Lab Guide for HJS (HTML & JS) for OS

|  |  |
| --- | --- |
| **Author(s)** |  |
| **Authorized by** |  |
| **Creation/Revision Date** |  |
| **Version** | 1.0 |
|  | |

# COPYRIGHT NOTICE

*All ideas and information contained in this document are the intellectual property of Education and Research Department, Asreet Technologies Limited. This document is not for general distribution and is meant for use only for the person they are specifically issued to. This document shall not be loaned to anyone, within or outside Asreet, including its customers. Copying or unauthorized distribution of this document, in any form or means including electronic, mechanical, photocopying or otherwise is illegal.*

# Document Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Author(s)** | **Reviewer(s)** | **Comments** |
| 1.0 |  |  |  | Draft Version |

# Contents

[COPYRIGHT NOTICE 2](#_TOC_250010)

[DOCUMENT REVISION HISTORY 3](#_TOC_250009)

[CONTENTS 4](#_TOC_250008)

CONTEXT 5

DAY 1 ASSIGNMENTS OF HTML 5

ASSIGNMENT 1: UNDERSTANDING BROWSERS AND HTML 5

ASSIGNMENT 2: WRITING AND VIEWING YOUR FIRST HTML PAGE 7

ASSIGNMENT 3: USING FORMATTING STYLES AND COLOR 9

ASSIGNMENT 4: USING HYPER LINKS IN HTML 13

ASSIGNMENT 5: USING LISTS IN HTML 16

ASSIGNMENT 6: USING IMAGES IN HTML 19

ASSIGNMENT 7: CREATING TABLES IN HTML 21

ASSIGNMENT 8: CREATING INTERACTIVE FORM USING HTML. 27

ASSIGNMENT 9: WORKING WITH FRAMESET AND FRAMES 31

ASSIGNMENT 10: UNDERSTANDING CSS 38

ASSIGNMENT 11 : CSS SYNTAX 40

ASSIGNMENT 12: SOME MORE SELECTORS 42

ASSIGNMENT 13: EXTERNAL STYLE SHEET, <DIV> AND <SPAN> 46

ASSIGNMENT 14: USING TEXT, FONT, BOX AND POSITION PROPERTIES 48

ASSIGNMENT 15: APPLYING STYLES TO TABLES AND FORM ELEMENTS 53

DAY 2 ASSIGNMENTS OF JAVA SCRIPT 55

[ASSIGNMENT 1: UNDERSTANDING CLIENT SIDE JAVASCRIPT 55](#_TOC_250007)

[ASSIGNMENT 2: USING VARIABLES AND OPERATORS 58](#_TOC_250006)

[ASSIGNMENT 3: USING CONTROL STRUCTURES AND LOOPS 60](#_TOC_250005)

[ASSIGNMENT 4: FUNCTIONS AND DIALOG BOXES 64](#_TOC_250004)

[ASSIGNMENT 5: EXTERNAL FILE AND BUILT IN OBJECTS 67](#_TOC_250003)

[ASSIGNMENT 6:WORKING WITH DOM AND IMPORTANT OBJECTS 71](#_TOC_250002)

[ASSIGNMENT 7: WORKING WITH FORM AND ELEMENTS 75](#_TOC_250001)

[ASSIGNMENT 8: REUSABLE FUNCTIONS 81](#_TOC_250000)

**Context**

This document contains the assignments to be completed as part of the hands on for the subject HJS.

**Note:** All assignments in this document must be completed in the sequence in this document in order to complete the course.

**Day 1 Assignments of HTML**

All the assignments in this section must be completed on Day 1 of your HJS course.

**Assignment 1: Understanding Browsers and HTML**

**Objective:** To understand the software requirements for learning HTML and work with different browsers and editors.

**Problem Description:** You have learnt the basics of HTML from IWT course. HTML file is a text file containing markup tags. HTML file can be created using a text editor. You can view the pages with a program called a Web browser.

Popular browsers are **Internet Explorer**(from Microsoft) and **Netscape Navigator** now known as **Mozilla Firefox**(from Netscape Communications). **Opera** is a very popular alternative browser. **Lynx** is a text browser for the World Wide Web.

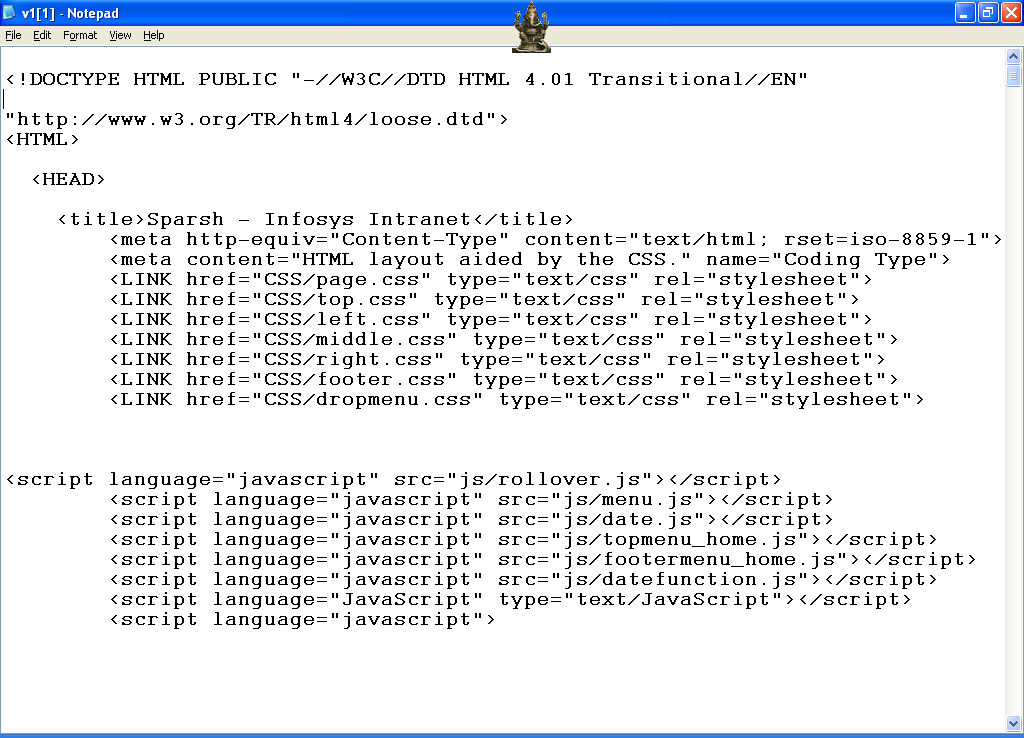
**Note:** You can easily edit HTML files using a WYSIWYG (what you see is what you get) editor like FrontPage, Visual Interdev etc. However, these tools do not generate much optimized HTML code.

As a programmer, you should be able to create and edit HTML files using text editors like Notepad or Textpad.

### Estimated Time:

**Step 1:** Open Internet Explorer(Select the Start Menu-> Internet Explorer)and type <http://sparsh/V1>on the Address bar.

**Step 2:** Select menu option, View -> Source. You can find the HTML tag that creates the Sparsh home page.



**Step 3:** Select menu option, Help -> About Internet Explorer. This should display the version of the Internet Explorer browser you are using.

**Step 4**: What is the version of Internet Explorer you are using?

**Step 5**: Search for a link called DART in the Sparsh home page. (Hint: Use menu option Edit-> Find)

### Summary of this exercise:

You have just learnt

* Using a Web Browser
* Using ―View Source‖ feature of Web browsers to see the actual HTML code
* Finding the version of the browser
* Finding within a web page using the browser‘s Find feature

Questions based on self study slides:

Text Formatting Tags (Heading Tags):

Q1. For which tag, Font size is bigger, <H2> or <H5>? Q2. How many levels of Heading Tags are there?

Q3. If you will write <H1> Heading 1 </H1> <H2> Heading 2 </H2>, Heading 1 Heading 2 will be printed in the same line or in different lines?

**Assignment 2: Writing and Viewing Your First HTML Page**

**Objective:** To write and view your first HTML page and to understand heading elements like

<h1> to <h6> and formatting elements like <b>,<i> and <u>.

**Problem Description:** You have learnt how to use a browser. The HTML display may depend on your browser and version you are using.

**Step 1:** Create a folder assign2 under your work directory (TBD: The directory structure)

**Step 2:** Open a text editor (In Windows, notepad.exe or textpad and in case of UNIX, vi) and type the following:

**<html>**

**<head>**

**<title> First Page</title>**

**</head>**

**<body>**

**<!-- Comment: The text below appears in heading style 2 -->**

**<h2> This is my first HTML page </h2>**

**</body>**

**</html>**

Here, the text enclosed between <h2> and </h2> will appear in heading style2 font. Save the file as ―firstpage.html‖ in the directory Assign2.

(While saving choose **Save As** type to be ―**All files**‖)

**Note:** An HTML file must have an .htm or .html file extension. HTML tags are not case sensitive, <b> means the same as <B>. The World Wide Web Consortium (W3C) recommends lowercase tags in their HTML 4 recommendation, and XHTML (the next generation HTML) demands lowercase tags.

**Step 3:** Double click on the firstpage.html (or) Open the browser, and select menu option File -> Open.

**Step 4:** Type the following code in a text editor and save the file as ―firsterror.html‖.

**<html>**

**<head>**

**<!-- Comment: There may be an error here -->**

**<title> First Page With Errors<title>**

**</head>**

**<body>**

**<h2> This is my first HTML page </h2>**

**</body>**

**</html>**

**Step 5:** Open the firsterror.html file in a browser. What is the output? Why?

**Step 6**: Find the Errors in ―firsterror.html‖ and correct it. Change the title as ―First Page Without errors‖.

**Hint:** All the tags are not closed. Try and find out which tag is not closed before going to Step 7.

**Step 7:** Notice that the closing „title‟ tag has a missing „/‟. Change the closing title tag to </title>

**Note:** All browsers contain an HTML interpreter, which interprets HTML tags so that headings, hyperlinks, tables, etc. appear as intended on the page. Interpreter is a program that reads source code one statement at a time, translates that statement to machine language, executes the machine language statement, then continues with the next statement.

**Step 6:** Create a new HTML page with your name and Address and save it as ―address.html‖, use <br> tag for line breaks.

**Step 7:** Display name in **BOLD** , address in *ITALIC* and underline the pin code.

Text between ―<b> ― and ―</b>‖ tags appear in **bold.**

Text between ―<i> ― and ―</i>‖ tags appear in *italic.*

Text between ―<u> ― and ―</u>‖ tags appear in underline.

**Step 8:** Create a HTML page and display name and address of three of your friends in a readable format. Save it as ―addressfriends.html‖.

**Step 9:** Create a page with the following details and save it as ―headings.html‖.

President -> Display in <h1> heading Style Managing Director -> Display in <h2> heading Style HOD -> Display in <h3> heading Style

Manager -> Display in <h5> heading Style

Worker -> Display in <h6> heading Style

### Summary of this exercise:

You have just learnt

Creating HTML page. Debugging HTML pages.

<br> , <i>, <u>and <b> tag. Heading Tags (h1 to h6).

### Deliverables of the exercise:

1. firstpage.html
2. firsterror.html
3. address.html
4. addressfriends.html
5. headings.html

**Assignment 3: Using Formatting Styles and Color.**

**Objective:** To understand adding of background color to a page, adding color to a text and usage of <font> tag.

**Step 1:** Type the following code and save it as ―bgcolor.html‖ in a folder called assign3.

**<html>**

**<head>**

**<title> Page with background Color</title>**

**</head>**

**<!—Comment: The bgcolor attribute value is a RGB value -->**

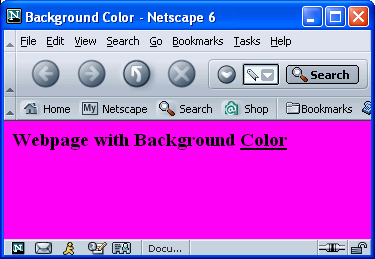
**<body bgcolor="#FF00F0">**

**<h3> Webpage with Background <u>Color</u></h3>**

**</body>**

**</html>**

Verify your output?



