S.N.J.B.'s

SHRI H. H. J. B. POLYTECHNIC, CHANDWAD

CLASS TEST I / II (201 - 201)

Course & Semester :		Roll No.:	Date : / /201
Name of Subject :			Marks obtained :
Sign. of Supervisor :		Sign. of Sub. Examiner :	
Q. No. 1	Q. No. 2	Q. No. 3	Total Marks

Supplied By : SNJB

Chapter 3 Form and Event Handling

0594

Basics of Form

* Building Blocks of Form

Form is typical layout on the webpage by which user can interact with webpage.

The component of forms are text, textarea, checkbox, radio button & push button

These component of form are also called as form controls or controls.

HTML allows us to place these form components on the web page & send the desired information to the destination server.

All these form contents appear in the <form> tag. The form has an attribute action which gets executed when user clicks a button on the form.

Use of form

- to collect info from customer/students for online registrations.
- for online survey.
- for conducting online exam,
- for getting feedback

The info present in the form is submitted to the server for further processing.

Properties & Methods of form :-

(*) Attribute :-

① action :- It specifies the url where the form should be submitted.

② Method :- It specifies the HTTP methods such as GET, POST

get :- Default. Appends the form data to URL in name/value pairs:
URL?name=value & name=value

post :- send the form data as an HTTP post transaction

③ name :- This attribute denotes the name of form.

④ target :- It specifies the target of the address in action attribute. The target values can be as follows:

_blank :- opens in new window.

_self :- opens in the same frame as it was clicked (default)

_parent :- opens in the parent frame

_top :- opens in the full body of the window

framename :- opens in a named frame.

The form can be written inside as <body> tag.

```
<body>
   <form name="myform" action="/myserverPage.
    php" method="GET" target="_blank">
      // code for placing form controls here
      ....
      .....
      .....
   </form>
</body>
```

(*) Text

It is typically required to place one line text.

The text field can be set using
<input type="text" size="30" name="username" value=" ">

- The input type is text & the value of this text field is " ". That means the blank text field is displayed initially & we can enter the text of our choice into it.
- There is size parameter which allows us to enter some size of the text field.
some other parameters or attributes
maxlength that allows us to enter the text of some maximum length
name indicates name of the text field.
align denotes the alignment of the text in the text field. The alignment can be left, right, bottom & top.

eg.-
```
<html>
<body>
<form>
 Input String :   _label
 <input type ="text" size="25" value="">
</form>
</body>
</html>
```

We can specify label using <label> tag as follow:

```
<label>Input String :
<input type="text" size="25" value="">
</label>
```

- The ~~to~~ label get bound to the text box. This aspect is always beneficial for web programmer bcz using label control we can focus on the corresponding text box contents.
- Initially the text box field is blank. We can type some text inside this text box.

* create password field in HTML form :-

```
<form name="form1">
Password: <input type="password">
</form>
```

Text Area :-

It is form component which allows us to enter single line text, what if we want to have multiple line text. Then we can use Text area component

eg.-
```
<html>
<body>
<form>
 Enter the text here
 <textarea cols="40" row="5" name="
   myname">
 </textarea>
 </form>
 </body>
 </html>
```

Various parameter that can be set for the text area can be ~~set text~~

col : specifies total number of columns in text area

row : denote total number of rows in ~~column to~~ text area

name : denote the name of the text area which can be utilized for handling that component for some specific purpose.

wrap can be virtual or physical.
If wrap is virtual then the line breaks get disappeared when the text is actually submitted to the server.
But if the wrap is assigned to the physical then the line breaks (if any) appear as it is in the text.

checkbox

- It is simplest component which is used particularly when we want to make some selection from several options.

- for having the checkbox we have to specify the i/p type as checkbox.

