Digicoders technologies (P) Ltd.

LECTURE NOTES

ON

INTERNET AND WEB TECHNOLOGY

Unit-7

Compiled by

Team Digicoders
B-36, Sector O, Near Ram Ram Bank Chauraha, Aliganj,
Lucknow Uttar Pradesh 226021

CONTENTS

S.NO	CHAPTER NAME	PAGE NO
1	Internet Basics	1-20
2	Internet Connectivity & WWW	21-30
3	Internet Security	31-39
4	Internet Security	40-45
5	Website Classifications	46-52
6	Development of Portals Using HTML	53-61
7	Client-side Scripting with JavaScript	62-71
8	Server-Side Scripting	72-76
9	Server-Side Programming using PHP	77-93

UNIT-7

Client side Scripting with JavaScript

Introduction to script

A scripting or script language is a programming language for a special run-time environment that automates the execution of tasks, the tasks could alternatively be executed one-by-one by a human operator. Scripting languages areoften interpreted, rather than compiled.

JavaScript

JavaScript (js) is a light-weight object-oriented programming language which is used by several websites for scripting the webpages. It is an interpreted, full-fledged programming language thatenables dynamic interactivity on websites when applied to an HTML document.

Advantages of JavaScript:

1. JavaScript is a client side language.
2. JavaScript is an easy language to learn.
3. JavaScript is comparatively fast for the end user.
4. Extended functionality to web pages
5. No compilation needed
6. Easy to debug and test
7. Platform independent
8. Event-Based Programming language
9. Procedural programming capabilities

Client-side scripting

Client-side scripting (embedded scripts) is code that exists inside the client's HTML page. This code will be processed on the client machine and the HTML page will NOT perform a Post Back to the web-server.
Traditionally, client-side scriptingis used for page navigation, data validation and formatting. The language used in this scripting is JavaScript. JavaScript is compatible and is able to run on any internet browser.

Advantages of Client-side scripting

- More interactive since it responds immediately to the user action.
- Quick Execution because they don't require a trip to the server.
- improve the user experience for the user whose browser support script.
- An alternative option is available for the user whose browser didn't support script.
- Reusable and obtainable from many resources.

Disadvantages of client-side scripting.

- it is not supported by all browsers. if no alternative is given for the script than the user might get an error.
- Need more testing. because of the different browser and its version support script differently.
- If the Script is not available through other resources than more development time and effort required.
- Developers have more control over the look and behaviour of their Web widgets; however, usability issuescan arise if a Web widget seems like a standard control but behaves different way or vice-versa.

JavaScript Variables

Like many other programming languages, JavaScript has variables. Variables can be thought of as named containers. You can place data into these containers and then refer to the data simply by naming the container. Before you use a variable in a JavaScript program, you must declare it. Variables are declared with the var keywordas follows.

```
<script type="text/javascript">
<!--
var m
oney;var
nam e;
//-->
</script>
```

Common Built-in Functions

- JavaScript provides number of built-in functions.

Functions	Description
isNan()	**Returns true,** if the object is Not a Number. **Returns false,** if the object is a number.
parseFloa t (string)	If the string begins with a number, the function reads through the string until it finds the end of the number; cuts off the remainder of the string and returns theresult. If the string does not begin with a number, the function returns NaN.
parseIn t (string)	If the string begins with an integer, the function reads through the string until itfinds the end of the integer, cuts off the remainder of the string and returns theresult. If the string does not begin with an integer, the function returns NaN (Not a Number).
String (object)	Converts the object into a string.
eval()	Returns the result of evaluating an arithmetic expression.

JavaScript Array

JavaScript array is an object that represents a collection of similar type of elements.The syntax of creating array directly is given below:

```
var arrayname=new Array();
```

Here, **new keyword** is used to create instance of array.

Javascript Operators

JavaScript includes operators as in other languages. An operator performs some operation on single or multiple operands(data value) and produces a result.
JavaScript includes following categories of operators.

1. Arithmetic Operators(
2. Comparison Operators
3. Logical Operators
4. Assignment Operators
5. Conditional Operators

Arithmetic Operators

Arithmetic operators are used to perform mathematical operations between numeric operands.

Operator	Description
+	Adds two numeric operands.
-	Subtract right operand from left operand
*	Multiply two numeric operands.
/	Divide left operand by right operand.
%	Modulus operator. Returns remainder of two operands.
++	Increment operator. Increase operand value by one.
--	Decrement operator. Decrease value by one.

Comparison Operators

JavaScript language includes operators that compare two operands and return Boolean value true or false.