**Note:** RGB Color Components.

Color attribute values give a color definition. Colors are defined using a hexadecimal notation for the combination of Red, Green, and Blue color values (RGB). The lowest value that can be given to one light source is 0 (hex #00). The highest value is 255 (hex #FF). A collection of color names like blue, green, cyan etc. are also supported by most of the browsers.

**Step 2:** Modify your HTML page change the bgcolor attribute value as "#009192".

**Step 3:** Type the following code and save it as ―fontcolor.html‖.

**<html>**

**<head>**

**<title> Paragraphs and Color</title>**

**</head>**

**<body bgcolor="cyan">**

**<!-- Comment : <p> Defines Paragraphs in HTML -->**

**<p>**

**All browsers contain an HTML interpreter, which interprets**

**HTML**

**tags so that headings, hyperlinks, tables, etc. appear as intended on the page.**

**</p>**

**<!-- Comment : The <font> tag specifies the font properties -->**

**<font color="#0000FF" face="Monotype Corsiva" size="5">**

**Interpreter is a program that reads source code**

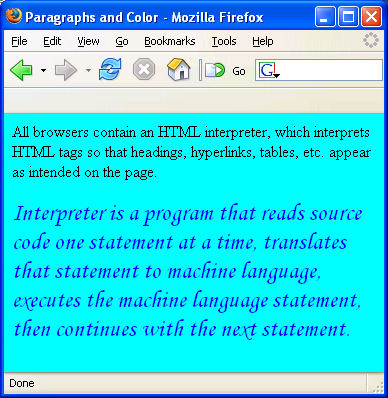
**one statement at a time, translates that statement to machine language, executes the machine language statement, then continues with the next statement.**

**</font>**

**</body>**

**</html>**

**Step 5:** Open the ―fontcolor.html‖ in a browser.



Here <p> is used to display paragraphs and font used to set the font properties like color, face and size. The Font size can take a number from 1 to 7.

**Note:** In HTML the font size is limited to 1 to 7. In future you can use **CSS**(Cascading Style Sheets) for changing the properties ,.i.e., you can have a font with size 10 cm. The font face value depends on the fonts available on the system. Default fonts like ―Arial‖, ―Times New Roman‖, and ―Courier‖ are available in all Systems.

**Note:** There are several reasons why HTML attributes values should be inside double quotes in HTML, although you can use single quotes or without quotes. In XHTML attributes should be inside double quotes. The entire course you are supposed to use attributes values inside double quotes.

**Step 5:** Modify the ―fontcolor.html‖, add align attribute to the <font> tag. ? What is the output? Try all values for align attribute.

**Note:** HTML elements permitted within the [BODY](http://www.htmlhelp.com/reference/html40/html/body.html) are classified as either **block-level elements (**or) [**inline elements**](http://www.htmlhelp.com/reference/html40/inline.html). Block-level elements typically contain inline elements and other block-level elements. Block-level elements usually begin on a new line.

**Step 6:** Verify <font> tag is an inline element or block-level element with an example.

**Step 7:** Modify the ―fontcolor.html‖ add the following code.

**<center>**

**<font color="#0000FF" face="Monotype Corsiva" size="5">**

**</font>**

**</center>**

**Note:** <center> tag centers its enclosed text horizontally.

**Step 8:** Modify the ―fontcolor.html‖ add a <hr> tag and save as ―hr.html‖.

<hr> tag is used to display a horizontal line.

**Step 9:** Include the width, size and color attribute to the <hr> tag.

**Ex: <hr width=‖70%‖ size=‖10‖ color=‖#00FF00‖>**

### Summary of this exercise:

You have just learnt

Using background color for html page. Using <p> and <font> tag.

The <hr> tag.

### Deliverables of the exercise:

1. bgcolor.html
2. fontcolor.html
3. hr.html

Questions based on self study slides:

Physical Text Formatting Tags:

Q1. In order to write H2O which tag will you use? Q2. In order to write A3 which tag will you use? Q3. What is the use of <strike> tag?

Q4. How will you underline the text using HTML?

Q5. To bring an image in the center of the page which tag will you use?

**Assignment 4: Using Hyper Links in HTML**

**Objective:** To understand and create hyperlinks in HTML and using image as hyperlinks.

**Step 1:** Create an HTML page with some contents and save it as ―linkdisp.html‖ in a folder called assign4.

**Step 2:** Create an HTML page with a hyperlink to the ―linkdisp.html‖ and save it as

―link.html‖.

**Ex: <a href=‖linkdisp.html‖>Link Disp Page </a>**

**Note:** When you are creating links to documents and images on the Web the path can be two ways 1) Absolute path and 2) Relative path.

Absolute paths refer to the very specific location like ―c:\html\assign2\fontcolor.html‖ (or)

―<http://sparsh/WebApps/dart/dart1.htm>‖. It is a good idea to use absolute paths, without the domain name,. i.e., like ―/WebApps/dart/dart1.htm‖. Always use forward slash(―/‖) to separate path to achieve platform independent.( In all Operating Systems).

Relative paths change depending upon what page the links are located. Links in the same directory as the page have no path information listed. Sub-directories are listed without any preceding slashes like ―Assign2\_sub/test.html‖. Links up one directory are listed like

―../Assign2/firstpage.html‖.

**Step 3:** Create a HTML page called ―linkpage.html‖ and store it in Assign2 folder. Give relative links to the following files.

1. link.html (Hint: Same folder)
2. assign2/firstpage.html (Hint: Links up one directory)

**Step 4**: Modify the ―linkpage.html‖ and add two more links to the following sites.

1. [http://kshop](http://kshop/)
2. [http://www.yahoo.com](http://www.yahoo.com/)

**Step 5:** Add link, alink and vlink attributes to the <body> tag.

**Ex: <body link=‖blue‖ vlink=‖red‖ alink=‖green‖>**

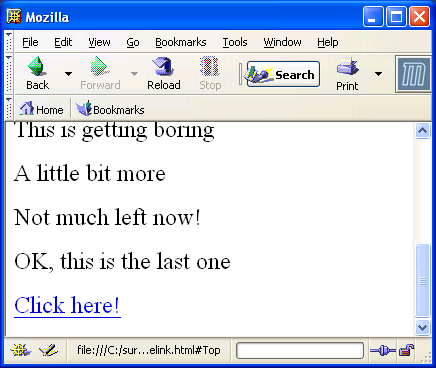
**<a** [**href=‖http://kshop/kshop/default.asp‖>**](http://kshop/kshop/default.asp)

Here link stands for the color of the link, vlink stands for the color of the visited link and alink stands for the active link. (All these apply only to that particular page).

**Note:** If the link is displaying still in visited color, Select menu option Tools-> Internet Options -> Clear History and refresh the page (Press F5 Button). The standard color for a link is blue in all browsers. It is always advisable to use standard color only for the links.

**Step 6:** Create an html file with more than one page content, save it as ―samelink.html‖. Open the page in Browser and make sure that the vertical scrollbar is enabled.

**Step 7:** Create a link on the bottom of the page so that it can go back to the top of the page when you click on it.



**Hint:**

**<a name=‖top‖>Top of the Page</a>**

**………………………………………………………………………… Contents……………………………………………………**

**…………………………………………………………………………**

**<a href=‖#top‖>Click here!</a>**

**<a name=‖bot1‖>Bottom of the Page</a>**

**Step 8:** Open ―link.html‖ and add a link to samelink.html, when it opens it should show the bottom of ―samelink.html‖?

### Hint: Use <a href=‖samelink.html#bot1‖>Open Samelink Bottom</a>

**Step 9:** Type the following code and what is the Error in that?

**<a href=‖samelik.html>Same link <a>**

**Step 10:** Check all the double quotes are closed?

### Summary of this exercise:

You have just learnt

Using hyperlink.

Absolute path and Relative path. Hyperlink to the same document.

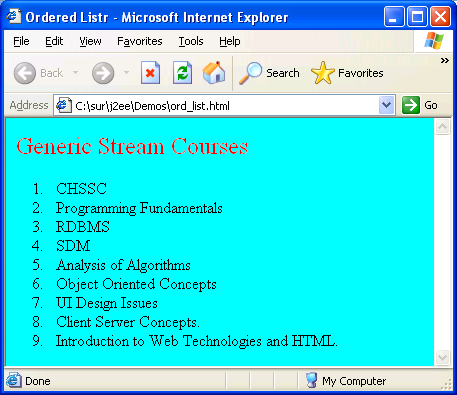
### Deliverables of the exercise:

1. linkdisp.html
2. link.html
3. linkpage.html
4. samelink.html

**Assignment 5: Using Lists in HTML**

**Objective:** To understand how to create ordered, unordered and definition lists in HTML.

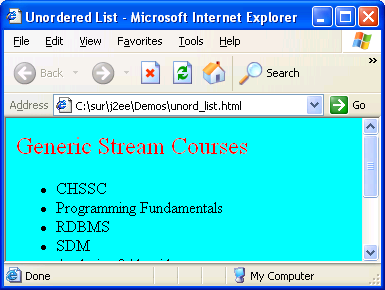
**Step1:** Create an html page and save it as ―ord\_list.html‖ which should look like the following demo.



**Step2:** Display the above list numbering in a,b,c instead of 1,2,3.

**Hint:** Change the type attribute of <ol> tag, the type attribute values are **A, a, i, I, 1.**

**Step3:** Open ―ord\_list.html‖ and save as ―unord\_list.html‖, change the numbering list to bulleted list.



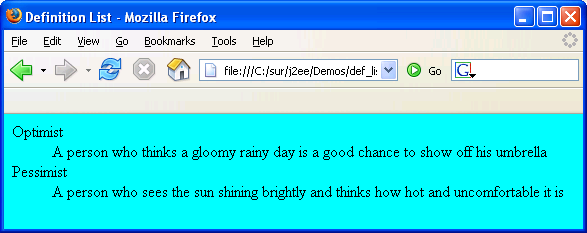
**Step 4:** Change the <ul> type attribute to ―square‖. The possible values are square, disc and circle.

**Step 5:** Change the <ul> type attribute to ―mystyle‖. What is the out put? Why?

**Note:** If the attribute value is not defined in HTML specification the default value will be assigned. The type attribute values are disc, square, circle.

**Step 6:** Create a Definition list that look like the following demo and save it as

―def\_list.html‖.



**Step 7:** Create a html page should look like the following and save as ―nested\_list.html‖.



To provide the font style put the entire list inside a <font> tag.

### Summary of this exercise:

You have just learnt

Ordered List Unordered List Definition Lists Nested List.

### Deliverables of the exercise:

1. ord\_list.html
2. unord\_list.html
3. def\_list.html
4. nested\_list.html

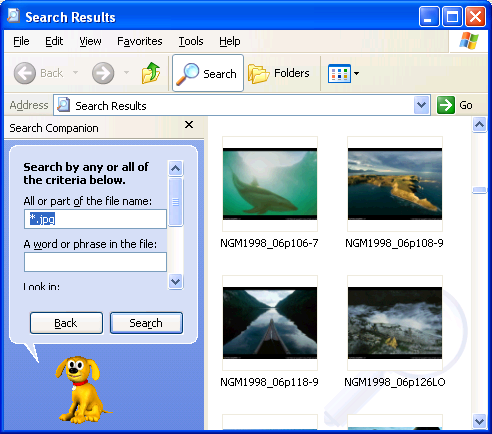
**Assignment 6: Using Images in HTML**

**Objective:** ―A picture is worth a thousand words‖. We will be learning how to use images in your HTML page.

**Step 1**: Create a folder called assign5.

**Step 2:** Select Start Button->Search->For Files or Folders.

**Step 3** : Choose the link ―All Files and Folders‖. Type **\*.jpg** in ―all or part of the filename.‖



**Step 4:** Copy some image files to the Assign5 folder**.**

**Note:** Web pages require JPG (or) GIF (or) PNG image types. On the web, JPG is the best choice (smallest file) for photo images, and GIF is common for graphic images because GIF supports transparent color feature. PNG was designed recently, but it's appeal is growing as people discover what it can do. Most of the Browsers can not show BMP images.

JPEG: Joint Photographic Experts Group.

GIF: Graphic Interchange Format.

PNG: Portable Network Graphics Format.

