.Echo ("Total Physical Memory (kb): " + e.item().TotalPhysicalMemory);
```

JavaScript Shell Scripting...



```
sRegTypes = new Array(
    "REG_SZ", "REG_EXPAND_SZ", "REG_DWORD", "REG_LINK", "REG_RESOURCE_LIST", "REG_RESOURCE_REQUIREMENTS_LIST", "REG_BINARY", "REG_DWORD_BIG_ENDIAN", "REG_MULTI_SZ", "REG_FULL_RESOURCE_DESCRIPTOR", "REG_QWORD");
```

```
HKLM = 0x80000002;
sRegPath = "SYSTEM\\CurrentControlSet\\Services\\Eventlog\\System";
```

```
oLoc = new ActiveXObject("WbemScripting.SWbemLocator");
oSvc = oLoc.ConnectServer(null, "root\\default");
oReg = oSvc.Get("StdRegProv");
oMethod = oReg.Methods_.Item("EnumValues");
oInParam = oMethod.InParameters.SpawnInstance();
oInParam.hDefKey = HKLM;
oInParam.sSubKeyName = sRegPath;
oOutParam = oReg.ExecMethod_(oMethod.Name, oInParam);
aNames = oOutParam.sNames.toArray();
aTypes = oOutParam.Types.toArray();
```

```
for (i = 0; i < aNames.length; i++)
    WScript.Echo("Type: ", sRegTypes[aTypes[i]], " KeyName: ", aNames[i]);
```

```
oMethod = oReg.Methods_.Item("GetMultiStringValue");
oInParam = oMethod.InParameters.SpawnInstance();
oInParam.hDefKey = HKLM;
oInParam.sSubKeyName = sRegPath;
oInParam.sValueName = "Sources";
oOutParam = oReg.ExecMethod_(oMethod.Name, oInParam);
aNames = oOutParam.sValue.toArray();
```

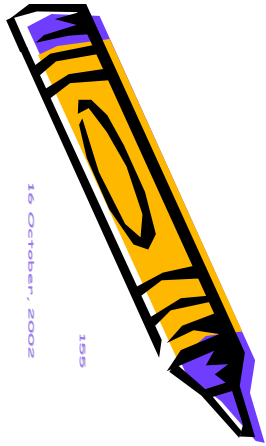
```
for (i = 0; i < aNames.length; i++)
    WScript.Echo("KeyName: ", aNames[i]);
```

```
function hex(nmb) {
    if (nmb > 0)
        return nmb.toString(16);
    else
        return (nmb + 0x100000000).toString(16);
}
```

display all keys under
HKLM\SYSTEM\CurrentControlSet\
Services\Eventlog\System

```
C:\WINDOWS\System32\cmd.exe
C:\Documents and Settings\Bob\Desktop\New Folder>escript uni.js
Microsoft (R) Windows Script Host Version 5.6
Copyright (C) Microsoft Corporation 1996-2001. All rights reserved.

