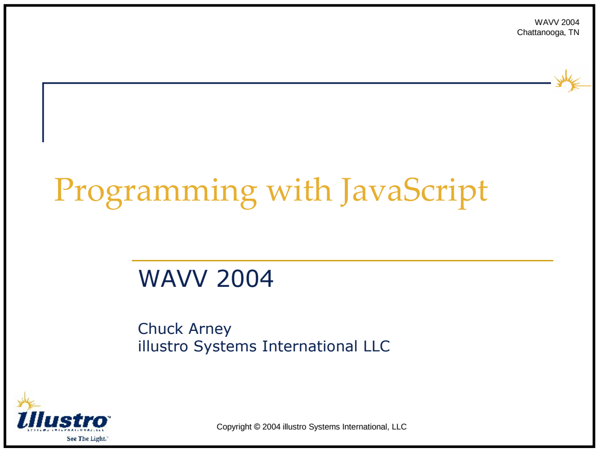


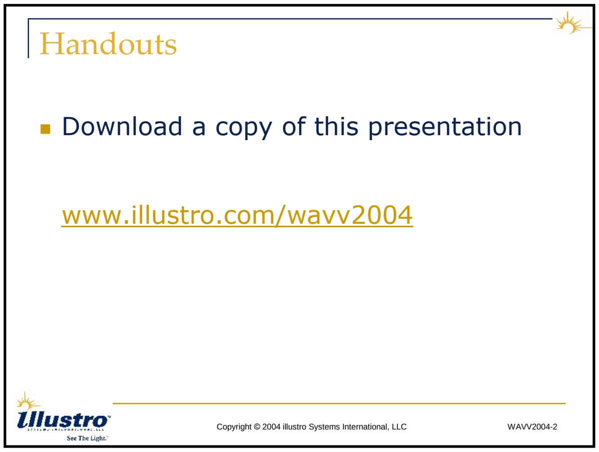
WAVV 2004
Chattanooga, TN

Programming with JavaScript

WAVV 2004

Chuck Arney
illustro Systems International LLC

Copyright © 2004 illustro Systems International, LLC

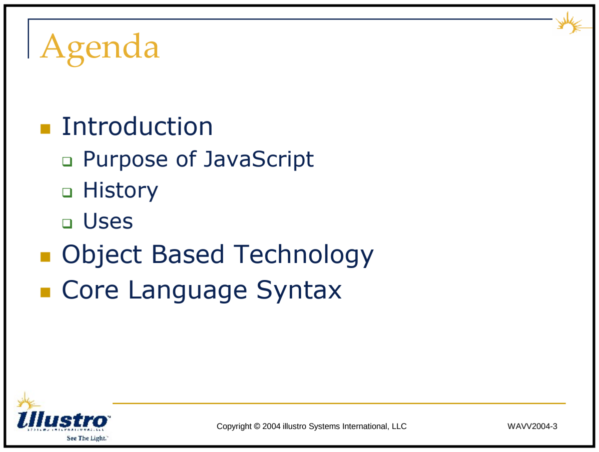


Handouts

- Download a copy of this presentation

  www.illustro.com/wavv2004

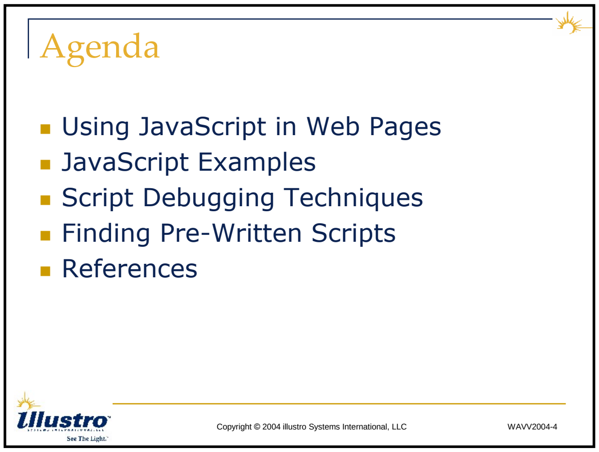
Copyright © 2004 illustro Systems International, LLC WAVV2004-2



Agenda

- Introduction
  - Purpose of JavaScript
  - History
  - Uses
- Object Based Technology
- Core Language Syntax

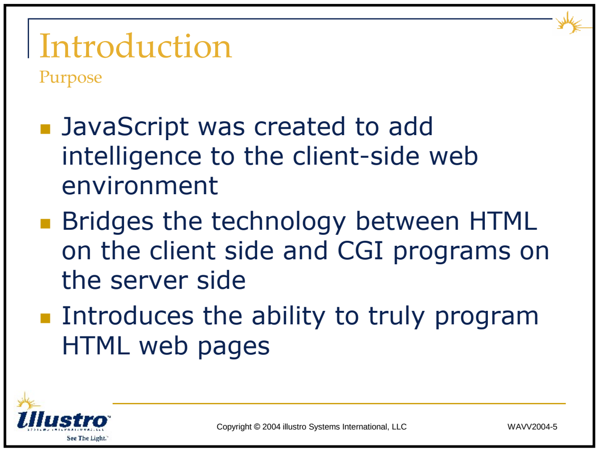
Copyright © 2004 illustro Systems International, LLC WAVV2004-3



Agenda

- Using JavaScript in Web Pages
- JavaScript Examples
- Script Debugging Techniques
- Finding Pre-Written Scripts
- References

Copyright © 2004 illustro Systems International, LLC WAVV2004-4

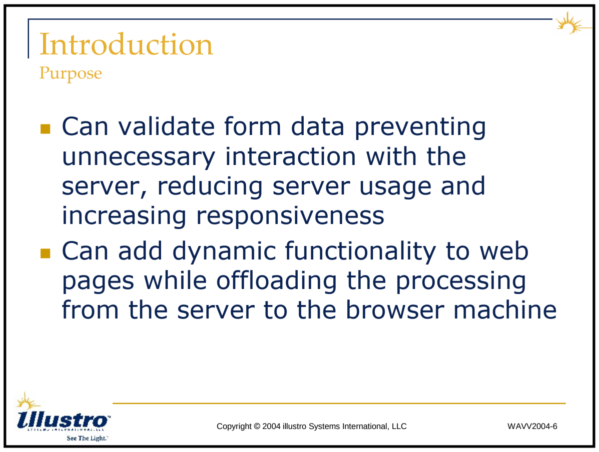


Introduction

Purpose

- JavaScript was created to add intelligence to the client-side web environment
- Bridges the technology between HTML on the client side and CGI programs on the server side
- Introduces the ability to truly program HTML web pages

Copyright © 2004 illustro Systems International, LLC WAVV2004-5

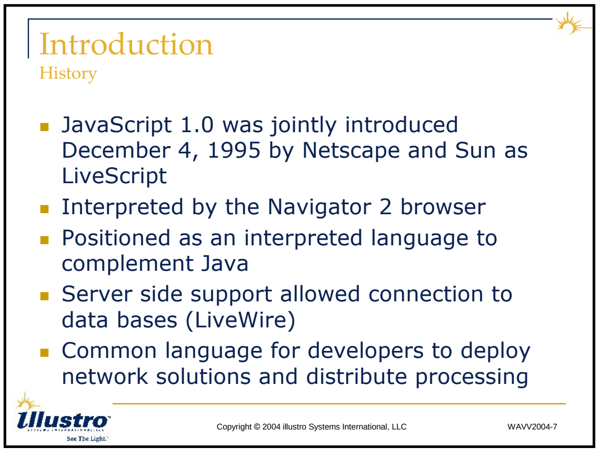


Introduction

Purpose

- Can validate form data preventing unnecessary interaction with the server, reducing server usage and increasing responsiveness
- Can add dynamic functionality to web pages while offloading the processing from the server to the browser machine