-for eg:-

```
<input type="checkbox" name=" option1"
  value ="mango" checked="checked">
```

If we want to get the checkbox displayed as checked then checked =" checked".

eg:-

HTML document into divide sections

```
<html>
<body>
< form name="checkboxform">
 <div align="center">
 <br> select the fruit of your choice </br>
 <input type="checkbox" name="option1"
   value="mango" checked="checked">Mango
<br/> <input type="checkbox" name="options2"
  value="apple"> <br/>
 <input type= "checkbox" name="options3"
   value=" guava"> <br/>
  </div>
  </form>
 </body>
 </html>
```

Radio Button

- Use to indicate the selection from several choices

- using input type="radio" we can place radio button on the webpage.

- This component allows us to make only one selection at a time.

- We can create a group of some radio button component.

eg:-

```
<html>
<body>
<form name="myform">
<div align ="left"> <br>
<b> select fruit which you like the most</br>
<br/>
<input type="radio" name="groups1" value=
 "Mango"> Mango <br/>
<input type="radio" name="group1" value=
 "Apple"} checked > Apple <br/>
<input type=" radio" name= "group1" value=
 " Greapes"> Grapes. <br/>
<b> select Flower which you like the most
</b> <br/>
 <input type="radio" name="group2" value =
 "Rose"> Rose <br/>
 <input type="radio" name ="group2" value=
 "Lotus"> Lotus <br/>
 <input type ="radio" name=" group2" value =
 "Jasmine" checked> Jasmine <br/>
 </div>
 </form>
 </body>
 </html>
```

Difference between group of Checkbox buttons & radio buttons.

- Both are used for making the selection from group of choices.

-when user selects checkbox, it values get assigned as the current value of the checkbox group's control name.

-A checkbox group's control name may get paired with several current values if the user selects more than one checkbox.

-Radio buttons work just like checkboxes except they are typically set up to be mutually exclusive of one another, ie when one is selected, all the others are automatically deselected.

Button

- we can create the button using <input type="button"> tag.

-There are two types of buttons that can be created in HTML

one is called submit button & another one is reset button.

Various parameters of submit button are

name denotes the name of the submit button.

value is for writing some text on the text on the button.

align specifies alignment of the button.

2

000492

S. N. J. B's
SHRI H. H. J. B. POLYTECHNIC, CHANDWAD

CLASS TEST I / II 20 - 20

Course & Semester :		Roll No. :	Date : / /20
Name of Subject :		Marks obtained :	
Sign. of Supervisor :		Sign. of Sub. Examiner :	
Q. No. 1	Q. No. 2	Q. No. 3	Total Marks

cgi=common gateway interface

```
<html>
    <body>
     <form name="myform" action="http://www.
       localhost.com/cgi.bin/hello.cgi" method
       ="POST">
     <div align="center">  <br>
     <input type="text" size="35" value="">
     <br>
     <input type="submit" value="send">
     <input type="reset" value="Reset">
     <br>
     </div>
      </form>
      </body>
      </html>
```

There are two attributes associated with the form tag & those are action & method. The action parameter indicates the address & the cgi script where the contents should go & method parameter is for the methods for submitting the data.

The method can be get or post. Thus by specifying the action & method for a form we can send the desired data at desired location.

Select Element :-

- HTML allows us to have pop down menu on webpage so that the desired selection can be made
- The parameter select is for the menu component & option parameter is for setting the values to the options of drop down menu.
- We can make some specific option selectedby selected value=.

eg:-

```
    <html>
        <body>
          <form name="myform">
          <div align="center">
           <select name=" My_Menu">
       <option value="Mango">Mango </option>
<option value="Strawberry"> Strawberry </
   option>
<option selected value="Banana"> Banana
</option>
  </select>
   </div>
    </form>
     </body>
    </html>
```

