Unit III : Form and Event Handling

(weightage - 10 marks)

TRANJAL SAVE

Building Blocks of a Form

1) Form is an HTML element which takes user input using various controls like textField, textArea etc...
2) The data can be validated on client browser using the Javascript. After validation of data when user clicks on the "SUBMIT" Button form data is sent to server for further processing.
3) HTML Form is used for development of dynamic web applications where user enters the input and based on the user input server sends response to client.
4) The <form> element is used to create form for user. There are many other elements which are used within form tag.
5) For example, <input>, <textarea>, <button>, <select>, <option> etc....

Syntax

```
<form>
  //form elements
</form>
```

Properties and Methods of Form

Attributes can be added to an HTML element to provide more information about how the element should appear and behave.

Form attributes :-

1) Name - it specifies name used to identify the form.
2) Action - It is used to specify an address (url) where submit the form.
3) method - It is used to specify the HTTP method when submitting the form data on server.

a) GET - Default method, form data is appended to URL when submitted.

b) POST - The form data is not appended to URL. seperately submitted to server.

4) Target - It is used to specify the target of address in action attribute.

i) _blank = the target url will open in new blank window.

ii) _self = default, open in same window

iii) _parent = open in parent frameset.

iv) _top = the url will open in full body of window

```
<form name = "formLogin" action = " "
      method = "GET" >
  //other elements.
</form>
```

Form contains four elements. 1) Input 2) textarea 3) button 4) Label.

Form methods

1) reset() - this method of form object is used to reset form.

Event name = onreset()

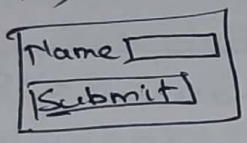
2) submit() - this method of form object is used to submit form

Event name = onsubmit()

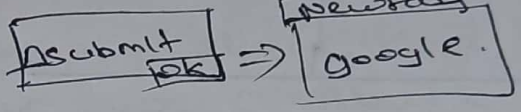
program -onsubmit()

```
<HTML>
<HEAD>
<TITLE>Submit</TITLE>
<script type="text/javascript">
    function pass()
    {
    alert("submit");
    }
</script>
</head>
<body>
<form name:"Login form" action="www.google.com"
 method="GET" target="_blank">
Name<input type="text" name="text"><br>
  <input type="submit" name="submit"
         value="submit">
</form>
</body>
</html>
```

Output



onclick



---

program reset()

```
<script>
   function clean()
   {
    alert("Data Reset");
   }
</script>

<form onreset="clean()">
 Name <input type="text" name="t1">
 <br><br>
 <input type="Reset" name="r1" value="Reset">
</form>
```

Button Element

In HTML following are three types of button that we create using <input> element.

1. Submit
2. Reset
3. Button

Attribute	value	Explanation
Type=" "	button submit reset	Creates a general purpose push button
value=" "	Button text	value display on button
Name=" "	field name	Identity in form Field

~~value~~ <input type="submit" value="SUBMIT" name="abc" />

Text

INPUT element defines an input field. A textbox is created by specifying the type. attribute as "text"

Attribute	value
Type=" "	text
value=" "	Initial default value
name=" "	fieldname
size=" "	width
maxlength=" "	No. of characters can be entered.

<input type="text" size="10" maxlength="5" />

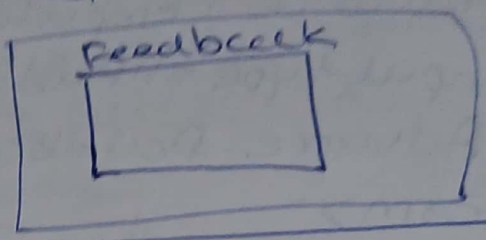
12345

Textarea

The Textarea element defines a multiline text area.

Attributes	value
name	field name
cols	number
rows	number
wrap	hard (line breaks)
	soft (no line breaks)
	off (not wrapped)

```
<form>
feedback <br>
<textarea name="fb"
cols="20" rows="5"
wrap="hard">
enter feedback //by default
</textarea>
```

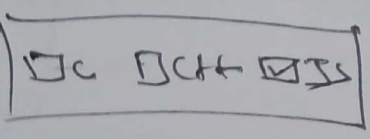


CheckBox

INPUT element defines an input field. when we specify "checkbox" for type attribute of this element, a checkbox is created.

