JavaScript client-side

JavaScript History

- Java was born as “LiveScript” at the beginning of the 94’s.
- Name changed into JavaScript (name owned by Netscape)
- Microsoft responds with Vbscript
- Microsoft introduces JScript (dialect of Javascript)
- A standard is defined: ECMAScript (ECMA-262, ISO-16262)

JavaScript Myths

- JavaScript is NOT simple
  - Simple tasks are indeed simple
- JavaScript is NOT Java

	Java	JavaScript
Browser Control	NO	YES
Networking	YES	NO
Graphics	YES	Partial

JavaScript is…

- Scripted (not compiled)
- Powerful
- Object-based
- Cross-Platform
- Client and Server

JavaScript allows…

- Dynamic Web Sites
- Dynamic HTML (DHTML)
- Interactive Pages/Forms
- Server-Side CGI Functionality
- Application Development

JavaScript can…

- Build Objects
- Use Events
- Enforce Security
- Embed or Componentize

Core JavaScript Base

- Syntax is C-like (C++-like, Java-like)
  case-sensitive, statements end with (optional) semicolon ;

  //comment /*comment*/

  operators (=,*,+,++,+=,!=,==,&&,…)
- Basic data types
  integer, floating point, strings (more later)
- Loosely typed variables (Basic-like) var x=3;

Core JavaScript Statements

- if (expression) {statements} else {statements}
- switch (expression) {
  case value: statements; break;
  …
  default: statements; break;
  }
- while (expression) {statements}
- do (expression) while {statements}
- for (initialize ; test ; increment) {statements}

JavaScript and HTML

- Between <SCRIPT> and </SCRIPT> tags
- Between <SERVER> and </SERVER> tags
- In a <SCRIPT SRC="url"></SCRIPT> tag

- In an event handler:
  - <INPUT TYPE="button" VALUE="Ok" onClick="js code">
  - <B onMouseOver="Jscode">hello<B>

Strings

a="foo"; b='tball'; a+b => football

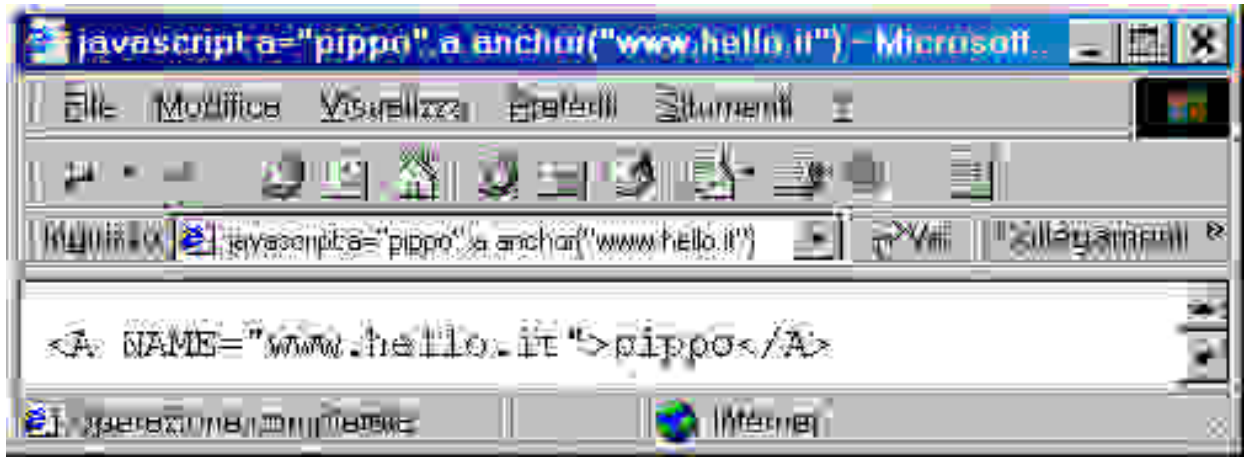
a<b => true; a.charAt(0) => f

Useful methods:

indexOf(substring), lastIndexOf(substring), charCodeAt(n), fromCharCode(value,…), concat(value,…), slice(start,end), toLowerCase(), toUpperCase(), replace(regexp,string), search(regexp)

