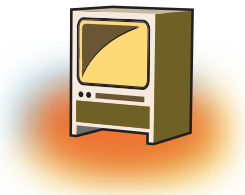


Chapter 4
Forms

Learning Objectives

After learning this chapter learner will be able to:-

- State the purpose of the FORM tag.
- List the attributes that can be used with the FORM tag to make web page more interactive.
- Learn to post the details filled in the form on the server.
- Differentiate the working of different controls.
- Understand the use of all the controls placed on the form.
- Assign attributes to the <input> element to create the different FORM objects

INTRODUCTION

A form is an object that is used for collecting data from the user. We generally come across such forms whenever we are creating a new account either in bank (manually) or for an email id (computerized).

In HTML, a form is a window that consists of the elements of a form called the form fields. These fields may be text field, text area, drop-down box, radio buttons, checkbox and/or a command button. HTML forms are used to pass data to a server. A form can be inserted in HTML documents using the HTML form element which acts as a container for all the input elements. All the information collected by a form can be submitted to a processing agent (a file containing a script made to process this information) that's usually specified in the "action" attribute of the Form tag.

The basic syntax is of a form is:

```
<form method = "get | post" action = "">
 <input>
 <input>
</form>
```

The above code contains both the start tag and the end tag of <form> to indicate the beginning of form object and end of form object. The <input> tag inside the form tag

does not have an end tag. There can be as many <input> tags as form fields you want in your web page.

1. METHOD ATTRIBUTE OF FORM

The method attribute specifies how to send form-data (the form-data is sent to the page specified in the action attribute). The form-data can be sent as URL variables (with method="get") or as HTTP post transaction (with method="post").

1.1 GET method:

- Appends form-data into the URL in name/value pairs
- The length of a URL is limited (about 3000 characters)
- Never use GET to send sensitive data! (will be visible in the URL)
- Useful for form submissions where a user want to bookmark the result
- GET is better for non-secure data, like query strings in Google

1.2 POST method:

- Appends form-data inside the body of the HTTP request (data is not shown is in URL)
- It has no size limitations
- Form submissions with POST cannot be bookmarked

  Syntax

  <form method="get|post" action= "address">
- action=address
- method=post or method=get

1.3 Attribute Values

Value	Description
Get	Default. Appends the form-data to the URL in name/value pairs: URL?name=value&name=value
Post	Sends the form-data as an HTTP post transaction

2. INPUT TAG

The <input> tag collects the information from the user.

The attributes are:

Attribute Name	Description
Name	takes a string of characters as internal name of the field, to be used as a reference later
Size	takes the value of a number in quotes which is equal to the width of the field.
Maxlength	takes the value of a number in quotes which is equal to the maximum number of characters that can be entered.
Type	takes the value of the field. It can take the value as "text" or "radio" or "checkbox" or "submit".

2.1 TextBox Field

If the value of the **type** attribute is "text" i.e. **<input type= "Text">,** the form will show a textbox. This textbox accepts the input in one line.

Along with this the TextBox field accepts value, size, name, maxlength, align and tabindex within the <input> tag.

Attributes of Text Field	Description
Size	It defines the width of the field. It contains no. of visible characters.
Maxlength	It defines the maximum length of the field. It contains maximum no. of characters that can be entered in the field.
Name	It adds an internal name to the field so the program that handles the form can identify the fields.
Value	It defines what will appear in the box as the default value.
Align	It defines how the text field will be aligned on the form. Valid entries are: TOP, MIDDLE, BOTTOM, RIGHT, LEFT, TEXTTOP, BASELINE, ABSMIDDLE, ABSBOTTOM.
Tabindex	It defines in which order the different fields should be activated when the visitor clicks the tab key.