Operato r s	Description
==	Compares the equality of two operands without considering type.
===	Compares equality of two operands with type.
!=	Compares inequality of two operands.
>	Checks whether left side value is greater than right side value. If yes then returns true otherwisefalse.
<	Checks whether left operand is less than right operand. If yes then returns true otherwise false.
>=	Checks whether left operand is greater than or equal to right operand. If yes then returns trueotherwise false.
<=	Checks whether left operand is less than or equal to right operand. If yes then returns true otherwise false.

Logical Operators

Logical operators are used to combine two or more conditions. JavaScript includes following logicaloperators.

Operator	Description
&&	&& is known as AND operator. It checks whether two operands are non-zero (0, false,undefined, null or "" are considered as zero), if yes then returns 1 otherwise 0.
||	|| is known as OR operator. It checks whether any one of the two operands is non-zero (0,false, undefined, null or "" is considered as zero).
!	! is known as NOT operator. It reverses the boolean result of the operand (or condition)

Assignment Operators

JavaScript includes assignment operators to assign values to variables with less key strokes.

Assignment operators	Description
=	Assigns right operand value to left operand.
+=	Sums up left and right operand values and assign the result to the left operand.
-=	Subtract right operand value from left operand value and assign the result to theleft operand.
*=	Multiply left and right operand values and assign the result to the left operand.
/=	Divide left operand value by right operand value and assign the result to the leftoperand.
%=	Get the modulus of left operand divide by right operand and assign resultedmodulus to the left operand.

Ternary Operator

JavaScript includes special operator called ternary operator :? that assigns a value to a variable based onsome condition. This is like short form of if-else condition.

Syntax:

<condition> ? <value1> : <value2>;

Ternary operator starts with conditional expression followed by ? operator. Second part (after ? and before : operator) will be executed if condition turns out to be true. If condition becomes false then thirdpart (after :) will be executed.

Conditional statements

In JavaScript we have the following conditional statements:

- Use if to specify a block of code to be executed, if a specified condition is
- true Use else to specify a block of code to be executed, if the same
- condition is false Use else if to specify a new condition to test, if the first
- condition is false
  Use switch to specify many alternative blocks of code to be executed

The if Statement

Use the if statement to specify a block of JavaScript code to be executed if a condition is true.

Syntax

```
if (condition) {
  // block of code to be executed if the condition is true
}
```

The else Statement

Use the else statement to specify a block of code to be executed if the condition is false.if (condition) {
```
  // block of code to be executed if the condition is true
} else {
  // block of code to be executed if the condition is false
}
```

The else if Statement

Use the else if statement to specify a new condition if the first condition is false.

Syntax

```
if (condition1) {
  // block of code to be executed if condition1 is true
} else if (condition2) {
  // block of code to be executed if the condition1 is false and condition2 is true
} else {
  // block of code to be executed if the condition1 is false and condition2 is false
}
```

The JavaScript Switch Statement

Use the switch statement to select one of many code blocks to be executed.
Syntax

```
switch(expression

 ) {case x:
  // code
 blockbrea
 k; case y:
  // code
 blockbrea
 k; default:
  // code block
}
```

JavaScript Loops
Loops are handy, if you want to run the same code over and over again, each time with a different value.The statements for loops provided in JavaScript are:

- for statement
- do...while
- statement while
  statement

for statement

A for loop repeats until a specified condition evaluates to false. The JavaScript for loop is similar to the Java and C for loop.

A for statement looks as follows:

```
for ([initialExpression]; [conditionExpression];

 [incrementExpression]) statement
```

do...while statement
The do...while statement repeats until a specified condition evaluates to false. A do...while statement looks as follows:

```
do
 statement
while (condition);
```

while statement
A while statement executes its statements as long as a specified condition evaluates to true. A while statementlooks as follows:

```
while
 (condition
 )
 statement
```

Document Object Model

Every web page resides inside a browser window which can be considered as an object.
A Document object represents the HTML document that is displayed in that window. The Document object hasvarious properties that refer to other objects which allow access to and modification of document content.
The way a document content is accessed and modified is called the **Document Object Model**, or **DOM**. The Objectsare organized in a hierarchy. This hierarchical structure applies to the organization of objects in a Web document.

- **Window object** − Top of the hierarchy. It is the outmost element of the object hierarchy.
- **Document object** − Each HTML document that gets loaded into a window becomes a document object. Thedocument contains the contents of the page.
- **Form object** − Everything enclosed in the <form>...</form> tags sets the form object.
- **Form control elements** − The form object contains all the elements defined for that object such as textfields, buttons, radio buttons, and checkboxes.