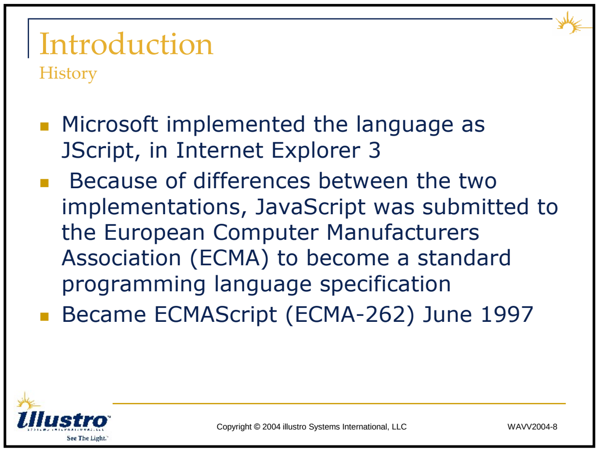
Copyright © 2004 illustro Systems International, LLC WAVV2004-6



Introduction
History

- JavaScript 1.0 was jointly introduced December 4, 1995 by Netscape and Sun as LiveScript
- Interpreted by the Navigator 2 browser
- Positioned as an interpreted language to complement Java
- Server side support allowed connection to data bases (LiveWire)
- Common language for developers to deploy network solutions and distribute processing

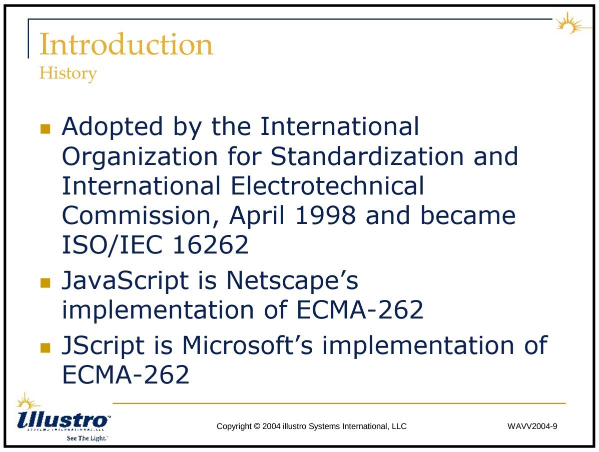
WAVV2004-7



Introduction
History

- Microsoft implemented the language as JScript, in Internet Explorer 3
- Because of differences between the two implementations, JavaScript was submitted to the European Computer Manufacturers Association (ECMA) to become a standard programming language specification
- Became ECMAScript (ECMA-262) June 1997

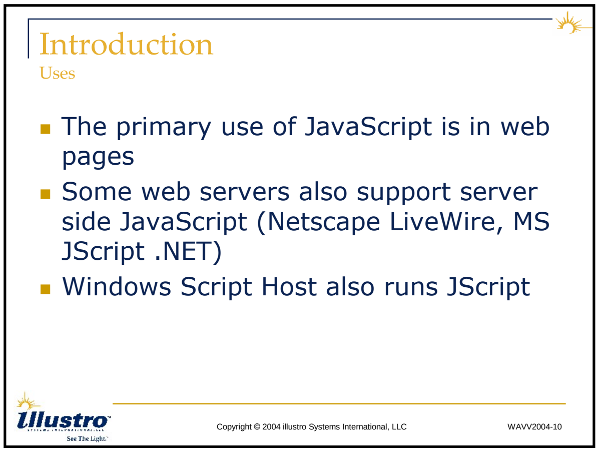
WAVV2004-8



Introduction
History

- Adopted by the International Organization for Standardization and International Electrotechnical Commission, April 1998 and became ISO/IEC 16262
- JavaScript is Netscape’s implementation of ECMA-262
- JScript is Microsoft’s implementation of ECMA-262

WAVV2004-9

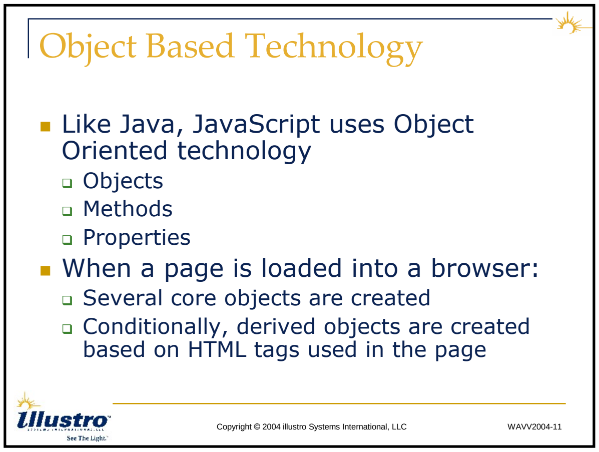


Introduction

Uses

- The primary use of JavaScript is in web pages
- Some web servers also support server side JavaScript (Netscape LiveWire, MS JScript .NET)
- Windows Script Host also runs JScript

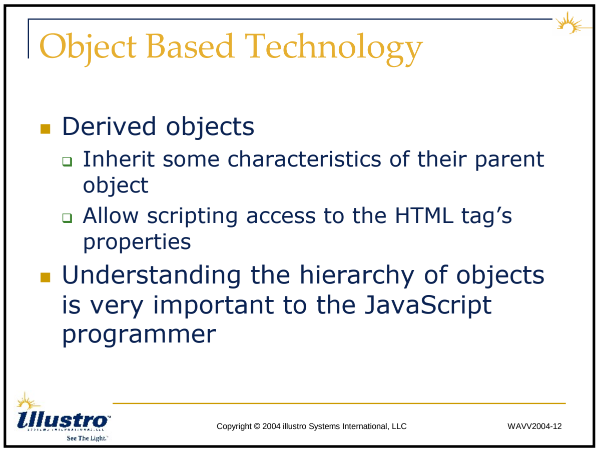
Copyright © 2004 illustro Systems International, LLC WAVV2004-10



Object Based Technology

- Like Java, JavaScript uses Object Oriented technology
  - Objects
  - Methods
  - Properties
- When a page is loaded into a browser:
  - Several core objects are created
  - Conditionally, derived objects are created based on HTML tags used in the page

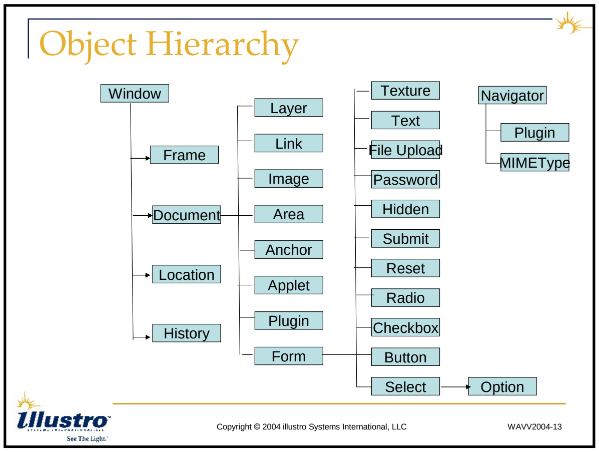
Copyright © 2004 illustro Systems International, LLC WAVV2004-11



Object Based Technology

- Derived objects
  - Inherit some characteristics of their parent object
  - Allow scripting access to the HTML tag’s properties
- Understanding the hierarchy of objects is very important to the JavaScript programmer

