

SAVITRIBAI PHULE PUNE UNIVERSITY

LAB COURSE III

**INTERNET PROGRAMMING,
NETWORKING & PROJECT**

(COURSE CODE:CS-349)

T.Y.B.Sc.(COMPUTER SCIENCE)

SEMESTER – II

Name_____

College Name_____

Roll No_____ Division_____

Academic Year_____

Internal Examiner : -----External Examiner : -----

PREPARED BY PHP ASSIGNMENT

MRS. SHILPA KHADILKAR (KAVERI COLLEGE)
MRS. SANGEETA RAUT (MODERN COLLEGE)
MR. BHUPESH TAUNK (FERGUSSON COLLEGE)
MRS. SARITA BYAGAR (INDIRA COLLEGE)
MR. SARFARAZ SHAIKH (INDAPUR COLLEGE)
MRS PRITI EDGAONKAR (SP COLLEGE)

Reviewed By :
MR JEEVAN LIMAYE(FERGUSSON COLLEGE)

BOARD OF STUDY (COMPUTER SCIENCE) MEMBERS: DR. SHAILAJA C. SHIRWAIKAR

About The Work Book

Objectives –

1. The scope of the course.
2. Bringing uniformity in the way course is conducted across different Colleges.
3. Continuous assessment of the students.
4. Providing ready references for students while working in the lab.

How to use this book?

This book is mandatory for the completion of the laboratory course. It is a measure of the performance of the student in the laboratory for the entire duration of the course.

Instructions to the students

- 1) Students should carry this book during practical sessions of Computer Science.
- 2) Printouts of the source code and output is not compulsory but optional.
- 3) Students should read the topics mentioned in reading section of this Book before coming for practical.
- 4) Students should solve all exercises which are selected by Practical in-charge.
- 5) Students will be assessed for each exercise on a scale of 5

1	Not done	0
2	Incomplete	1
3	Late complete	2
4	Needs improvement	3
5	Complete	4
6	Well-done	5

PHP Assignment Completion Sheet

Sr. No.	Assignment Name	Marks (out of 5)	Sign
6	Form designing (HTML & CSS) and cookies & sessions		
7	Drupal – a content management system		
8	Email Handling With PHP		
9	XML		
10	Javascript		
11	AJAX		
	Total out of 30		
	Total out of 05		

Head,

Dept. of Computer Science

ASSIGNMENT NO. 6 : FORM DESIGNING (HTML & CSS) AND COOKIES & SESSIONS

While designing web pages main focus is on frequently changing part of the web page and not changing part of the webpage. To decide such things

Header of Web Page		
Left Side Bar	Body of Web Page	Right Side Bar
Footer of Web Page		

designer has to think over layout of the web page.

This is one of the page layouts. This can be implemented using HTML Table tag.

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
  <table>
    <tr> <td> Header of Web Page </td>
    </tr>
    <tr>
      <td> Left Side Bar </td>
      <td> Body of Web Page </td>
      <td> Right Side Bar </td>
    </tr>
    <tr>
      <td> Footer of Web Page </td>
    </tr>
  </table>
</body>
</html>
```

If we execute this code these borders are changing according to size of the content in it . So avoid variations we can fix the borders of the tables and its rows and columns. For this and look and feel CSS actively works.

Table Borders

To specify table borders in CSS, use the border property.

The example below specifies a black border for <table>, <th>, and <td> elements:

```
table, th, td {  
    border: 1px solid black;  
}
```

Collapse Borders

The border-collapse property sets whether the table borders are collapsed into a single border or separated:

```
table {  
    border-collapse: collapse;  
}
```

```
table, th, td {  
    border: 1px solid black;  
}
```

Table Width and Height

Width and height of a table is defined by the width and height properties.

The example below sets the width of the table to 100%, and the height of the <th> elements to 50px:

```
table {  
    width: 100%;  
}
```

```
th {  
    height: 50px;  
}
```

Horizontal Text Alignment

The text-align property sets the horizontal alignment, like left, right, or center.

By default, the text in <th> elements are center-aligned and the text in <td> elements are left-aligned.

The following example left-aligns the text in <th> elements:

```
th {  
    text-align: left;  
}
```

Table Padding

To control the space between the border and content in a table, use the padding property on <td> and <th> elements:

```
td {  
    padding: 15px;  
}
```

Table Color

The example below specifies the color of the borders, and the text and background color of <th> elements:

