UNIT-IV

DHTML with Javascript : Data Validation, Opening a new window, messages and confirmations, the status bar, different frames, rollover buttons, moving images.

DHTML

- ❖ DHTML stands for Dynamic HTML.
- ❖ Using DHTML we can develop interactive web pages i.e., the content of the web page can be changed according to the user input.
- ❖ A separate HTML file will be generated for each and every user when they try to use a static web page.
- ❖ To overcome this problem DHTML came into the existence.
- ❖ DHTML included Javascript along with HTML and CSS to make the page dynamic.

DATA VALIDATION

- ❖ Data validation is the process of ensuring that users submit only the set of required characters.
- ❖ It is not the process of ensuring that the data is accurate.
- ❖ Most of the data that the scripts get from users will be textual and not possible to verify.
- ❖ We just ensure that the user has provided formatted input or not.

Consider a login form which accepts username and password from a user. The following constraints are applied on form.

1. Username should contain only characters either upper case or lowercase.
2. Username contains only underscore (_) symbol.
3. Password must contain a uppercase character
4. Password must contain a digit
5. Password should be minimum 8 characters and maximum 14 characters.

Now the validation procedure performs the following operations.

a. First it will validate whether the user specified values for the required fields or not.
b. Username field contains a character or not.
c. Username contains any other special symbol other than underscore (_)
d. Password length should be between 8 and 14 or not.
e. Password contains a digit or not.
f. Password contains a any uppercase character or not.

If the user doesn’t follow any of the constraints then the validation procedure will be failed.

a. Validation can be defined by many validation methods, and deploy in many different ways.
b. Server Side Validation : It is performed by a web browser after input has been specified.
c. Client Side validation : It is performed by a web browser before input sent to the web browser.

Example:

```
<html>
<head>
<title>Data Validation Demo</title>
<script language="javascript">
function validate()
{
```

```
var uname=document.getElementById("txt1").value;
var pwd=document.getElementById("txt2").value;
if(uname=="" || pwd=="")
{
        alert("UserName/Password should not be empty");
}
else if(pwd.length<8 || pwd.length>14)
{
        alert("Password Should be Minimum 8 and Maximum 14 characters");
}
else
{
        alert("Log In Successful");
}

}
</script>
</head>
<body>
<h1 align=center>Log In Form</h1>
<form name="loginform">
<table align=center border=0>
<tr><th>UserName</th><td><input type="text" name="uname" id="txt1"></td></tr>
<tr><th>Password</th><td><input type="password" name="pwd" id="txt2"></td></tr>
<tr><td colspan=2 align=center><input type="button" value="LogIn" onclick="validate()"></td></tr>
</table>
</body>
</html>
```

OPENING A NEW WINDOW

1. To display data dynamically we can open a new window.
2. A new window has a set of options to define what is displayed on the window or not.
3. A new window can be opened using open( ) method of window object.
4. Open( ) method takes three parameters.
    a. URL
    b. Name
    c. List of attributes.
5. URL specifies what should be displayed on the new window.
6. Name specifies the name of the window.
7. List of attributes specify various options and all these options are enclosed in a single or double quotes and all options are separated by a comma.
8. The following are the various options and all these options takes the value 0 or 1, or yes or no.

i.	toolbar	-	whether to display toolbar in the new window or not.
ii.	location	-	whether to display address bar in the new window or not.
iii.	directories	-	whether to display buttons for favorites in the new window or not.
iv.	status	-	whether to display status bar in the new window or not.
v.	menubar	-	whether to display menu bar in the new window or not.
vi.	scrollbar	-	whether to display scrollbars in the new window or not.
vii.	resizable	-	whether to change the size of the new window or not.
viii.	height	-	specifies the height of the new window.
ix.	width	-	specifies the width of the new windows.