**Note:** In Windows XP and beyond you can create image files(.jpg, .png, .gif)using Microsoft paint. Use Start button -> Programs -> accessories -> paint. For saving the image use menu option -> File ->Save, Select the save as type for selecting the particular format.( .bmp, .jpg,

.gif, .png, etc.)

**Step 5:** Create a HTML page with two images and save it as ―img.html‖ in assign5 folder. **Hint:** <img src=”img11.jpg” alt=”Image1 1 image‖>. The name of the image should exist in your folder.(You can use absolute or relative path for images)



**Note:** Some elements do not have closing tags (known as empty tag). They do not affect a block of the document in some way. These elements do not require an ending *tag*. <img> is an empty tag. You can use like <img src=”s1.jpg” />

In case the image is missing or the browser is not a graphical browser (e.g.Lynx), the alt text is displayed in place of the image. Text specified inside alt is shown as tool tip text.

**Step 6:** Identify two empty elements from your previous assignments.

**Note:** If width and height is different from the actual size of the image, quality of the image may be degraded.

**Step 7:** Include two more images with suitable size. Use width and height attribute for resizing the image size and save the file as ―img1.html‖.

**<img src=‖s1.jpg‖ width=‖300‖ height=‖400‖>**

**Step 8:** Add some text to the img1.html and align the text messages in to left of the image. Hint: Use align attribute of <img> tag the values can be left, right, top, bottom etc.

**<img src=‖s1.jpg‖ align=‖right‖>**

**Step 9:** Create an image link to the img.html and save the file as ―imagelink.html‖.

**Hint:** place the <img> tag inside the <a> tag.

**<a href=‖img.html‖><img src=‖kid.jpg‖></a>**

**Step10:** What about making an image as background for the entire page and save it as

―imgbackground.html‖. Try like this

**<body background=‖kids.jpg‖>**

### Summary of this exercise:

You have just learnt

JPEG and GIF Iamges.

<img> tag and attributes. Aligning the images.

Creating an image link

### Deliverables of the exercise:

1. Img.html
2. img1.html
3. imagelink.html
4. imgbackground.html

**Assignment 7: Creating Tables in HTML**

**Objective:** How to create tables in HTML and layout the pages using invisible tables.

**Note:** To create table use the combination of the following tag.

<table> The main tag. Used to tell the browser "this is a table".

<tr> TableRow defines a horizontal row.

<th> TableHead defines a cell (or) heading of the column.

<td> TableData defines a cell (or) column.

**Step1:** Type the following code into an html file and save it as ―tab1.html‖.

**<!--Comment: border=‖2‖, indicating the border width of the table-->**

**<table border=‖2‖>**

**<tr>**

**<!-- Comment: Tag th is for giving a table heading-->**

**<th> Name </th><th>Empno</th>**

**</tr>**

**<tr>**

**<td>Jack</td><td>1001</td>**

**</tr>**

**<tr>**

**<td>John</td><td>1002</td>**

**</tr>**

**</table>**

**Step 2:** Change the border attribute value to ―0‖ and observe what will be the output?

**Step 3:** Change the table tag to <table border=”2” width="50%">.

**Step 4:** Right click on the desktop, select properties menu and settings tag. Find out the resolution of your PC. (It may be 1024 x 768 pixels)

**Note:** Table width will determine how much of the browser space (horizontally) will be used to display your table. There are two basic ways to accomplish this - by percent and by pixels. Pixels can be thought of as the smallest logical unit for display. Pixel resolution can vary from PC to PC. Tables built with percents will occupy that percentage of the browser‘s visible area or the container area. (A table which is inside another table will take the percentage width based on the container table).

**Step 5:** Include a background color for the table. Can you? Save the file as ―tabcolor.html‖

**<table border=‖2‖ width="50%" bgcolor="#CDCA8F">**

**Step 6:** If it is possible to add a background color, try to give a background image for the table.

**<table border=‖2‖ width="50%" background="s1.jpg">**

**Step 7:** Include a background color for a particular row.

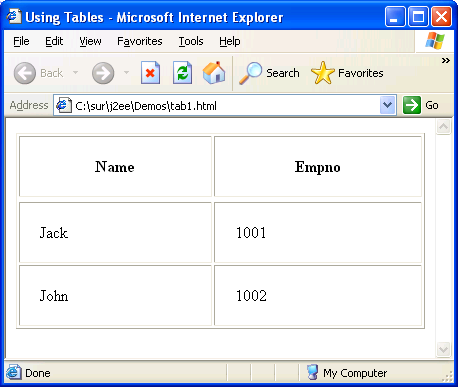
**Hint:** Use bgcolor attribute for the <tr> tag

**Step 8:** Increase the space between two cells and save it as ―tabspace.html‖.

### <table border=‖2‖ width="50%" cellspacing="10">

**Step 9:** Increase the space between the cell and the content.

### <table border=‖2‖ width="50%" cellpadding="20">



**Step 10:** Can you give some color and size for the table elements?

**Hint:** Use <font> inside a <td> tag.

**Note:** Do you need to give <font> tag for each cell? **Yes,if you want font effect for all the cells you need to give.** In future you can use CSS for avoiding the repetition of the HTML tag for providing style.

**Step 11:** Create the following HTML table and save it as ―tabspan.html‖.



### Verify the code.

**<table border=‖2‖ width="90%" cellpadding="10" bordercolor="blue" >**

**<tr>**

**<!-- Comment: colspan = ‖3‖ indicates that the table heading Marks will span three columns-->**

**<th colspan="3">Marks</th>**

**</tr>**

**<tr>**

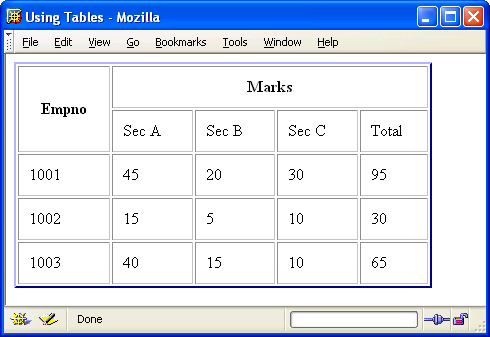
**<td>Sec A</td><td>Sec B</td><td>Sec C</td>**

**</tr>**

**</table>**

**Note:** The colspan attribute of the <td> (or) <th> tag Indicates the number of columns this cell should span and rowspan indicates the number of rows this cell should span.

**Step 12.** Create the following HTML table and save it as ―tabspannew.html‖.

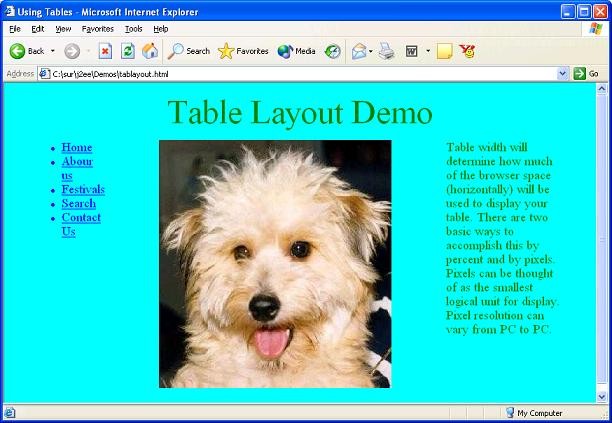


**Hint:** Use rowspan for the 1st cell and colspan for the 2nd cell.

**Note:** Tables are used in websites for two major purposes.

1. The primary purpose of arranging information in a table
2. The more widely used purpose of creating a page layout with the use of hidden tables. (border attribute values set to ―0‖)

**Step 13.** Create the following HTML table and save it as tablayout.html.



**Hint:** The above webpage created using a table with one row and three columns. The border of the table is invisible to the users.

The <center> tag is used align the content to the center of the container. The width attribute of the <td> tag used to specify the width of the column and valign tag is used to align the content vertically in a cell. The valign values are top, middle, bottom and baseline.

1. Replace the first cell content with unordered list of hyper links.
2. Replace the second cell content with an image.
3. Replace the third cell content with a paragraph.

### Summary of this exercise:

You have just learnt

Creating tables and using Attributes Using rowspan and colspan with <td> The align, width and valign tag.

The table layout.

### Deliverables of the exercise:

* 1. tab1.html
  2. tabcolor.html
  3. imgbackground.html
  4. tabspace.html
  5. tabspan.html
  6. tabspannew.html
  7. tablayout.html

**Assignment 8: Creating Interactive form using HTML.**

**Objective:** To understand designing interactive forms in HTML.

**Note**: Forms enable users to submit information that can be used to create an interactive web application ranging from an order entry system to an email application.

**Step 1:** Type the following code into a HTML page and save it as ―login.html‖.

**<h3>Login Screen</h3>**

**<!-- Comment : form elements should be inside a form tag -->**

**<form name="f1" method="post" action="Login.jsp">**

**<!-- Comment : size attribute specifies the size of the text box--> Username<input type="text" name="txtuname" size="20"><br><br>**

**<!-- Comment : password is similar to text, but displays only ―\*‖ --> Password<input type="password" name="txtpasswd"**

**size="20"><br><br>**

**<!-- Comment : submit button used to submit the form to the server -->**

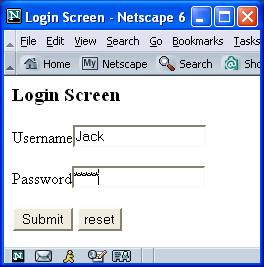
**<input type="submit" value="Submit" name="sub">**

**<!-- Comment : reset button used to reset the form elements to the default vale -->**

**<input type="reset" value="reset" name="res">**

**</form>**

Verify your output.



**Note :** In a <form> the method attribute value is either POST or GET. Data in any form submitted with the GET method is sent as part of the URL and form data submitted with POST is sent in the HTTP request body. Action attribute indicates a program on the server that will be executed when this form is submitted.

The Submit button calls the action page on the server and reset button resets the form element values to the default value.

**Step 2:** Create a HTML page that will display a multiple choice questions with four choices and save it as ―formquestion.html‖.

**Hint:** Use Radio Buttons.

Here “Answer” is visible to the user and the value “1” is not visible to the user, which is used by either client side program or server side program.

**Note:** All the radio buttons should have the same name (known as radio group), so that you can select only one choice at a time.

**Step 3:** Include a descriptive type question in to the ―formquestion.html‖.

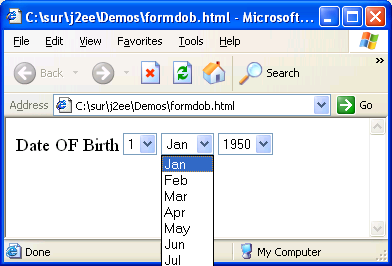
**Hint:** Use a text area for accepting the result.

Here the cols attribute will define the width of the textarea and rows attribute will define the height of the textarea.

**Note:** <textarea>is a separate tag unlike <input> tags. <input> tag without any attribute will result in to a textbox.

The text between <textarea> and </textarea>, will appear as the default value for the textarea. To set a default value for a text box you can use <input type=”text” value=”Asreet”>.

**Step 4:** Create the following form and save it as ―formdob.html‖.



**Hint:** Use three select boxes.

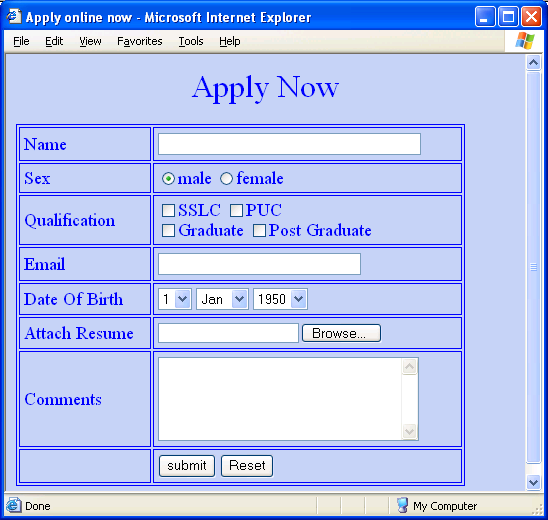
**Note:** The multiple attribute of <select> tag specifies that, multiple items can be selected at a time. The size attribute defines the number of visible items in the drop-down list.

<select name=”seldate” size=”4” multiple>

The selected attribute of <option> tag specifies that the option should appear by default selected.