Strings

- a=“foo”;
- TAG-related methods:
  - a.bold() => <B>foo</B>
  - big(), blink(), fontcolor(), fontsize(), small(), strike(), sup() , anchor(),link()



Functions

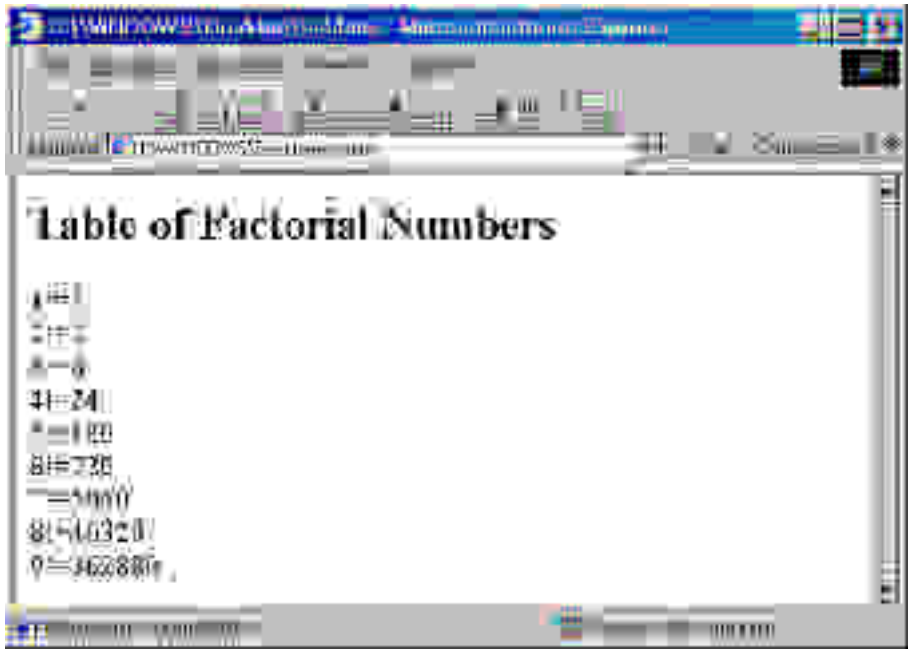
- function f(x) {return x*x}
- function add(x,y) {return x+y};
- function multiply(x,y) {return x*y};
- function operate(op,x,y) {return op(x,y)};
- operate(add,3,2); => 5

Example

```
<HTML>
<HEAD>
<SCRIPT>
function fact(n) {
    if (n==1) return n;
    return n*fact(n-1);
}
</SCRIPT>
</HEAD>
…
```

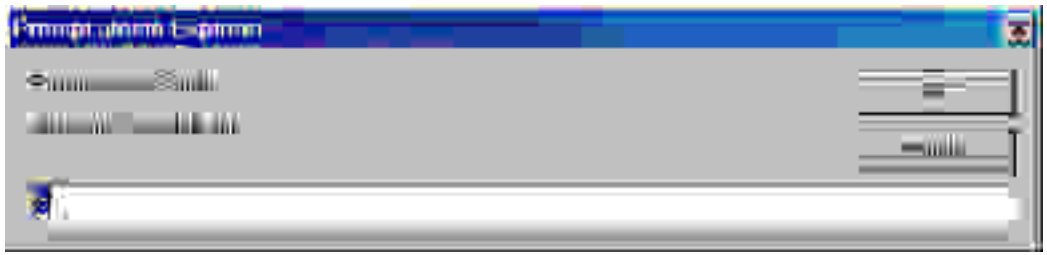
Example

```
<BODY>
<H2>Table of  Factorial Numbers </H2>
<SCRIPT>
for (i=1; i<10; i++) {
    document.write(i+"!="+fact(i));
    document.write("<BR>");
}
</SCRIPT>
</BODY>
</HTML>
```



