Adding the Bahavioral to your Web Site

Interactivity added using scripting languages - JavaScript

JavaScript

- Scripting languages add functionality – e.g. popup windows, form validation
- Client-side scripting language – interpreted language (i.e not compiled) – no software purchase needed.
- Note: JavaScript has nothing to do with the Java programming language
- Other scripting languages (only work on Internet Explorer)
  - Jscript
  - VBScript (based on Visual Basic)

What JavaScript does

- Clocks
- Mouse trailers
- Drop down menus
- Alert messages
- Popup windows
- HTML form validation
- …..etc

Implementing JavaScript

- Scripts embedded in HTML document or in separate file (as with CSS). File containing scripts is text file with suffix `.js` – created using text editor
- `<script> …..</script>` elements establishes script block in HTML file
- A document can have as many scripts as you like
- To use JavaScript:

```
<script language="javascript" type="text/javascript">
<!--
```

Scripts go here

```
//-->
</script>
```

Note: HTML comment hides Javascript within <script> tags

Where to put JavaScript

- JavaScript is a sequence of statements to be executed by the browser
- Scripts in external file or inserted in the `<head>` execute when called
- Scripts inserted in `<body>` execute when page loads – generate content
- To call external file:

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

JavaScript Syntax

- End of line of code implies end of statement
- Two statements on one line separated by a semi-colon.
- Useful to always end a statement with semi-colon
- Examples:

```
first statement
second statement

first statement; second statement;

first statement;
second statement;
```

JavaScript Syntax

- JavaScript is case sensitive
  - `Myvar` is different from `myvar` and different from `MYVAR or myVar`
- Comments:
  - Single line comments

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

  - Multi-line comments

```
/* This is a
   longer comment */
```

JavaScript Syntax

- Variables: a variable is a container – can contain any value assigned to it, as in algebra
  - Variable names contain any letter or digit 0 – 9 or underscore, no spaces, no punctuation; first character cannot be a digit
  - Multi-word names often written with underscore or in 'camelCase'
  - E.g. `my_var` or `myVat`
- Variables declared (defined) using keyword `var`
  - E.g. `var myVar;`
- Values may be assigned to variables at declaration
  - E.g. `var myVar=10;`

Data Types

- Variable types
  - Scalar (one value at a time)
  - Array (multiple values)
- Strings:
  - Text information, enclosed in single or double quotes, contain characters. E.g.
  - `var my_string = 'here is my text';`
  - `var my_string = "here is my text";`

Data Types

- Numbers (self-explanatory!)
  - `var my_age = 21;`
  - `var negNum = -1.9553;`
- Booleans(true/false values)

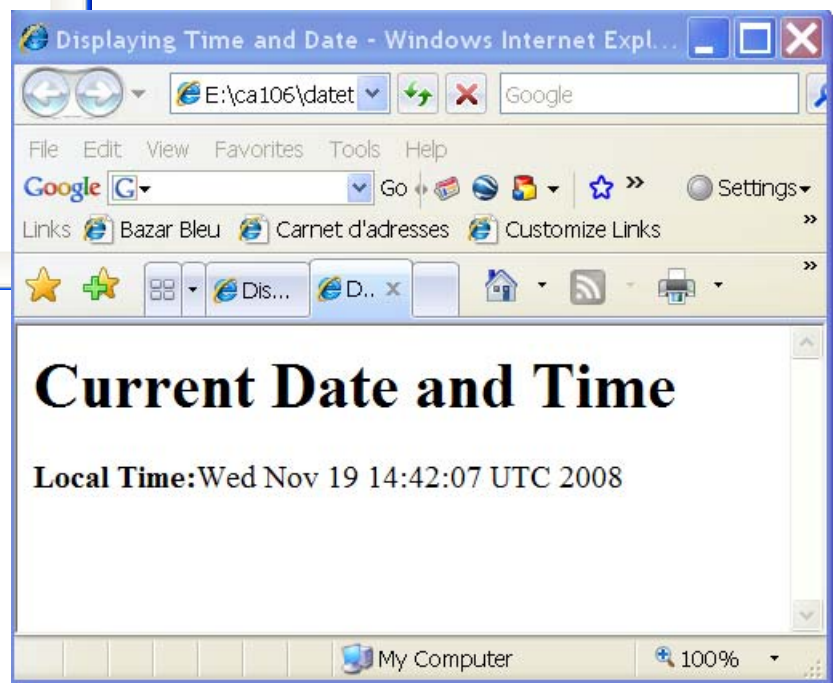
```
var wearglasses = true; // I wear glasses
var bearded = false; //I don't have a beard!
```