Form Events

- It is activity that represents a change in the environment.
  - for eg:- mouse click, pressing a particular key of keyboard represent the events
  - A javascript event is an action that can be detected by javascript.
  - Many of theme are initiated by user actions but some are generated by the browser itself.
  - We say then an event is triggered & then it can be caught by Javascript functions, which then do something in response.
  - Event handler is a script that gets executed in response to these events.
  - Thus event handler enables the web document to respond the user activities through the browser window
  - Events are specified in lowercase letters & these are case-sensitive

commonly used events & tag attribute

Event	Intrinsic event attribute	Meaning	Associated tags
① blur	Onblur	losing the focus	<button> <input> <a> <textarea> <select>
② change	onchange	on occurance of some change	<input> <textarea> <select>

③ click	onclick	When user clicks the mouse button	<a> <input>
④ dblclick	ondblclick	when user double clicks the mouse button	<a> <input> <button>
⑤ focus	onfocus	when user acquires the input focus.	<a> <input> <select> <textarea>
⑥ keyup	onkeyup	when user releases the ~~the~~ key from the keyboard.	Form elements such as input, button, text textarea etc
⑦ keydown	onkeydown		—11—
⑧ keypress	onkeypress		—11—
⑨ mousedown	onmousedown		—11—
⑩ mouseup	onmouseup		—11—
⑪ mousemove	onmousemove		—11—
⑫ mouseout	onmouseout		—11—
⑬ mouseover	onmouseover		
⑭ load	onload	after getting the document loaded	<body>
⑮ reset	onreset	when the reset button is clicked	<form>
⑯ submit	onsubmit	when the submit button is clicked	<form>
⑰ select	onselect	On selection	<input> <textarea>
⑱ unload	onunload	when user exits the document	<body>

Event Handling eg:-

```
<html>
 <head>
 <script type="text/javascript">
  function A()
   {   alert("Welcome");
   }
  </script>
  <body onload="A()">   ⟹ when web document gets loaded on the browser window then A() will be called
 </body>
 </html>
```

Mouse Event

are used to capture the interactions made by user using mouse

Event
onclick → The mouse was clicked on element
ondblclick → —11— doubleclicked —11—
onmousedown → —11— pressed down over element
onmouseup → —11— released over element
onmouseover → —11— moved over element
onmouseout → —11— moved off of element
onmousemove → —11— moved while over element.

eg:-

```
<html><head><script type="text/javascript">
function A()  { alert("Hello"); }
</script></head><body><form>
<input type="button" value="Click" onclick
 ="A()">
</form>
</body>
</html>
```

Key Event

the events that occur when the user interact using the keyboard

Event :-

onkeydown : when user presses the key on keyboard that was down

onkeypress : The user presses a key

onkeyup : The user releases a key ~~that~~ ~~was~~ on keyboard.

eg:-

```
<html>
  <head>
    <script type="text/javascript">
       function A()
       {
          alert("Hi");
       }
    </script>
    </head>
    <body>
     <form>
    Enter Name:
     <input type="text" onkeypress=
        "A()">
     </form>
     </body>
     </html>
```

Form objects & Elements :

- website contains various objects. The very first object which we see is window.
  - The window object contains the HTML document which is called as document object.
- This document object has several important functionalilities
- out of which, the most commonly used functionality is write() method.
  - We can display any message on web page using following statement

```
document.write ("Hello");
```

Once can omit the use of window. That means instead of writing window.document.write then it is perfectly allowed.

- A form is placed on document. The form objects are stored in array called forms. They appear in the order in which the forms appear in document
- each form can be refered by using the index of the forms array.

for eg:-
suppose we want to refer second form then we write

```
document.forms[1]
```

- A form can be referred by its name.
- For instance - if a form has name form1 then we can refer this form as

```
document.forms.form1.
```