Example

```
<BODY>
<SCRIPT>
n=window.prompt("Give me the value of n",3)
document.write("fact("+n+")="+fact(n));
document.write("<BR>");
</SCRIPT>
</BODY>
</HTML>
```



Objects

Object: A data structure with methods: a special method is the “constructor”.

```
function Rectangle(w, h) {
    this.width=w;
    this.height=h;
    this.area=function(){return this.w*this.h}
    }
```

a=new Rectangle(3,4); a.area() => 12

a.width => 3

Objects

- Actually, JavaScript does NOT have classes and inheritance.
- Moreover, the approach we have shown is not the most efficient in terms of memory allocation.
- It would be better to use the “prototype” feature.

```
Rectangle.prototype.area=function(){return
    this.w*this.h}
```

Arrays

a = new Array()

a[0]=3; a[1]="hello"; a[10]=new Rectangle(2,2);

a.length() => 11

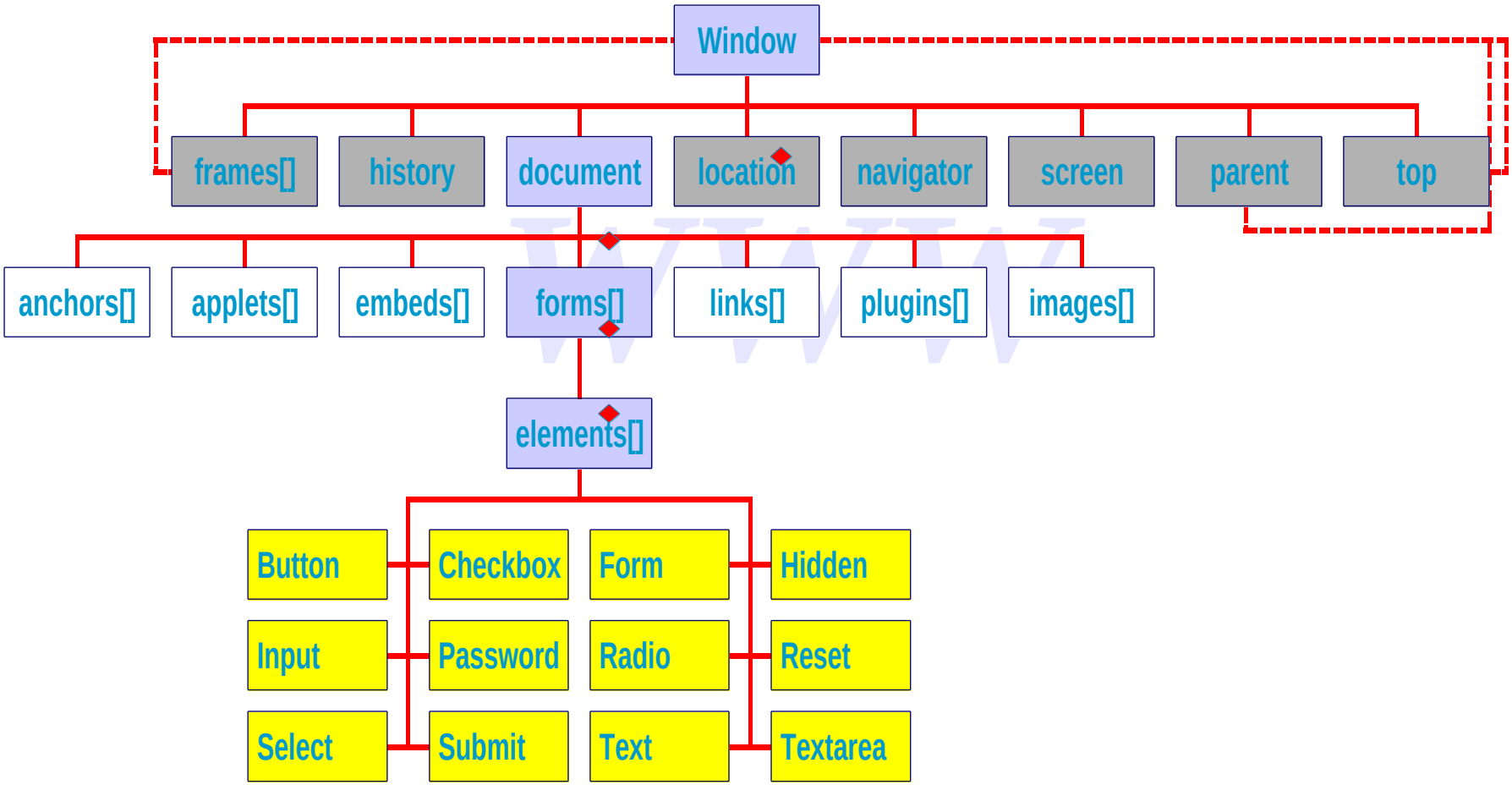
Arrays can be

SPARSE, INHOMOGENEOUS , ASSOCIATIVE

a["name"]="Jaric"

z=new Rectangle(3,4); z["width"] ⇔ z.width

Object hierarchy



Window

Main properties

Objects

- history
- frames[]
- document
- location
- navigator
- screen
- parent – top

Window

Main methods

- alert(), prompt(), confirm()
- focus(), blur()
- moveBy(), moveTo()
- resizeBy(), resizeTo()
- scroll(), scrollBy(), scrollTo()
- setInterval(), clearInterval()
- setTimeout(), clearTimeout()

Screen

Main properties:

- availHeight, availWidth
- height, width
- colorDepth, pixelDepth
- hash

Navigator

Main properties

- appName
- appVersion
- Platform

Main methods

- javaEnabled()

Other properties

Info on available plugins, but only in Netscape Navigator

History

Main properties

- lenght

Main methods

- back()
- forward()
- go(+/-n)
- go(target_substring)

Location