<option value=‖25‖ selected>25

**Step 5:** Create the following form and save it as ―applyform.html‖.



**<body>**

**<form>**

**<! – Comment: The entire table should be inside the table tag-->**

**<table border=‖2‖>**

**<tr>**

**<td>Name</td>**

**<td><input type=‖text‖ name=‖txtname‖></td>**

**</tr>**

**……………………………………………………………………………**

**……………………………………………………………………………**

**</table>**

**</form><!—Comment: close the form only after the </table> tag -->**

**</body>**

**Note: Some special elements.**

<input type=”file”> : This control is used for file uploading.

<input type=”button”>: This control used to create a button that has no predetermined actions (rest (or) submit).

<input type=”hidden”>: Used to store some temporary data(hidden) that can be accessed by client side or server side programs.

<input type=”image”>: This control is used to create a graphical version of the submit button.

**Note: Some special attributes.**

name and value are the common attribute for all the <input> elements.

maxlength : Sets the maximum number of characters allowed in the field.( Applicable only to text, hidden and password elements.)

checked : To set a checkbox (or) radio button to be selected by default.

selected : An occurrence of the selected attribute in the <option> element sets the form select control to select this item by default.

**Step 6:** Open ―applyform.html‖ and change border=‖0‖ in the table tag.

### Summary of this exercise:

You have just learnt

Creating a login screen. Designing GUI using HTML. Using <textarea> and <select>

Aligning form elements using tables.

### Deliverables of the exercise:

1. login.html
2. formquestion.html
3. formdob.html
4. tablayout.html
5. applyform.html

**Assignment 9: Working with frameset and frames**

**Objective:** To understand how to display more than one HTML document in the same browser window. With frames, you can display more than one Web page in the same browser window.

**Note:** The frameset element defines a frameset. It is used to organize multiple windows (frames). Each frame holds a separate document. The frameset element states only how many columns or rows there will be in the frameset. You must use the cols or the rows attribute. The frame element defines a sub window (a frame).

**Step 1:** Open a text editor and type the following:

**<html>**

**<!--comment: Two Columns of 30% and 70% each -->**

**<frameset cols="30%,\*">**

**<!--comment: \* is indicating the remaining space (here 70%) -->**

**<!-- Comment: frame1.html is for the first frame -->**

**<frame src="frame1.html">**

**<!—Comment: frame2.html for second frame -->**

**<frame src="frame2.html">**

**</frameset>**

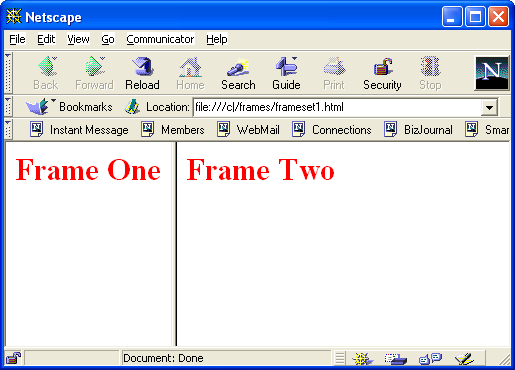
**</html>**

**<!--comment: no need to use body tag at all-->**

Save the file as ―frameset1.html‖.

**Note:** The two html pages frame1.html and frame2.html should be present and resides in the same directory. The <frameset> tag is used instead of the <body> tag.

**Step 2:** Open the **“**frameset1.html‖ in the browser.



**Step 3:** Type the following code in a text editor and save the file as ―frameset2.html‖.

**<html>**

**<frameset cols="40%,\*">**

**<!—Comments: Divides the window into two cols -->**

**<frameset rows="35%,\*">**

**<!—Comments: Divides the first frame(Col of width 40%)into two rows-->**

**<frame src="frame1.html" name=‖f1‖>**

**<frame src="frame2.html" name=‖f2‖>**

**</frameset>**

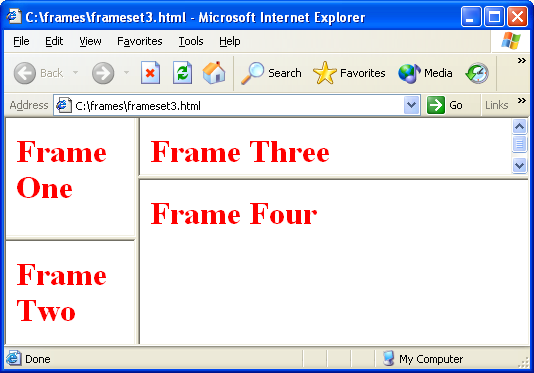
**<frame src="frame3.html" name=‖f3‖>**

**</frameset>**

**</html>**

**Step 4:** Open the ―frameset2.html‖ file in a browser.

**Step 5**: Create the following HTML page and save it as ―frameset3.html‖



**Step 6**: Modify the ―frameset3.html‖ file and save it as ―frameset4.html‖. Do the following

* 1. Make the frame border invisible to the user.
  2. Prevent the user from resizing the frame.
  3. Disable the scrollbars.

**Hint:** Set the attribute frameborder=0 for the <frameset> to disable the border. Set the attribute NORESIZE for the <frame> to disable resizing of the frame Set the attribute SCROLLING=NO for the <frame> to disable scrolling.

**Step 7:** Type the following code and save it as ―mainframe.html‖

**<html>**

**<frameset cols="30%,\*">**

**<frame name=‖frame1‖ src="doc1.html">**

**<frame name="frame2" src="doc2.html">**

**</frameset>**

**</html>**

**Note:** The name attribute values are case sensitive.

**Step 8:** Type the following code and save it as ―doc1.html‖

**<html>**

**<body>**

**<h1><font color=red>Frame One</font></h1>**

**<a** [**href="http://sparsh/v1"**](http://sparsh/v1) **target="frame2">Go To Sparsh</a>**

**</body>**

**</html>**

**Step 9:** Type the following code and save it as ―doc2.html‖

**<html>**

**<body>**

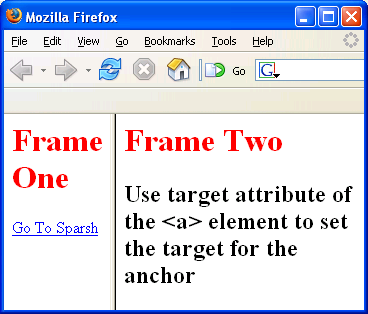
**<h1><font color=red>Frame Two</font></h1>**

**<h2> Use target attribute of the &lt;a> element to set the target for the anchor</h2>**

**</body>**

**</html>**

**Step 10:** Open ―mainframe.html‖ in the browser.



**Step 11:** Click on the link ―Go To Sparsh‖.

**Note:** Ensure frame naming in frame tag. Predefined target values for <a> are

\_blank – Loads link into new blank window

\_parent - Loads the link into the immediate parent of the document the link is in

\_self - Loads the link into the same window. (Default)

\_top - Loads the link into the full body of the window

**Step 11:** Modify doc1.html and set the target value as ―\_parent‖ and check for the output.

**<a** [**href="http://sparsh"**](http://sparsh/) **target="\_parent">Go To Sparsh</a>**

**Step 12:** Type the following code and save it as ―top.html‖.

**<html>**

**<frameset cols="30%,\*">**

**<!-- Comments : mainframe.html is a frame, contains doc1.html and doc2.html-->**

**<frame name="frame1" src="mainframe.html">**

**<frame name="frame2" src="doc2.html">**

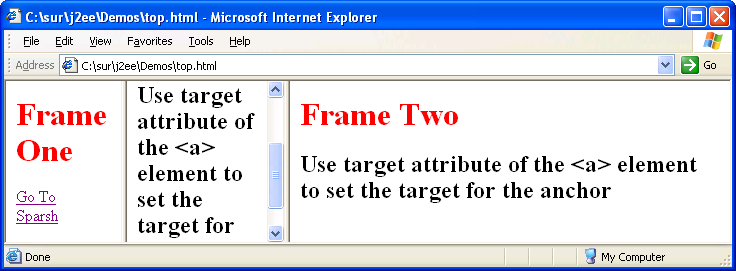
**</frameset>**

**</html>**

In ―doc1.html‖ change the target as ―\_top‖.

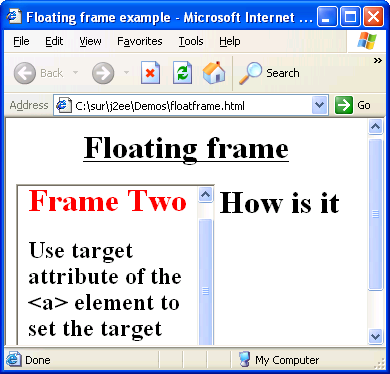
**<a** [**href="http://sparsh"**](http://sparsh/) **target="\_top">Go To Sparsh</a>**

Open in the browser and verify the output.



**Note:** The iframe element creates an inline frame that contains another document. This is another form of frame, called as floating frame. An inline frame is created by <iframe> element and can occur anywhere the <body> tag.

**Step 13:** Type the following code and save it as ―floatframe.html‖.



**<html>**

**<head>**

**<title>Floating frame example</title>**

**</head>**

**<body>**

**<h1 align="center"><u> floating frame </u></h1>**

**<!—Comment: The following statement creats an inline frame with doc2.html as the content -->**

**<iframe name="floatframe1" src="doc2.html" height=200 width=200 align="left">**

**There will be a floating frame if your browser supports it**

**</iframe>**

**<h1>How is it</h1>**

**</body>**

**</html>**

**Step 14:** Modify ―floatframe.html‖ set frameborder=0 for the <iframe> element.

### Summary of this exercise:

You have just learnt

Working with frames

<frameset> and <frame> tag.

Using target attribute of <a> tag with frames. Floating frames.

### Deliverables of the exercise:

1. frameset1.html
2. frame1.html
3. frame2.html
4. frameset2.html
5. frameset3.html
6. mainframe.html
7. doc1.html
8. doc2.html
9. top.html
10. floatframe.html

**Assignment 10: Understanding CSS**

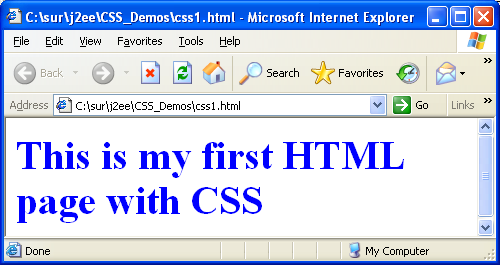
**Objective:** To understand the usage of CSS and provide cascading style for your HTML documents.

**Problem Description:** You have learnt how to create HTML documents. Style Sheets is a breakthrough in Web design because it allows developers to control the style and layout of multiple Web pages all at once.

**Note:** CSS1 Accepted by the W3C Consortium in 1996

In 1998, CSS2 was introduced to add more power to CSS.

However, no browsers support all of the CSS2 features; although, many support parts of it. Internet Explorer 6, Netscape 6, and Opera 6 have almost full support of CSS2.

**Step 1:** Type the following code and save it as css1.html.

**<html>**

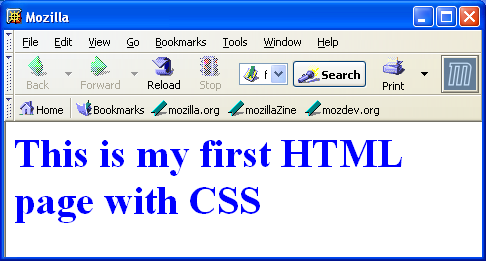
**<body>**

**<!-- Comment: Using Inline Style -->**

**<h2 style=‖color:red‖> This is my first HTML page with CSS</h2>**

**</body>**

**</html>**



**Step 2:** In the above example replace the <h2> tag with the following code and save the file as css2.html

**<h2 style="color:blue; font-size:10pt;"> This is my first HTML page with CSS</h2>**

**Step 3**: Save the file as css3.html and change the color of the font to green and size of the font to 50 pt;

**Step 4**: Using style Tag, save the file as css4.html

**<html><head>**

**<!-- Comment:**

**<style>**

**<style> tag is used to define styles**

**-->**

**<!-- Comment: Style defined for <h2> tag -->**

**h2{**

**color:red;**

**font-size:30pt;**

**}**

**</style><head>**

**<body>**

**<h2> This is my first HTML page with CSS</h2>**

**</body>**

**</html>**

### Summary of this exercise:

You have just learnt

* Using inline style sheet.

