

Web Application Development (WAD)

Vth Sem – BBAITM (Unit – 4)

By: Binit Patel

Working with Forms:

A very popular way to make a web site interactive is using HTML based forms by the site. Using HTML forms, a website can interact with a visitor and:

- Obtain registration information prior access to different site offerings
- Permit a visitor to submit information to the website
- Allow a visitor to upload files to the website

and much more.

Forms are generally used for:

- User Registration
- Receiving feedback
- Online catalogs
- Request for information
- Logging in user
- Online shopping
- Surveys
- Conferencing

Introducing HTML Form Tags and Elements

- **Form Tag**

<FORM></FORM> tags indicate the boundaries of a form on the HTML page. <FORM> indicates where the form starts and </FORM> indicates where the form ends. All other form elements must be enclosed between <FORM></FORM> tags.

Syntax:

```
<form name = "formname" action = "server side php file" method = "POST/GET">
```

Name: Uniquely identifies the form.

Action: name of the web server side program which will process the data captured by the form when the data captured is returned to the web server by the Browser.

Method: Get – transmit data using url. Size is limited to 1KB. Post – transmit data using http data stream. There is no limit to the size.

Form Elements

- **Text Box**

When placed on HTML form the textbox permits the capture of one line of text. A user will be able to key in any textual information into a Text Box.

Syntax:

```
<input type= "text" name = "textboxname" size = "" maxlength = "">
```

Input: is an HTML keyword that indicates the start of an HTML object being described.

Type: accepts the type of a form element, in this case a textbox.

Name: accepts any name which is then bound to the textbox like: txt1

- **Text Area**

This is a larger textbox that allows a user to key in multiple lines of text. The entered text can span across multiple lines. The size of a Text Area is specified in Rows and Columns.

Column means number of character & Rows means number of lines.

Syntax:

```
<textarea name = "textareaname" Rows = "" Cols = ""></textarea>
```

Name: is the name of the text area

Rows: How many lines of text will be accepted

Cols: how many characters will fit in each row of text captured

- **Password**

The password field looks identical to a text box. However, when text is typed in it, the letters are hidden. This provides a level of security when keying in password into form.

Syntax:

```
<input type= "password" name = "passwordname" size = "" maxlength  
= "">
```

Input: is an HTML keyword that indicates the start of an HTML object being described.

Type: accepts the type of a form element, in this case a password type textbox.

Name: accepts any name which is then bound to the password textbox like: pwd1

- **Radio Button (Option Buttons)**

Radio button indicate a fixed set of choices. A radio button is an empty circle with textual prompt display beside it. When selected, the circle is partially filled in. only one radio button from set of radio buttons can be selected at any time. Radio buttons are grouped together by simply naming all the radio buttons used on the form with the same name.

Syntax:

```
<input type= "radio" name = "radiobuttonname" value = "caption"  
checked = "true/false">
```

Input: is an HTML keyword that indicates the start of an HTML object being described.

Type: accepts the type of a form element, in this case a radio button.

Name: accepts any name which is then bound to the radio button like: rbtn1

Checked: accepts either true or false. It is used to select a default radio button.

Value: accepts a caption for a radio button. It is text placed adjacent to the radio button.

- **Check Box**

A checkbox also looks similar to a radio button but can be used for selecting multiple choices from a set. The HTML object placed on the page is a square. The square holds a tick mark in it when selected. The major difference between radio button and check box is that with checkboxes multiple choices can be made from a list of choices.

Syntax:

```
<input type= "checkbox" name = "checkboxname" value = "caption" checked = "true/false">
```

Input: is an HTML keyword that indicates the start of an HTML object being described.

Type: accepts the type of a form element, in this case a checkbox.

Name: accepts any name which is then bound to the checkbox like: chk1

Checked: accepts either true or false.

Value: accepts a caption for a checkbox. It is text placed adjacent to the checkbox.

- **Combo Box or Drop Down List Box**

The combo box or drop down list box displays multiple options available. The combo box or drop down list box initially displays one item. When clicked on, it drops down to display a list of item to select from.

Syntax:

```
<select name = "selectboxname" size = "1" >  
<option> opt1 </option>  
<option> opt2 </option>
```

```
<option> opt3 </option>  
</select>
```

Select: indicates the start of the combo box.

Name: accepts any name which is then bound to the combo box like:
cmb1

Size: accepts the size of a box. Here, 1 means one line is shown by default.

Option: accepts a string that will be displayed when the combo box is clicked and drops down.

- **Submit Button**

After placing all the form elements required on an HTML page within the `<form></form>` tags, there should be some way to have the contents of these elements returned to a particular destination. Submit button sends the request and form data back to web server.

Syntax:

```
<input type= "button" name = "submitbuttonname" value = "caption">
```

Input: is an HTML keyword that indicates the start of an HTML object being described.

Type: accepts the type of a form element, in this case a submit button.

Name: accepts any name which is then bound to the submit button like:
sbn1

Value: accepts a string that will display on a submit button.

- **Reset Button**

Some forms also allow erasing all the entries made in form elements prior filling in the form all over again. This functionality is brought to the form by a Reset button. The reset button resets all form elements to their default values.

Syntax:

```
<input type= "reset" name = "resetbuttonname" value = "caption">
```

Input: is an HTML keyword that indicates the start of an HTML object being described.

Type: accepts the type of a form element, in this case a reset button.

Name: accepts any name which is then bound to the reset button like: rst1

Value: accepts a string that will display on a reset button.

Basic SQL Commands

- **Insert:**

Insert into statement is used to insert a new row in a database table.