```
table, td, th {  
    border: 1px solid green;  
}
```

```
th {  
    background-color: green;  
    color: white;  
}
```

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
table, td, th {  
    border: 1px solid green;  
}  
  
th {  
    background-color: green;  
    color: white;  
}  
  
</style>  
  
</head>  
<body>  
<table>  
<tr>  
    <td> Header of Web Page </td>  
</tr>  
<tr>  
    <td> Left Side Bar </td>  
<td> Body of Web Page </td>
```

```

<td> Right Side Bar </td>
</tr>
<tr>
<td> Footer of Web Page </td>
</tr>
</table>
</body>
</html>

```

You can use CSS tag in your HTML code by using <style> tag or you can define CSS file where all properties are present.

Ex.

Table.css

```

.table100{
    font-family: Arial, Helvetica, sans-serif;
    width: 100%;
    border: 0px solid #3499cd;
}
.td25{
    width : 25%;
}
.td50{
    width : 50%;
}

```

```

<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" href="table.css ">
</head>
<body>
<table class = table100>
<tr>
    <td> Header of Web Page </td>
</tr>
<tr>
    <td class = td25> Left Side Bar </td>
    <td class = td50> Body of Web Page </td>
    <td class = td25> Right Side Bar </td>
</tr>
<tr>
    <td> Footer of Web Page </td>
</tr>

```

```
</table>
</body>
</html>
```

<http://www.w3schools.com> provides CSS as well as HTML information.

Cookie is a small amount of data stored by the user's browser in compliance with a request from a server or script.

Session allow us to easily create multi page forms, save user authentication information from page to page, and store persistent user preferences on a site. A session can be defined as a series of related interactions between a single client and the Web server. The session may consist of multiple requests to the same script or a variety of different resources on the same web site.

Function	Purpose	Example
setcookie (name [, value [, expire [, path [, domain [, secure]]]]]); name – A unique name for a particular cookie. Value – string value attached to this cookie. Expire – expiration date is specified as no of second since midnight Jan, 1, 1970 GMT. Path – the browser will return the cookie only for URL, below this path. Domain – the browser will return the cookie only for URLs within this domain.	Create cookie to remember data in traversal multiple pages	Secure – the default is false. <pre><?php setcookie("item", "TV", time()+3600, "/", ".yourdomain.com", 0); if(isset(\$_COOKIE["item"])) { echo "Hello again, you have chosen : " . \$_COOKIE["item"]; } else { echo "Hello you. This may be your first visit."; } ?></pre>
bool session_start ([array \$options = []])	to enable session for a page. This function assigns a new session ID to the new session	<pre><?php session_start(); ?></pre>
bool session_register (<u>mixed</u> \$name [, <u>mixed</u> \$...])	– to register a variable with the session by passing the name of the variable. When a session is started, you can store any number of variables in the \$_SESSION	<pre><?php session_start(); session_register('hits'); ++\$hits; ?></pre> This page has been viewed <?= \$hits ?> times.

	superglobal array and then access them on any session enabled page.	
string session_id ([string \$id])	session_id() is used to get or set the session id for the current session.	<pre><?php session_start(); \$a = session_id(); echo \$a; ?></pre>
bool session_destroy (void)	session_destroy() destroys all of the data associated with the current session. It does not unset any of the global variables associated with the session, or unset the session cookie. To use the session variables again, <u>session_start()</u> has to be called.	<pre><?php session_start(); ?> <!DOCTYPE html> <html> <body> <?php // remove all session variables session_unset(); // destroy the session session_destroy(); ?> </body> </html></pre>
bool session_unregister (string \$name)	session_unregister() unregisters the global variable named name from the current session.	

Cookies are very powerful and reliable method of storing small piece of data needed to be stored on client's machine but not permanently between separate visits to a Web site

- A Cookie is a small part of data that can be used to store a variable's name, its value along with the information on the site from which it came and its expiry time.
- Cookies provide client-side storage in files present on client machine's hard drive.
- Cookies can be accessed and changed by the web server from which they were sent originally.