Syntax:

```
window.open("URL", name, "attribute list");
```

Example:

```
<html>
<head>
<title>Open Demo</title>
<script language="javascript">
function load(url)
{
        window.open(url,"newwin",'status=0,toolbar=0,esizable=0,width=300,height=3000');
}
</script>
<body>
<a href=" " onclick="load('aditya.jpg')">Next Page</a>
</body>
</html>
```

MESSAGES AND CONFIRMATIONS:

1. Java script provides three built-in window types that can be used to display alert messages and user confirmations.
2. The three kinds of popup windows in javascript are
   a. alert box
   b. confirm box
   c. prompt box

ALERT BOX:

1. alert box is used to display alert messages to the user for doing some action.
2. When a alert box is invoked, a small window is displayed along with a message and a button.
3. The message is the user specified text message and the button is OK button.

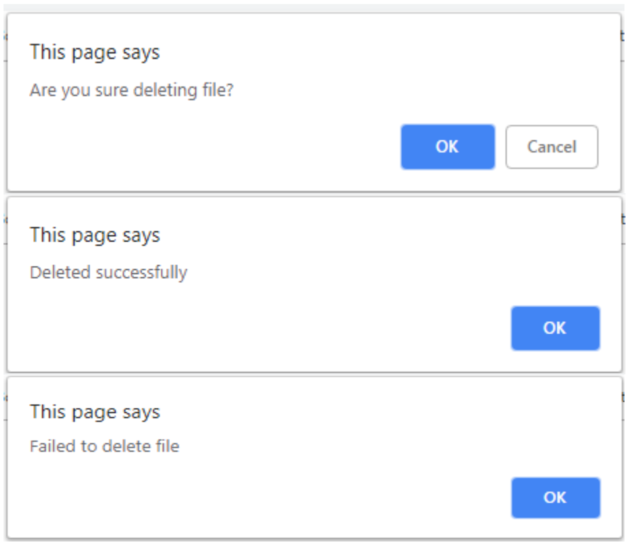
```
        alert(“Message);
        alert(“Welcome”);
```

CONFIRM BOX:

1. Confirm box is used to take user confirmation whether to perform a task or not.
2. Confirm box display a user specified message along with two buttons i.e., OK and CANCEL.
3. Confirm box returns a boolean value i.e., true / false.
4. When user select OK button it return true and return false when CANCEL button is selected.

```
        confirm(“message”);
```

```
<html>
<head>
<title>Messages and Confirmations</title>
</head>
<body>
<script language="javascript">
var x=confirm("Are you sure deleting
file?");
if(x)
alert("Deleted successfully");
else
alert("Failed to delete file");
</script>
</body>
</html>
```


This page says
Are you sure deleting file?

This page says
Deleted successfully

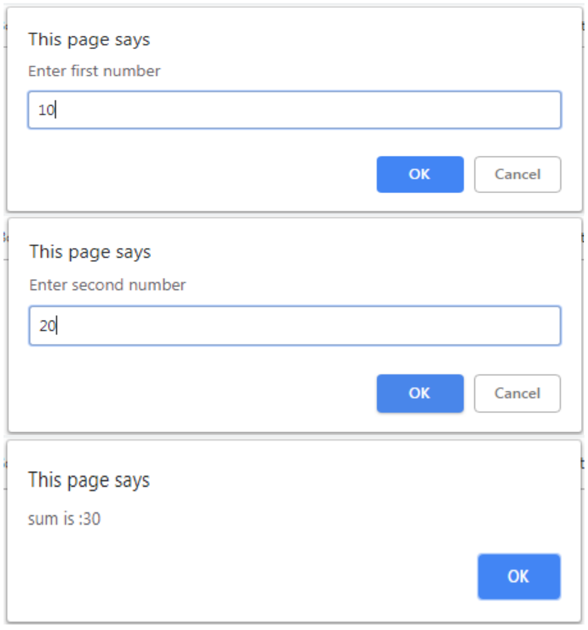
This page says
Failed to delete file

PROMPT BOX:

1. Prompt box is used to accept input from the user through keyboard.
2. Prompt box contains a text field and two buttons OK and CANCEL.
3. Any type of data read using prompt box will be treated as string.