Script to display time and date

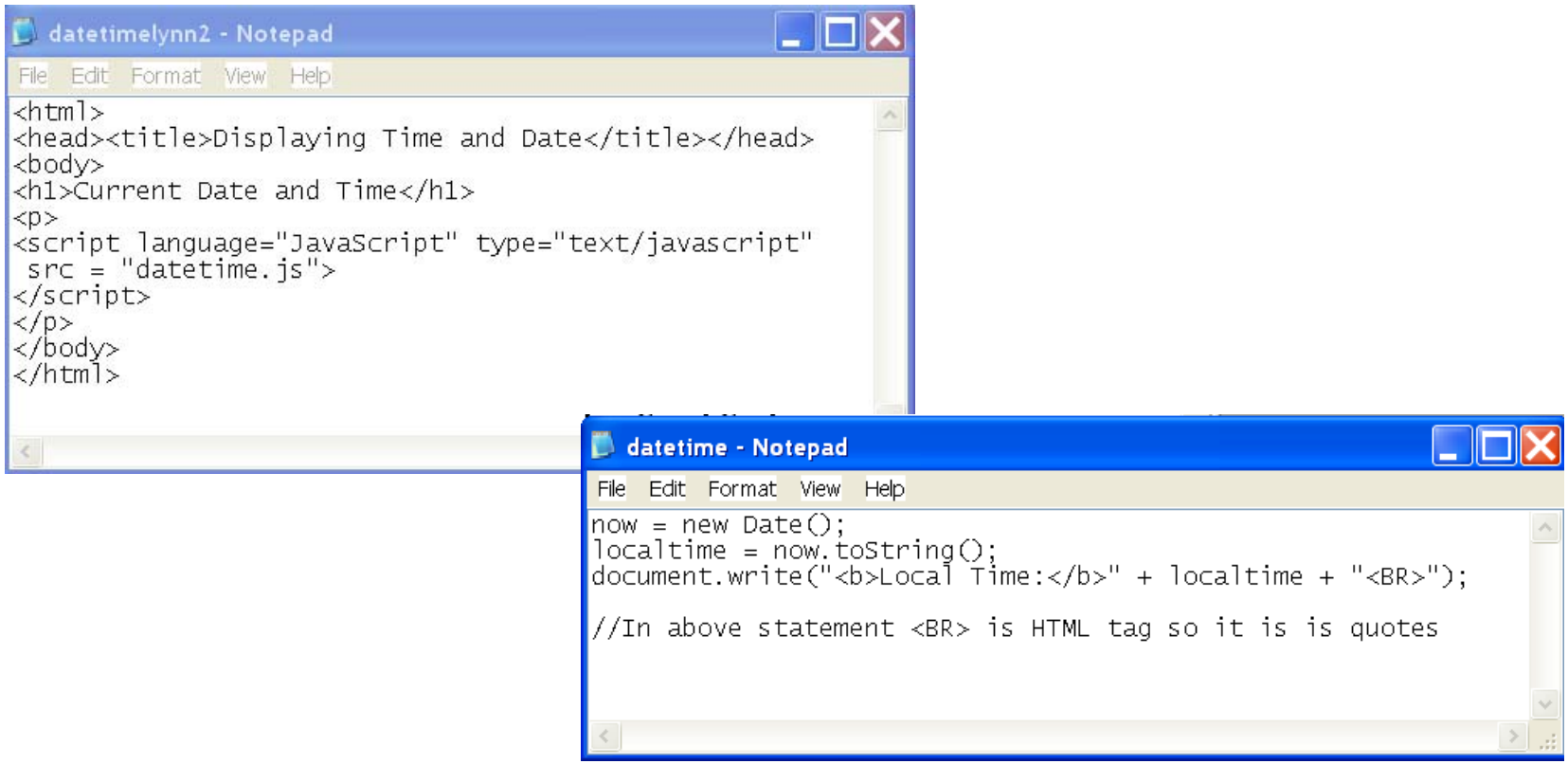
datetimelynn11 - Notepad

File Edit Format View Help

```
<html>
<head><title>Displaying Time and Date</title></head>
<body>
<h1>Current Date and Time</h1>
<p>
<script language="JavaScript">
now = new Date();
localtime = now.toString();
document.write("<b>Local Time:</b>" + localtime + "<BR>");

//In above statement <BR> is HTML tag so it is is quotes

</script>
</p>
</body>
</html>
```



Displaying date and time – script in external file

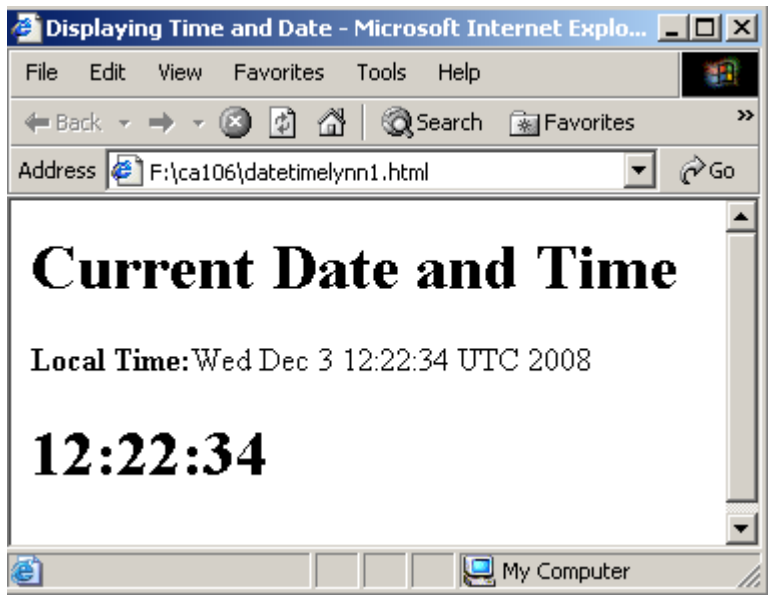


Script to display time and date also as heading

datetimelynn1 - Notepad

File Edit Format Help

```
<html>
<head><title>Displaying Time and Date</title></head>
<body>
<h1>Current Date and Time</h1>
<p>
<script language="JavaScript">
now = new Date();
localtime = now.toString();
document.write("<b>Local Time:</b>" + localtime + "<BR>");
hours = now.getHours();
mins = now.getMinutes();
secs = now.getSeconds();
document.write("<h1>");
document.write(hours + ":" + mins + ":" + secs);
document.write("</h1>");
</script>
</p>
</body>
</html>
```