Set A :

- 1) Design a webpage for the following layout
For Student profile – where student rno, name, contact, photo, class in column 1 and area of interest, subjects , topics(has read more link) in column 2

Banner (Logo of website and Name of website with image) (consider 20% of the web page height as this part)	
Column 1 40 % of web page should be consider of total width 50 % of height	Column 2 This part will be 40 % 50 % of height
Company name copy rights... 25 of height	

- 2) Create a login form with a username and password. Once the user logs in, the second form should be displayed to accept user details (name, city, phoneno). If the user doesn't enter information within a specified time limit, expire his session and give a warning.

Set B :

- 1) Design a webpage of three columns with following data
Information for placement Like personal details – in one column,
Educational details – in second column, technical skills-in third column
- 2) Change the preferences of your web page like font style, font size, font color, background color using cookie. Display selected settings on next web page and actual implementation (with new settings) on third web page.
- 3) Create a form to accept student information (name, class, address).
Once the student information is accepted, accept marks in next form (Phy, Bio , Chem, Maths, Marathi, English) .Display the mark sheet for the student in the next form containing name, class, marks of the subject, total and percentage.

Set C:

- 1) Design layout similar to your department webpage.
- 2) Write a program to create a shopping mall. User must be allowed to do purchase from two pages. Each page should have a page total. The third page should display a bill, which consists of a page total of whatever the purchase has been done and print the total. (Use http session tracking).
- 3) Create a form to accept customer information(name, address, ph-no).Once the customer information is accepted, accept product information in the next form(Product name, qty, rate). Display the bill for the customer in the next form. Bill should contain the customer information and the information of the products entered.

Signature of the instructor:----- Date:-----

Assignment Evaluation

0:Not Done	<input type="text"/>	2: Late Complete	<input type="text"/>	4:Complete	<input type="text"/>
1: Incomplete	<input type="text"/>	3: Needs Improvement	<input type="text"/>	5:Well Done	<input type="text"/>

(Designed by : Shilpa Khadilkar, Kaveri College)

ASSIGNMENT NO. 7: DRUPAL – A CONTENT MANAGEMENT SYSTEM

Drupal is open source software that allows publishing, managing and organizing a wide variety of content on a website easier. Drupal is used to easily manage, update and publish the content in the website. Many individuals and organizations are using Drupal to create professional websites to suit their custom requirements. Because of easy creating sites, application and management, Drupal is used by many organizations. We can enhance the functionality of Drupal by adding available add-on modules.

Creating Contents

You can add two types of contents in your website: Article and Basic Page. To create a content click the link “Add content.” From the short cut menu. Choose between Article and Basic page.

Creating Articles

Content type Article has the following features:

- Summary posted to the front page of the Web site.
- Comments enabled.
- An image can be displayed with the article.
- User name of the article author as well as the time it was originally published.
- Tags enabled, allowing you to categorize articles.

To create an article, do the following steps:

- From the shortcut menu, click the link “Add content.” An overlay will appear prompting you to choose between Article and Basic page.
- Click “Article”.
- Enter a title and body for your page.
- Scroll to the bottom and click Save.

Creating Basic Page

Content type Basic Page has the following features:

- Are not published to the front page of your Web site.
- Do not allow visitors to post comments.
- Do not have tagging enabled.
- Do not have an image upload widget.
- Are not date-stamped.

To create a Basic Page, do the following steps:

- From the shortcut menu, click the link “Add content.” An overlay will appear prompting you to choose between Article and Basic page.
- Click “Basic page”.
- Enter a title and body for your page.
- Scroll to the bottom and click Save.

Customizing the Display

Use the following steps to change the theme and logo image of your website:

- Using the administrative dashboard, click the tab Appearance.
- Scroll down to the bottom of the screen (where all the disabled themes live), and beneath your theme’s screen shot, click the link Enable and set default.”

- After the screen refreshes, click the settings link for your theme.
- Scroll to the fieldset “Logo image settings.” Unselect the check box “Use the default logo.” A new set of settings will be revealed.
- Click Browse and find your logo image for this theme on your hard drive.
- Scroll to the bottom of the screen and click “Save configuration.”

Blocks

Blocks can be placed into any region in your theme.

To create Block, do the following steps:

- Using the administrative dashboard, navigate to Structure > Blocks.
- Click the link “Add block”.
- Enter description and the text.
- Scroll to the bottom and click “Save block.”

Modules

The modules are used to create, edit, and delete content; convert URLs into specific database requests to retrieve content; and create the menus you use to navigate your Web site.