DOM compatibility

If you want to write a script with the flexibility to use either W3C DOM or IE 4 DOM depending on their availability,then you can use a capability-testing approach that first checks for the existence of a method or property to
determine whether the browser has the capability you desire.
For example −if (document.getElementById) {

```
  // If the W3C method exists, use it
} else if (document.all) {
  // If the all[] array exists, use it
} else {
  // Otherwise use the legacy DOM
}
```

Creating Functions,

A JavaScript function is defined with the function keyword, followed by a name, followed by

parentheses ().Function names can contain letters, digits, underscores, and dollar signs (same

rules as variables).

The parentheses may include parameter names separated by commas:
(parameter1, parameter2, ...)

The code to be executed, by the function, is placed inside curly brackets:

```
{}function name(parameter1, parameter2, parameter3) {
 // code to be executed
}
```

Event handling in JavaScript

Introduction to Event Handling

- Event Handling is a software routine that processes actions, such as keystrokes and mouse movements.
- It is the receipt of an event at some event handler from an event producer and subsequent processes.

Functions of Event Handling

- Event Handling identifies where an event should be forwarded.
- It makes the forward event.
- It receives the forwarded event.
- It takes some kind of appropriate action in response, such as writing to a log, sending an error or recovery routineor sending a message.
- The event handler may ultimately forward the event to an event consumer.

Event Handlers

Event Handler	Descriptio n
onAbort	It executes when the user aborts loading an image.
onBlur	It executes when the input focus leaves the field of a text, textarea or a select option.
onChange	It executes when the input focus exits the field after the user modifies its text.
onClick	In this, a function is called when an object in a button is clicked, a link is pushed, a checkboxis checked or an image map is selected. It can return false to cancel the action.
onError	It executes when an error occurs while loading a document or an image.
onFocus	It executes when input focus enters the field by tabbing in or by clicking but not selectinginput from the field.
onLoad	It executes when a window or image finishes loading.
onMouseOver	The JavaScript code is called when the mouse is placed over a specific link or an object.
onMouseOut	The JavaScript code is called when the mouse leaves a specific link or an object.

onReset	It executes when the user resets a form by clicking on the reset button.
onSelect	It executes when the user selects some of the text within a text or textarea field.
onSubmit	It calls when the form is submitted.
onUnload	It calls when a document is exited.

Example : Simple Program on onload() Event handler

```
<html>
   <head>
   <script
   type="text/javascript"
   > function time()
   {
       var d = new
       Date(); var ty =
       d.getHours() +
       ":"+d.getMinutes()+":"+d.getSeconds();
       document.frmty.timetxt.value=ty;
       setInterval("time()",1000)
   }
   </script>
   </head>
<body onload="time()">
   <center><h2>Displaying Time</h2>
       <form name="frmty">
          <input type=text name=timetxt size="8">
       </form>
   </center>
</body>
</html>
```

Embedding JavaScript in HTML

Client-side JavaScript code is embedded within HTML documents in a number of ways:

- Between a pair of `<script>` and `</script>` tags
- From an external file specified by the `src` attribute of a `<script>` tag
- In an event handler, specified as the value of an HTML attribute such as `onclick` or `onmouseover`
- As the body of a URL that uses the special `javascript:` protocol

The <script> Tag

Client-side JavaScript scripts are part of an HTML file and are coded within `<script>` and `</script>` tags

You may place any number of JavaScript statements between these tags; they are executed in order of appearance, as part of the document loading process. `<script>` tags may appear in either the `<head>` or `<body>` of an HTML docume

Including JavaScript Files

As of JavaScript 1.1, the `<script>` tag supports a `src` attribute. The value of this attribute specifies the URL of a file containing JavaScript code. It is used like this:

```
<script src="../../javascript/util.js"></script>
```

A JavaScript file typically has a *.js* extension and contains pure JavaScript, without `<script>` tags or any other HTML.

Event Handlers

JavaScript code in a script is executed once, when the HTML file that contains it is read into the web browser. A program that uses only this sort of static script cannot dynamically respond to the user. More dynamic programs define event handlers that are automatically invoked by the web browser when certain events occur

```
<input type="checkbox" name="opts" value="ignore-case"
       onclick="ignore-case = this.checked;"
>
```

Cookies

- Cookies are data, stored in small text files, on your computer.
- When a web server has sent a web page to a browser, the connection is shut down, and the server forgetseverything about the user.
- Cookies were invented to solve the problem "how to remember information about the
- user": When a user visits a web page, his/her name can be stored in a cookie.
- Next time the user visits the page, the cookie "remembers" his/her name.

Create a Cookie with JavaScript

JavaScript can create, read, and delete cookies with the document.cookie property. With JavaScript, a cookie can becreated like this:

document.cookie = "username=John Doe";