Arrays

- Used to group multiple values (members) in a single variable
- Declaring arrays

```
var array_1 = new Array(); //empty array

var array_2 = new Array(2); //array with two
  undefined members

var array_3 = []; //shorthand for empty array

var array_1[0] = 100; //assigns value of 100 to
  first element in array_1
```

Objects

- Objects store data – data items called 'properties'
  - e.g. object: address book; properties for each person: name, address, phone number etc.
- Period used to separate object names and property names
  - E.g. `lynn.address` and `lynn.phone`
- Objects can have methods

  E.g. `lynn.display` could display Lynn's details
- `document.write` → `write` is method of `document` object

JavaScript Objects

- Built-in objects e.g. date, string
- DOM (Document Object Model) objects – represent components of browser and current HTML document
- Custom objects – objects you create yourself

Common JavaScript Operators

Operator	***Example***
+ concantenate	`"this is" + " a test";`
+ add	`result = 5 + 7;`
- subtract	`score = 100 - 10`
* multiply	`total = quantity * price;`
/ divide	`average = sum/4;`
% modulo (remainder)	`remainder = sum % 4;`
++ increment	`counter++;`
-- decrement	`total--;`

Assignment statements

- Example:

  `x = y + z` → x takes on the value of (y + z)

- But could have:

  `j = j + 1` → j is incremented by 1

  - as in Java, C++ etc.

Popup Boxes

- Alert box – user clicks “OK” to proceed

```
alert(“sometext”)
```

- Confirm box – user clicks “OK” or “Cancel” to proceed – OK returns true, Cancel returns false.

```
confirm(“sometext”)
```

- Prompt box – used when input from user required – user enters a value and clicks “OK” or “Cancel” – OK returns input value, Cancel returns null.

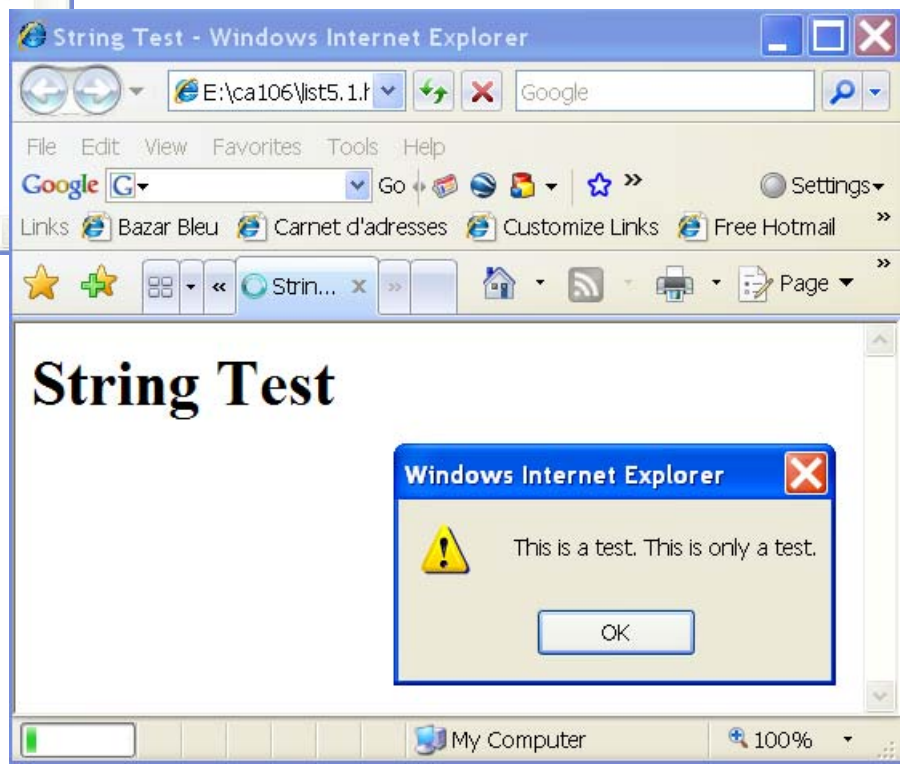
```
prompt(“sometext”, “defaultvalue”)
```