Modules are little programs that allow you to do more things with your Web site. Modules are set of files contained in a Drupal folder. These files may include the following:

- An information file that describes the module to Drupal. This file lists the version, files within the module directory, configuration screen shots, and a short description of the module. This file is required.
- Installation instructions for Drupal that create the necessary database tables for the module. This file is required.
- PHP scripts that hook into Drupal and allow you to perform specific tasks.
- Template files responsible for the output of the module. These template files can be altered by your theme. These files are optional.
- CSS files, JavaScript files, and images. These files are optional.

SET A

1. Create a Basic Page in Drupal titled “About Me”. Add the details about yourself in the page. Also place this page link in the Main Menu. Display this menu link before all the menu items. Show text “This is <your name>” when move the mouse pointer at this menu link.
2. Create a front-page article in Drupal titled “My Article”. Write an article about PHP programming Language and add to the article page. Display an Image appropriate to the Article at the bottom of the Article. Also place this page link in the Main Menu. Display this menu link before all the menu items. Show text “This is <your name>” when move the mouse pointer at this menu link. Also post a comment about your article.

SET B

1. Create a Block in Drupal titled “Tutorials”. The block should be displayed in the left side of each page. The block should contain announcement about an upcoming event in your college. Also change the theme of your website by following properties:
 - a) Change the background colour.
 - b) Change the logo image.
2. Develop a module in Drupal to create a page showing your contact details (name, roll_no, address, phone). Also add Navigation on the Home Page called “Contact Details”.

SET C

1. Develop a module in Drupal to design a registration form with the following fields:
 - Text Field – First Name, Last Name, email, city
 - List Boxes – Select Country, Date of Birth (Separate Select Boxes for month, day, and year)
 - Radio Buttons – Gender - Male/Female
 - Check Boxes – Technology Known – Java, PHP
 - One Browse button to upload picture.
 - Perform validation to check if the First Name and Last Name are not empty and the email is valid. If that is not the case display error message and the form will not be submitted.
 - Display message “Form has been submitted successfully” after clicking on the Submit button.
 - Also add a Navigation on the Home Page called “Registration”.
2. Create a module in Drupal To design a form with the following components:
 - Text Fields - Roll No, Name, and Address
 - One submit button.
 - After submitting the form insert a student record into a table named „student“. Also display a message when the record is inserted successfully, and fetch the name of student from the table and display “Hello: <student name>”. Also add a Navigation on the Home Page called “Student Form”.

Signature of the instructor:----- Date:-----

Assignment Evaluation

0:Not Done	<input type="text"/>	2: Late Complete	<input type="text"/>	4:Complete	<input type="text"/>
1: Incomplete	<input type="text"/>	3: Needs Improvement	<input type="text"/>	5:Well Done	<input type="text"/>

(Designed by : Mr. Bhupesh Taunk, Fergusson College, Pune)

ASSIGNMENT NO. 8 : EMAIL HANDLING WITH PHP

Objective : To understand Email Handling in PHP

Email : Email or Electronic mail is the transmission of messages over electronic networks like the internet.

The **mail()** function allows you to send emails directly from a PHP script.

This function requires three mandatory arguments that specify the recipient's *email address*, the *subject* of the message and the actual *message* additionally there are other two optional parameters.

Syntax :

```
mail( to, subject, message, [headers], [parameters] );
```

Parameter	Required/ Optional	Description
To	Required	Specifies the receiver / receivers of the email
Subject	Required	Specifies the subject of the email. This parameter cannot contain any newline characters
Message	Required	Defines the message to be sent. Each line should be separated with a LF (\n). Lines should not exceed 70 characters
Headers	Optional	Specifies additional headers, like From, Cc, and Bcc. The additional headers should be separated with a CRLF (\r\n)
Parameters	Optional	Specifies an additional parameter to the sendmail program

As soon as the mail function is called PHP will attempt to send the email then it will return true if successful or false if it is failed.

Example :

```
$to = "sar@yahoo.com";  
$subject = "Sending email using PHP";  
$message = "This is simple text message sending via php script";  
mail($to,$subject,$message);
```

```
<html>
  <head>
    <title>Sending email using PHP</title>
  </head>
  <body>
    <?php
      $to = "sar@yahoo.com";
      $subject = "Sending email using PHP";
      $message = "This is simple text message sending via php script";
      $header = "From:unipune@yahoo.com \r\n";
      $mail= mail ($to,$subject,$message,$header);
      if( $mail==true)
      {
        echo"Message sent successfully...";
      }
      else
      {
        echo"Message sending Error...";
      }
    ?>
  </body>
</html>
```

SET A :

1. Write a PHP script to accept following details form a user (To, From, Subject, Text) and send email to particular user.
2. Write a PHP script to design a form to sending an email to more than one recipient.