### Deliverables of the exercise:

1. css1.html
2. css2.html
3. css3.html
4. css4.html

**Assignment 11 : CSS syntax**

**Objective:** To understand the CSS syntax and different kind of selectors.

**Note**: CSS syntax has two components, the selector and the style declaration. The *selector* specifies what the style rule is acting upon. A selector can be one of two things. It can be an HTML element name or it can be an attribute reference.

**Step 1:** Type the following code and save it as ―csselement.html‖.

**<html><head>**

**<style> h1{**

**color: red; font-size:30pt;**

**}**

**</style><head>**

**<body>**

**<h1> First H1 Tag</h1>**

**<h1> Second H1 Tag</h1>**

**</body>**

**</html>**

**Step 2:** In the above code add <h2> tag and <h3> tag. Provide necessary style using element selector.

**Note**: HTML has an attribute called the class attribute. It is specifically for use with style sheets. Classes allow you to assign style rules only to selected elements that have the correct class attribute value in their tag.

**Step 3:** Type the following code and save the file as ―class1.html‖

**<html><head>**

**<style>**

**h1{**

**color:red;**

**font-size:30pt;**

**}**

**<!-- Comment: Style defined for h1 tag with classname ‗c‘ --> h1.c**

**{**

**color:blue;**

**}**

**</style><head>**

**<body>**

**<!-- Comment: Style defined for h1 tag -->**

**<h1> First H1 Tag</h1>**

**<!-- Comment: Takes the Style defined for h1 tag with class name ‗c‘**

**-->**

**<h1 class="c"> Second H1 Tag</h1>**

**</body></html>**



**Step 4:** In the above example add the following line and save it as ―class2.html‖. Check the output.

**<h2 class="c"> new h2 Tag with class</h2>**

**Step 5**: Type the following code and save the file as ―class3.html‖.

**<html><head>**

**<style>**

**h1{**

**color:red;**

**font-size:30pt;**

**}**

**<!-- Comment: A generic style created with class name ‗c‘ -->**

**.c**

**{**

**color:blue;**

**}**

**</style><head>**

**<body>**

**<h1> First H1 Tag</h1>**

**<h1 class="c"> Second H1 Tag</h1>**

**<h2 class="c"> Second H1 Tag</h2>**

**</body>**

**</html>**

### Summary of this exercise:

You have just learnt

CSS syntax. Element selectors. Class selectors

### Deliverables of the excercise:

1. csselement.html
2. class1.html
3. class2.html
4. class3.html

**Assignment 12: Some more selectors.**

**Objective:** To understand different kind of selectors like Inheritance, Contextual Selectors**,** Grouping and Pseudo selectors.

**Note**: Style properties are inherited from the parent element to the child element.

**Step 1:** Type the following code and save it as ―cssinher.html‖.

**<html><head>**

**<style>**

**<!-- Comment: An example for Inheritance selector -->**

**body{**

**color: red;**

**}**

**h1**

**{**

**font-size:10pt;**

**}**

**</style></head>**

**<body>**

**<h1> Heading</h1> Some text**

**</body>**

**</html>**

**Step 2:** Type the following code and save the file as ―csscon.html‖

**<html><head>**

**<style>**

**<!-- Comment: An example for Contextual selector -->**

**p font{**

**color:red;**

**}**

**</style><head>**

**<body>**

**<p> Para <font> Inside font</font></p>**

**<font>Outside Para</font>**

**</body>**

**</html>**

**Note**: The above style rule enforce that the text inside the <font> tag which is inside <p> tag will display in red color.

**Step 3:** Type the following code and save the file as ―cssgroup.html‖.

**<html><head>**

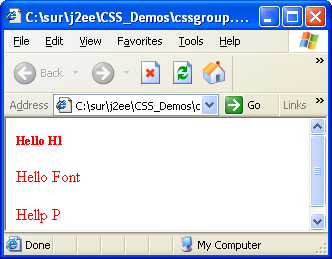
**<style>**

**<!-- Comment: An example for Grouping selectors -->**

**p,font,h1{**

**color:red;**

**}**



**h1**

**{**

**font-size: 10pt;**

**}**

**</style><head>**

**<body>**

**<h1> Hello H1</h1>**

**<font> Hello Font</font>**

**<p> Hello P</p>**

**</body>**

**</html>**

**Note**: The above style rule enforces that all the group elements be displayed in same style.

**Step 4:** Modify the above program and use class selectors instead of group selector and save as ―cssclassnew.html‖.

**Note**: Pseudo-classes are special "classes" that are automatically recognized by CSS- supporting browsers. Pseudo-classes can be assigned to the A element to display links, visited links and active links differently. The pseudo classes are

A:link -- link

A:active -- active link

A:visited -- visited link

A:hover -- Mouse Over

**Step 5:** Type the following code and save it as ―csspsedo.html‖.

**<html>**

**<head>**

**<style>**

**<!-- Comment: An example for Pseudo selectors -->**

**a:link**

**{**

**color:blue;**

**}**

**a:visited**

**{**

**color:red;**

**}**

**</style>**

**<head>**

**<body>**

**<h2> Go to sparsh <h2><a** [**href="http://sparsh">Sparsh</a>**](http://sparsh/)

**</body>**

**</html>**

**Step 6:** In the above program include the following code inside the <style> tag.

**a:hover**

**{**

**color:green;**

**}**

**Step 7:** Open the page in browser and move mouse over the link and verify the output.

**Step 8:** In ―psedo.html‖ add style for a:active pseudo class.

### Summary of this exercise:

You have just learnt

Inheritance selector. Contextual selector. pseudo selector.

Group selector.

### Deliverables of the excercise:

1. cssinher.html
2. csscon.html
3. cssgroup.html
4. csspsedo.html

**Assignment 13: External Style sheet, <DIV> and <SPAN>**

**Objective:** To understand the advantage of External style sheet and usage of <DIV> and

<SPAN> tag.

**Note**: An external style sheet is ideal when the style is applied to many pages. The external style sheet file extension should be ―.css‖.

**Step 1:** Type the following code and save it as ―ext.css‖.

**/\* Comments: do not use <style> tag here h1**

**{**

**color:red;**

**}**

**h2**

**{**

**color:blue;**

**}**

**\*/**

**Step 2:** Type the following code and save is as ―extcss.html‖. ( css file and html files should be inside the same folder).

**<html><head>**

**<!-- Comment: The document refers to a separate external file**

**‗ext.css‘ -->**

**<link rel="stylesheet" type="text/css" href="ext.css">**

**<head>**

**<body>**

**<h1> style from external css</h1>**

**<h2> style from external css</h2>**

**</body>**

**</html>**

**Note**: The <link> element provides a way to define the relationship between linked objects. The <link> element allows a style sheet for a document to be referenced from a separate file.

**Note:** DIV was introduced as a generic block level element who's original purpose was to serve as a container(**block**) for the logical 'parts' or divisions of a document. The Span tag is a generic inline container element. If you want to group text without using a block element, use the <span> element.

**Step 3:** Type the following code and save it as ―cssdiv.html‖.

**<html>**

**<head>**

**<style>**

**<!-- Comment: A style defined for Div tag --> div{**

**color:blue;**

**}**

**</style>**

**<head>**

**<body>**

**<div>**

**<h1>Division heading </h1>**

**<p>many paragraphs and block elements</p>**

**<p>Notice that color is blue</p>**

**<div>**

**</body>**

**</html>**

**Step 4:** Modify the above program, include a <font> tag with some text inside the <div> tag. Save the file as ―cssdiv1.html‖.

**Step 5:** Type the following code and save the file as ―cssspan.html‖.

**<html>**

**<head>**

**<style>**

**<!-- Comment: A style defined for span tag with class name as ‗imp‘ --**

**>**

**span.imp**

**{**

**color:blue;**

**}**

**</style>**

**<head>**

**<body>**

**<p> In this sentence <span class="imp">Some of the text is important</span></p>**

**</body>**

**</html>**

### Summary of this exercise:

You have just learnt

External style sheet. Usage of <div> Usage of <span>

### Deliverables of the excercise:

1. ext.css
2. extcss.html
3. cssdiv.html
4. csspspan.html

Questions based on self study slides:

Style Properties:

Q1. Name a few background properties. Q2. Name a few font properties.

Q3. What are the different values for text-align property?

Q4. What is the difference between word-spacing and letter-spacing in Text properties? Q5. What is the use of Z-index position property?

**Assignment 14: Using text, font, box and position properties**

**Objective:** To understand the usage of defining font in text and appearance of the text. Defining box properties and

**Step 1:** Type the following code and save it as ―cssfont1.html‖.

**<html>**

**<head>**

**<style>**

**<!-- Comment: Defining style for tag h2 considering few font related properties -->**

**h2**

**{**

**font-style:italic; font-size:28pt; color:cyan;**

**}**

**</style>**

**<head>**

**<body>**

**<h2> Using Font Style</h2>**

**</body>**

**</html>**

**Step 2:** Modify the above code and provide the font family as ―Courier New‖. Hint: use font-family property.

**Step 3:** Type the following code and save it as ―csstext.html‖.

**<html>**

**<head>**

**<style> h2**

**{**

**letter-spacing:4pt; font-size:40pt; color:blue;**

**text-align:center;**

**}**

**</style>**

**<head>**

**<body>**

**<h2>Asreet Mysore</h2>**

**</body>**

**</html>**

**Step 4:** Modify the above program and display the text with underline. Hint: Use text-decoration property.

**Step 5:** Type the following code and save it as ―cssborder.html‖.

**<html>**

**<head>**

**<style>**

**<!-- Comment: Defining style for tag h2 considering few border related properties -->**

**h2**

**{**

**text-align:center; color:blue;**

**border-width:thick; border-color:green; Border-style:solid;**

**}**

**</style>**

**<head>**

**<body>**

**<h2>Asreet<sup>&reg;</sup> Mysore</h2>**

**</body>**

**</html>**

**Note**: The **border** property is shorthand for setting the border-width, border-style and border-color of an element's border.

**Step 6:** Modify the above program with the following code and save it as ―cssborder1.html‖.

**<style> h2**

**{**

**text-align : center; color : blue;**

**border : thick dotted red;**

**}**

**</style>**

**Step 7:** Type the following code and save the file as ―csspos.html‖.

**<html>**

**<head>**

**<style> h2**

**{**

**letter-spacing:4pt; font-size:40pt; color:blue;**

**text-align:center; position: absolute; top:0px;**

**}**

**h3**

**{**

**letter-spacing:4pt; font-size:40pt; color:blue;**

**text-align:center; position: absolute; top:20px; left:20px;**

**}**

**</style>**

**<head>**

**<body>**

**<h2>Asreet Mysore</h2>**

**<h3>Asreet Mysore</h3>**

**</body>**

**</html>**

**Step 8:** Modify the above code, display a border around the text and save the file as

―cssposborder.html‖.

**Step 9:** Type the following code and save it as ―cssabove.html‖.

**<html>**

**<head>**

**<style> h2**

**{**

**position:absolute; Z-index:1; color:red;**

**}**

**<!-- Comment: Defining style for img tag --> img**

**{**

**position:relative; top:-50px;**

**}**

**</style>**

**<head>**

**<body>**

**<h2>Asreet Mysore</h2>**

**<img src="gec.jpg" width="200" height="150">**

**</body>**

**</html>**

**Step 10:** Modify the above program and set the Z-index value as ―-1‖.

### Summary of this exercise:

You have just learnt

Text properties. Font properties. Border properties. Position properties.

### Deliverables of the exercise:

1. cssfont1.html
2. csstext.html
3. cssborder.html
4. cssborder1.html
5. csspos.html
6. cssabove.html

**Assignment 15: Applying styles to Tables and Form Elements**

**Objective:** To understand the usage of CSS with tables and form elements.

**Note**: It‘s possible to change the default look of a table and form elements by styling their html tags:

**Step 1:** Create the following html file and save it as ―csstab1.html‖.

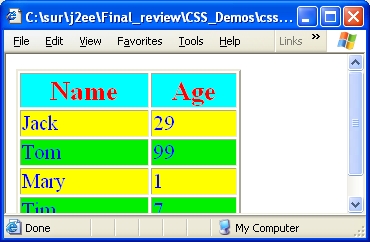
### Hint:

**Create separate class styles for the following:**

### Table heading

1. **Even numbered rows**

### Odd numbered rows



**Step 2:** Modify the above code by including the above created style classes in a separate external file cssexternal.css.