Alert box – click OK to continue

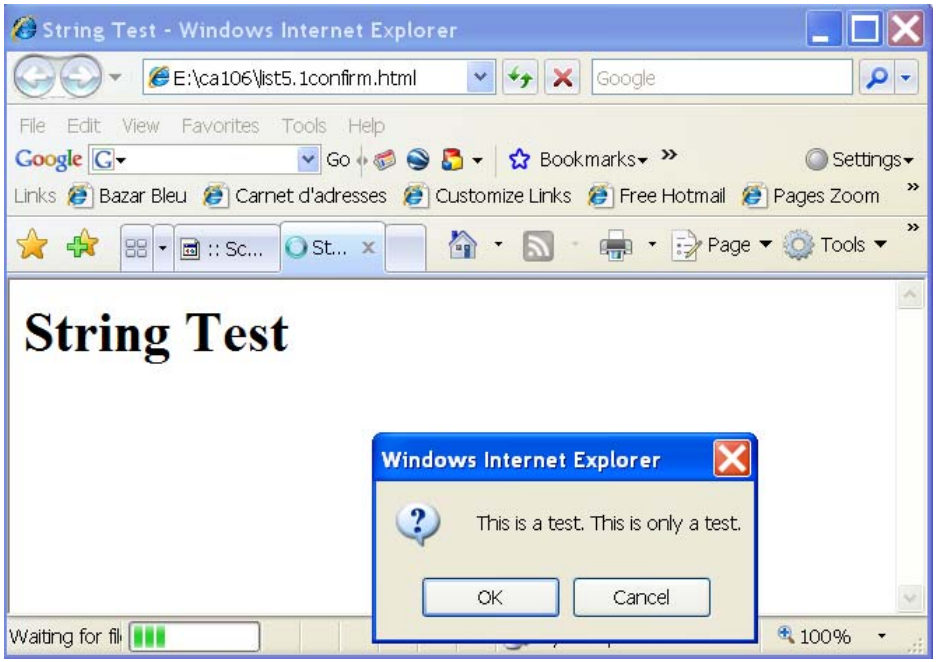
list5.1 - Notepad

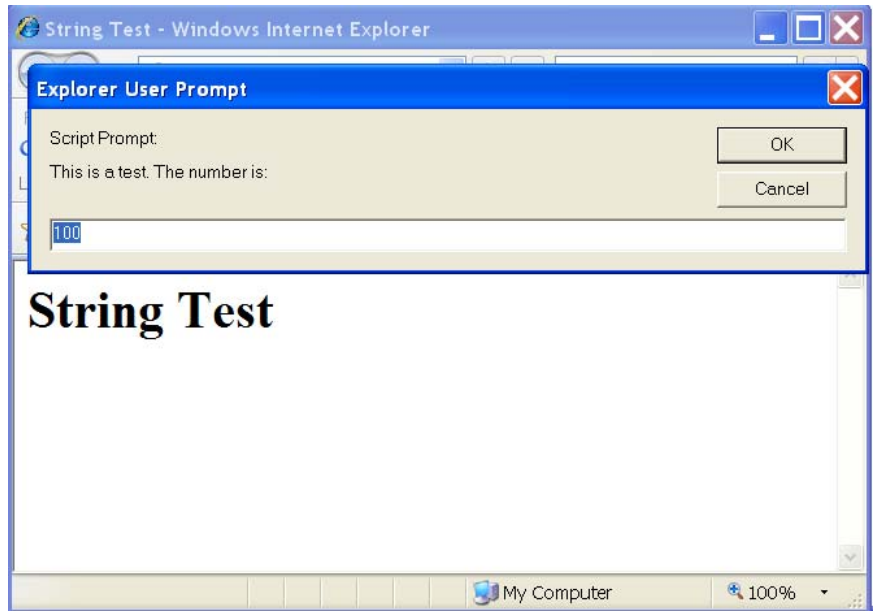
File Edit Format View Help

```
<html>
<head>
<title>String Test</title>
</head>
<body>
<h1>String Test</h1>
<script language="JavaScript" type="text/javascript">;
test1 = "This is a test. ";
test2 = "This is only a test.";
both = test1 + test2;
alert(both);
</script>
</body>
</html>
```



Confirm and Prompt box






```
<h1>String Test</h1>
<script language="JavaScript" type="text/javascript">;
test1 = "This is a test. ";
test2 = "The number is: ";
both = test1 + test2;
prompt(both, 100);
```

JavaScript Functions

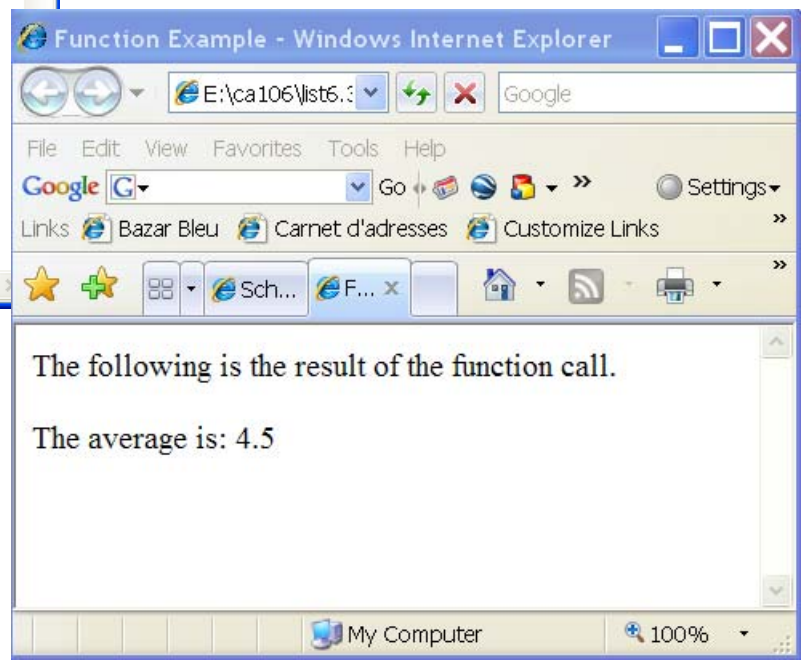
- Contain code executed when called
- Functions should be defined in `<head>` section
- Example: create function to find average of numbers, call it as required