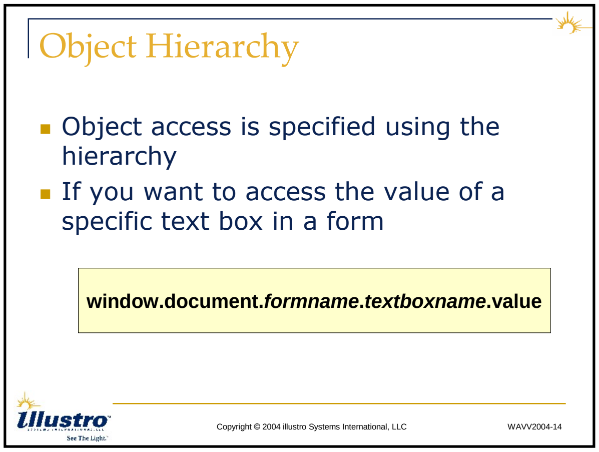
Copyright © 2004 illustro Systems International, LLC WAVV2004-12



Object Hierarchy

- Window
  - Frame
  - Document
    - Layer
    - Link
    - Image
    - Area
    - Anchor
    - Applet
    - Plugin
    - Form
      - Texture
      - Text
      - File Upload
      - Password
      - Hidden
      - Submit
      - Reset
      - Radio
      - Checkbox
      - Button
      - Select
        - Option
  - Location
  - History
- Navigator
  - Plugin
  - MIMEType

Copyright © 2004 illustro Systems International, LLC WAVV2004-13

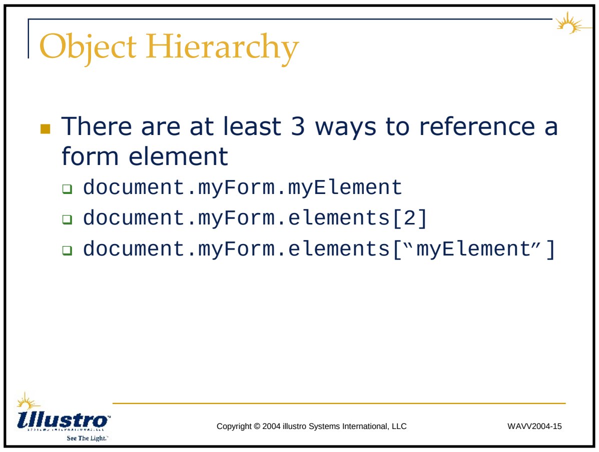


Object Hierarchy

- Object access is specified using the hierarchy
- If you want to access the value of a specific text box in a form

window.document.*formname.textboxname*.value

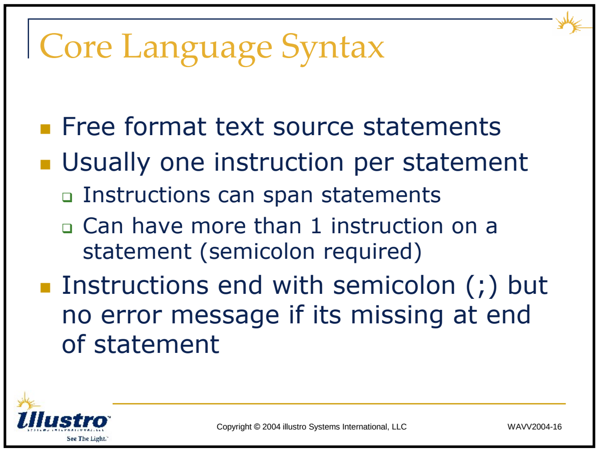
Copyright © 2004 illustro Systems International, LLC WAVV2004-14



Object Hierarchy

- There are at least 3 ways to reference a form element
  - `document.myForm.myElement`
  - `document.myForm.elements[2]`
  - `document.myForm.elements[“myElement”]`

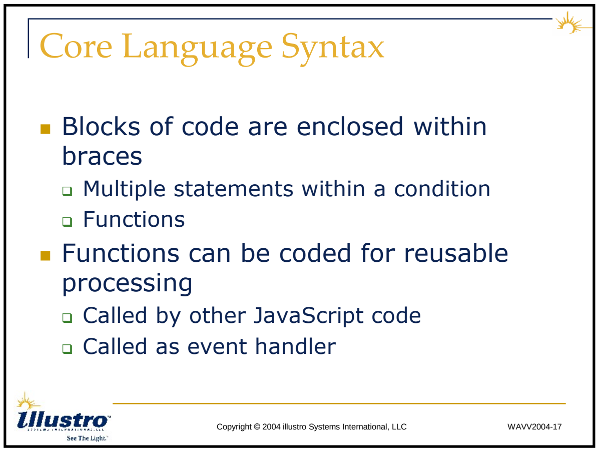
Copyright © 2004 illustro Systems International, LLC WAVV2004-15



Core Language Syntax

- Free format text source statements
- Usually one instruction per statement
  - Instructions can span statements
  - Can have more than 1 instruction on a statement (semicolon required)
- Instructions end with semicolon (;) but no error message if its missing at end of statement

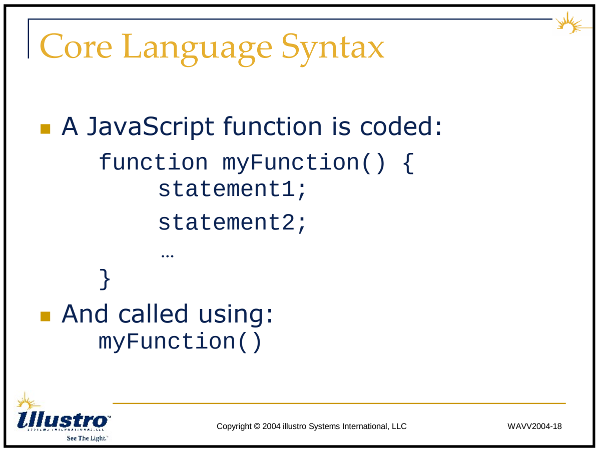
Copyright © 2004 illustro Systems International, LLC WAVV2004-16



Core Language Syntax

- Blocks of code are enclosed within braces
  - Multiple statements within a condition
  - Functions
- Functions can be coded for reusable processing
  - Called by other JavaScript code
  - Called as event handler

Copyright © 2004 illustro Systems International, LLC WAVV2004-17



Core Language Syntax

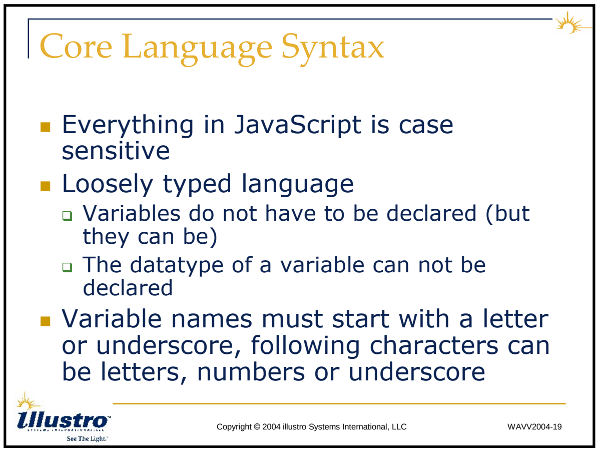
- A JavaScript function is coded:

```
function myFunction() {
     statement1;
     statement2;
     …
}
```

- And called using:

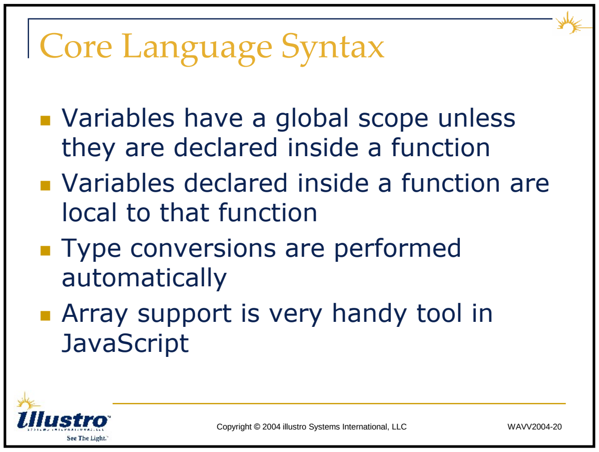
```
myFunction()
```

Copyright © 2004 illustro Systems International, LLC WAVV2004-18



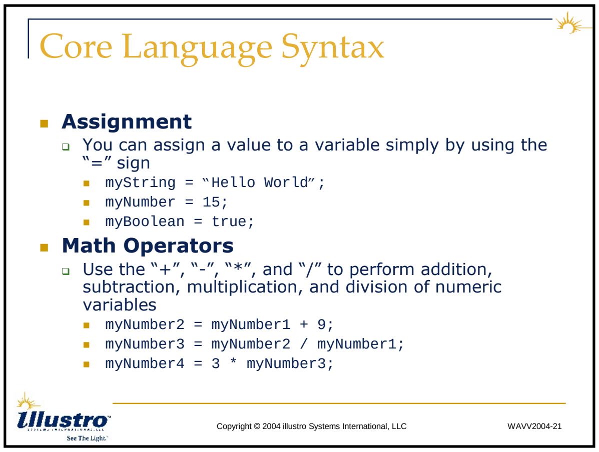
Core Language Syntax

- Everything in JavaScript is case sensitive
- Loosely typed language
  - Variables do not have to be declared (but they can be)
  - The datatype of a variable can not be declared
- Variable names must start with a letter or underscore, following characters can be letters, numbers or underscore



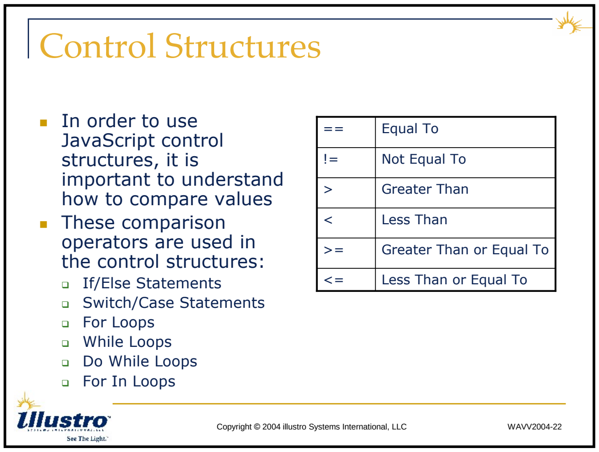
Core Language Syntax

- Variables have a global scope unless they are declared inside a function
- Variables declared inside a function are local to that function
- Type conversions are performed automatically
- Array support is very handy tool in JavaScript



Core Language Syntax

- **Assignment**
  - You can assign a value to a variable simply by using the “=” sign
    - `myString = “Hello World”;`
    - `myNumber = 15;`
    - `myBoolean = true;`
- **Math Operators**
  - Use the “+”, “-”, “*”, and “/” to perform addition, subtraction, multiplication, and division of numeric variables
    - `myNumber2 = myNumber1 + 9;`
    - `myNumber3 = myNumber2 / myNumber1;`
    - `myNumber4 = 3 * myNumber3;`

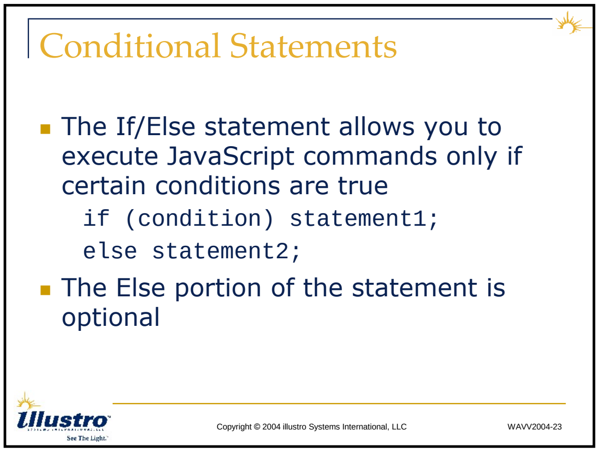


Control Structures

- In order to use JavaScript control structures, it is important to understand how to compare values
- These comparison operators are used in the control structures:
  - If/Else Statements
  - Switch/Case Statements
  - For Loops
  - While Loops
  - Do While Loops
  - For In Loops

==	Equal To
!=	Not Equal To
>	Greater Than
<	Less Than
>=	Greater Than or Equal To
<=	Less Than or Equal To

Copyright © 2004 illustro Systems International, LLC WAVV2004-22



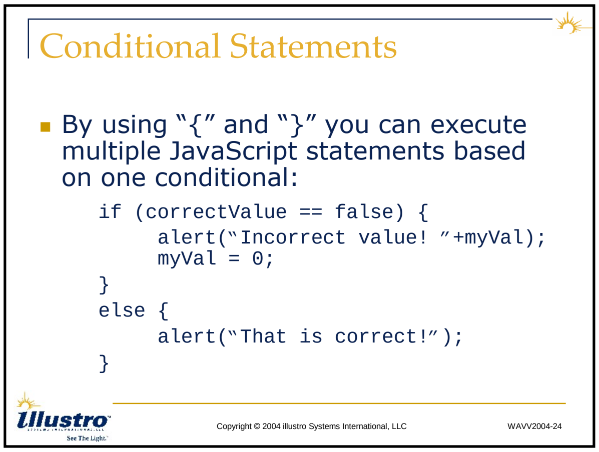
Conditional Statements

- The If/Else statement allows you to execute JavaScript commands only if certain conditions are true

```
if (condition) statement1;
else statement2;
```

- The Else portion of the statement is optional

Copyright © 2004 illustro Systems International, LLC WAVV2004-23

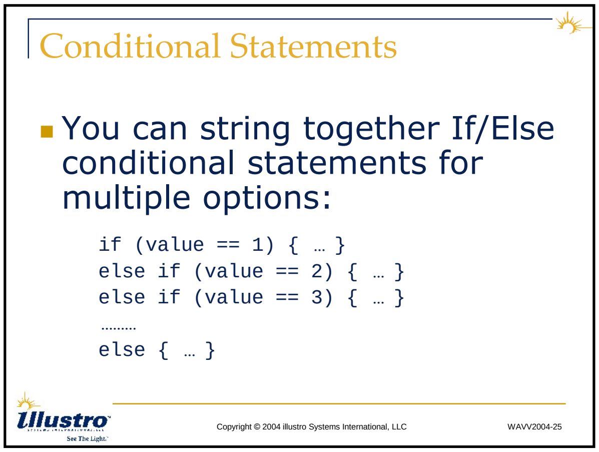


Conditional Statements

- By using “{” and “}” you can execute multiple JavaScript statements based on one conditional:

```
if (correctValue == false) {
     alert(“Incorrect value! ”+myVal);
     myVal = 0;
}
else {
     alert(“That is correct!”);
}
```

Copyright © 2004 illustro Systems International, LLC WAVV2004-24

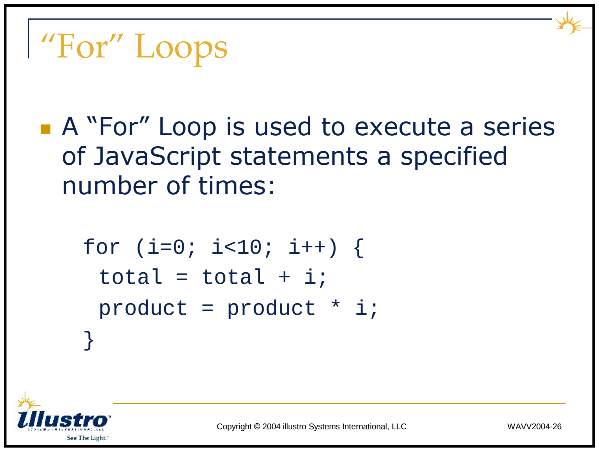


Conditional Statements

- You can string together If/Else conditional statements for multiple options:

```
if (value == 1) { … }
else if (value == 2) { … }
else if (value == 3) { … }
………
else { … }
```