Main properties

- href
- protocol, hostname, port
- search
- hash

Main methods

- reload()
- replace()

Document

Main properties

Arrays of Component Objects

- anchors[]
- applets[]
- embeds[]
- forms[]
- links[]
- plugins[]

Document

Main methods

- open()
- close()
- clear()
- write()

Other properties

bgColor, fgColor, linkColor, vlinkColor,
lastModified, title, URL, referrer, cookie

Image

Main properties

- border [width in pixels]
- height
- width
- src [URL of the image to be displayed]

Events

onClick	User clicks once	Link, button
onDblClick	User clicks twice	Document, Image, Link, button
onMouseDown	User presses mouse button	Document, Image, Link, button
onMouseUp	User releases mouse button	Document, Image, Link, button
onMouseOver	Mouse moves over element	Link, Image, Layer
onMouseOut	Mouse moves off element	Link, Image, Layer
onKeyDown	User presses key	Document, Image, Link, Text elements
onKeyUp	User releases key	Document, Image, Link, Text elements
onKeyPress	KeyDown+KeyUp	Document, Image, Link, Text elements

Events

onFocus	Element gains focus	TextElement, Window, all form elements
onBlur	Element loses focus	TextElement, Window, all form elements
onChange	User selects/deselects a text and moves focus away	Select, text input elements
onError	Error while loading image	Image
onAbort	Loading interrupted	Image
onLoad	Document or image finishes loading	Window, Image
onUnload	Document is unloaded	Window
onResize	Window is resized	Window
onReset	Form reset requested	Form
onSubmit	Form submission requested	Form

Form

Main properties

- action [destination URL]
- method [get/post]
- name [name of Form]
- name [destination Window]
- Elements[] [list ;of contained elements]