Function which returns a value – input parameters *a,b,c,d*, return value = *result*

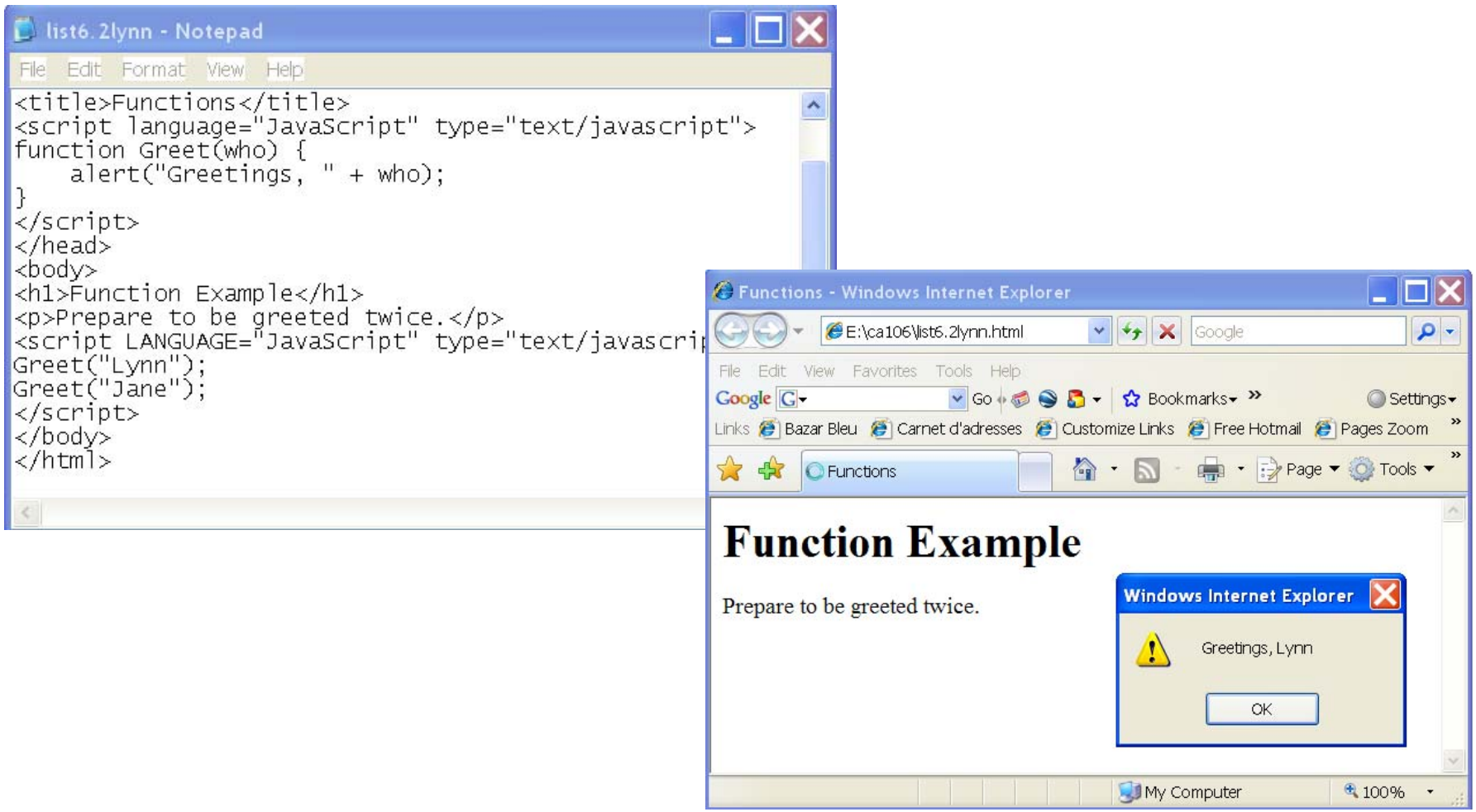
list6.3 - Notepad

File Edit Format View Help

```
<html>
<head>
<title>Function Example</title>
<script LANGUAGE="JavaScript" type="text/javascript">
function Average(a,b,c,d) {
result = (a + b + c + d) / 4;
return result;
}
</script>
</head>
<body>
<p>The following is the result of the function call.</p>
<script language="JavaScript" type="text/javascript">
score = Average(3,4,5,6);
document.write("The average is: " + score);
</script>
</body>
</html>
```



Function – note input parameter 'who' is given values *Lynn* and *Jane*



Conditional operators

- **==** is equal (idenical) to
- **!=** is not equal(identical) to
- **<** is less than
- **>** is greater than
- **<=** is less than or equal to
- **>=** is greater than or equal to

Controlling Flow - Conditions

- Testing and comparing values – behaviour of scripts depends on result
- Example: *If the phone rings, answer it.*
- Syntax:

```
if (condition)
{
code to be executed
}
```

Example:

```
if(a == 1){
window.alert("Found a 1!");
}
```

Note: `{ ..... ....}` used to group statements into a block

Logical (Boolean) Operators

- And - &&
- Or - ||
- Not - !

```
if ((a == 1) || (a == 2)){alert("a less than 3");}

if ((a == 1) && (b == 1)) {alert ("a and b = 1");}
```

if …. else ….