Copyright © 2004 illustro Systems International, LLC WAVV2004-25

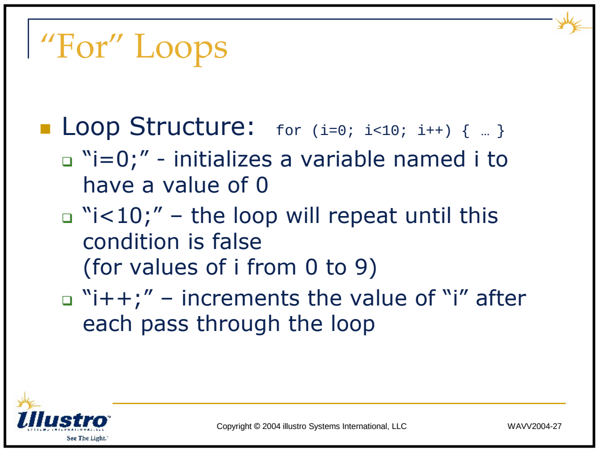


"For" Loops

- A "For" Loop is used to execute a series of JavaScript statements a specified number of times:

```
for (i=0; i<10; i++) {
 total = total + i;
 product = product * i;
}
```

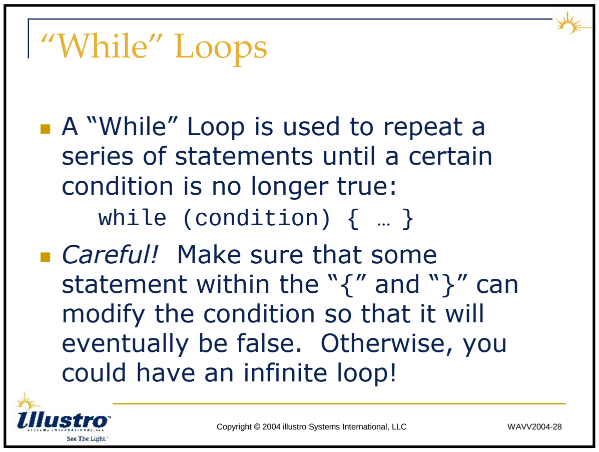
Copyright © 2004 illustro Systems International, LLC WAVV2004-26



"For" Loops

- Loop Structure: `for (i=0; i<10; i++) { … }`
  - "i=0;" - initializes a variable named i to have a value of 0
  - "i<10;" – the loop will repeat until this condition is false (for values of i from 0 to 9)
  - "i++;" – increments the value of "i" after each pass through the loop

Copyright © 2004 illustro Systems International, LLC WAVV2004-27



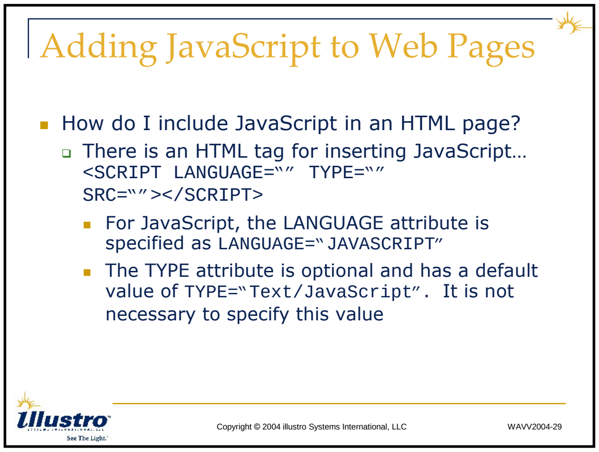
“While” Loops

- A “While” Loop is used to repeat a series of statements until a certain condition is no longer true:

  ```
  while (condition) { … }
  ```

- *Careful!* Make sure that some statement within the “{” and “}” can modify the condition so that it will eventually be false. Otherwise, you could have an infinite loop!

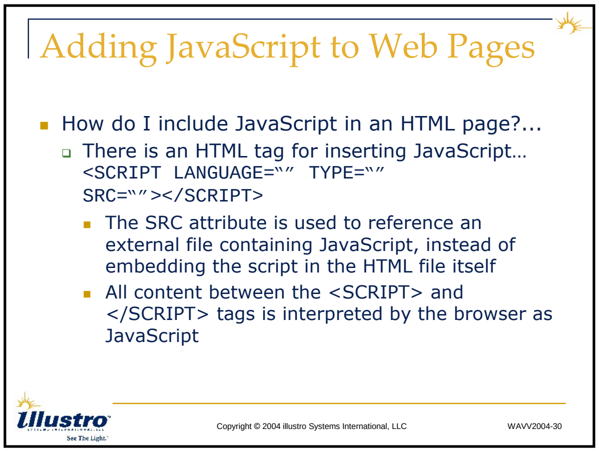
Copyright © 2004 illustro Systems International, LLC WAVV2004-28



Adding JavaScript to Web Pages

- How do I include JavaScript in an HTML page?
  - There is an HTML tag for inserting JavaScript… `<SCRIPT LANGUAGE=“” TYPE=“” SRC=“”></SCRIPT>`
    - For JavaScript, the LANGUAGE attribute is specified as `LANGUAGE=“JAVASCRIPT”`
    - The TYPE attribute is optional and has a default value of `TYPE=“Text/JavaScript”`. It is not necessary to specify this value

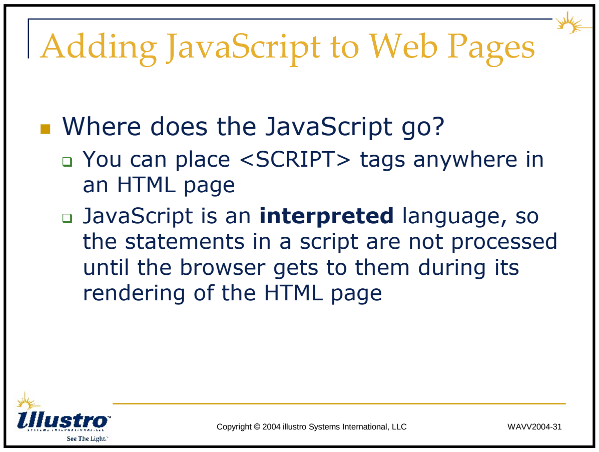
Copyright © 2004 illustro Systems International, LLC WAVV2004-29



Adding JavaScript to Web Pages

- How do I include JavaScript in an HTML page?…
  - There is an HTML tag for inserting JavaScript… `<SCRIPT LANGUAGE=“” TYPE=“” SRC=“”></SCRIPT>`
    - The SRC attribute is used to reference an external file containing JavaScript, instead of embedding the script in the HTML file itself
    - All content between the <SCRIPT> and </SCRIPT> tags is interpreted by the browser as JavaScript