Main methods

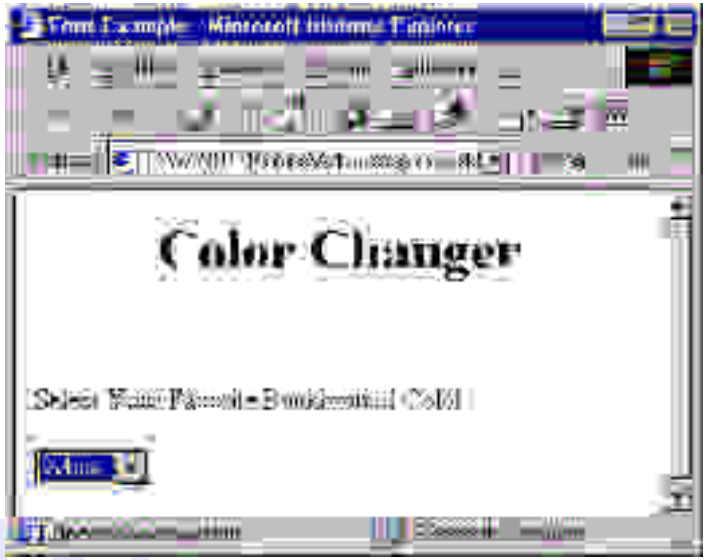
- reset()
- submit()

Input

Object	Propriety										Method				Event			
	checked	DefaultChecked	DefaultValue	form	length	name	Options[]	selectedIndex	type	value	Blur()	Click()	Focus()	Select()	onblur	onchange	onclick	onfocus
Button				X		X			X	X	X	X	X		X		X	X
Checkbox	X	X		X		X			X	X	X	X	X		X		X	X
Radio	X	X		X		X			X	X	X	X	X		X		X	X
Reset				X		X			X	X	X	X	X		X		X	X
Submit				X		X			X	X	X	X	X		X		X	X
Text			X	X		X			X	X	X		X	X	X	X		X
Textarea			X	X		X			X	X	X		X	X	X	X		X
Password			X	X		X			X	X	X		X	X	X	X		X
FileUpload			X	X		X			X	X	X		X	X	X	X		X
Select				X	X	X	X	X	X		X	X	X		X	X		X
Hidden				X		X			X	X								

Events

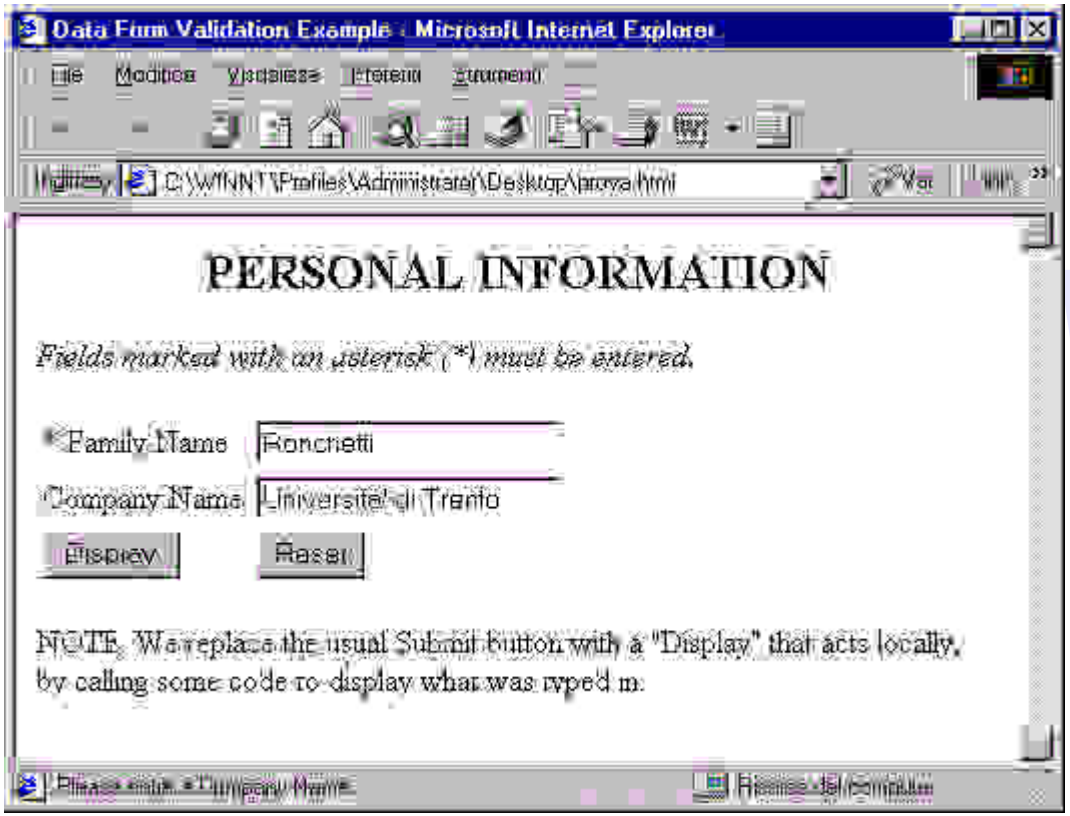
```
<HTML>
<HEAD>
<TITLE>Form Example</TITLE>
<SCRIPT LANGUAGE="JavaScript1.2">
function setColor() {
  var choice;
  choice = document.colorForm.color.selectedIndex;
  switch(choice) {
    case 0: document.bgColor = "FF0000"; break;
    case 1: document.bgColor = "00FF00"; break;
    case 2: document.bgColor = "0000FF"; break;
    case 3: document.bgColor = "FFFFFF"; break;
    case 4: document.bgColor = "FFFF00"; break;
    case 5: document.bgColor = "FF00FF"; break;
  }
}
</SCRIPT>
```