Type: REG_EXPAND_SZ      KeyName: DisplayNameFile
Type: REG_DWORD          KeyName: DisplayNameID
Type: REG_EXPAND_SZ      KeyName: File
Type: REG_DWORD          KeyName: MaxSize
Type: REG_SZ             KeyName: PrimaryModule
Type: REG_DWORD          KeyName: Retention
Type: REG_MULTI_SZ       KeyName: Sources
Type: REG_DWORD          KeyName: RestrictGuestAccess
Type: REG_SZ             KeyName: EventMessageFile
Type: REG_BINARY         KeyName: TypesSupported
KeyName: WZCSUC
KeyName: Workstation
KeyName: Windows Script Host
KeyName: Windows File Protection
KeyName: Win32k
KeyName: W32Time
KeyName: UsbSnap
KeyName: viaide
KeyName: UgaSave
KeyName: USER32
KeyName: URS
KeyName: ultra
KeyName: udfs
KeyName: tosido
KeyName: TermServSessDir
KeyName: TermService
KeyName: TermServDevices
KeyName: TermDD
KeyName: tdi
KeyName: TCPMon
KeyName: Tcpip
KeyName: System Error
KeyName: sym_u3
KeyName: sym_h1
KeyName: sym8xx
KeyName: sym810
KeyName: StillImage
KeyName: SSDPSRV
KeyName: Srv
KeyName: srsservice
```

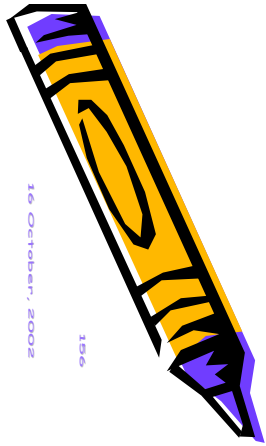


Rhino

- Open-source implementation of JavaScript written entirely in Java
 - Intended to be embedded into Java applications to provide a scripting facility to end users
 - Rhino 1.5 implements JavaScript 1.5
 - Contains the *jsc* JavaScript compiler

```
public class RunScript {
    public static void main(String [] args)
        throws JavaScriptException
    {
        Context cx = Context.enter();
        try {
            Scriptable scope = cx.initStandardObjects(null);
            String s = "";
            for (int i=0; i < args.length; i++)
                s += args[i] + " ";
            Object result = cx.evaluateString(scope, s, "<cmd>", 1, null);
            System.err.println(cx.toString(result));
        }
        finally {
            Context.exit();
        }
    }
}
```





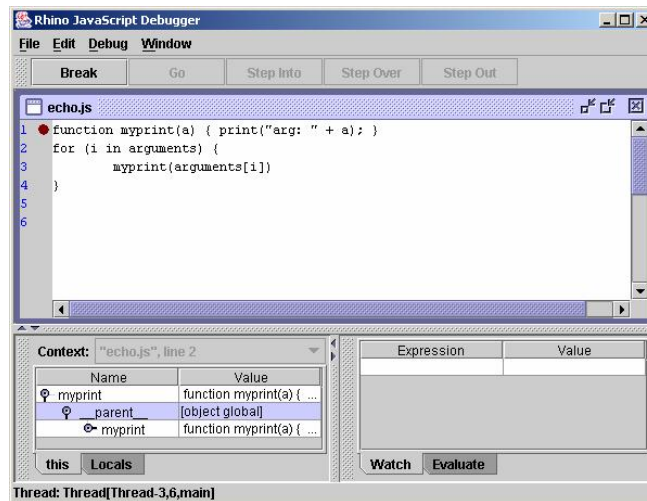
Rhino...

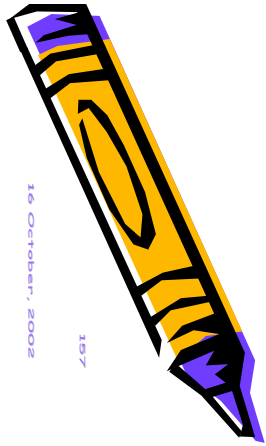
- Interactive mode makes it easy to do interactive program development