Syntax:

It is possible to write the insert into statement in two forms.

- 1) The first does not specify the column names where the data will be inserted, only their values

Insert into table_name values (value1, value2, value3, ...)

- 2) The second form specifies both the column names and the values to be inserted.

Insert into table_name (column1, column2, column3, ...) values (value1, value2, value3,...)

For e.g.:

```
insert into student values('1', "XYZ", "TYITM", "SEMCOM")
```

- **Update:**

The update statement is used to update existing records in a database table.

Syntax:

update table_name set column1=value , column2=value, column3=value,... where some_column=some_value

where clause specifies which record or records that should be updated. If you omit the where clause, all records will be updated !

For e.g.:

update student set name="ABC", class="TYITM", college="SEMCOM" where id= '1')

- **Delete:**

The delete statement is used to delete existing records from a database table.

Syntax:

Delete from table_name where some_column=some_value

where clause specifies which record or records to be deleted. If you omit the where clause, all records will be deleted !

For e.g.:

delete from student where id= '1'

- **Select:**

The select statement is used to select and fetch the records from a database table.

Syntax:

Select column_names from table_name

Or

Select * from table_name

Or

Select * from table_name where some_column=some_value

where clause specifies which record or records that should be selected or fetched. If you omit the where clause, all records will be selected or fetched!

For e.g.:

Select * from student

Database functions

PHP MySQL connect to a Database

The free MySQL database is very often used with PHP.

- **Create a connection to a MySQL Database**

Before you can access data in a database, you must create a connection to the database.

In PHP, this is done with the `mysqli_connect()` function.

Syntax:

`mysqli_connect (servername, username, password, databasename)`

Servername: specifies server to connect to. (Default is localhost)

Username: specifies the username to log in with. (Default is root)

Password: specifies the password to log in with. (Default is "" - NULL)

Databasename: specifies the name of the database to which we want to connect.

Example:

```
<?php
```

```
$con = mysqli_connect("localhost", "root", "", "tyitm")
```

```
If (!$con)
```

```
{
```

```
Echo "Error connecting the database !!!";
```

```

}
Else
{
// code to execute on successful connection
}
?>

```

- **Closing a connection**

The connection will be closed automatically when the script ends. To close the connection before, use the `mysqli_close()` function

Example:

```

<?php
$con = mysqli_connect("localhost", "root", "", "tyitm")
If (!$con)
{
Echo "Error connecting the database !!!";
}
Else
{
// code to execute on successful connection
}

```

```

mysqli_close ($con);
?>

```

- **Executing a query**

The `mysqli_query()` function executes a query on a MySQL database.

This function returns the query handle for SELECT, TRUE/FALSE for other queries, or FALSE on failure.

Syntax:

mysqli_query (connection,query)

query: specifies the SQL query to send

connection: specifies MySQL connection

Example:

```
<?php
$con = mysqli_connect("localhost", "root", "", "tyitm")
If (!$con)
{
Echo "Error connecting the database !!!";
}
Else
{
$ins = "insert into student values ('1', 'xyz', 'tyitm', 'semcom')";
mysqli_query ($con, $ins);
}

Mysqli_close ($con);
?>
```

- **Getting number of rows**

The `mysqli_num_rows()` function returns the number of rows in a recordset.

This function returns FALSE on failure.

Syntax:

mysqli_num_rows (data)

data: specifies which data pointer to use. The data pointer is the result from the `mysqli_query()` function.

Example:

```
<?php
$con = mysqli_connect("localhost", "root", "", "tyitm")
If (!$con)
{
Echo "Error connecting the database !!!";
}
Else
```

```
{  
$sel = "select * from student";  
$result=mysqli_query ($con, $ins);  
echo mysqli_num_rows ($result);  
}
```

```
Mysqli_close ($con);  
?>
```

If we have 4 rows in student table then we will have following output.

Output

4

- **Fetching Rows**

The `mysqli_fetch_row()` function returns a row from a recordset.

This function gets a row from the `mysqli_query()` function and returns a row on success, or `FALSE` on failure or when there are no more rows.

Syntax:

mysqli_fetch_row(data)

data: specifies which data pointer to use. The data pointer is the result from the `mysqli_query()` function.

Example:

```
<?php  
$con = mysqli_connect("localhost", "root", "", "tyitm")  
If (!$con)  
{  
Echo "Error connecting the database !!!";  
}  
Else  
{  
$sel = "select * from student";  
$result=mysqli_query ($con, $ins);
```

```

While ($row=mysqli_fetch_row($result))
{
echo $row[0]. "<br>";
echo $row[1]. "<br>";
echo $row[2]. "<br>";
echo $row[3]. "<br>";
}
}

```

```

Mysqli_close ($con);
?>

```

If we have 1 row in student table then we will have following output.

Output

```

1
XYZ
TYITM
SEMCOM

```

- **Selecting a specific database at later stage**

To select a specific database at a later stage, `mysqli_select_db()` function can be used.

Syntax:

```

mysqli_select_db(connection,databasename)

```

connection: specifies MySQL connection

databasename: specifies the name of the database to which we want to connect the application.

Example:

```

<?php
$con = mysqli_connect("localhost", "root", "")
If (!$con)
{
Echo "Error connecting the database !!!";
}

```

```
}  
Else  
{  
mysqli_select_db($con, "tyitm");  
// code to execute on successful connection  
}  
mysqli_close($con);  
?>
```

DISCLAIMER :

This study material is prepared by **Mr. Binit Patel**. The objective of this material is to supplement teaching and discussion in the class room in the subject. Students are required to go for extra reading in the subject through library book.