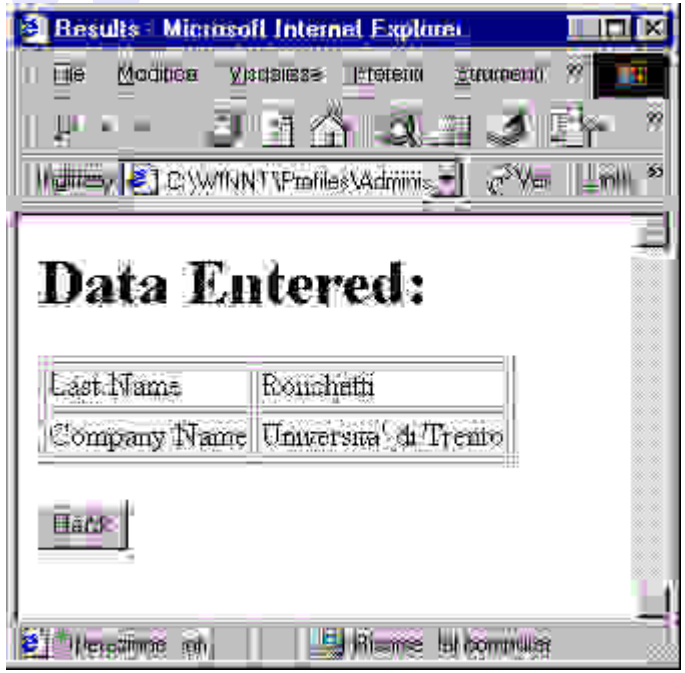
Events

```
<BODY>
<CENTER><H1>Color Changer</H1></CENTER>
<BR><BR>
Select Your Favorite Background Color:
<FORM NAME="colorForm">
<SELECT NAME="color" onChange=setColor()>
    <OPTION VALUE="red">Red
    <OPTION VALUE="green">Green
    <OPTION VALUE="blue">Blue
    <OPTION VALUE="white">White
    <OPTION VALUE="yellow">Yellow
    <OPTION VALUE="purple">Purple
</SELECT>
</FORM>
</BODY>
</HTML>
```

A more complex example



A simple data entry validation page



A more complex example

Start of file “FormValidation.html”


```
<HTML>
<HEAD>
<TITLE>Data Form Validation Example</TITLE>
<SCRIPT LANGUAGE="JavaScript1.1"
    SRC="FormCheck.js"></SCRIPT>
```