SET B :

1. Write a PHP script to Validate given email ID. Design necessary screen layouts.
2. Write a PHP script to design a form to compose/write an email with file attachment.

SET C :

1. Write a PHP project script to design HTML Login form, validate it and compose/write an email with the following details:
To:
From:
Subject:
Message:
2. Design interactive WebPages in PHP to login, validation, and email sending with file attachment to one or more recipient.

Signature of the instructor:----- Date:-----

Assignment Evaluation

0:Not Done	<input type="checkbox"/>	2: Late Complete	<input type="checkbox"/>	4:Complete	<input type="checkbox"/>
1:Incomplete	<input type="checkbox"/>	3: Needs Improvement	<input type="checkbox"/>	5:Well Done	<input type="checkbox"/>

(Designed by :Mr. Sarfaraz Shaikh, Indapur College)

ASSIGNMENT NO. 9 : XML

XML is a data format for standardized structured document exchange.

simplexml_load_string()	<p>This function is used to read XML data from a string.</p> <p>Ex. <?php \$myXMLData = "<?xml version='1.0' encoding='UTF-8'?> <note> <to>Tove</to> <from>Jani</from> <heading>Reminder</heading> <body>Don't forget me this weekend!</body> </note>";</p> <p>\$xml=simplexml_load_string(\$myXMLData) or die("Error: Cannot create object"); print_r(\$xml); ?></p>
-------------------------	---

Set A:

- 1) Write a script to create XML file named "Course.xml"

```
<Course>  
  <Computer Science>  
    <Student name>.....</Student name>  
    <Class name>.....</Class name>  
    <percentage>.....</percentage>  
  </Computer Science>  
</Course>
```

Store the details of 5 students who are in TYBSc.

- 2) Link "Course.xml" file to the CSS style sheet and get well formatted output as given below

Property Name	Student Name	Class Name	Percentage
Color	Blue	Green	Red
Font-family	Bodoni MT	Arial, Helvetica, sans-serif	Impact, Charcoal, sans-serif
Font-style:	italic	Normal	oblique
Font Size	16 pts	12pts	14pts

Set B:

- 1) Write PHP script to generate an XML code in the following format

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<CD Store>
  <Movie>
    <Title>Mr. India</Title>
    <Release Year>1987</ Release Year >
  </Movie>
  <Movie>
    <Title>Holiday</Title>
    <Release Year>2014</ Release Year >
  </Movie>
  <Movie>
    <Title>LOC</Title>
    <Release Year>2003</ Release Year >
  </Movie>
</CD Store>
```

- 2) Create a XML file which gives details of movies available in “Mayanagari CD Store” from following categories

- a) Classical
- b) Action
- c) Horror

Elements in each category are in the following format

```
<Category>
  <Movie Name>----</Movie Name>
  <Release Year>----</Release Year>
</Category>
```

Save the file with name “movies.xml”.

Set C

- 1) Create an application that reads “book.xml” file into simple XML object. Display attributes and elements (Hint:simple_xml_load_file() function)

2) Write a script to create “cricket.xml” file with multiple elements as given below