Attributes	value
Type =	checkbox
name =	fieldname
value =	Initial value
checked	true checked

```
<form>
Select subjects.
<input type="checkbox"
name="subject"
value="C"> C
<input type="checkbox"
name="subject"
value="C++"> C++
<input type="checkbox"
name="subject"
value="JS" checked=
"checked"> JS
</form>
```

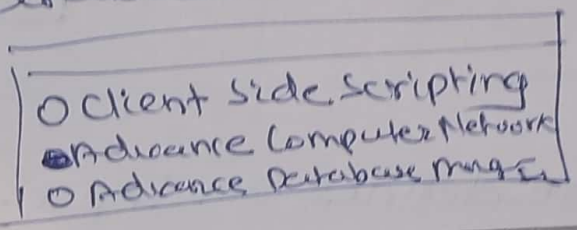


Radio Button

- INPUT elements defines an input field.
- When we specify "radio" for the type attribute of this element, a radio is created.

Attributes. Type, name, value, checked.

```
<form>
  <b> Subject Elective </b> <br>
<input type="Radio" name="q1" value="yes">
  Client Side scripting Language
<input type="Radio" name="q2" value="no">
  Advance Computer Network
<input type="Radio" name="q1" value="no">
  Advance Database management system.
</form>
```



Select Element

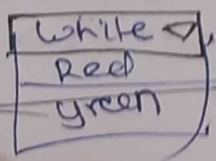
The SELECT element defines a selectable list, and the option element is used to define a list item.

Attributes of select : name, size, multiple

Attributes of option : value="", selected

```
<select name="color">
<option value="white"> white </option>
<option value="Red"> Red </option>
<option value="Green"> Green </option>
</select>
```

```
<select size="3">
<multiple=multiple>
```



2. Form events - mouse event, key events.

orm Events

There are following four events that can be sed to trigger any JS code when there is n event occurs on form level.

Event Name	Description
onchange	Javascript runs this event when an element changes.
onselect	Javascript runs this event when an element is selected.
onblur	Javascript runs this event when an element loses focus.
onfocus	Javascript runs this event when an element gets focus.

```
<script type="text/javascript">
    function validate()
    {
    alert("is selected");
    }
function msg()
    {
    alert("Please enter value");
    }
</script>

<form>
Name=<input type="text" name="t1" value="" onselect="validate()"> <br><br>
Age=<input type="text" name="t2" onblur="msg()">

// Age=<input type="text" name="t3" onfocus="msg()">
```

Mouse Events

The object mouse has numerous event associated with it which depends on the user's actions.

There are following 8 events which are generated by mouse when it comes in contact of any HTML tag.

Event Name	Description
onclick	Javascript runs when a mouse click
ondblclick	Javascript runs when mouse doubleclick
oncontextmenu	Javascript runs when user right click on an element to open context menu
onmousedown	runs when mouse button is pressed
onmousemove	runs when mouse button is move
onmouseout	Javascript runs when mouse pointer out of an element
onmouseover	Javascript runs when mouse pointer over an element.
onmouseup	Javascript run when mouse is released

over & out example()

```
<script>
    function over()
    {
    document.frm1.b1.value="mouse over";
    }
    function out()
    {
    document.frm1.b1.value="mouse out";
    }
</script>
<form name="frm1">
<input type="button" name="b1"
 value="click" onmouseover="over()"
 onmouseout="out()">
</form>
```

Key Events

Following are the three event which care generated by keyboard.

Attribute	Description
onkeydown	keypress
onkeypress	key press & released.
onkeyup	key released

program

```
<script>
    function down()
      {
      document.frm1.b1.value = "key Down!";
      }
    function up()
    {
    document.frm1.b1.value = "key up!";
      }
</script>

<form name = "frm1" >
<input type="text" name="t1" onkeydown=
"down()" on key up = "up()" >

<input type = "button" name: "b1" value = "Press
key" >
```

3.3 : Form objects and Elements

- Webpage is a collection of various elements including window as first element.
- A window contain an HTML document which is known as document object.
- Document object has various properties like document.write() which allow access to and modification of document content.
- Document can have more than one form and form can have multiple elements.