- Syntax:

```
if (condition)
{code executed if condition true
}
else
{code executed if condition false
}
```

- Example:

```
if (time < 12)
{
document.write (“Good morning”);
}
else
document.write(“Good day”);
}
```

if….else if…..else

- Syntax:

```
if (condition1)
{code executed if condition1 true
}
else if (condition2)
{code executed if condition2 true
}
else
{code executed if condition1 and
  condition2 not true
}
```

Example – `if` statements

list7.1 - Notepad

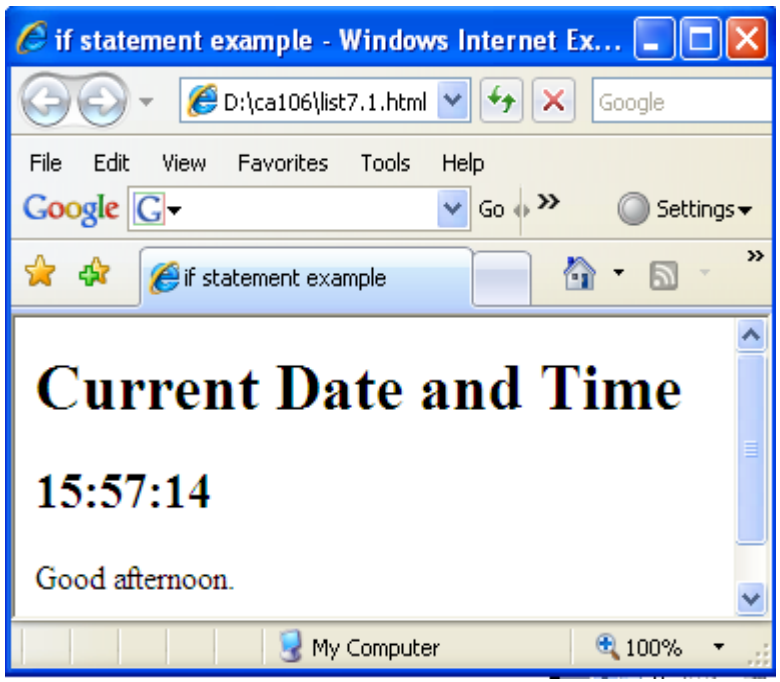
File Edit Format View Help

```
<html>
<head><title>if statement example</title></head>
<body>
<h1>Current Date and Time</h1>
<p>
<script language="JavaScript" type="text/javascript"
 src = "timegreet.js">
</script>
</p>
</body>
</html>
```

timegreet - Notepad

File Edit Format Help

```
// Get the current date
now = new Date();

// Split into hours, minutes, seconds
hours = now.getHours();

//hours = 9;
mins = now.getMinutes();
secs = now.getSeconds();

// Display the time in large <h2> type - note use of quotes for HTML tags
document.write("<h2>");
document.write(hours + ":" + mins + ":" + secs);
document.write("</h2>");

// Display a greeting
document.write("<p>");
if (hours < 10) document.write("Good morning.");
else if (hours >= 14 && hours <= 17) document.write("Good afternoon.");
else if (hours > 17) document.write("Good evening.");
else document.write("Good day.");
document.write("</p>");
```



`switch` – select one of several blocks to be executed

- Syntax:

```
switch(n)
{case 1:
   execute code block 1
   break;
case 2:
   execute code block 2
   break;
default:
   code executed if n is different from case 1 or case 2
}
```

Note: There can be as many cases as required. Also note `break` causes execution to skip out of switch.

if….else if… or switch

```
if(height > 6){
alert("Tall");
} else if (height > 5.5){
alert("Average height");
}else{
alert("Shorter than
   average");
}
```

```
switch(true){
   case (height > 6):
alert("Tall");
break;
case ( height > 5.5)
alert("Average height");
break;
default:
alert("Shorter than
   average")
break;
}
```

JavaScript Loops

- May need to repeat execution of same code
- **`for`** loop:

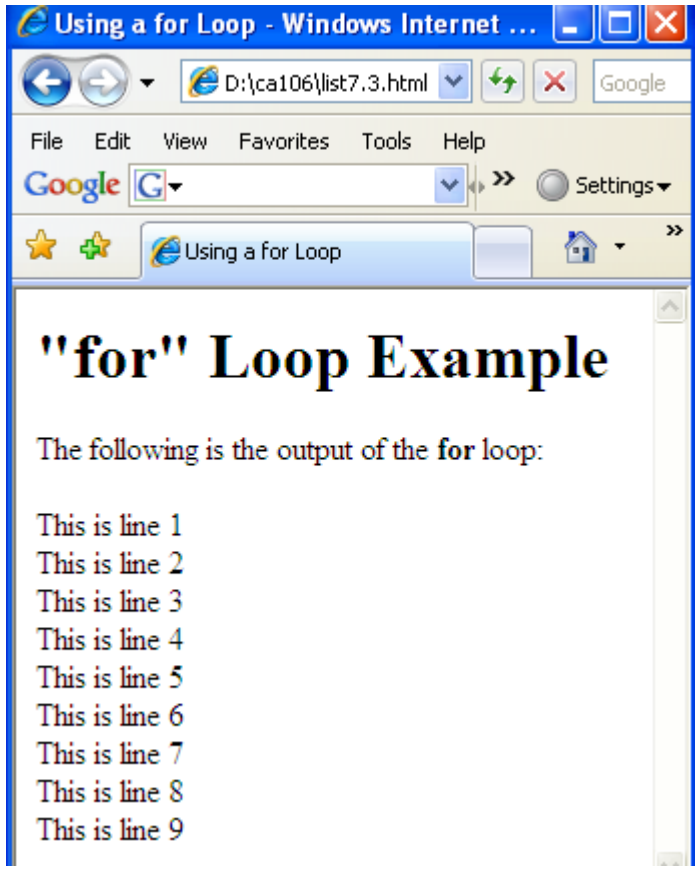
```
for (var=startvalue; var<=endvalue; var =
  var + increment)
{
code executed
}
```

`for` loop

list7.3 - Notepad

File Edit Format View Help

```
<html>
<head>
<title>Using a for Loop</title>
</head>
<body>
<h1>"for" Loop Example</h1>
<p>The following is the output of the
<b>for</b> loop:</p>
<script language="JavaScript" type="text/javascript">
for (i=1;i<10;i++) {
   document.write("This is line " + i + "<br>");
}
</script>
</body>
```



Value of i starts at 1 and increments by 1 on each pass through the loop (i++, can also be written i = i+1), as long as i < 10

JavaScript Loops

- **`while`** loop:

```
while (var <= endvalue)
{
code executed
}
```

- **`do`**….**`while`** loop:

```
do
{
code executed
}
while (var <= endvalue);
```

Note: **`do`**….**`while`** loop is always executes code block at least once.

`do while` loop

list7.4 - Notepad

File Edit Format View Help

```
<html>
<head>
<title>Loops Example</title>
</head>
<body>
<h1>Loop Example</h1>
<p>Enter a series of names. I will then
display them in a nifty numbered list.</p>
<script language="JavaScript" type="text/javascript"
src="loops.js">
</script>
</body>
</html>
```

loops - Notepad

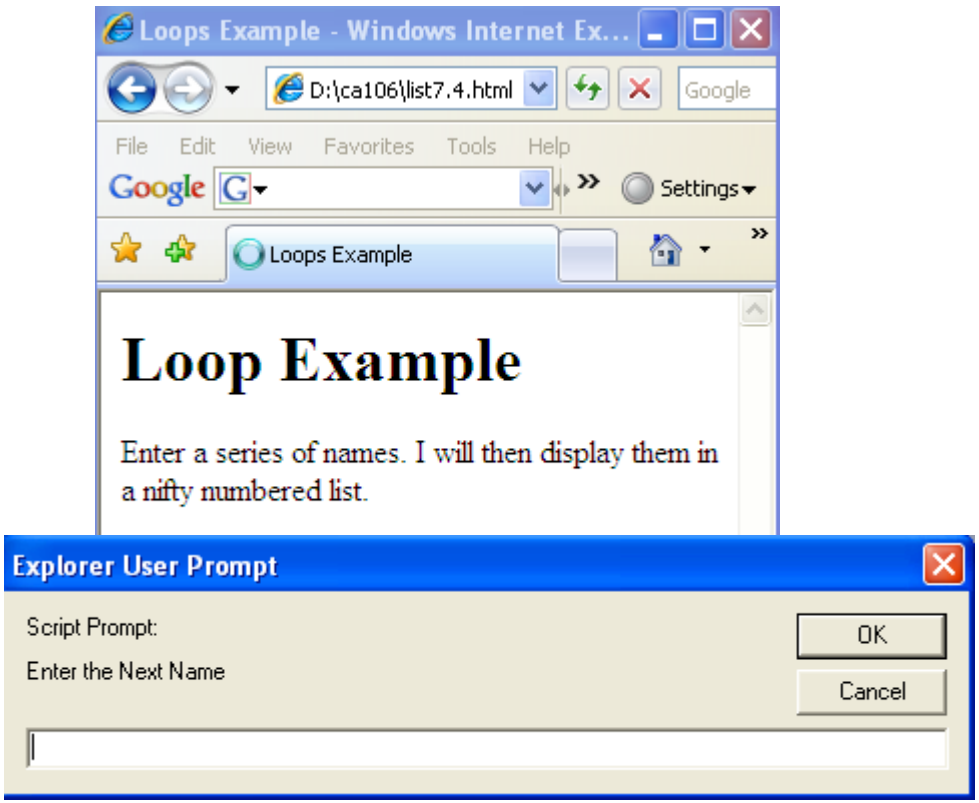
File Edit Format Help

```
// create the array for the names
names = new Array();
i = 0;

// loop and prompt for names
do {
    next = window.prompt("Enter the Next Name", "");
    if (next > " ") names[i] = next;
    i = i + 1;
   } while (next > " ");
document.write("<h2>" + (names.length) + " names entered.</h2>");

// display all of the names
document.write("<ol>");
for (i in names) {
    document.write("<li>" + names[i] + "<br>");
}
document.write("</ol>");
```