```
<Cricket team>  
<Country = India>  
    <Player Name >----- <Player Name >  
    <Wickets>----- </Wickets>  
    <Runs>-----</Runs>  
</Country>  
</Cricket team>
```

Also add country = “England” and its elements

Signature of the instructor:-----Date:-----

Assignment Evaluation

0:Not Done	<input type="checkbox"/>	2: Late Complete	<input type="checkbox"/>	4:Complete	<input type="checkbox"/>
1: Incomplete	<input type="checkbox"/>	3: Needs Improvement	<input type="checkbox"/>	5:Well Done	<input type="checkbox"/>

(Designed by : Ms. Shilpa Khadilkar, Kaveri College and Ms. Priti Edgaonkar, SP College)

ASSIGNMENT NO. 10 : JAVASCRIPT

Javascript is basically designed to create interactivity with HTML pages.

Javascript enables you to read and change the content of HTML controls.

Javascript enables you to load a specific page depending upon the client's request.

Javascript helps you to do certain validations on client side.

Data types: Number, String, Boolean, Undefined, NULL

Variables: Value of a variable can change during the script. No need to declare a variable.

Operators: Operators available in Javascript are same as that of PHP , C programming.

Conditional statements and loops: In Javascript , the syntax of Conditional statements and loops are same as that of PHP, C programming.

Javascript Object

Objects in Javascript are divided into three categories.

1. Built-in objects
2. Browser Objects
3. User-defined objects.

1. Built-in Objects:

Array, string, math, Date are commonly used built-in objects.

Array: Using Keyword **new**, you can create an instance of the object.

For ex. `Var myarray=new Aarray();`

String: String object is used to manipulate a stored piece of text.

`Var text ="PHP and Javascript"`

`Document.write(text.length);`

Math: This object is used to perform common mathematical tasks .

Javascript provides eight mathematical values that can be accessed from the math object.

These are

Math.E

Math.PI

Math.SQRT2

Math.SQRT 1_2

Math.LN2

Math.LN10

Math.LOG2E

Math.LOG10E

Ex. Round method is used to round a number

```
Document.write(Math.round(4.7))
```

Date: This object works with date and time

```
Var myDate=new Date()
```

```
myDate.setFullYear(2015, 0, 20)
```

2. Browser Objects :

BOM is a collection of objects that interact with the browser window.

These objects include the **Window object, history object, location object, navigator object, screen object and document object.**

The window object method is the top object in BOM hierarchy. The window object is used to move, resize windows , create a new windows. Window object is also used to create dialogue boxes such as alert boxes. Some commonly used **methods** of window object are **open, close, confirm, alert,prompt etc.**

Document Object Model

The document Object Model (DOM) is a tree- based representation of a document. The DOM was created by **World Wide Web Consortium(w3c)** for XML and HTML/XHTML. The DOM provides a set of objects for representing the structure of the document, as well as for accessing those objects.

Methods available in DOM for accessing objects:

1) getElementById() method:

This method returns the element with the specified ID.

Refer the examples given below.

2) getElementByTagName() method:

This method returns all elements with the specified tag name.

Refer the examples given below.

1) simple javascript example

```
<!DOCTYPE html>
<html>
<body>
<script type="text/javascript">
Document.write("Hello World!")
</script>
</body>
</html>
```

2) javascript example using 'alert box ' window object.

```
<!DOCTYPE html>
<html>
  <body>
    <h1>My First Web Page</h1>
    <p>My first paragraph.</p>
    <script>
      window.alert(5 + 6);
    </script>
  </body>
</html>
```

3) javascript example using 'document.write method'.

```
<!DOCTYPE html>
<html>
  <body>
    <h1>My First Web Page</h1>
    <p>My first paragraph.</p>
    <button onclick="document.write(5 + 6)">Try it</button>
  </body>
</html>
```

Popup boxes :JavaScript has three kind of popup boxes: Alert box, Confirm box, and Prompt box.

Alert Box: An alert box is often used if you want to make sure information comes through to the user.

When an alert box pops up, the user will have to click "OK" to proceed.

Syntax

```
window.alert("sometext");
```

The window.alert() method can be written without the window prefix.

Example

```
<!DOCTYPE html>
<html>
  <body>
    <p>Click the button to display an alert box:</p>
    <button onclick="myFunction()">Try it</button>
    <script>
      function myFunction() {
        alert("I am an alert box!");
      }
    </script>
  </body>
</html>
```

Confirm Box : A confirm box is often used if you want the user to verify or accept something. When a confirm box pops up, the user will have to click either "OK" or "Cancel" to proceed. If the user clicks "OK", the box returns true. If the user clicks "Cancel", the box returns false.

Syntax

```
window.confirm("sometext");
```

The window.confirm() method can be written without the window prefix.

Example

```
var r = confirm("Press a button");
if (r == true) {
  x = "You pressed OK!";
} else {
```

```
x = "You pressed Cancel!";  
}
```

Prompt Box : A prompt box is often used if you want the user to input a value before entering a page.

When a prompt box pops up, the user will have to click either "OK" or "Cancel" to proceed after entering an input value.

If the user clicks "OK" the box returns the input value. If the user clicks "Cancel" the box returns null.

Syntax