```
for (i in arguments) {  
    print(arguments[i])  
}
```

```
C:\WINDOWS\System32\cmd.exe  
C:\rhino1_5R3>java -cp js.jar org.mozilla.javascript.tools.shell.Main echo.js on  
e two three  
one  
two  
three  
C:\rhino1_5R3>_
```

- Has a (quite primitive) debugger





Rhino...

- Interacts very well with Java
 - Illustrates Netscape's LiveConnect idea
 - Also found in the Netscape Browser
 - *Almost* a Java interpreter...(?)

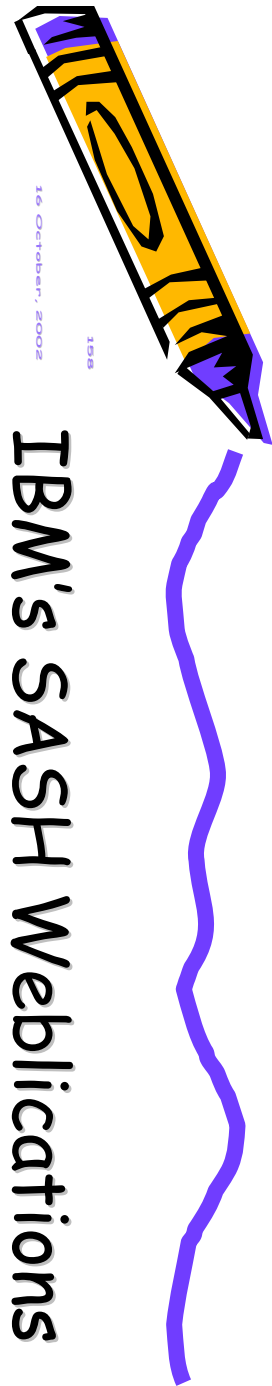
The screenshot shows a Java IDE with a file named 'C:\Documents and Settings\Bob\Desktop\rj.txt'. The code in the editor is:

```
importPackage(java.awt);
frame = new Frame("Rhino/Java Test");
frame.show();
frame.setSize(new Dimension(200,100));
button = new Button("Hello from Rhino");
frame.add(button);
frame.show();
```

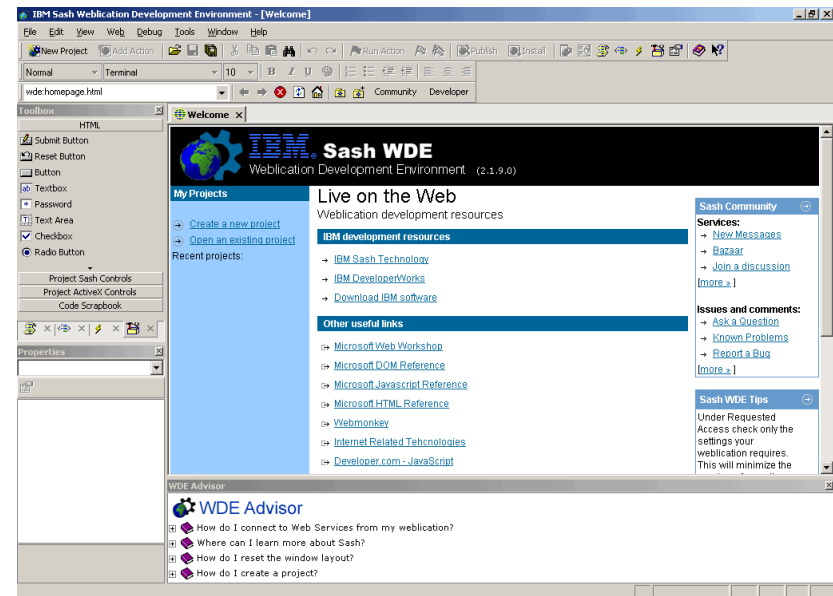
Below the code editor is a small window titled 'Rhino/Java Test' with the text 'Hello from Rhino'.

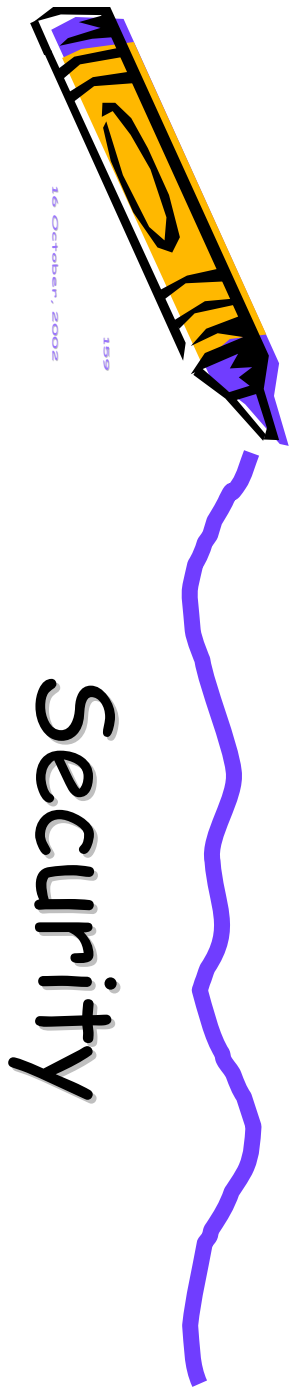
To the right of the IDE is a command prompt window titled 'C:\WINDOWS\System32\cmd.exe - java -jar js.jar'. It shows the following commands and output:

```
C:\RHINO1~1>java -jar js.jar
Rhino 1.5 release 3 2002 01 27
js> importPackage(java.awt);
js> frame = new Frame("Rhino/Java Test");
js> frame.show();
js> frame.setSize(new Dimension(200,100));
js> button = new Button("Hello from Rhino");
js> frame.add(button);
js> java.awt.Frame[frame0,0,0,0x0,invalid,hidden,layout=java.awt.BorderLayout,title=Rhino/Java Test,resizable,normal]
js> js> js> java.awt.Button[button0,0,0,0x0,invalid,label=Hello from Rhino]
js> java.awt.Button[button0,0,0,0x0,invalid,label=Hello from Rhino]
js> frame.show();
js>
```



- *AlphaWorks* project
 - A way to build feature-rich client applications on top of an existing Web back-end infrastructure
 - a dynamically configurable programming environment for building and deploying platform-integrated desktop applications using JavaScript and DHTML





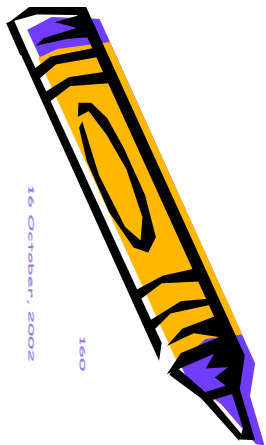
- Various facets
 - Security holes, etc.
 - See AusCERT, www.aucert.org.au
 - Signed scripts
 - I.P. leaks
 - "JMyth" Code Encryption Utility
 - Microsoft script encoder
- Many SOEs mandate that JavaScript be turned off

"JavaScript has a troubling history of security holes, and is known to be vulnerable to privacy disclosure. These holes are numerous, and different for each version of web browsers. And although many of them have been caught, new ones are appearing at a steady rate."

"A surprisingly large portion of the entire infrastructure must be trustworthy, including pieces you might not have realized were critical."

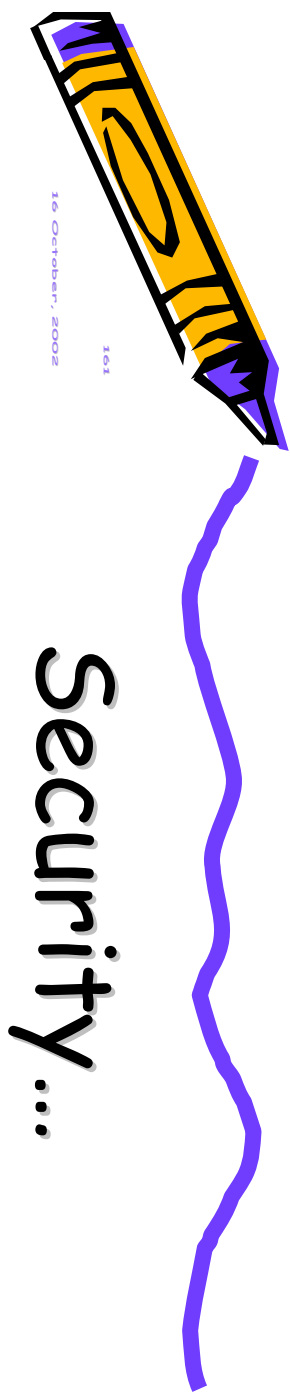
"JavaScript's first line of defense against malicious code is that the language simply doesn't support certain capabilities. For example, client-side JavaScript does not provide any way to read, write, create, delete, or list files or directories on the client computer. Since there is no File object, and no file access functions, a JavaScript program obviously cannot delete a user's data, or plant viruses on the user's system, for example."

Security...