4. If user selects OK button then input specified data will be returned as a string.
5. If user selects CANCEL then null value will be returned.
   prompt("message","default value");

```
<html>
<head>
<title>Prompt Demo</title>
</head>
<body>
<h1 align=center>Prompt Demo</h1>
<script language="javascript">
var x=parseInt(prompt("Enter first number"));
var y=parseInt(prompt("Enter second number"));
var z=x+y;
alert("sum is :"+z);
</script>
</body>
</html>
```



This page says
Enter first number
10

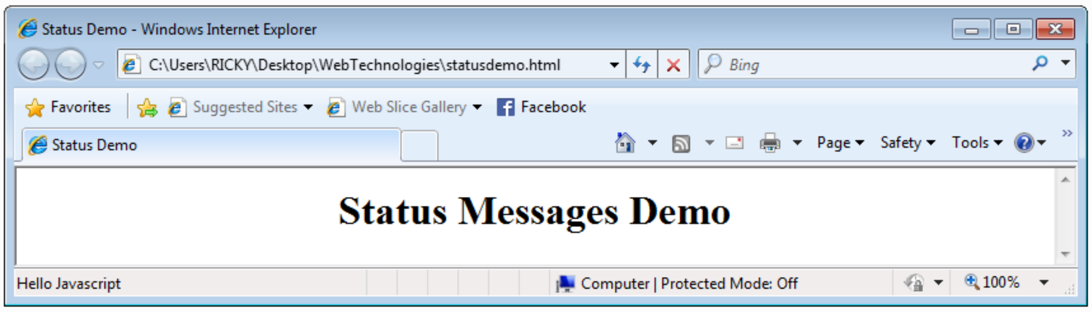
This page says
Enter second number
20

This page says
sum is :30

THE STATUS BAR:

1. Javascript provides you the ability to modify the status bar.
2. The status property sets the text in the status bar at the bottom of the browser or return the previously set status.
3. The status bar basically displays helpful information about the operation of the browser.

```
<html>
        <head>
                <title>Status Demo</title>
                <script language="javascript">
                        self.status="Hello Javascript";
                </script>
        </head>
        <body>
                <h1 align=center>Status Messages Demo</h1>
        </body>
</html>
```



Status Demo - Windows Internet Explorer
C:\Users\RICKY\Desktop\WebTechnologies\statusdemo.html

Status Messages Demo

Hello Javascript

Computer | Protected Mode: Off

DIFFERENT FRAMES:

wdf.html

```
<html>
<head>
<title>Writing to Different Frames</title>
<frameset rows="50,50">
<frame name="top" src="add.html">
<frame name="bottom" src="result.html">
</frameset>
</html>
```

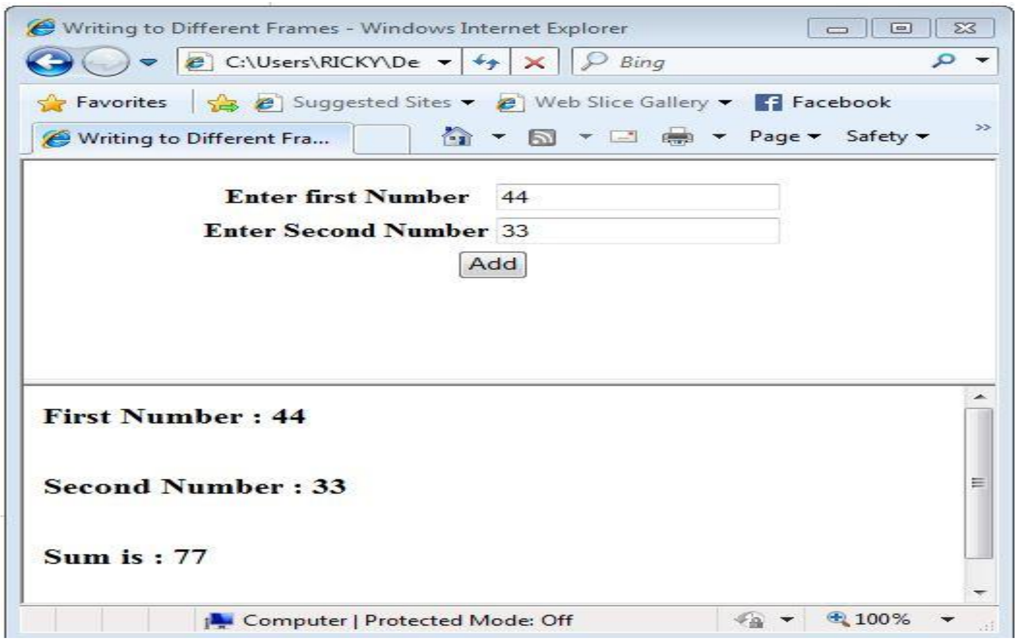
result.html

```
<html>
<head>
<title> Result Page</title>
</head>
<body>
</body>
</html>
```