S. N. J. B's
SHRI H. H. J. B. POLYTECHNIC, CHANDWAD
CLASS TEST I / II 20 - 20

Course & Semester :		Roll No. :	Date : / / 20
Name of Subject :		Marks obtained :	
Sign. of Supervisor :		Sign. of Sub. Examiner :	
Q. No. 1	Q. No. 2	Q. No. 3	Total Marks

eg:-

```
<html>
<head>
<script type="text/javascript">
 function A()
 {   alert(" You have entered the name: "
   +document.forms.myform.username.value);
  }
</script>
 </head>
 <body>
  <form name="myform">
 Enter name: <input type="text" name=
        "username" />
<input type="button" value="Click me"
   onclick=" A()"/>
</form>
</body>
</html>
```

Document.forms.myform.username.value

- Document.forms → Refering to forms array
- myform → Named the form
- username → Name of text box
- value → value within the text box,

Using the alert popup box the text typed within the text box will be displayed.
We can modify the above given A() as follows for efficient use

```
function A()
{
  with(document.forms.myform.username)
  {
      alert("You have entered the name:"
       + value);
  }
}
```

Changing Attribute Value Dynamically

during form feeling process itself, the color or font of the text field can be changed. This dynamic change helps the user to notify the importance changes in the form fields.

eg:- changing the background color of text box as the user enters some data in it.

eg:-

```
<html>
<head>
  <script type="text/javascript">
   function changeMe(Ele~~ment~~)
   {
      Ele~~ment~~.style.background='pink'
   }
  </script>
</head>
<body>
<form name="myform">
Enter RollNo : <input type="text"
name="roll" onchange="changeMe(this)"/>
Enter Name: <input type="text" name=
 "name" onchange="changeMe(this)" />
<input type="submit" value="Submit"/>
 </form>
</body>
</html>
```

⊛ Changing Option List Dynamically

- option list represents the list of one or more than one items which can be chosen by user.
- In web application, it is common practice to change the contents of the option list base on some category chosen.
- Javascript allows to change the items present in the list dynamically.

eg:-
```
<html>
  <head>
  < script type="text/javascript">
   function Myselection(val)
   {
       with(document.forms.myform)
       {
           if(val==1)
           {
           choices[0].text="Mango";
           Choices[0].values=1;
           Choices[1].text="Orange";
           choices[1].value=2;
           choices[2].text="Banana";
           Choices[2].value=3;
           }
```

```
        if(val==2)
        {
          choices[0]. text ="Rose";
          choices[0]. value=1;
          choices[1].text =" Jasmine";
          choices[1]. value =2;
          choices[2]. text="Lotus";
          choices[2]. value=3;
        }
        if( val==3)
        {
           choices[0].text="Red!"
           choices[0].value =1;
           Choices[1].text =" Green";
           Choices[1].value=2;
           choices[2].text="Blue";
           choices[2].value=3;
        }
      }
    }
</script>
</head>
<body>
<form name="myform">
<input type ="radio" name="items"
   checked="true"  value =1  onclick=
   Myselection(this.value)>Fruit
<input type="radio" name="items"
 ~~che~~ value= 2  onclick=Myselection(this.
 value> Flower
<input type="radio" name="items"
  value=3  onclick=Myselection(this.
  value) > Colors
   <select  name="choices">
        <option Value=1> Mango
     < option  Value =2> Orange
      < option  value =3> Banana
   </select>
   </form>
   </body>
   </html>
```

❋ changing a Label Dynamically

we can change the label of any form element dynamically. The same element can be used for multiple purpose by simply Changing the label.

Q. We changing the label of the reset button. Hence same button can be used for Fruit & Flowec list ~~dg~~ display. The options of the List box also changed.