```
<HTML>
<HEAD>
<TITLE>Script Encoder Sample Page</TITLE>
<SCRIPT LANGUAGE="JScript">
<!--//
//Copyright© 1998 Microsoft Corporation. All Rights Reserved.
/**Start Encode**
function verifyCorrectBrowser(){
    if(navigator.appName == "Microsoft Internet Explorer")
        if (navigator.appVersion.indexOf ("5.") >= 0)
            return(true);
        else
            return(false);
}
function getAppropriatePage(){
    var str1 = "Had this been an actual Web site, a page compatible with ";
    var str2 = "browsers other than ";
    var str3 = "Microsoft Internet Explorer 5.0 ";
    var str4 = "would have been loaded.";
    if (verifyCorrectBrowser())
        document.write(str1 + str3 + str4);
    else
        document.write(str1 + str2 + str3 + str4);
}
//-->
</SCRIPT>
</HEAD>
<BODY onload="getAppropriatePage()">
</BODY>
</HTML>
```

```
<HTML>
<HEAD>
<TITLE>Script Encoder Sample Page</TITLE>
<SCRIPT LANGUAGE="JScript.Encode">
<!--//
//Copyright© 1998 Microsoft Corporation. All
Rights Reserved.
/**Start
Encode***@~^QwIAAA==@#&Q;mDkWp7nDbOzZKD.n1YAMG
hk+Dvb`@#&P;kW`UC7kL1DGdc122gl:n~{ '~Jtr1DGkWbY
P&xDnD+0PA62sKD+ME#@#&P;~~kBPvxC\rLmYGDcCwa.n.
kkWU
bxI+X6bPcr*cJ#@*{~!*P~P,P~. YEMU`DDE bIP,P+s
/n@#&P~P;~PM+0;Mx`WC^/n#pN6EU1YbWx;o 0baw.WaDr
CD)+nmL+v#@#&~P71MPdY.q;'~J_CN;Y4rkP4nnPCx;C1Y;
mV; +(PkrY ~~1;wCL PmKhwmYk(snPSkDt~Ji@#&P~\
m.PkY.+;'PE&MWA/ .kPGDt DPDtmUPri@#&P-
CMP/D.&;'Pr\rmMwkwWY~(YnDnY;2a2^Wdn.;*
!;Ep@#&P71D;/D.c;'~JSW;S9Ptm-
+;4+ U~VK19+EREI;PrD;c\ DrWHZW.. m0AMGS/nM`*#
@#&P;~PqW^Es+U0chDb0+v/YMq~_/DDfPQ~kY.c*IP;+s
d @#&~~;PcW1;s+UDRSdkD+vdYMF~_/0.yP_1dYM&P3~d
YMc*iNz&R @*^#~@
//-->
</SCRIPT>
</HEAD>
<BODY onload="getAppropriatePage()">
</BODY>
</HTML>
```

- Browsers treat JavaScript security differently
 - IE has concept of 'zones' and can (dis)allow scripts based on origin
 - NS has two basic policies
 - Same origin policy: when loading a document from one origin, a script loaded from a different origin cannot get or set certain predefined properties of certain browser and HTML objects in a window or frame
 - Signed script policy: relies on LiveConnect and the standard Java SecurityManager class
 - Gives fine-grained control over activities
 - NS also experimented with "Data Tainting"
 - Developer explicitly marks entities that can be manipulated by JavaScript
 - Only in JavaScript 1.1