





Ground Rules for Onsite Classrooms

- Everyone participates
- Respect individual opinions and diversities
- Be open and honest
- Give headlines, be concise
- One speaker at a time
- Make language a non-issue
- Stick to time contracts
- Seek first to understand, and then to be understood
- Clean desk / room policy
- No mobile phone, no computer
- Clients & Leadership are often here, please remember that
- Maintain a spirit of fun and enthusiasm

























Objectives

By the end of this session, you will be able to:

- Understand what is JavaScript
- Describe Language Syntax
- State the features of Document Object Model (DOM)
- Explain what is Event handling
- List the features of Form Validation
- State the five JavaScript Events







JavaScript – Introduction

- JavaScript is an interpreted programming language that can be embedded into an HTML web page.
- Why JavaScript?





JavaScript - Common Usage

JavaScript extends the usefulness of web page.

Common usage:

- To ensure that user enters data into forms in valid format.
- To create interesting buttons with mouse rollover effects.
- To create pop-up windows.
- changing content of a certain area on an user event.
- Adding / removing fields or making fields editable
 / read only based on user input.







JavaScript with HTML

How

- JavaScript is added to an HTML page using the SCRIPT tag.
- •JavaScript statements may be placed inside the html document as shown in example code or it can be in a .js extension file separately and used as:
- <SCRIPT SRC="/common.js"> </SCRIPT>

Example

```
<HTML>
 <HFAD>
 <TITLE>Web Page containing
 JavaScript</TITLE>
 <SCRIPT
 LANGUAGE="JAVASCRIPT">
<!-- hide JavaScript code from browsers
 that are not JavaScript enabled
(JavaScript Statements goes here)
//end hiding of JavaScript code -->
</SCRIPT>
 </HEAD>
 <BODY>
 (HTML document goes here)
 </BODY>
```





</HTML>

Rules & Conventions

- Case-sensitivity: JavaScript is a case-sensitive language, means the language will treat these words differently: example, Example, EXAMPLE.
- Semicolons: All statements should end in a semicolon. The semicolon separates one statement from another.
- String: A string is a sequence of zero or more characters enclosed within single or double quotes ('single', "double").
- Comments: You can create a comment like this:
 - Single line comment using the double forward slashes "//".
 - multiple line comments you can open the comment with a forward slash and an asterisk "/*", and close it with an asterisk followed by a forward slash "*/".





Variables: A variable is a name assigned to a location in a computer's memory to store data. Variables are declared with the var keyword and loosely bound with data types, like var x; or var x = 1, y = 3, sum = 0;

Rules for the names of the variables:

- The first character must be a letter of the alphabet (lowercase or uppercase), an underscore (_) or a dollar sign (\$). The dollar sign is not recommended as it is not supported prior to JavaScript ver 1.1.
- You CANNOT use a number as the first character of the name.
- Names CANNOT contain spaces.
- Names CANNOT match any of the reserved words.





Operators: Operators are the things that act on variables like:

```
var x = 1, y = 3, sum = 0;
sum = x + y;
```

- Computational like Unary negation (), Increment (++)
- Logical like Logical NOT (!), Less than (<) etc.</p>
- Bitwise like AND (&) , OR (|) etc.
- Assignment like = , Addition (+=) etc.





If

The "if" statement is a fundamental control statement. It allows your program to perform a test, and act based on the results of that test.

```
if ( (x == 1) && (y == 3) ) {sum = x + y;
}
else {
  subtotal = sum;
}
```





Switch

The switch statement is handy when a variable may take on a number of values and we want to do different things for some of those values.

```
switch(n) {
         case 1: //start here if n equals 1.
                        // place code here
                  break:
         case 2: //start here if n equals 2.
                  // place code here
                  break; // stop here
         default: // if all other conditions fail do this
                         // place code here
                  break;
```