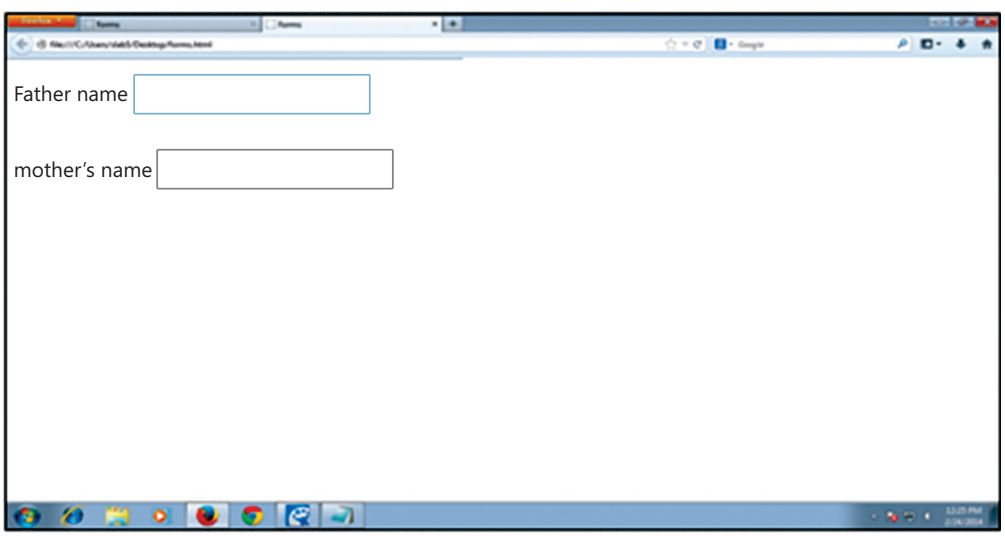
For Example:

Two textboxes whose internal name (computer can understand the internal name) is "Fname" and "Mname" are created using the value "text" for the **type** attribute. The internal name is different from the label.

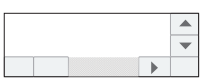
```
HTML Code:
<form>
Father's Name:
<input type="text" size = "20" name="FName" >
<br>
Mother's name:
<input type="text" size = "20" name="MName">
</form>
```

The above code creates two textboxes, each of which is 20 characters wide. The two textboxes are arranged vertically one below the other. The <br> tag does the work of Enter key.

Output:



2.2 Text Area



Text areas are the fields that displays several text lines at a time. Textarea field is generally used to prepare the body of the email or use it to take comment from the user. The <textarea> tag has both start and the end tag indicating from where the textarea begins and where the area where you were writing text ends.

The attributes are:

Attribute Name	Description
Cols and Rows	It takes the value as number. While cols indicate the length of the textarea, the rows indicate the number of rows with text that will be visible at a time.
Name	It provides an internal name to the textarea field as understood by the computer.
Tabindex	It defines the order number of activation of this field when the visitor clicks on the tab key.
Wrap	It helps in flowing the text to the next row in a paragraph when the text reaches the right hand border of the text area. (i) It can take three types of values as discussed below: (ii) If the text is not wrapped, i.e. its value is set to "**wrap off**", it will be placed in the first row of the text area. In such a case if the sentence doesn't get completed at the right hand border of the text area, the text after the border will not be visible. (iii) The value **Virtual** recognizes the text with line breaks when textarea is loaded on the web page. (iv) **Physical** defines the format of the text. It will appear on the webpage as inserted by the user.

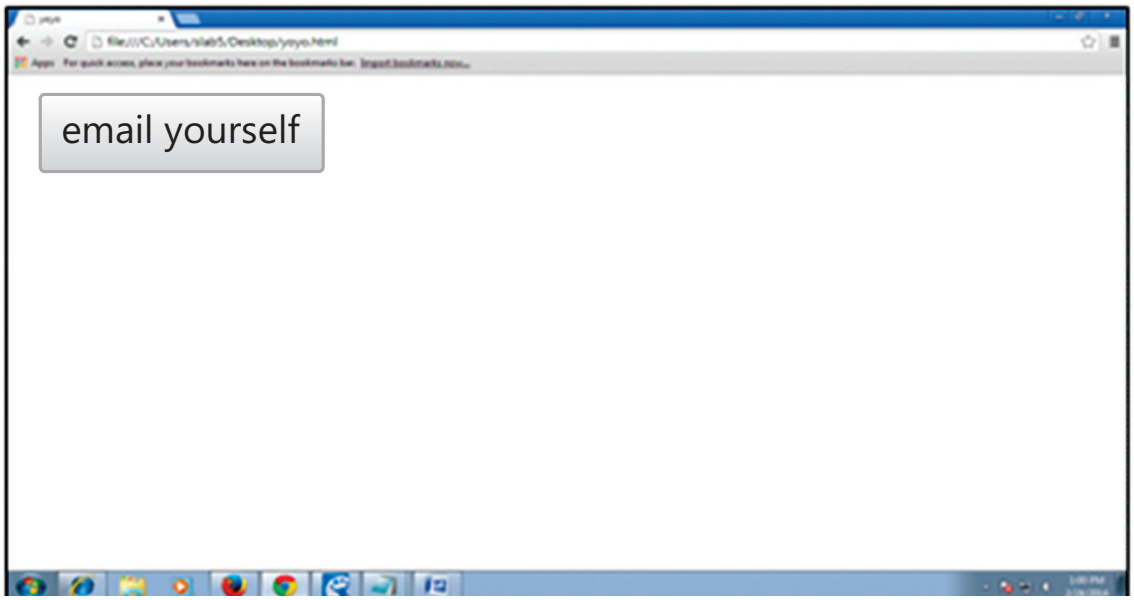
For Example:

The following code displays the use of <textarea> tag.

```
<form method="post" action="mailto:youremail@gmail.com">
    <textarea rows="5" cols="30" wrap="physical" name="comments">
            Enter Comments Here
    </textarea>
<input type="submit" value="Email Yourself">
</form>
```

The above code produces the following output which displays a text area of 5 lines visible at a time and 30 characters long with a word wrap facility where the text flows to the next line if the active text area is smaller than 30 characters due to small size of the webpage opened by the user.

Output:



2.3. Radio Buttons

If the value of the **type** attribute is "radio" i.e. **<input type= "Radio">,** the form will show a radio button. This button is also called toggle button.

Radio button enables the selection of one of the options out of the many.

Attributes of Text Field	Description
Name	It adds an internal name to the field so the program that handles the form can identify the fields. To group the radio buttons so that only one could be selected at a time, same name is given to all the buttons.
Value	It defines what will be submitted if checked.
Align	It defines how the text field will be aligned on the form. Valid entries are: TOP, MIDDLE, BOTTOM, RIGHT, LEFT, TEXTTOP, BASELINE, ABSMIDDLE, ABSBOTTOM.
Tabindex	It defines in which order the different fields should be activated when the visitor clicks the tab key.
Checked	This attribute can have value (e.g., checked="checked | unchecked").

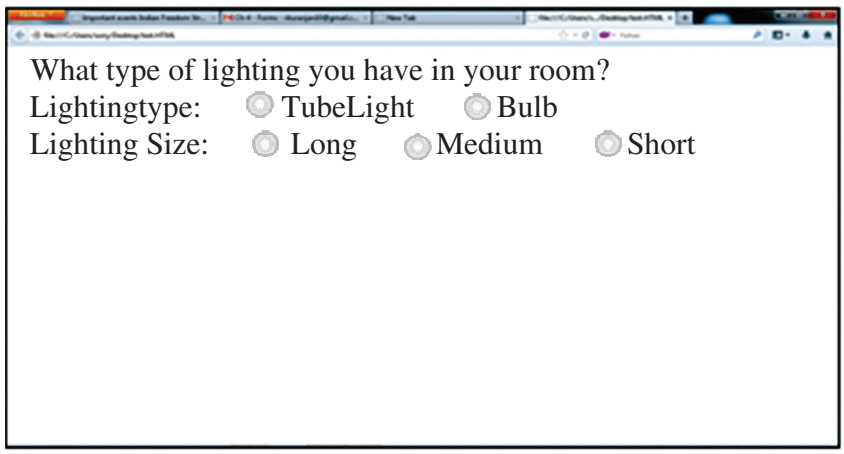
For Example:

```
<form>
What type of lighting you have in your room? <br>

Lightingtype:
<input type=”radio” name=”Ltype” value=”tube”>TubeLight
<input type=”radio” name=”Ltype” value=”bulb”>Bulb <br>

Lighting Size:
<input type=”radio” name=”LSize” value=”Long”>Long
<input type=”radio” name=”LSize” value=”medium”>Medium
<input type=”radio” name=”LSize” value=”short”>Short <br></form>
```

The above code produces the following output:



The user is allowed to select one of the two from Lighting type i.e. either TubeLight or Bulb. Similarly, the user can select any one of the three from Lighting Size i.e. Long, Medium and Short. This is achieved by naming all options of a set the same. Here all light type has the name Ltype while all light size is name LSize. This is where the id attribute comes handy, especially if the value has to be used later on.