**Step 3:** Type the following code and save the file as ―cssform1.html‖.

**<html>**

**<head>**

**<style> input{**

**color:blue;**

**letter-spacing:3pt;**

**font-size:14pt; }**

**form{**

**border: thick solid blue; }**

**</style>**

**<head>**

**<body>**

**<form>**

**Name <input type="text" value="style"><br>**

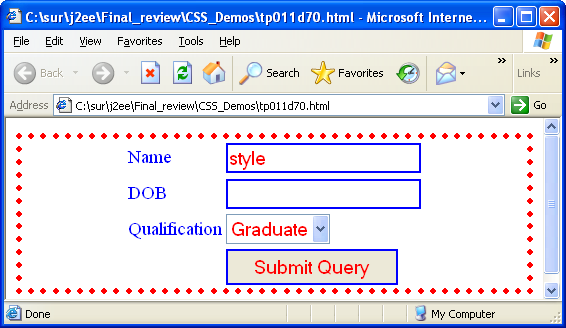
**<input type="submit">**

**</form>**

**</body>**

**</html>**

**Step 4:** Create the following html file and save it as ―cssform2.html‖.



### Summary of this exercise:

You have just learnt

Applying styles to <table>, <tr> and <td> tag. Applying styles to <form>, <input> and <select> tag. Applying external styles.

### Deliverables of the exercise:

* 1. csstab1.html
  2. cssexternal.css
  3. cssform1.html
  4. cssform2.html

**Day 2 Assignments of Java Script**

All the assignments in this section must be completed on Day 2 of your HJS course.

## Assignment 1: Understanding Client Side JavaScript

**Objective:** To understand the usage of Client Side JavaScript.

**Background:** You have learned HTML and Cascading Style Sheet (CSS). Using HTML and CSS you can create only static pages. Now you can create interactive pages using client side JavaScript.



**Note:** JavaScript is embedded as a small program in a web page that is interpreted and executed by the Web client. JavaScript is a scripting language - a scripting language is different from programming language. JavaScript is an interpreted language (means that scripts execute without preliminary compilation). JavaScript is supported by all major browsers like Netscape, Internet Explorer, Mozilla etc.

### Estimated time: 10 minutes

**Step 1:** Create a folder named ―assign1‖ under your working directory

**Step 2:** Type the following code into a text editor and save it as ―firstjs.html‖

<html>

<head>

<!-- Comments : To include a client side script use <script> tag -->

<script language="JavaScript">

<!--

document.write("Displaying using JavaScript");

//-->

</script>

</head>

<body>

<h1> First JavaScript Program</h1>

</body>

</html>



**Note:** To hide the script from the browsers, which are not supporting JavaScript use HTML comments (<!-- , //-->). // is the JavaScript single line comment.

**Step 3:** Modify the ―firstjs.html‖ remove the language attribute of the <script> tag. And execute the program.

Eg. <script>



**Note:** If you are getting the proper output which is indicating that, the default scripting language for your browser is JavaScript. But it is good practice that always use the language attribute.

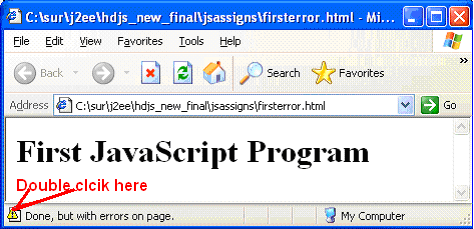
**Step 4:** Modify the ―firstjs.html‖ and change the <script> tag like the following

<script language="**Java Script**">

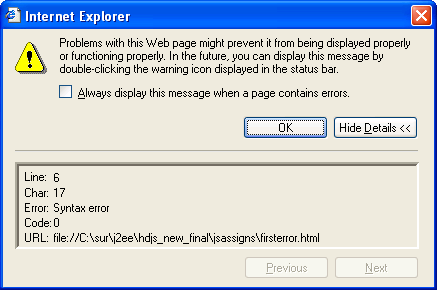
The script will not execute because the browser does not understand the script language. Note down the space between Java and Script and remove it.

**Step 5:** Modify the ―firstjs.html‖ like the following and save as ―firsterror.html‖. Open the page in the browser.

document.write(<h1>”Displaying using JavaScript"</h1>);



**Step 6:** The page is displayed with errors. To find out the error in JavaScript double-click on the icon on the status bar. A small Dialog window will display the error messages with the line number. ( if the error message is not displaying properly select the address(URL) from the address bar and press enter, check the status bar).



**Step 7:** The Dialog box is showing that the error is at line number 6 and the message is

―Syntax error‖. Let us examine the code at line number 6

document.write(<h1>"Displaying using JavaScript"</h1>)

We can find that the syntax is wrong. The correct syntax is

document.write(“<h1>Displaying using JavaScript</h1>”)

**Step8:** Correct the error and save the file as ―errorfree.html‖

### Summary of this exercise:

You have just learnt

How to use Javascript in HTML documents. Debug and fix the error.

### Deliverables of the exercise:

1. firstjs.html
2. firsterror.html
3. errorfree.html

## Assignment 2: Using Variables and Operators.

**Objective:** To work with JavaScript we need to know the different types of variables and operators.

### Estimated time: 15 minutes

**Step 1:** Type the following code in an editor and save it as ―varjs.html‖.

<html>

<head>

<script language="JavaScript">

/\* Comments : Variables declaration \*/

/\* Comments : Inline scripting \*/

var num1=10; var num2=20;

var num3=num1+num2;

/\* Comments : document.write is a method to write the contents on the browseer \*/

document.write("The sum is "+num3);

</script>

</head>

<body>

<h3> JavaScript variable demo</h3>

</body>

</html>



**Note:** Variable names are case-sensitive and they must begin with a letter or the underscore character.

**Step 2:** Modify the ―varjs.html‖, place the entire Javascript part inside the <body> tag and save it as ―varjsbody.html‖. Open the page in the browser.

**Step 3:** Remove the **var** keyword from ―varjs.html‖ and save it as ―remvar.html‖. Open the page in the browser and check the output.



**Note:** var is used to declare a variable. In JavaScript you can use variables without declaring it. If a variable is prefixed by the keyword ―*var*‖ within a function then it is a local variable, otherwise it is a global variable.

**Step 4:** Modify the ―remvar.html‖, change the variable name ―**num3**‖ as ―**3num**‖. Save it as

―varerror.html‖ Verify the output.

**Step 5:** Debug the error in ―varerror.html‖ and correct it.

**Step 6:** Include the following code inside <head> and save it as ―assign.html‖.

<script language="JavaScript"> inum=55;

msg="4"; sum=msg+inum;

alert(sum);

/\* Comments : The above statement displays a message dialog box \*/

</script>

**Step 7:** Write a program to find out 10.5 % 2.5, Save it as ―mod.html‖, what is the output?

**Step 8:** Include the following code inside <head> and save it as ―unary.html‖.

<script language="JavaScript"> num1 = 10; num2 = 10; num3 = num1++;

num4 = ++num2;

</script>

Display the values of num1, num2, num3 and num4.

**Step 9:** Find out what is the error in the following code, save it as ―varerror2.html‖

<script language="JavaScript"> num1=100;

Num1+=20;

</script>

Debug the error and display the value of ―num1‖.

**Step 10:** Find out what is the value of num1 in the following code and save it as ―local.html‖.

<script language="JavaScript">

/\* Comments : Defining a function by name „fun‟ \*/

function fun()

{

/\* Comments : Declaring a local variable „num1‟ \*/

var num1=100;

}

fun()

/\* Comments : calling function \*/

document.write("the value of num1 is "+num1);

</script>

**Step 11:** Getting Error? Why? Debug it. (Refer the note in page-no 4)

**Step 12:** Modify the ―local.html‖, remove the ―**var**‖ keyword and execute it.

### Summary of this exercise:

You have just learnt

Declare and how to use variables. Different operators in JavaScript

Difference between local and global variables.

### Deliverables of the exercise:

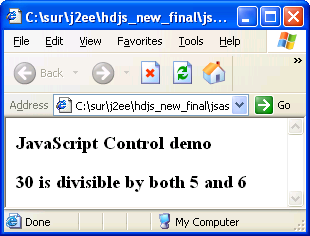
1. varjs.html
2. varjsbody.html
3. remvar.html
4. varerror.html
5. assign.html
6. mod.html
7. unary.html
8. varerror2.html
9. local.html

## Assignment 3: Using Control Structures and Loops

**Objective:** Any logic can be created using control structures and loops. Here we are learning about the usage of control structures and loops.

### Estimated time: 20 minutes

**Step 1:** Write a program to find whether a given number is divisible by 5 and 6 And save it as ―control1.html‖



**Step 2:** Write a program to find the largest of three numbers using if-else control structure and save it as ―control2.html‖.

**Step 3:** Write a program to find, whether the trainee has cleared the ―Generic Comprehensive‖ exam or not with the following condition. To pass the exam the total mark should be greater than 65% and Sec A, Sec B, Sec C mark should be greater than 50%. Save the file as ―control3.html‖.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ie, | Total | >= | 65 | /100 |
|  | Sec A | >= | 25 | / 50 |
|  | Sec B | >= | 15 | / 30 |
|  | Sec C | >= | 10 | / 20 |

**Step 4:** Find out what is the output of the following code. Save the file as ―control4.html‖

<script language="JavaScript"> var num=100;

var str="100";

/\* Comments : Equality operator used for comparison \*/

if(num==str){

document.write("<h1> Equal</h1>");

}

else{

document.write("<h1> Not Equal</h1>");

}

</script>

**Step 5:** Modify the above code, replace ―**==**‖ with ―**===**‖. What will be the output?

**Step 6:** Write a program to find whether the given character is vowel or not. Use Switch case control structure. Save the file as ―switch.html‖

Remember that JavaScript does not support character data type. Use String data type for doing the same. Eg letter=”A”.

**Step 7:** Write a program to display numbers from 1 to 10 using for loop in a table format and save it as ―for1.html‖.



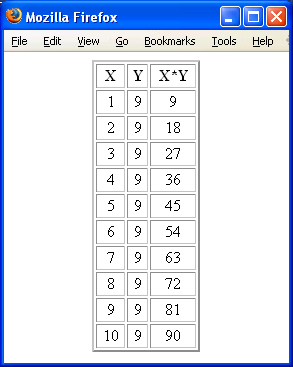
**Note:** When you are using attribute values inside the write method, it is better to use inside single codes.

document.write(“<table border=‟1‟>”);

Otherwise use escape character \‖ like

document.write(“<table border=\"1\”>”);

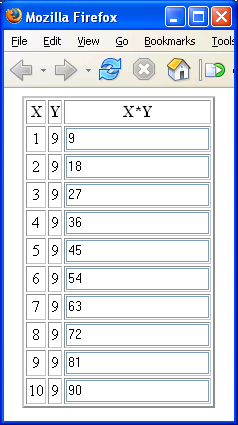
**Step 8:** Write a program to display the multiplication table of the given number using a loop. Display the output in a table format. Save the file as ―multtable.html‖.



**Step 9:** Modify the above program and display the result inside a textbox. Save the file as

―multtabtext.html‖. Verify the output.

Place the <table> tag inside a <form> tag and use ―text‖ input type. The value of the input type should be assigned dynamically.



### Summary of this exercise:

You have just learnt

Using control structure if-else and switch Using for loop, while and do-while loop. Dynamic table creation.

### Deliverables of the exercise:

1. control1.html
2. control2.html
3. control3.html
4. control4.html
5. switch.html
6. for1.html
7. multtable.html
8. multtabtext.html

## Assignment 4: Functions and Dialog Boxes

**Objective:** Functions are one of the fundamental building blocks in JavaScript. A function is a JavaScript procedure—a set of statements that performs a specific task. Here we will learn how to create user defined functions and make use of in built functions.