Loops

- •while: While a condition is true, execute one or more statements. "While loops" are especially useful when you do not know how many times you have to loop, but you know you should stop when you meet the condition.
- •for: "For loops" are useful when you know exactly how many times you want the loop to execute.

```
var x = 1;
 while ( x <= 10 ) {
  // loop until x is greater than 10
   x++; // add one to the value of x
var x:
 for (x = 1; x \le 10; x++)
       // loop while x is \leq 10
        // do something ten times
```





Functions

- •Functions are an important part of programming as they allow you to create chunks of code that perform a specific task.
 - Built-in functions
 - Programmer created functions

```
var x = 1;
 while ( x <= 10 ) {
  // loop until x is greater than 10
   x++; // add one to the value of x
var x:
 for (x = 1; x \le 10; x++)
       // loop while x is \leq 10
        // do something ten times
```

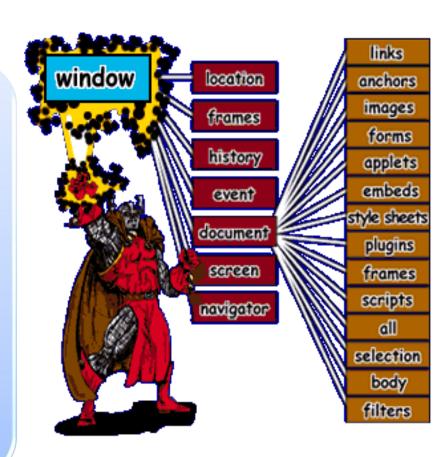




Document Object Model

The Document Object Model (DOM)

• The browser provides us with a series of objects. The browser window that the page is displayed in is known as the window object. The HTML page displayed by your browser is known as the document object.







DOM – Example

DOM-Example

- An example is the FORM element and the elements that reside inside the form. window.document.forms[0]
- Refers to the first form in the document. Forms are implemented as arrays in the DOM. If there is more than one form on page the numbers will start at zero and go up.
- window.document.Form1
- Refers to the form by name Form1.
- window.document.Form1.FirstName.value
- Refers to the value typed into the textbox named FirstName by the client, in the form named Form1.

```
<HTML>
<HFAD>
     <TITLE>Simple Form</TITLE>
</HFAD>
<BODY><FORM NAME="Form1">
Name: <INPUT TYPE="TEXT"
NAME="FirstName"><BR>
<INPUT TYPE="Button" VALUE="Submit Info" >
</FORM><FORM NAME="Form2">
Name: <INPUT TYPE="TEXT"
NAME="LastName"><BR>
<INPUT TYPE="Button" VALUE="Submit Info" >
</FORM></BODY>
</HTML>
```



Functions

- •onClick: A click event occurs when a button, checkbox, link, radio button, reset button, or submit button is clicked. This event is regularly used with these components to start script execution.
- •onSubmit: A submit event occurs when the user submits a form. This event is regularly used with a form and a submit button to start the form validation script.

Example

```
<HTML>
 <HFAD>
 <TITLE>Click Events</TITLE>
 </HFAD>
<BODY>
<FORM action="http://www.kanbay.com/form.jsp"
onSubmit="return checkform();">
<INPUT TYPE="Button" VALUE="Click Me"</pre>
onClick="window.alert('You Clicked me');">
</FORM>
</BODY>
```





</HTML>

Mouse Events

onMouseOver

 An onMouseOver event occurs when the user positions their mouse over a hyperlink, or a linked region of a client-side image.

onMouseOut

 An onMouseOut event occurs when the user moves their mouse off of a hyperlink, or a linked region of a client-side image map.

```
<HTML>
 <HFAD>
 <TITLE>Mouse Events
</TITLE>
 </HEAD>
 <BODY><FORM
 action="http://www.kanbay.com/form.jsp"
<a href="http://www.kanbay.com/mouse"
onMouseOver="window.status='Your mouse is
 over the link!'; return true;"
 onMouseOut="window.status='Your mouse is out
 of the link!'; return true;">Click for
 information.</a>
</BODY>
 </HTML>
```





Focus Event

onFocus

 This event occurs when a user tabs into or clicks on a password field, a text field, a Textarea, or a FileUpload field in an HTML form. If a user clicks on one of these elements in a form, it is receiving the user's focus.