2.4. Checkboxes ☐☑☐

If the value of the type attribute is “checkbox” i.e. <input type= “Checkbox”>, the form will show a checkbox. As compared to radio button, a checkbox allows for multiple selections of items.

The check box’s attributes namely, name, align, value and tabindex behave the same as a radio button’s attributes.

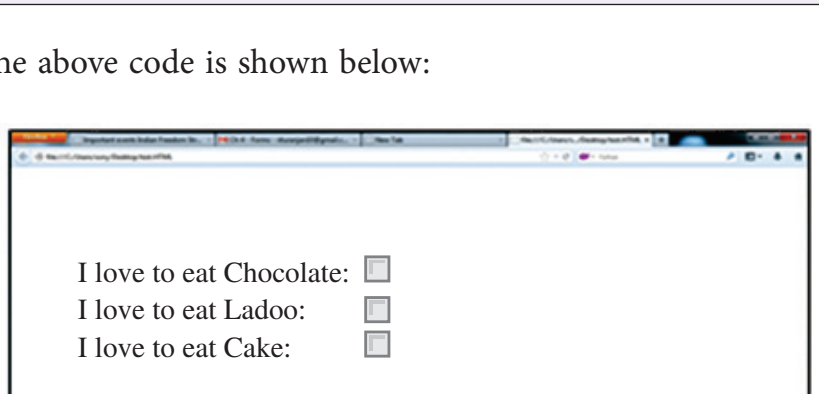
Every checkbox does not get a different name, rather a collection of checkboxes get the same name so as to refer to a group as shown in the following code.

For example:

All the checkboxes have the same name i.e. "sweet", though the value for each one of them is different.

```
<form>
I love to eat Chocolate:
<input type="checkbox" name="sweet" value="Chocolate">
<br>
I love to eat Ladoo:
<input type="checkbox" name="sweet" value="Ladoo">
<br>
I love to eat Cake:
<input type="checkbox" name="sweet" value="Cake">
</form>
```

The output of the above code is shown below:



2.5. Command Button

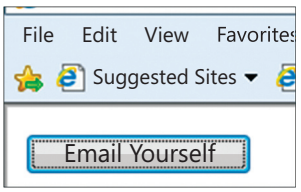
If the value of the **type** attribute is "Submit" i.e. **<input type= "submit | reset">,** the form will show a command button.

- **Submit buttons**: When activated, a submit button submits a form. A form may contain more than one submit button.
- **Reset buttons:** When activated, a reset button resets all controls to their initial values.

This command button will perform some action when the user clicks on it. The action is defined by **action** attribute in <form> tag along with the **method** attribute as shown in the following code:

```
<form method=”post” action=”mailto:youremail@email.com”>
<input type=”submit” value=”Email Yourself”>
</form>
```

The above code displays the following output:



When the user click on the command button called “Email Yourself”, the action specified in <form> tag is performed. It opens your email inbox in email.com.

2.6. Drop down Box

Drop down Box contains a list that prompts the user to select one item from the list. It is created by using <select> and <option> tags. Both <Select> and <option> tag have start and an end tag. A SELECT element must contain at least one OPTION element.

The attributes used are:

Attribute	Description
Name	It adds an internal name to the field so the program that handles the form can identify the fields
Size	It defines the number of items to be visible when user clicks on the drop down box
Multiple	It allows for multiple selections
Value	It defines what will be submitted to the computer when an item is selected. If the **value** attribute takes the string as “CHO”, then “CHO” will only be understood by the computer when the item selected is “Chocolate”.

The <option> tag is used for creating a list inside a <select> tag as shown under:

```
<select>
<option>Chocolate</option>
<option>Ladoo</option>
<option>Cake</option>
</select>
```