Load file “FormCheck.js”,
which contains several JavaScript functions

A more complex example

```
function isEmpty(s){
   return ((s == null) || (s.length == 0))
}

function warnEmpty (theField, s){
    var mPrefix = "You did not enter a value into the ";
    var mSuffix = " field. This is a required field. Please enter it now.";
    theField.focus();
    alert(mPrefix + s + mSuffix);
    return false;
}
```

All this is contained in the file “FormCheck.js”

A more complex example

```
function promptEntry (s){
   window.status = "Please enter a " + s;
}

function validatePersonalInfo(form){
   return (checkString(form.elements["LastName"],sLastName))
}

function checkString (theField, s)
{
    if (isEmpty(theField.value)) return warnEmpty (theField, s);
    else return true;
}
```

All this is contained in the file “FormCheck.js”

A more complex example

```
<SCRIPT>
var sCompany="Company Name";  var sLastName="Last Name";
var form="PersonalInfo";
function displayPersonalInfo(form){
    var outputTable =
    "<HTML><HEAD><TITLE>Results</TITLE></HEAD>" +
    "<BODY><H1>Data Entered:</H1><TABLE BORDER=1>" +
    "<TR><TD>" + sLastName + "</TD><TD>" +
    form.elements["LastName"].value + "</TD></TR>" +
    "<TR><TD>" + sCompany + "</TD><TD>" +
    form.elements["Company"].value +
    "</TD></TR></TABLE><FORM>“ +
    "<INPUT TYPE=\"BUTTON\" NAME=\"Back\" VALUE=\"Back\"
    onClick=\"history.back()\"></FORM></BODY></HTML>"
    document.writeln(outputTable)
    document.close()
    return true
}</SCRIPT></HEAD>
```

End of “HEAD” portion of “FormValidation.html”

A more complex example

```
<BODY BGCOLOR="#ffffff">
<CENTER><H2>PERSONAL INFORMATION </H2></CENTER>
<P><P><I>Fields marked with an asterisk (*) must be entered.</I>
<FORM NAME="PersonalInfo">
<TABLE>
<TR>
   <TD>* Family Name:</TD>
   <TD><INPUT TYPE="text" NAME="LastName"
    onFocus="promptEntry(sLastName)"
    onChange="checkString(this,sLastName)" ></TD>
</TR>
<TR>
   <TD>Company Name:</TD>
   <TD><INPUT TYPE="text" NAME="Company"
    onFocus="promptEntry(sCompany)"></TD>
</TR>
```

Start of “BODY” portion of “FormValidation.html”

A more complex example

```
<TR>
   <TD>
   <INPUT TYPE="BUTTON" NAME="fakeSubmit"
    VALUE="Display"
    onClick="if (validatePersonalInfo(this.form))
    displayPersonalInfo(this.form); ">
   </TD>
   <TD><INPUT TYPE = "reset" VALUE = "Reset">
   </TD>
</TR>
</TABLE>
<P> NOTE: We replace the usual Submit button with a "Display" that acts
    locally,
<BR>by calling some code to display what was typed in.
</FORM>
</BODY>
</HTML>
```

End of file “FormValidation.html”

Applet

Properties

- Same as the public fields
- of the Java applet

Methods

- Same as the public methods
- of the Java applet

LiveConnect

A two-faced, (Netscape-only) technology to let JavaScript interact with Java, so that:

- A JavaScript script can control and coordinate Java applets, and let Java applets interact with plugins.
- A Java Applet can execute JavaScript code.

Server-Side JavaScript

Server-dependent technology to process the

A substitute for CGI.

Web page before passing it to the client.

(The Netscape SSJS object model is different from the Microsoft ASP object model, although JavaScript can be used as SSSLanguage for ASP)

References

D.Flanagan “Javascript. The definitive guide” O’Reilly.

D.Goodman “Dynamic HTML. The definitive reference” O’Reilly