Example

<HTML>

<HEAD>
<TITLE>Simple Form</TITLE>
</HEAD>

<BODY>MONTH

<INPUT TYPE="TEXT" NAME="Month"</pre>

onFocus="window.status=('Please enter the Month as two digits 01 through 12'); return true;">

</BODY>





Change Event

onChange

 The change event happens when the user leaves a Password, Text, Textarea, FileUpload field or selects an option of dropdown list in an HTML form, and its value has changed.

Example

<HTML>

<HEAD>
<TITLE>Simple Form</TITLE>
</HEAD>

<BODY><INPUT TYPE="TEXT" NAME="Month"

onChange="window.status=('The value of the Month Changed!!!!'); return true;"

</BODY>

</HTML>





Load Events

onLoad

•The load event triggers when the browser finishes loading an image, window, document, all the frame pages within a FRAMESET or any particular frame.

onUnload

The unload event occurs when you move to a new document. For example if you use the back button, or click on a link to another page the unload event will occur.

Example

<HTML>

<HEAD>

<TITLE>Simple Form</TITLE> </HEAD>

<BODY onLoad="alert ('Welcome to our website!');">

</BODY>

</HTML>





Form Validation

- Form validation is accomplished by using JavaScript to preprocess the information the user types into a form before the data is sent to a server application.
- This practice is more efficient than allowing incorrectly formatted data to be sent to the server.
- If the information is incorrectly formatted you can alert the user with a pop-up and force them to correct the mistake.







Weakness & Recap

Weakness of JavaScript:

- All browser versions do not support javascripts. This makes it difficult to create a complicated JavaScript code that work across all browsers and all versions.
- The user can disable the javascript option of the browser, he is using. This not only stops javascript functioning, but affects look and feel of the static web pages also.
- It's difficult to hide the code that makes your application work. Anyone can right click and choose "View Source" to see exactly how your software operates.
- The size of your javascripts due effect the loading of your webpage. That means the a lengthy script also means a performance hit for html web page too.

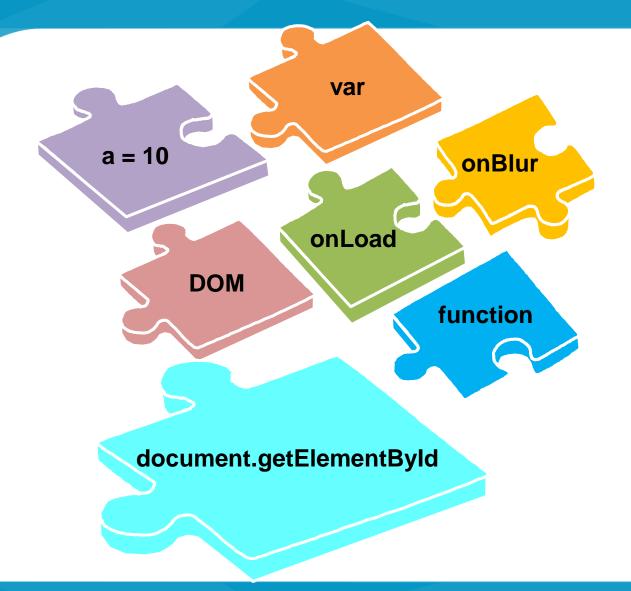
Topics we covered:

- Introduction to JavaScript
- Language Syntax
- Document Object Model (DOM)
- Event handling
- Form Validation





Recap







Thank You







About Capgemini

With more than 120,000 people in 40 countries, Capgemini is one of the world's foremost providers of consulting, technology and outsourcing services. The group reported 2011 global revenues of EUR 9.7 billion.

Together with its clients, Capgemini creates and delivers business and technology solutions that fit their needs and drive the results they want. A deeply multicultural organization, Capgemini has developed its own way of working, the Collaborative Business Experiencetm, and draws on Rightshore[®], its worldwide delivery model.

Rightshore® is a trademark belonging to Capgemini



www.capgemini.com