The objects are represented as hierarchical order.

1) Window object - this is top of hierarchy. It is outmost element of object hierarchy
2) Document object - Every HTML document loads into window is document object and it contains the element of page.
3) Form object: all tags those enclosed in <form> ... </form> tags set the form object.
4) Form control element - textbox, button .. etc.

Note:
All the form object stored in array known as form and it keep the order they appered in document.

Example

window.document.forms.invoice
(forms name)

or

window.document.form[1]
↳ 2 position

gram 1 Textbox value in alert box

```
<script>
   function display()
    {
    alert("value = "+ document.forms.entry.name.
                      value);
    }
</script>  <body>
          <form name="entry">
Country <input type="text" name="
<input type="button" name = "b1" value=
    "Print" onclick="display()">
    </form> </body>
```

Program 2

```
<script>
    function display ()
    {
    with(document.forms.entry)
      {
    alert("value=" +element[0].value +element[1].
        value +element[2].value);
      }
    }
</script>
<body>
   <form name="entry">
Country <input type="text" name="Cname">
<br>
State <input type="text" name="Sname">
<br>
City <input type="text" name="Ciname">
<br>
<input type="button" name="b1" value="
    Print" onclick="display()">
</form>
```

getElementById()

- method returns element with specified value
- this DOM method is used for accessing any element on page via its ID attribute

Syntax: getElementById("ID")

program

```
<script>
    function check()
    {
    var i= document.getElementById("myText");
      if( i.value!="" )
        {
        alert("You Entered="+i.value);
        }
      else
        {
        alert("Enter some text");
        }
    }

<form name="entry">
  <input type="text" id="myText">
  <br><br><br>
  <input type="button" value="checker"
    onclick="check()">
  </form>
```

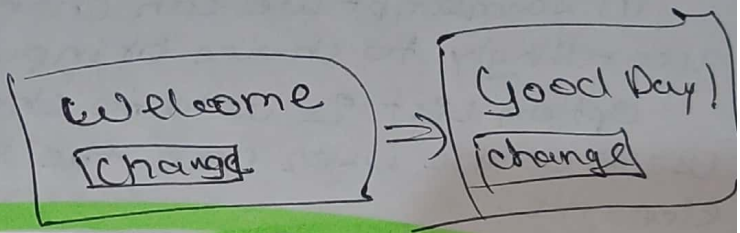
innerHTML:

Each HTML element has an 'innerHTML property that defines both HTML code and text that occurs between that element's opening and closing tag.

program

```
<script>
    function changeText()
    {
document.getElementById("myText").innerHTML =
            "GoodDay";
    }
</script>

<body> <h4 id="myText">welcome</h4>
  <form>
   <input type="button" value="change"
    onclick="changeText()" >
  </form>
</body>
```



3.4 Changing Attribute value Dynamically

In Javascript we can change value of any form elements dynamically.

For example - if element is loaded with default value at time of page load, then the values can be changed at run time.

Onchange event

It executes a Javascript when a user changes the value of an element.

program (textbox color & font color changes) onchange.

```
<form name="entry">
Country: <input type="text" name="cname"
        value="India" onchange="change(this)"/>
State: <input type="text" name="sname"
        value="Assam" onchange="change(this)"/>
</form>
<script>
    function change(Element)
    Element.style.color="blue";
    Element.style.backgroundColor="silver";
</script>
```

3.5

3.5 changing option List Dynamically

In Javascript we can change the values at runtime accordingly to choice or input from user.

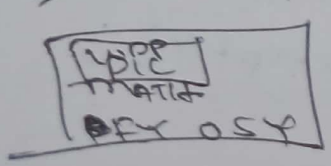
Option List is used to show list of items to users where user can the select one or more elements.