Highlighting one item in the dropdown box

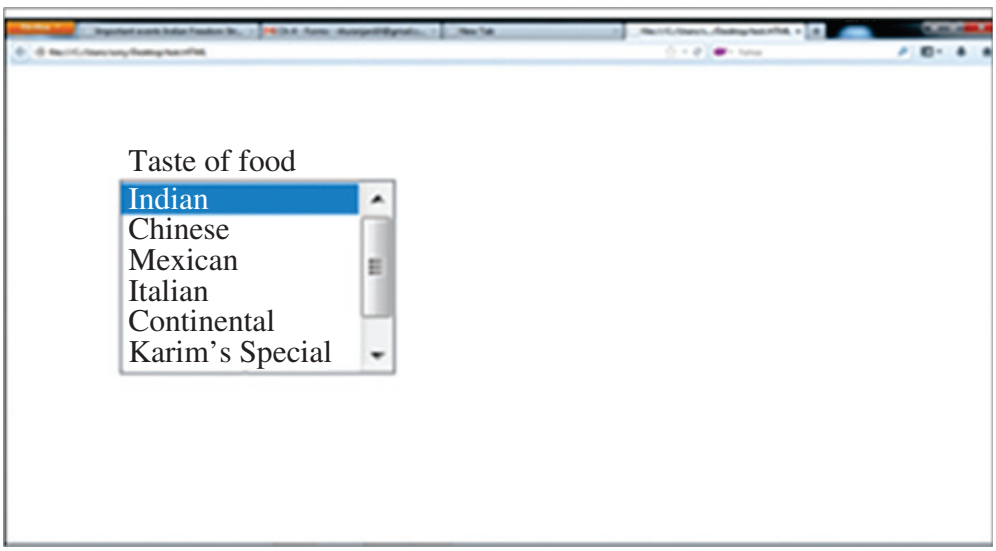
You can force an item to be selected by default by adding the "selected" in the <option> tag as <option selected>.

For Example:

The following code displays the use of <select> and <option> tag to show the dropdown box in the browser. The **size attribute takes the value as number "4", specifying the dropdown box to show 4 items at a time in the list box.**

```
<form method="post" action="mailto:youremail@email.com">
Taste of food
<select multiple name="taste" size="4">
<option value="indian" selected>Indian</option>
<option value="chinese" >Chinese</option>
<option value="mexican" >Mexican</option>
<option value="italian" >Italian</option>
<option value="continental" >Continental</option>
<option value="karim" >Karim's Special</option>
<option value="japanese" >Japanese Sweets</option>
</select>
<input type="submit" value="Email Yourself">
</form>
```

The following output will display 6 items in the dropdown box, which is already open and where the first item is already selected.



Summary

1. A form is an object that is used for collecting data from the user.
2. A The method attribute specifies how to send form-data using URL variables (with method="get") or as HTTP post transaction (with method="post").
3. A The <input> tag collects the information from the user.
4. A The textfield control accepts the input in one line.
5. A Text areas are the fields that displays several text lines at a time. The <textarea> tag has both start and the end tag.
6. A Radio button enables the selection of one of the options out of the many.
7. A Checkbox allows for multiple selections of items.
8. A The command button placed on the form performs some action when the user clicks on it.
9. A Submit buttons: When activated, a submit button submits a form. A form may contain more than one submit button.
10. Reset buttons: When activated, a reset button resets all controls to their initial values.
11. Drop down Box contains a list that prompts the user to select one item from the list. It is created by using <select> and <option> tags.

EXERCISE

A. Multiple choice questions

1. A _____ can be inserted in HTML document which can act as a container for all the input elements.
   (a) Text field (b) Teaxt area
   (c) Form (d) Command Button

2. _____ method is used to sent form data as URL variables.
   (a) get (b) set
   (c) post (d) none of them

3. _____ method is used to sent form data as HTTP post.
   (a) get (b) set
   (c) post (d) none of them

4. What is the purpose of a web form
   (a) An outdated feature still used to help the page load faster
   (b) An useful way to send information from the user directly to the search engines
   (c) A way to input data into a website or an application
   (d) To enable the user to navigate the website with ease

5. Which element allows for the creation of groups of options in a select menu?
   (a) <select> (b) <group>
   (c) <option> (d) <optgroup>

6. Which of the option will be selected with the following code snippet?

```
<select>
<option selected value="Fiat">Fiat</option>
<option value="selected">Saab</option>
<option value="opel">selected</option>
<option value="audi">Audi</option>
</select>
```