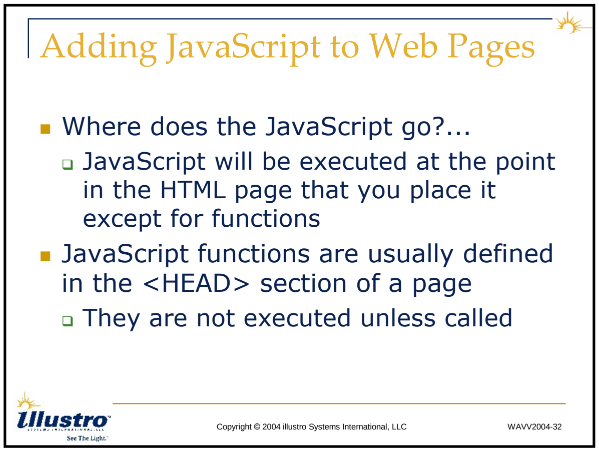
Copyright © 2004 illustro Systems International, LLC WAVV2004-30



Adding JavaScript to Web Pages

- Where does the JavaScript go?
  - You can place <SCRIPT> tags anywhere in an HTML page
  - JavaScript is an **interpreted** language, so the statements in a script are not processed until the browser gets to them during its rendering of the HTML page

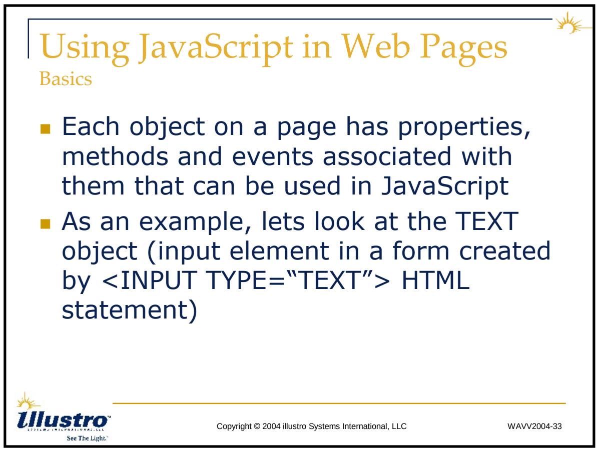
Copyright © 2004 illustro Systems International, LLC WAVV2004-31



Adding JavaScript to Web Pages

- Where does the JavaScript go?…
  - JavaScript will be executed at the point in the HTML page that you place it except for functions
- JavaScript functions are usually defined in the <HEAD> section of a page
  - They are not executed unless called

Copyright © 2004 illustro Systems International, LLC WAVV2004-32

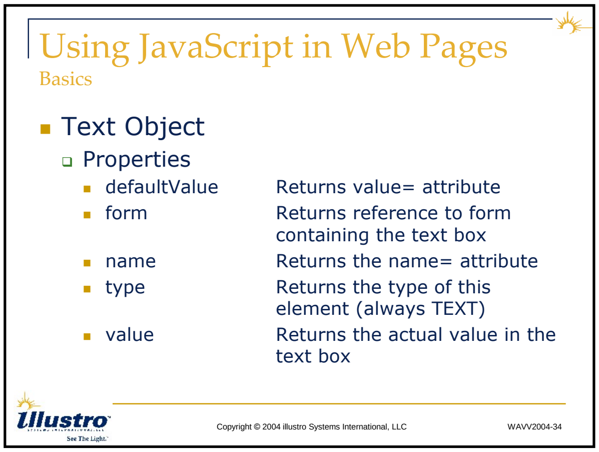


Using JavaScript in Web Pages

Basics

- Each object on a page has properties, methods and events associated with them that can be used in JavaScript
- As an example, lets look at the TEXT object (input element in a form created by <INPUT TYPE="TEXT"> HTML statement)

Copyright © 2004 illustro Systems International, LLC WAVV2004-33



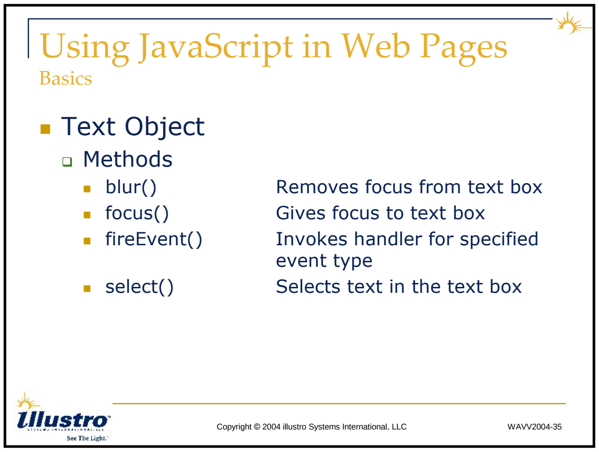
Using JavaScript in Web Pages

Basics

- Text Object
  - Properties

defaultValue	Returns value= attribute
form	Returns reference to form containing the text box
name	Returns the name= attribute
type	Returns the type of this element (always TEXT)
value	Returns the actual value in the text box

WAVV2004-34



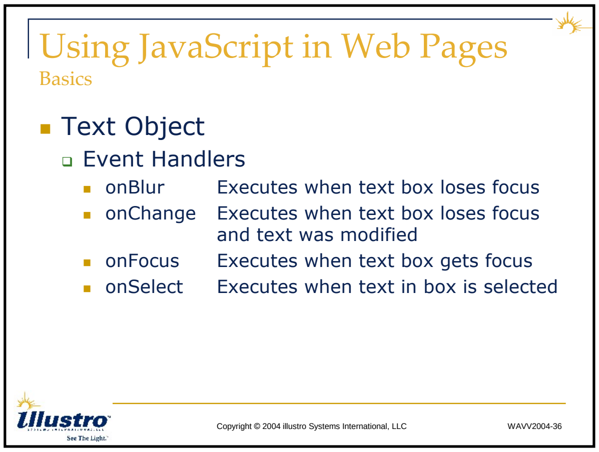
Using JavaScript in Web Pages

Basics

- Text Object
  - Methods

blur()	Removes focus from text box
focus()	Gives focus to text box
fireEvent()	Invokes handler for specified event type
select()	Selects text in the text box

WAVV2004-35



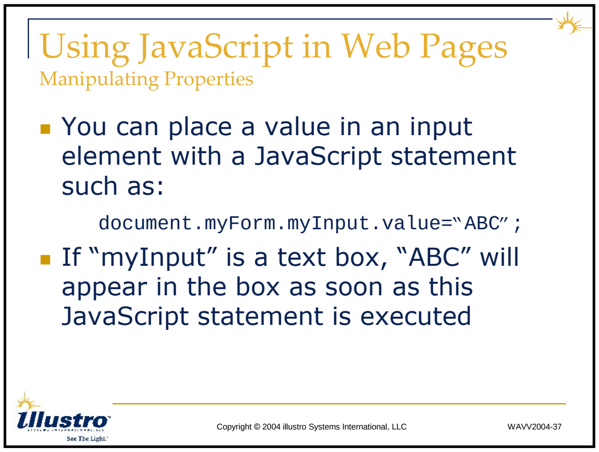
Using JavaScript in Web Pages

Basics

- Text Object
  - Event Handlers

onBlur	Executes when text box loses focus
onChange	Executes when text box loses focus and text was modified
onFocus	Executes when text box gets focus
onSelect	Executes when text in box is selected

WAVV2004-36



Using JavaScript in Web Pages

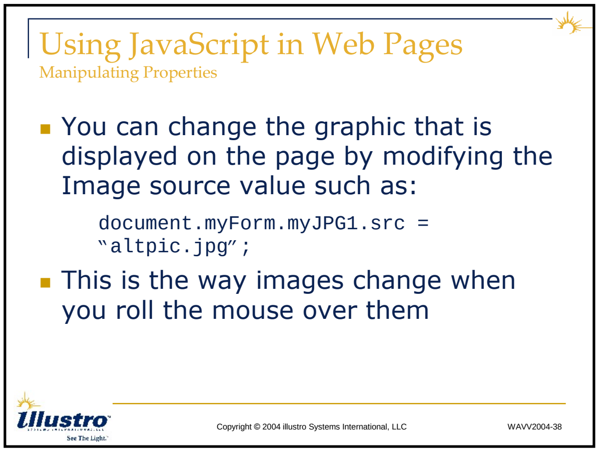
Manipulating Properties

- You can place a value in an input element with a JavaScript statement such as:

```
document.myForm.myInput.value=“ABC”;
```