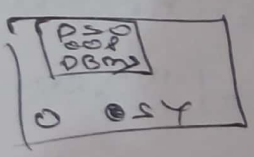
However you can change the values of option list accordingly to choice or input from user using Javascript functions.

program

```
<script>
  function Display(Element.value)
   {
  with(document.forms.frm1)
    {
  if(Elementvalue==1)
    {
    op1[0].text="WPD"
    op1[0].value=1
```

```
op[1].text = "PIC"
op[1].value = 2
op[2].text = "maths"
op[2].value = 3
}
if(Element value == 2)
{
  op[0].text = "OOP"
  op[0].value = 1
  op[1].text = "DSU"
  op[1].value = 2
  op[2].text = "DBMS"
  op[2].value = 3
}
}
</script>
<body>
<form name="frm1">
<select name="op1" size="3">
<option value=1> WPD
<option value=2> PIC
<option value=3> maths
</select>
<input type="radio" name="sub" checked="true"
value=1 onclick="Display(this.value)">
First Year
<input type="radio" name="sub" value=2
onclick="Display(this.value)"> Second
Year
</form>
```





3.6 Evaluating checkBox selection

CheckBox is used to select one or more items from set of choices.

We can write a Javascript function to evaluate wheather checkbox is selected or not and processes as per need of application

program

```
<form name="frm1">
  <input type="checkbox" name="c1" value="C">
    C <br>
  <input type="checkbox" name="c2" value="Ctt">
    Ctt <br>
  <input type="checkbox" name="c3" value="Js">
    Javascript <br>
</form>
<input type="button" name="b" value="show"
       onclick="show()">
</script>

<script>
function show()
{
with(document.forms.frm1)
  {
if(c1.checked==true)
{
alert("C");
}
if(c2.checked==true)
{
alert("Ctt");
}
if(c3.checked==true)
{
alert("Javascript");
}
}
}
</script>
```

3.8 manipulating Form Elements.

- Sometimes it is mandatory to manipulate the form elements after clicking or before form is being submitted to CGI application / DB.
- To validate whether each field on the form is filled with data or not using Javascript we can call such functions or events like onsubmit, onclick etc....
  - many times in form some fields are hidden and at time of submission these fields are assigned hidden value should be submitted.
  - HTML hidden elements are similar to other html elements only the element does not appear on screen.
  - It has name and value attributes that need to send CGI program along with other form elements.

Program

```
<form name="frm1">
Year <input type="text" name="year">
Branch <input type="text" name="branch">
Course code <input type="hidden" name="course">
   <input name="submit" value="submit"
    type="submit" onclick="assign()">
</form>
    <script>
        function assign()
{
with (document.forms.frm1)
{ if (year.value.length>0 && branch.value.length>0)
{
course.value=year.value+branch.value
}
}
}
```

Intrinsic functions

3.9

↳ Javascript provides some special set of built in function known as Intrinsic functions.

↳ In Javascript, when working with forms, intrinsic functions are methods or actions that can be directly applied to form elements like buttons for submitting or resetting a form. These methods are build into Javascript and tried to form element like <form> <inputs> or <button>

- **submit() method**: This is an intrinsic function of form element in Javascript, allowing you to programically submit the form as if the user had clicked a submit button. (Also can add image on button.
- **Reset() method**: This function clear the forms input and resets them to initial values.

Program - 1

```
<HTML>
<body>
  <form id="myform">
First name: <input type="text" name="fname"
  value="John"> <br><br>
Last name: <input type="text" name="lname"
  value="Doe"> <br><br>
  <input type="button" value="Submit
  form" onclick="submitForm()">
  <input type="button" value="Reset
  form" onclick="ResetForm()">
</form>
```

```
<script>
    function submitForm()
    {
    document.getElementById("myform").
                                submit();
    }
    function ResetForm()
      {
    document.getElementById("myform").
                                reset();
    }
</script> </body> </html>
```

Or

Program 2

```
<form name="frm1">
Enter Name:- <input type="text"
  name="txtname" >
<img src="submit.png" height="100"
  width="150"
  onclick="javascript:document.forms['frm1'].
    submit()">
</form>
```