→
```
 <html>
  <head>
    <script type="text/javascript">
     function A(val)
     {
       with(document.forms.myform)
        {
           if(val=="Fruit")
           {
            Button_Label.value="Flower";
            choices[0]. text ="Rose"
            choices[0].value=1;
            choices[1].text =" Jasmine";
            choices[1]. value= 2;
            Choices[2]. text =" Lotus";
            choices[2].value=3;
           }
```

```
if( val=="Flower")
  {
    Button_Label.value ="Fruit";
    choices[0].text ="Mango";
    Choices[0].value=1;
     choices[1].text ="orange";
      choices[1].value= 2;
    choices[2].~~value~~ text= "Banana";
     choices[2].value=3;
  }
 }
}
</script>
 </head>
  <body>
  <form name="myform">
    <select name="choices">
      <option value=1> Mango
       <option value=2> Orange
       < option value=3> Banana
     </select>
 <input type="reset" name="Button_Label"
  value="Fruit" onclick= "A(this.value)">
    </form>
    </ body>
    </html>
```

Manipulating Form Elements :-

We can ~~man~~ manipulate the form elements before submitting it to web server. For that purpose we can keep some of the fields hidden & at the time of submitting the form, the desired value can be set to the hidden field so that the assigned value for hidden field can be submitted.

Q. The user enters roll number & name. The registration id for the student can be formed by taking first two characters of name followed by the rollno. Initially the registration id field is kept hidden & at the time of submitting the form this value is assigned to the registration field.

→
```
<html>
 < head>
   <script type=" text/javascript">
     function A()
      {
       with( document.forms.myform)
        {
          if(name.value.length>0 &&
          roll.value.length>0)
          regid.value=name.value.charAt
         (0)+ name.value.charAt(1) +
          roll.value
        }
      }
   </script>
   </head>
   <~~/bd~~body>
   < form name =" myform">
   Roll Number: <input type=" text"
      name="roll">
   Name: <input type="text" name=
    "name">
```

```
Reg.ID :<input type ="hidden" name=
        " regid">
<input type ="submit" name="submit"
  value="submit" onclick ="AC)">
</form>
</body>
</html>
```

* Intrinsic Functions:-

— the built in functions that are provided by Javascript.

— for submit or reset button

— once can use these functionalities while submitting the form or resetting the form fields.

- The submit() method of the form object can be used to send the form to the server in exactly same way as if the user has pressed the submit button.

```
eg:-  <html>
      <body>
      <form name ="myform">
RollNumber: <input type="text" name="roll">
Name:<input type="text" name="name">
<img src ="submit.gif" onclick="javascript.
document.forms.myform.submit()">
 <form>
  </body>
  </html>
```

000384

S.N.J.B.'s

SHRI H. H. J. B. POLYTECHNIC, CHANDWAD

CLASS TEST I / II (201 - 201)

Course & Semester :		Roll No.	Date : / /201
Name of Subject :			Marks obtained :
Sign. of Supervisor :		Sign. of Sub. Examiner :	
Q. No. 1	Q. No. 2	Q. No. 3	Total Marks

Supplied By : SNJB

Disabling Elements

- We can restrict some fields on the form by using disabled.
- If disabled property of particular form element is set to true then user can not edit that element.
- Similarly on setting disabled property to false we can edit the field.

```
eg:-   <html>
       <head>
        <script type=" text/javascript">
         function AC)
         {
         document.forms.myform.name.disab
        led=false ;
         }
          function BC)
         {
         document.forms.myform.name.disable
         d= true;
         }
       </script>
       </head>
       <body>
       <form name ="my form">
```

```
User Name: <input type="text" name=
 "name>
<input type="button" value="Disable
   Name Field" onclick=" B()">
<input type="button" value=" Enable
  Name Field" onclick="A()">
</form>
 </body>
  </html>
```

* Read_only Elements

- sometimes we need to set some value to field which user should not change.

- To restrict user from changing the value of particular field we make that element readonly by setting readonly= true

- If the readonly attribute is set to false then anyone, including the user entering info into the form, can change the value of the element

eg:-

```
        <html>
        <head>
          <script type="text/javascript">
          function A()
           {
              document.forms.myform.name
               .readOnly=true;
           }
        </script>
        </head>
         <body>
         <form name="myform">
User Name:
  <input type="text" name="name">
<input type="button" value="Read Only
Name Field" onclick="A()">
 </form>
  </body>
   </html>
```