- If “myInput” is a text box, “ABC” will appear in the box as soon as this JavaScript statement is executed

WAVV2004-37



Using JavaScript in Web Pages

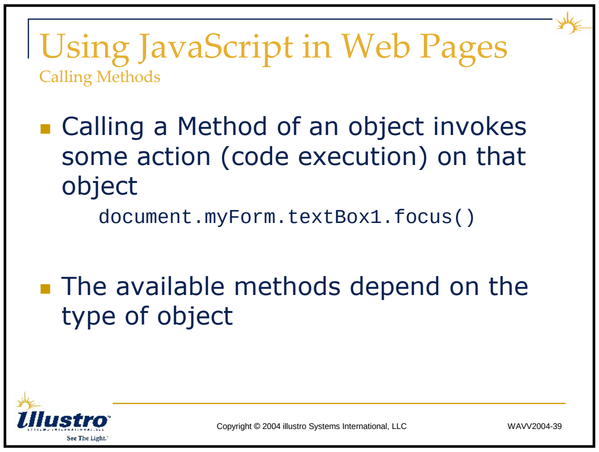
Manipulating Properties

- You can change the graphic that is displayed on the page by modifying the Image source value such as:

```
document.myForm.myJPG1.src =
“altpic.jpg”;
```

- This is the way images change when you roll the mouse over them

WAVV2004-38



Using JavaScript in Web Pages

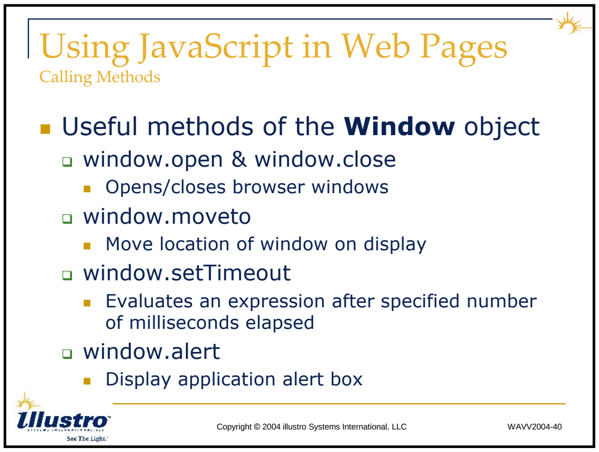
Calling Methods

- Calling a Method of an object invokes some action (code execution) on that object

```
document.myForm.textBox1.focus()
```

- The available methods depend on the type of object

WAVV2004-39

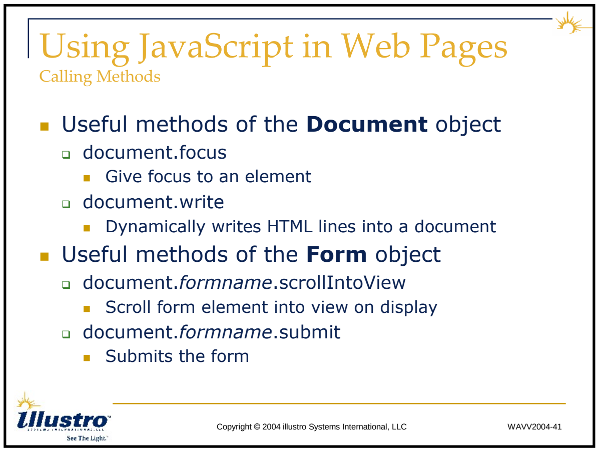


Using JavaScript in Web Pages

Calling Methods

- Useful methods of the **Window** object
  - window.open & window.close
    - Opens/closes browser windows
  - window.moveto
    - Move location of window on display
  - window.setTimeout
    - Evaluates an expression after specified number of milliseconds elapsed
  - window.alert
    - Display application alert box

Copyright © 2004 illustro Systems International, LLC WAVV2004-40

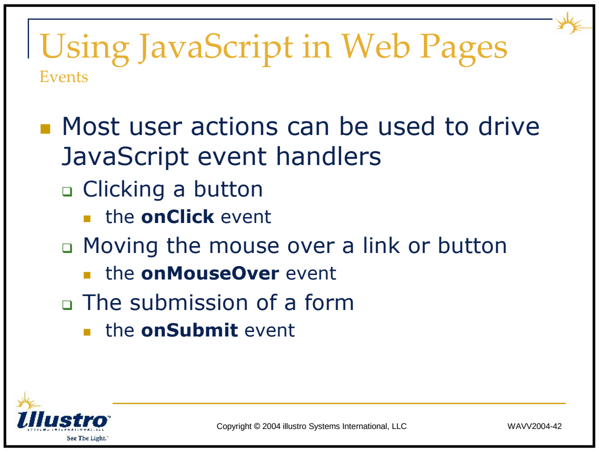


Using JavaScript in Web Pages

Calling Methods

- Useful methods of the **Document** object
  - document.focus
    - Give focus to an element
  - document.write
    - Dynamically writes HTML lines into a document
- Useful methods of the **Form** object
  - document.*formname*.scrollIntoView
    - Scroll form element into view on display
  - document.*formname*.submit
    - Submits the form

Copyright © 2004 illustro Systems International, LLC WAVV2004-41



Using JavaScript in Web Pages

Events

- Most user actions can be used to drive JavaScript event handlers
  - Clicking a button
    - the **onClick** event
  - Moving the mouse over a link or button
    - the **onMouseOver** event
  - The submission of a form
    - the **onSubmit** event

Copyright © 2004 illustro Systems International, LLC WAVV2004-42

Using JavaScript in Web Pages

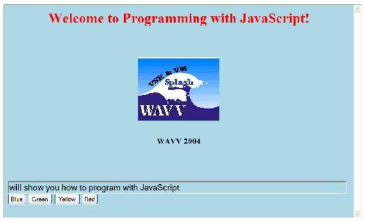
Events

- Events for HTML elements are defined in element tags, just like attributes

```
<input type=‶button″ value=‶Click Me″
     onClick=‶buttonWasClicked()″>
```

JavaScript Examples

- Lets look at some live examples



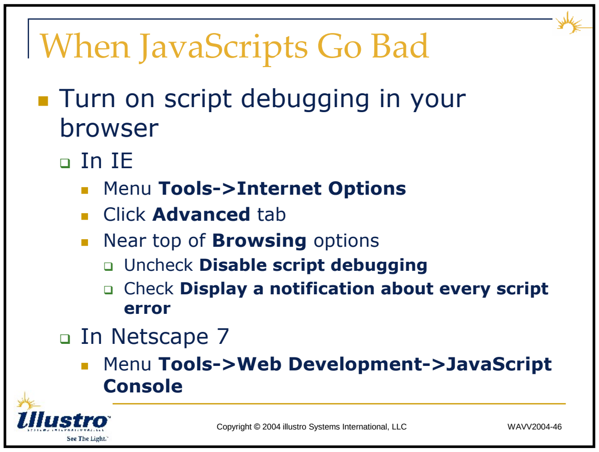
Welcome to Programming with JavaScript!

WAVV 2004

will show you how to program with JavaScript.