### Estimated time: 15 minutes

**Step 1:** Type the following code in to a text editor and save it as ―fun1.html‖.

<script language="JavaScript">

/\* Comments : function definition \*/ function addFun(num1,num2)

{

document.write("the sum is "+ (num1+num2));

}

/\* Comments: Calling function \*/ addFun(10,20);

</script>

**Step 2:** Modify the above program, add a function for multiplying two numbers.



**Note:** The main advantage of a function is reusability. So instead of displaying the output inside the function return the result to the caller using the **return** keyword.

**Step 3:** Type the following code into a text editor and save it as ―funreturn.html‖.

<html><head>

<script language="JavaScript"> function addFun(num1,num2)

{

/\* Comments : Returning value to the calling environment \*/

return num1+num2;

}

</script>

</head><body>

<h1> Function Demo</h1>

/\* Comments : More than one script tag can be used in a document \*/

<script language="Javascript">

document.write("<h1>Sum is "+addFun(20,50)+"</h1>");

</script>

</body></html>

**Step 4:** Write a function called power(arg1, arg2) to find the power of the given number and save it as ―power.html‖.



**Note:** The arguments of a function are maintained in an array. Within a function, you can address the parameters passed to it as follows: arguments[i]. The total number of arguments is indicated by arguments.length.

**Step 5:** Type the following code and save it as ―funarg.html‖.

<html><head>

<script language="JavaScript">

/\* Comments : function does not have any arguments, but arguments array will store the arguments \*/

function dispStr()

{

for(i=0;i<arguments.length;i++)

{

document.write(arguments[i]+" ");

}

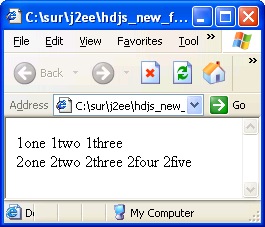
}

/\* Comments : Calling function with three arguments \*/ dispStr("1one","1two","1three"); document.write("<br>");

/\* Comments : Calling function with five arguments \*/ dispStr("2one","2two","2three","2four","2five");

</script>

</head><body></body></html>



**Step 6**: Write a function which takes n number of arguments and return the average of n numbers and save it as ―numarg.html‖



**Note:** There are three different types of dialog boxes (alert, prompt and confirm) which are the methods of ―window‖ object. All these three dialog boxes are modal i.e. The user must close it before continuing.



**Note:** Event handlers are very powerful and useful in client side scripting. They are JavaScript code that are not added inside the <script> tags, but rather, inside the html tags, that execute JavaScript when an event happens, such as pressing a button, moving your mouse over a link, submitting a form

,etc. For Example

<input type="button" value="Close" name="but2" onClick="closeWin()">

**Step 7**: type the following code and save it as ―dialog.html‖.

<html><head>

<script language="JavaScript"> function dispMsg()

{

/\* Comments : Using prompt dialog box data can be accepted through data entry field from the user \*/

num1=parseInt(prompt("enter the first number","0"));

/\* Comments : parseInt method used to convert string to a numeric value \*/

num2=parseInt(prompt("enter the second number")); alert("Sum is \n"+(num1+num2));

}

function closeWin()

{

/\* Comments : confirm dialog box used to get user confirmation about closing the window or not \*/

res=confirm("do you want to close the window?"); if(res==true) window.close();

}

</script>

</head><body>

<form name="f1">

/\* Comments : Event Onclick used with „Add‟ button . Hence on clicking the „Add‟ button „dispMsg()‟ function is invoked \*/

<input type="button" value="Add" name="but1" onClick="dispMsg()">

<input type="button" value="Close" name="but2" onClick="closeWin()">

</form>

</body></html>

### Summary of this exercise:

You have just learnt

Create and invoke user defined functions. The arguments array and its usage

Use Dialog boxes.

Use onClik event handler.

### Deliverables of the exercise:

1. fun1.html
2. funreturn.html
3. power.html
4. funarg.html
5. numarg.html
6. dilog.html

## Assignment 5: External File and Built in Objects

**Objective:** To understand the usage of External JavaScript file (.js) and learn the usage of some important built in objects.

### Estimated time: 25 minutes

**Step 1:** Type the following code in to a text editor and save it as ―ext.js‖.

/\* Comments : no need to use script tag here \*/ function extMsg()

{

alert("Message from ext.js");

}

**Step 2:** Type the following code into a text editor and save it as ―ext.html‖

<html><head>

<!-- Comments : Using external javascript file -->

<script language="JavaScript" src="ext.js"></script>

<!-- Comments : Normal mistake, forget to close the script tag -->

</head><body>

<input type="button" value=" Add " name="but1" onClick="extMsg()"><br>

</form>

</body></html>

**Step 3:** Write a program to display the current date, month and year and save it as

―date1.html‖. (display in a format ―dd-mm-yyyy‘) Use date object and its methods.

d=new Date();

/\* Date object created with current date \*/ year=d.getYear();

/\* getYear() returns the current year \*/

**Step 4:** Modify the above program and display the date in the following format.

“dd-mon-yyyy”.

**Step 5:** Write a program to generate a random number between 1 and 10. The Math.random() will generate a number between 0 and 1. Save the file as ―math.html‖.

**Step 6:** Modify the above program, display the number in words.(1 to 10). Use a switch case control structure.

**Step 7:** Find out what the output of the following code. Save the file as ―math1.html‖.

fnum = Math.ceil(-12.5111); snum = Math.floor(-12.5111); tnum = Math.round(-12.5111);

**Step 8:** Write a program to find the length of the following string and save it as

―strllen.html‖.

str = ”Asreet Technologies Limited”. Use the **length** property of the string object.

len=str.length

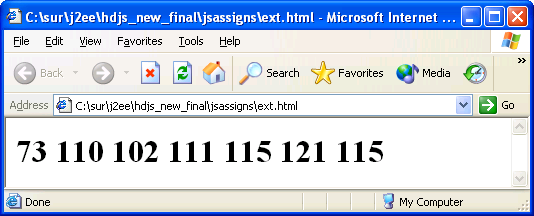
**Step 9:** Write a program to display the ASCII value of the following string. Save the file as

―strascii.html‖

str=”Asreet”.

Use the charCodeAt() function of the string object.

str.charCodeAt(0) will return the ASCII value of the first character ―I‖. Use a loop for displaying the entire string.





**Note:** In Javascript String is an object not an array of characters, so if you want to access individual characters you need to use the methods associated with the string object.

**Step 10:** Write a program to find the occurrence of ―@‖ and ―.‖ in the following email address. Make sure that there is only one occurrence of ―@‖ character. Save the file as

―email.html‖ [email=”jack@yahoo.com”](mailto:jack@yahoo.com) Use the following String methods

1. indexOf(searchValue[, fromIndex])
2. lastIndexOf(searchValue[, fromIndex])

Eg. Here num= email.indexOf(“@”) , the num value is ―4‖.

**Step 11:** Write the program to find the length of the following Array. Save the file as

―array.html‖.

var arr =new Array(1,2,55,66,33,22,1);

Use the length property of the Arary.

**Step 12:** Modify the above program and display individual elements.

**Step 13:** Modify the above program and reverse the ―Array‖. Use the reverse() method of the Array object. Save the file as ―arrayrev.html‖.

**Step 14:** Modify the above program and find the biggest and smallest elements in the Array and save it as ―bigsmall.html‖.

**Step 15:** Divide the following sentence into words and display in the ascending order of words. Save the file as ―split.html‖.

sent=”javascript is very easy to learn”.

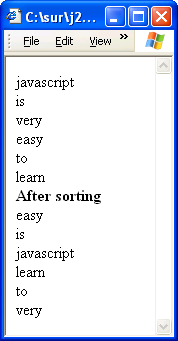
You can use the split() method of the String object to split the sentence to an Array of Strings.

wordarr=sent.split(“ “);

/\* splitting the sentence stored in „sent‟ upon each space \*/

wordarr is become an array of string.

Use the sort() method of the array for sorting the string



### Summary of this exercise:

You have just learnt

Using external Javascript file

Working with Date, Math, String and Array Object.

### Deliverables of the exercise:

* 1. ext.js
  2. ext.html
  3. date.html
  4. math.html
  5. math1.html
  6. strlen.html
  7. strascii.html
  8. email.html
  9. array.html
  10. bigsmall.html
  11. split.html

## Assignment 6:Working with DOM and Important objects

**Objective:** DOM defines the document object hierarchy, here we are learning about the usage of different objects and its usage.

### Estimated time: 20 minutes

**Step 1:** Type the following code and save it as ―navigator.html‖.

<script language="JavaScript">

document.write("the browser name is "+navigator.appName);

</script>



**Note:** “navigator‖ object contains information about the browser and it‘s version.

**Step 2:** Modify the above program and find the operating system of the system and version of the browser.

Use the appVersion and platform properties of the navigator object.

**Step 3:** Type the following code and save it as ―newwin.html‖.

<html>

<head><title> New Window</title><head>

</body>

<font color="red" size="5"> New Window</font>

</body>

</html>

**Step 4:** Type the following code and save it as ―mainwin.html‖.

<html>

<head><title> Main Window</title>

<script language="JavaScript"> function openNew(){

/\* Comments : To open a file „newwin.html‟ in a window by name

„newone‟ \*/

newwin=window.open("newwin.html","newone");

}

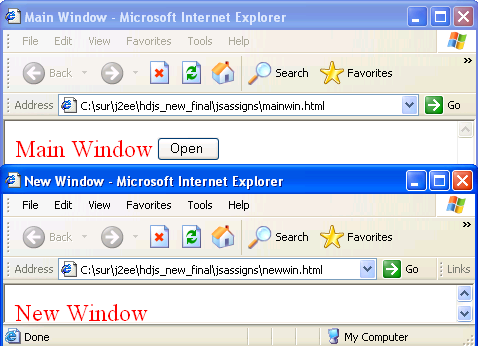
</script>

</head><body>

<font color="red" size="5"> Main Window</font>

<input type="button" value=" Open " onClick="openNew()">

</body></html>





**Note:** Consider newwin=window.open("newwin.html","newone"), ―newwin‖ is the object reference to the new window, ―newwin.html‖ is the html file you want to open and ―newone‖ is the name of the new window.

**Step 5:** Modify the above program, replace the function with the following code.

function openNew(){ newwin=window.open("newwin.html","newone","width=300; height=400;toolbar=no");

}

**Step 6:** Save the ―mainwin.html‖ as ―mainclose.html‖. And modify the following

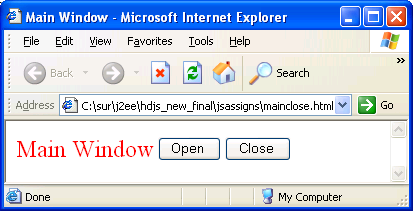
Insert a button with a value ―close‖ and onClick of the button call the following function.

function closeWin()

{

/\* Comments: newwin is the reference of the new window \*/ newwin.close();

}



**Step 7:** Type the following code and save it as ―status.html‖

<script language="JavaScript">

/\*Comments: set a string in the browser status bar \*/

window.status="Check the status bar";

/\*Comments: Status is a property of window \*/

</script>



**Note:** If the status bar is not visible, select menu option, view->status bar.

**Step 8:** Type the following code and save it as ―history.html‖.

<html><body>

<h1>History demo</h1>

<!-- Commnets : Using Inline JavaScript -->

<input type="button" value="back" onClick="history.go(-1)">

</body></html>



**Note:** If you want work with history object more than one page should be loaded in to the browser.

* + 1. Open ―status.html‖
    2. In the same browser open ―history.html‖

**Step 9:** Type the following code and save it as ―location.html‖.

<html><head>

<script language="JavaScript"> function fun() {

/\*Comments: href property of Location object used to specify the complete URL \*/

location.href="status.html";

}

</script>

</head><body>

<h3> Location Demo</h3>

<input type="button" value="change" onClick="fun()">

</body></html>

**Step 10:** Modify the above program, use location.replace(URL) method for replacing the page. Display the hostname and protocol. (Use protocol and hostname properties)

**Step 11:** Type the following code and save it as ―doc.html‖

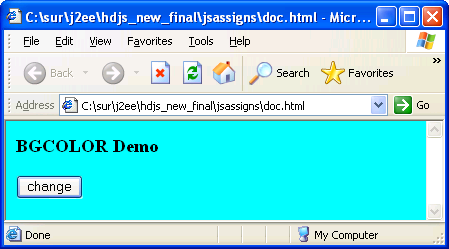
<body>

<h3> BGCOLOR Demo</h3>

<input type="button" value="change"

nClick="document.bgColor='cyan';">

</body>



**Step 12:** Modify the above program, add one more button to change the foreground color of the document. (Use fgColor property of document object). Save the file as ―fgcolor.html‖

**Step 13:** Write a program to open another document, when the user clicks on the change button. Save the file as ―url.html‖.