```
window.prompt("sometext","defaultText");
```

The window.prompt() method can be written without the window prefix.

Example

```
var person = prompt("Please enter your name", "Harry Potter");  
if (person != null) {  
    document.getElementById("demo").innerHTML =  
    "Hello " + person + "! How are you today?";  
}
```

HTML Events

An HTML event can be something the browser does, or something a user does.

Here are some examples of HTML events:

An HTML web page has finished loading

An HTML input field was changed

An HTML button was clicked

Often, when events happen, you may want to do something.

JavaScript lets you execute code when events are detected.

HTML allows event handler attributes, with JavaScript code, to be added to HTML elements.

With single quotes:

```
<some-HTML-element some-event='some JavaScript'>
```

With double quotes:

```
<some-HTML-element some-event="some JavaScript">
```

Common HTML Events : Here is a list of some common HTML events:

Event	Description
Onchange	An HTML element has been changed
OnClick	The user clicks an HTML element
Onmouseover	The user moves the mouse over an HTML element
Onmouseout	The user moves the mouse away from an HTML element
Onkeydown	The user pushes a keyboard key
onload	The browser has finished loading the page

In the following example, an **onclick** attribute (with code), is added to a button element:

Example

```
<!DOCTYPE html>
<html>
  <body>
    <button      onclick="getElementById('demo').innerHTML=Date()">The
    time is?</button>
    <p id="demo"></p>
  </body>
</html>
```

4)javascript example using 'document.getElementById(id) method.'

```
<!DOCTYPE html>
<html>
  <body>
    <h1>My First Web Page</h1>
    <p>My First Paragraph</p>
    <p id="demo"></p>
    <script>
      document.getElementById("demo").innerHTML = 5 + 6;
    </script>
  </body>
</html>
```

This is the easiest way to create a JavaScript Object.

Using an object literal, you both define and create an object in one statement.

An object literal is a list of name: value pairs (like age:50) inside curly braces {}.

The following example creates a new JavaScript object with four properties:

```
<!DOCTYPE html>
<html>
  <body>
    <p>Creating a JavaScript Object.</p>
    <p id="demo"></p>
    <script>
      var person = {
        firstName : "John",
        lastName  : "Doe",
        age       : 50,
        eyeColor  : "blue"
      };
      document.getElementById("demo").innerHTML =
        person.firstName + " is " + person.age + " years old.";
    </script>
  </body>
</html>
```

Using the JavaScript Keyword new

The following example also creates a new JavaScript object with four properties:<!DOCTYPE html>

```
<html>
  <body>
    <p id="demo"></p>
    <script>
      var person = new Object();
      person.firstName = "John";
```

```

        person.lastName = "Doe";
        person.age = 50;
        person.eyeColor = "blue";
        document.getElementById("demo").innerHTML =
            person.firstName + " is " + person.age + " years old.";
    </script>
</body>
</html>

```

Using an Object Constructor

The examples above are limited in many situations. They only create a single object.

Sometimes we like to have an "object type" that can be used to create many objects of one type.

The standard way to create an "object type" is to use an object constructor function:

Example

```

function person(first, last, age, eye) {
    this.firstName = first;
    this.lastName = last;
    this.age = age;
    this.eyeColor = eye;
}
var myFather = new person("John", "Doe", 50, "blue");
var myMother = new person("Sally", "Rally", 48, "green");

```

JavaScript Function Definitions

JavaScript functions are defined with the function keyword.

You can use a function declaration or a function expression.

Function Declarations

Earlier in this tutorial, you learned that functions are declared with the following syntax:

```

function functionName(parameters) {
    code to be executed
}

```

Declared functions are not executed immediately. They are "saved for later use", and will be executed later, when they are invoked (called upon).

Example