When JavaScripts Go Bad

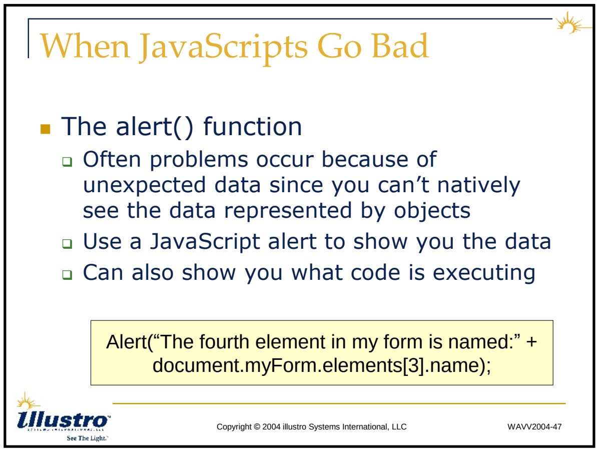
- Debugging tips
  - Turn on script debugging in your browser
  - Use the alert() function to display suspect data
  - Open a debug window and write trace information to it



When JavaScripts Go Bad

- Turn on script debugging in your browser
  - In IE
    - Menu **Tools->Internet Options**
    - Click **Advanced** tab
    - Near top of **Browsing** options
      - Uncheck **Disable script debugging**
      - Check **Display a notification about every script error**
  - In Netscape 7
    - Menu **Tools->Web Development->JavaScript Console**

Copyright © 2004 illustro Systems International, LLC WAVV2004-46

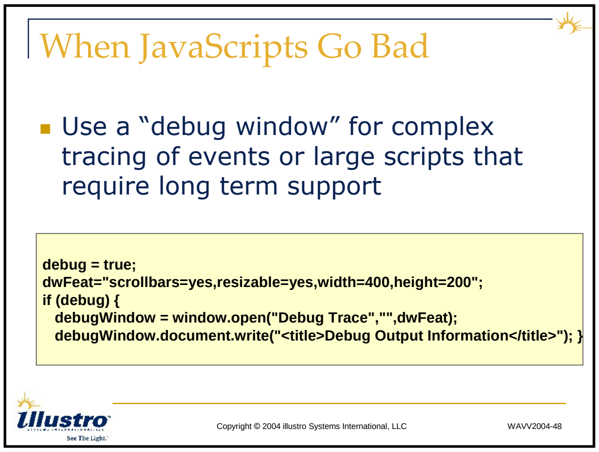


When JavaScripts Go Bad

- The alert() function
  - Often problems occur because of unexpected data since you can’t natively see the data represented by objects
  - Use a JavaScript alert to show you the data
  - Can also show you what code is executing

```
Alert(“The fourth element in my form is named:” +
document.myForm.elements[3].name);
```

Copyright © 2004 illustro Systems International, LLC WAVV2004-47

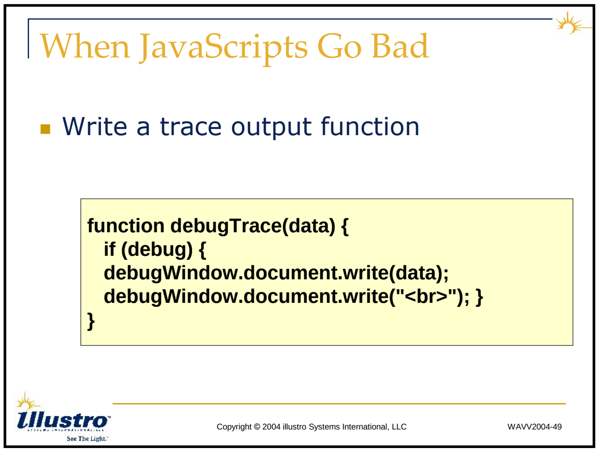


When JavaScripts Go Bad

- Use a “debug window” for complex tracing of events or large scripts that require long term support

```
debug = true;
dwFeat="scrollbars=yes,resizable=yes,width=400,height=200";
if (debug) {
   debugWindow = window.open("Debug Trace","",dwFeat);
   debugWindow.document.write("<title>Debug Output Information</title>"); }
```

Copyright © 2004 illustro Systems International, LLC WAVV2004-48

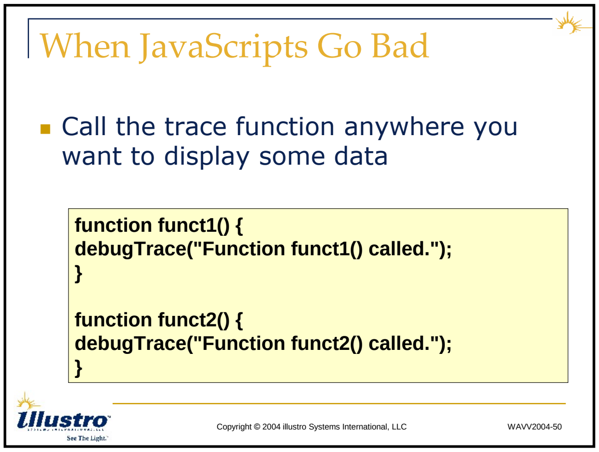


When JavaScripts Go Bad

- Write a trace output function

```
function debugTrace(data) {
   if (debug) {
   debugWindow.document.write(data);
   debugWindow.document.write("<br>"); }
}
```

WAVV2004-49



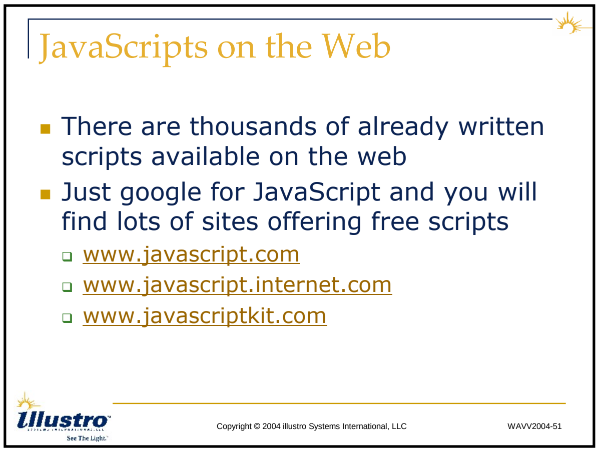
When JavaScripts Go Bad

- Call the trace function anywhere you want to display some data

```
function funct1() {
debugTrace("Function funct1() called.");
}

function funct2() {
debugTrace("Function funct2() called.");
}
```

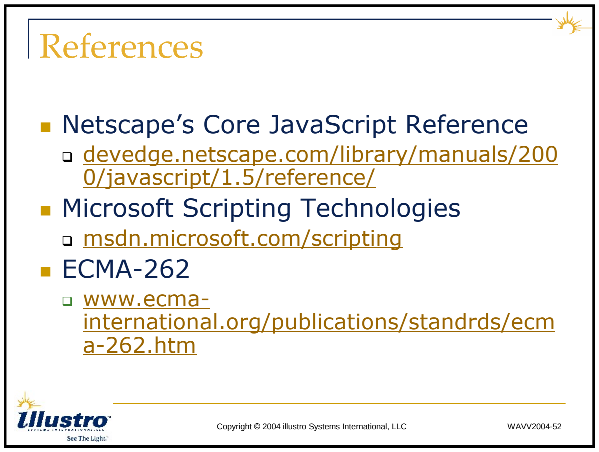
WAVV2004-50



JavaScripts on the Web

- There are thousands of already written scripts available on the web
- Just google for JavaScript and you will find lots of sites offering free scripts
  - www.javascript.com
  - www.javascript.internet.com
  - www.javascriptkit.com

WAVV2004-51



References

- Netscape’s Core JavaScript Reference
  - devedge.netscape.com/library/manuals/2000/javascript/1.5/reference/
- Microsoft Scripting Technologies
  - msdn.microsoft.com/scripting
- ECMA-262
  - www.ecma-international.org/publications/standrds/ecma-262.htm

WAVV2004-52