Use URL property of the document object.

**Step 14:** Write a program to display the last modified date of the HTML document. Save the file as ―lastmod.html‖. ( use lastModified property of document object).

### Summary of this exercise:

You have just learnt

How to use the following objects. navigator

window document history location

### Deliverables of the exercise:

1. navigator.html
2. newwin.html
3. mainwin.html
4. mainclose.html
5. status.html
6. history.html
7. location.html
8. doc.html
9. fgcolor.html
10. url.html

## Assignment 7: Working with Form and Elements

**Objective:** One of the Web‘s most interesting possibilities is the creation of interactive features. You can interact to the website using form elements. Here we are going to learn about how to access the form elements.



**Note:** In ―DOM‖ you can access each element in a form using the following hierarchy.

window -> document -> form -> element.

### Estimated time: 35 minutes

**Step 1:** Type the following code and save it as ―actionpage.html‖.

<html><body>

<h3> Action page demo</h3>

</body></html>

**Step 2:** Type the following code and save it as ―mainform.html‖. Open the page in a browser, click on the submit button verify the URL.

<html>

<body>

<h1> Form Demo</h1>

/\*Comments: Creating a form with name ‗frm1‘ \*/

<form name="frm1" action="actionpage.html" method="get">

<input type="text" name="txt1" value="Asreet"><br>

/\*Comments: onClick of the following submit button the action attribute of form reflects the functionality to be carried out further \*/

<input type="submit" value="submit">

</form></body>

</html>

**Step 3:** Modify the above program, set the method attribute value as ―post‖. Open the page in a browser, click on the submit button verify the URL.

<form name="frm1" action="actionpage.html" method="post">

**Step 4:** Modify the above program, change the ―submit‖ to ―button‖ and save the file as

‖mainform1.html‖. Click on the button. What is the Output?

<input type="button" value="submit">



**Note:** Button can not submit the form to the server unless and until you incorporate client side scripting. Use the submit method of the form for doing the same.

Modify the above program using the following code.

<input type="button" value="Submit" onClick="document.frm1.submit()">



**Note:** Submission of the form is required only for the server side programs. We are dealing with client side programs. So at the point of time we are not dealing with the submission of the form.

**Step 5:** Type the following code and save it as ―formelement.html‖

<html><head>

<script language="JavaScript">

/\*Comments: only three elements in the form, to access that we can use elemenmts array \*/

function funDisp()

{

alert("textbox val is "+document.forms[0].elements[0].value);

alert("textbox val is "+document.forms[0].elements[1].value); alert("textbox val is "+document.forms[0].elements[2].value);

}

</script></head><body>

<form name="frm1">

<input type="text" name="txt1" value="Asreet"><br>

<input type="radio" name="rad1" value="1" checked>Ok<br>

<input type="button" value="Submit" name="but1" onClick="funDisp()">

</form></body></html>

**Step 6:** Modify the above program, replace the function with the following function. Save the file as ―formnames.html‖.

/\* If we know the name of the form and elements, we can use it directly, “frm1” is the name of the form and “txt1” is the name of the textbox \*/

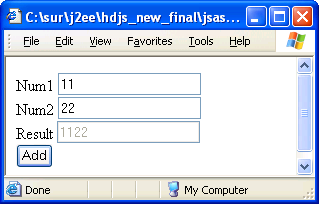
function funDisp()

{

alert("textbox val is "+document.frm1.txt1.value); alert("textbox val is "+document.frm1.rad1.value); alert("textbox val is "+document.frm1.but1.value);

}

**Step 7:** Write a program to accept two numbers from two text boxes and on click of Add button the resultant value should be placed in the third text box as shown below and save it as ―addnum.html‖.



**Step 8:** The sum of ―11‖ and ―22‖ is giving as ―1122‖. Who is responsible for this mistake? Yes it is our mistake, here two strings are getting concatenated and the result is a string. So before adding the numbers convert them to a number.



**Note:** Any value from a form element will treat as a String, so before doing mathematical calculation convert them to numbers.

Modify the above program using the parseInt method.

**Step 9:** Save the above file as ―simplecalc.html‖. Add three more buttons and functionality for subtracting, multiplying and dividing two numbers.

**Step 10:** If the user enters only numbers it works fine, but if the user enters a string then the result will be ―NaN‖. How can we avoid this. Why can‘t we check the validity for the data entered by the user. i.e check if the input is correct or not.

We can use the Top Level function ―isFinite()‖ and ―isNaN()‖for checking if the given string is a number or not.

Consider the following code for checking the first textbox value is a number or not.

/\*Comments : Not isFinite() will return true if num1 is a string \*/

if(**!**isFinite(**num1**))

{

alert("Please eneter a number!"); return false;

}

Modify the ―simplecalc.html‖ save as ―simplecalcval.html‖.

**Step 11:** Create a form with a username filed and a password field (A textbox and password type) and save it as ―user.html‖. Make sure that user is entering a minimum 8 characters and a maximum of 15 characters.

Use maxlength=15 attribute of <input> for checking the maximum length.

**Step 12:** Display a multiple choice question with four choices (Use a radio group with the name ―rad‖). When the user chooses the answer and click on the button it should display the answer is correct or not. Save the file as ―apti.html‖.

All radio buttons should have the same name(―rad‖), so use array of objects and access individual elements using rad[0], rad[1], rad[2], and rad[3]. Use the ―checked‖ property of the radio button to check the user is selected the radio or not.



**Step 13:** Type the following code and save it as ―select.html‖.

<html><head>

<script language="JavaScript"> function disp()

{

ind = document.f1.webs.selectedIndex;

/\*Comments : index of the selected Item \*/

alert("the value is \n"+document.f1.webs.options[ind].value); alert("the text is \n"+document.f1.webs.options[ind].text);

}

</script>

</head><body>

<form name="f1">

<select name="webs" onchange="disp()">

<option value="1"> [http://sparsh](http://sparsh/)

<option value="2"> [http://kshop](http://kshop/)

<option value="3"> <http://172.21.20.155/>

<option [value="4">http://enr.ad.asreet.com/](http://enr.ad.infosys.com/)

</select>

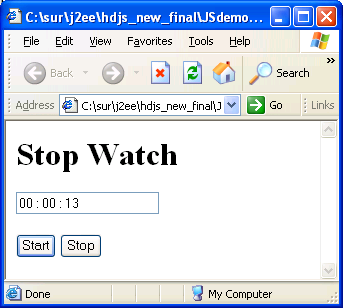
</form></body></html>

**Step 14:** Save the above program as ―selectwebs.html‖. When the user selects a particular website it should redirect to the particular site.

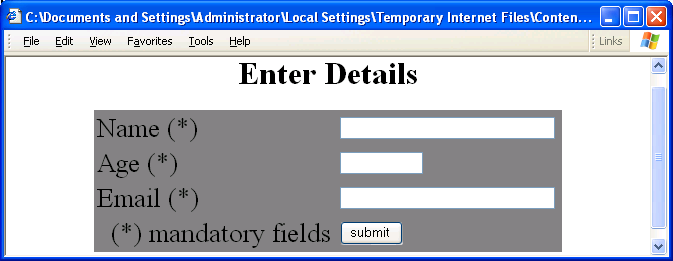
**Step 15:** Create a textarea for accepting feedback from the user, do not allow the user to enter more than 200 characters. Save the file as ―textarea.html‖.

**Step 16:** Write a program to display a stop watch. When user clicks on the start button it should start display the number of seconds and when user clicks on the stop button it should stop counting.

Use id = setTimeout(funName, time in MS) method of window object for calling a function recursively with a time delay. Use clearTimeout(id) for stop the function call. Save the file as ―stopwatch.html‖



**Step 17:** Create the following form, do the validations. Save the file as ―val.html‖.



* All the fields are mandatory.
* Name should be a sting with minimum 3 Character.
* Age should be a number between 1 and 100.
* Email should be a string with and ―@‖ and ―.‖ Character.

### Summary of this exercise:

You have just learnt

Working with form and its properties. Working with the following form elements

Text Password Textarea Radio Select

Form Validation

### Deliverables of the exercise:

1. actionpage.html
2. mainform.html
3. mainform1.html
4. formelement.html
5. formnames.html
6. addnum.html
7. simplecalc.html
8. simplecalcval.html
9. user.html
10. apti.html
11. select.html
12. selectwebs.html
13. textarea.html
14. stopwatch.html
15. val.html.

## Assignment 8: Reusable functions

**Objective:** To create reusable functions .This could be used for the validation of form elements.

### Estimated time:35 minutes

**Step 1:** Create a text box for accepting Employee Number, a submit button by name ‗Submit‘ and save it as ―empvalidate.html‖

**Step 2:** Write functions to validate the following.

* 1. To check whether Empno field is empty ,isEmpty() returns 0 if empty else 1.
  2. To check whether Empno field has ONLY spaces , onlySpaces() returns 0 if all the characters are spaces else 1
  3. To check whether Empno field has ONLY digits**,** onlyDigits() returns 0 if all the characters are digits else 1

**Step 3:** Write a function(validateEmpDetails()) to invoke the functions defined previously for the validation of Emp Number.

**Step 3.1:** Invoke function isEmpty(). If it returns 0 then display a suitable error message and return false from the function.

**Step 3.2:** If isEmpty() returns 1 then invoke the next function onlySpaces(). If

the function returns 0 then display a suitable error message and return false from the function.

**Step 3.3:** if onlySpaces() returns 1 then invoke the next function onlyDigits(). If

the function returns 0 then display a suitable error message and return false else return true from the function.

**Step 4:** if true is returned then action performed should be a message display ―successful validation‖ being written in another html i.e display.html.

**Hint:** Use action attribute for the above

Use Onsubmit event handler as mentioned below. onSubmit= ‖return validateEmpDetails()‖

**Step 5:** Write a separate function to check whether the name has ONLY alphabets[onlyAlphabets()].

**Step 6:** Modify the above HTMl file by adding another text field for accepting Employee Name.

**Step 7:** Append the function(validateEmpDetails()) with the following functionalities. Invoke the functions defined previously for the validation of EmpName.

**Step 7.1:** Invoke function isEmpty(). If it returns 0 then display a suitable error message and return false from the function.

**Step 7.2:** If isEmpty() returns 1 then invoke the next function onlySpaces(). If

the function returns 0 then display a suitable error message and return false from the function.

**Step 7.3:** if onlySpaces() returns 1 then invoke the next function onlyAlphabets (). If the function returns 0 then display a suitable error message and return false else true from the function validateEmpDetails().

**Step 8:** The action performed should invoke the html file display.html in case true is returned from validateEmpDetails() function.

**Step 9:** Include the given js file dateValidation.js in the script tag.

**Step 10:** Modify the above code by adding another text field for accepting DateOfJoining(dd- mon-yyyy).

**Step 11:** Append the function(validateEmpDetails()) with the following functionalities. Invoke the functions defined previously for the validation of DateOfJoining.

**Step 11.1:** Invoke function isEmpty(). If it returns 0 then display a suitable error message and return false from the function.

**Step 11.2:** If isEmpty() returns 1 then invoke the next function onlySpaces(). If

the function returns 0 then display a suitable error message and return false from the function.

**Step 11.3:** if onlySpaces() returns 1 then invoke the next function dateValidate () given in dateValidation.js. If the function returns 1 then display a suitable error message and return false else true from the function validateEmpDetails().

**Step 12:** The action performed should invoke the html file display.html in case true is returned from validateEmpDetails() function.

### Summary of this exercise:

You have learnt how to have reusable functions for client side validation Also have learnt how to use external js function.

### Deliverables of the exercise:

1. empvalidate.html
2. display.html 5)