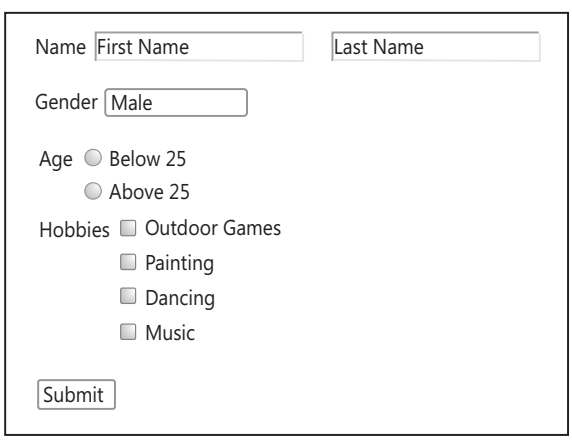
   (a) Fiat (b) Saab
   (c) Selected (d) Audi

B. Answer the following questions:

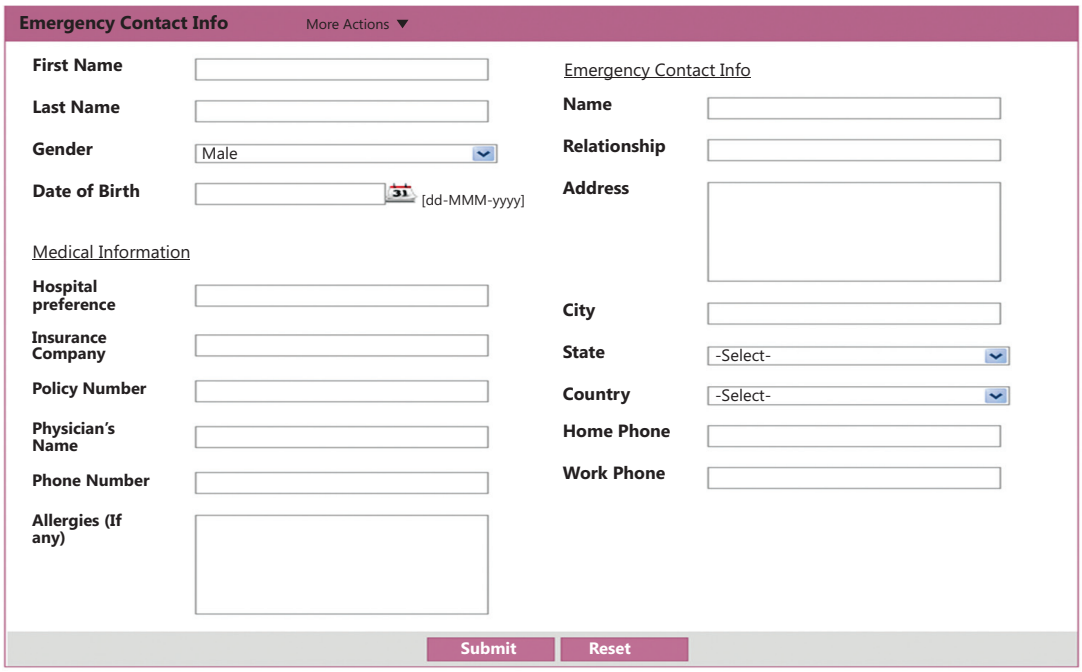
1. Why forms are used in web pages?
2. Explain all the attributes of Form tag.
3. Differentiate between Get & Post methods of Form tag.
4. How text field and text area controls are different from each other?
5. Explain the use of Radio buttons in HTML forms with the help of an suitable example.
6. Mention all the attributes of Check box. Justify how it is different from Radio button.
7. State the purpose of Submit and Reset button.
8. Which attributes are necessary to insert drop down list in a HTML page?
9. Sometimes it is better to use the text area element instead of an input element of type text. Write a short note to explain when and why?

C. Lab Session

1. Write HTML code to generate the following form. Save it as task1.html



2. Write HTML code to generate the following form. Save it as task2.html



3. Generate the output by using the following code:

```
<FORM action="http://prog/user" method="post">
    <P>
    <LABEL for="firstname">First name: </LABEL>
    <INPUT type="text" id="fname"><BR>
    <LABEL for="lastname">Last name: </LABEL>
    <INPUT type="text" id="lname"><BR>
    <LABEL for="email">email: </LABEL>
    <INPUT type="text" id="mail"><BR>
    <INPUT type="radio" name="Gender" value="Male"> Male<BR>
    <INPUT type="radio" name="Gender" value="Female"> Female<BR>
    <INPUT type="submit" value="Send"> <INPUT type="reset">
    </P>
</FORM>
```