add.html

```
<html>
<head>
<script language="javascript">
function sum()
{
        var x=parseInt(document.forms[0].n1.value);
        var y=parseInt(document.forms[0].n2.value);
        var z=x+y;
        alert("sum :"+z);
        var frm=parent.frames['bottom'].document;
        frm.open();
        frm.write("<body>");
        frm.write("<h3>First Number : "+x+"</h3><br>");
        frm.write("<h3>Second Number : "+y+"</h3><br>");
        frm.write("<h3>Sum is : "+z+"</h3><br>");
        frm.write("</body>");
        frm.close();
}
</script>
<body>
<form name="addition">
<table border=0 align=center>
<tr><th>Enter first Number</th><td><input type="text" name="n1"></td></tr>
<tr><th>Enter Second Number</th><td><input type="text" name="n2"></td></tr>
<tr><th colspan=2><input type="button" value="Add" onclick="sum()"></th></tr>
</table>
</form>
</body>
<html>
```

1. In the above example a web page contains two frames.
2. The upper frame contains a form which is used to accept data from user.
3. The lower frame contains a html page which display the user specified input and addition of those two values.


Writing to Different Frames - Windows Internet Explorer
C:\Users\RICKY\De
Bing
Favorites
Suggested Sites
Web Slice Gallery
Facebook
Writing to Different Fra...
Page
Safety
Add
Computer | Protected Mode: Off
100%

Enter first Number 44
Enter Second Number 33

First Number : 44

Second Number : 33

Sum is : 77

ROLLOVER BUTTONS

1. Javascript is commonly used to create rollover effects, using the onmouseover event handler to trigger the rollover graphic and the onmouseout event handler to return the graphic to its original state.
2. If just the navigation button changes, it is referred as a single rollover.
3. If the button changes and another graphic anywhere on the page also changes, it is referred as a multiple rollover.
4. Rollover buttons is typically done for navigation buttons.
5. An image rollover occurs when an image is moused over and that image is replaced by another image.
6. Rollovers indicate to the user what button is being selected.
7. To create a single image rollover all that is necessary to modify the “src” property of the image.

```
<html>
        <head>
                <title>Rollover Buttons Demo</title>
        </head>
        <body>
                <img src="image.jpg" onmouseover="this.src='image1.jpg'" onmouseout="this.src='hari.jpg'"
                  height=200 width=200>
        </body>
</html>
```

MOVING IMAGES:

1. Javascript can be used to move a number of elements around the page according to a pattern determined by a logical equation or function.
2. The setTimeout( ) function is used to call a user specified function after completion of specified time.
3. setTimeout( ) function accepts two parameters i.e., function name and time duration.
   Syntax:
   ```
           setTimeout(“function-name( )”, “time-duration”);
   ```
   Example:
   ```
           setTimeout(“display( )”,”20”);
   ```

```
<html>
<head>
<title>Moving Images Demo</title>
<script language="javascript">
var i=1;
function starttimer()
{
        document .getElementById('myimg').style.position='relative';
        document .getElementById('myimg').style.left=+i;
        i++;
        timer=setTimeout("starttimer()",10);
}
</script>
</head>
<body onload="starttimer()">
<img id="myimg" src="hari.jpg" height=200 width=200>
</body>
</html>
```