```
<!DOCTYPE html>
<html>
  <body>
    <p>This example calls a function which performs a calculation,
    and returns the result:</p>
    <p id="demo"></p>
    <script>
      function myFunction(a, b) {
        return a * b;
      }
      document.getElementById("demo").innerHTML =
      myFunction(4, 3);
    </script>
  </body>
</html>
```

SET A:

- 1) Write a javascript to display message 'Good Morning' using alert box.
- 2) Write a javascript to display message 'Good Afternoon' using function.(Hint: use Event 'Onload').

SET B:

- 1) Write a javascript function to validate username and password for a membership form.
- 2) Using Javascript function, display the string in different formatting styles(Bold, italic, underline, strikethrough, hypertext etc)

SET C:

- 1) Write a Javascript to create a FIFO queue . Insert new element in it (Hint: Use concept of Array)

2) Write a Javascript program to accept name of student, Change font color to red, font size to 18 if student name is present otherwise on clicking on empty text box display image which change its size (Use onblur, onload, onmouseover, onmouseleave, onmouseup)

Signature of the instructor:----- Date:-----

Assignment Evaluation

0:Not Done	<input type="checkbox"/>	2: Late Complete	<input type="checkbox"/>	4:Complete	<input type="checkbox"/>
1: Incomplete	<input type="checkbox"/>	3:Needs Improvement	<input type="checkbox"/>	5:Well Done	<input type="checkbox"/>

(Designed by: Ms. Sangita Raut, Modern College and Ms. Priti EdgaonKar, SP College)

ASSIGNMENT NO. 11 : AJAX

AJAX = Asynchronous JavaScript and XML.

AJAX is not a new programming language, but a new way to use existing standards.

AJAX is the art of exchanging data with a server, and updating parts of a web page - without reloading the whole page.

XMLHttpRequest is a JavaScript object capable of calling the server and capturing its response. It is used to send HTTP or HTTPS requests to a web server and load the server response data back into the script.

Create an XMLHttpRequest Object

All modern browsers (IE7+, Firefox, Chrome, Safari, and Opera) have a built-in XMLHttpRequest object.

Syntax for creating an XMLHttpRequest object:

```
xmlhttp=new XMLHttpRequest();
```

When a request to a server is sent, we want to perform some actions based on the response.

The onreadystatechange event is triggered every time the readyState changes.

The readyState property holds the status of the XMLHttpRequest.

Three important properties of the XMLHttpRequest object:

Property	Description
onreadystatechange	Stores a function (or the name of a function) to be called automatically each time the readyState property changes
readyState	Holds the status of the XMLHttpRequest. Changes from 0 to 4: 0: request not initialized 1: server connection established 2: request received 3: processing request 4: request finished and response is ready

status	200: "OK" 404: Page not found
--------	----------------------------------

In the `onreadystatechange` event, we specify what will happen when the server response is ready to be processed.

The `onreadystatechange` event is triggered five times (0-4), one time for each change in `readyState`.

To get the response from a server, use the `responseText` or `responseXML` property of the `XMLHttpRequest` object. The `responseText` property returns the response as a string.

To send a request to a server, we use the `open()` and `send()` methods of the `XMLHttpRequest` object:

Method	Description
<code>open(method,url,async)</code>	Specifies the type of request, the URL, and if the request should be handled asynchronously or not. <i>method</i> : the type of request: GET or POST <i>url</i> : the location of the file on the server <i>async</i> : true (asynchronous) or false (synchronous)
<code>send(string)</code>	Sends the request off to the server. <i>string</i> : Only used for POST requests

Set A :

- 1) Write Ajax program to read a textfile and print the contents of the file when the user clicks on the Print button.
- 2) Write Ajax program to carry out validation for a username entered in textbox. If the textbox is blank, print 'Enter username'. If the number of characters is less than three, print 'Username is too short'. If value entered is appropriate the print 'Valid username'.

Set B :

- 1) Create employee table as follows
EMP(eno, ename, designation, salary). Write Ajax program to select the employees name and print the selected employee's details.
- 2) Write Ajax program to print Movie details by selecting an Actor's name.
Create table MOVIE and ACTOR as follows with 1 : M cardinality
MOVIE (mno, mname, release_yr) and ACTOR(ano, aname)

Set C :

- 1) Write Ajax program to fetch suggestions when is user is typing in a textbox. (eg like google suggestions. Hint create array of suggestions and matching string will be displayed)
- 2) Write Ajax program to get book details from XML file when user select a book name. Create XML file for storing details of book(title, author, year, price).

Signature of the instructor:----- Date:-----

Assignment Evaluation

0:Not Done	<input type="checkbox"/>	2: Late Complete	<input type="checkbox"/>	4:Complete	<input type="checkbox"/>
1: Incomplete	<input type="checkbox"/>	3: Needs Improvement	<input type="checkbox"/>	5:Well Done	<input type="checkbox"/>

(Designed by : Sarita Byagar, Indira College)