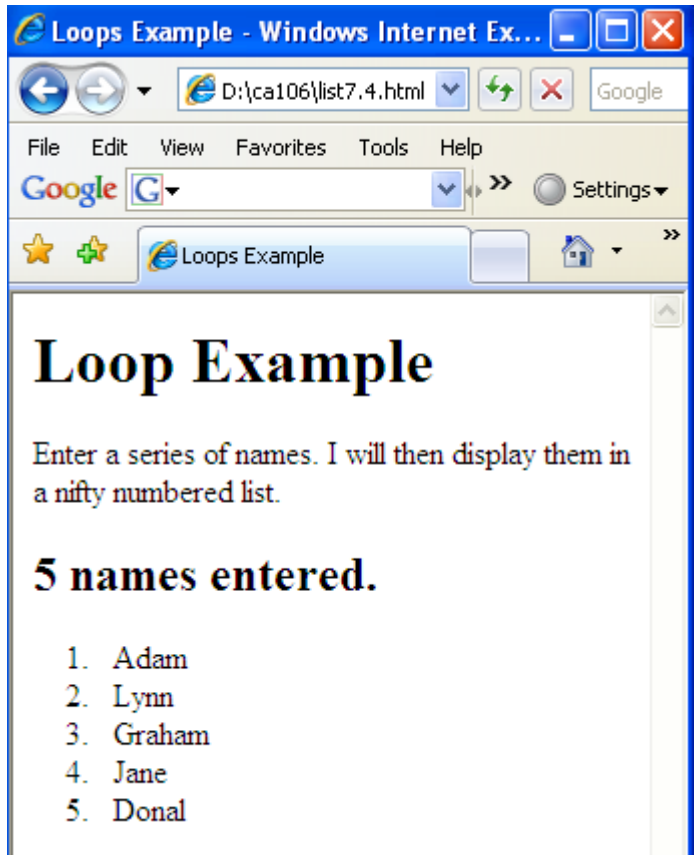
Note – Names are stored in array called `names`.

`.length` property = number items in array.

`do while` example - view



After five names have been entered



for …..in

- Used to iterate through elements in an array or through properties of an object
- Syntax:

```
for (variable in object)
{
code executed
}
```

continue and **break**

- **continue** breaks current loop and continues with next value

```
for (j=0; j<=5; j++)
{ if (j==3)
{
continue;
}
Output statements
}
```

This will output 0, 1, 2, 4 and 5

- **break** breaks the loop and continues executing the code that follows the loop

```
for (j=0; j<=5; j++)
{ if (j==3)
{
break;
}
Output statements
}
```

This will output 0, 1 and 2

Event Handling

- Events – things that happen to the browser – e.g. user clicks a button, mouse pointer moves, web page or image loads.
- Event handler – script used to detect and respond to events.
- Event handlers have ‘on’ before event name
- Good website to see how event handlers work:

http://docs.sun.com/source/816-6408-10/handlers.htm

Objects and Events

- Using the DOM, JavaScript uses set of objects to store information about web page – buttons, links, images, windows etc.
- Events can happen in several places on page, so each event is associted with an object

Common Events (there are more!)

Event Handler	***Event***
`onBlur`	Element loses focus
`onChange`	Content of a field changes
`OnClick`	Mouse clicks an object
`onError`	Error occurs when loading page/image
`onFocus`	Element gets in focus
`onKeydown`	Key is pressed
`onKeypress`	Key is pressed or held down
`onKeyup`	Key is released
`onLoad`	Page/image finishes loading
`onMousedown`	Mouse button is pressed
`onMousemove`	Mouse is moved
`onMouseout`	Mouse is moved off element
`onMouseover`	Mouse is moved over and element
`onMouseup`	Mouse button is released
`onSubmit`	Submit button is clicked

Event Handling Examples

- Alert box appears when mouse is over link:

```
<a href = "http://www.dcu.ie/"
onMouseover="alert('Demo of mouseOver');return false">
</a>
```

- Alert box appears when link clicked:

```
<a href= "http://www.dcu.ie/"
onClick="alert('Go to the DCU site');">Click here</a>
```

- Alert box when page has loaded:

```
<body onLoad="alert('This page has finished
  loading')"; onUnload="alert('Bye Bye!')">
```

Event Handling Examples

- `onSubmit` used to validate form fields:

```
<form method = "post" action = "xxxxxx.html"
onSubmit="return checkForm()";
```

  - `checkForm` function will be called when submit button clicked. `checkForm` validates form and returns true or false – if true, form is submitted, otherwise submit cancelled.

```
<form onSubmit="return confirm('Are you sure?')action = "xxx"">
```

Event Handling Examples– Status Bar

statusbar - Notepad

File Edit Format View Help

```
<html>
<head>
<title> Status Bar Message</title>
<script type="text/javascript">
<!--
function statwords(message){
window.status = message;
}
//-->
</script>
</head>
<body>
<form>
<input type="button" Value = "Write status bar message" onclick="statwords('This is a message');">
<input type="button" value="Remove status bar message" onclick="statwords(' ');">
</form>
```