

## CHAPTER - 6

### Working with HTML Form

classmate

Date \_\_\_\_\_

Page \_\_\_\_\_

(1)

Forms: - HTML form or web form are used to collect data from the website visitors. It allows the user to submit information to the server.

Each field in the "form" has a name (key) that is unique within the form, and the data (value) entered in that field by the user is associated with the field name.

1. a) Form contains attributes, elements.

b) Form's name "standard" is in English.

c) Form elements: "standard" is in English.

1. i. `<input>` name "standard" is in English.

2. `<select>`

3. ii. `<textarea>` is in English.

ii. Form has title as "Form Test Name".

1. `<input>` element: `<input>` element contains many fields. such as, text, password, radio, checkbox, file, submit, reset etc.

• Text field: The text field accepts the text as input. Text field is created using the `<input>` element with a `type` attribute.

Exa: `<input type="text" name="Username">`

, Password field: Password fields are a type of text field in which the text entered is masked using asterisk (\*) or dots (.) for prevention of user identity and small screen users.

`<input type="password" name="password">`

- Radio buttons: Radio buttons are used to select exactly one option from a list of predefined options. It is used to select one or more than one option from a list of options.
- `<input type="radio" name="gender" id="male">  
<input type="radio" name="gender" id="female">`
- check boxes are used to select one or more options from a predefined set of options.

```
<input type="checkbox" name="subject" id="Java">  
<input type="checkbox" name="subject" id="SE">  
<input type="checkbox" name="subject" id=".NET">  
<input type="checkbox" name="subject" id="DM">
```

- File field: File field is used to allow the user to select a local file and send it as an attachment to the web server.

```
<input type="file" name="upload" id="selectfile">
```

- Submit and Reset buttons: The submit button allows the user to send the form data to the web server. The reset button is used to reset the form data and use the default values.

```
<input type="submit" value="Submit">  
<input type="reset" value="Reset">
```

- Select element: Select boxes are used to allow users to select one or more than one option from a pull-down [drop-down] list of options.

Exa : <select>

<option> India </option>

<option> Sri Lanka </option>

<option> America </option>

</select>

3. <Text area> element : Text area is a multiple line text input control which allows user to provide a description or text in multiple lines.

<textarea rows="5" cols="50">

</textarea>

### \* Attributes Used in HTML Forms :

- Action
- Target
- Name
- Method.

Syntax : <form action = "Insert.php" method = "post"  
enctype = "application/x-www-form-urlencoded"  
target = "\_blank" >

The action attribute tells the web browser where to send the form data when the user fills out and submits the form.

This should either be an absolute URL or a relative URL.

The method attribute tells the browser how to send the form data. There are 2 methods Get & Post.

Get is useful for sending small amount of data

Post is useful for sending large amount of data.

The enctype attribute specifies how the web page browser encodes the data before it sends it to the server.

Value	Description
application/x-www-form-urlencoded	Default type. All characters are encoded before sent. Spaces are converted to "+" symbols, and special characters are converted to ASCII HEX values.
multipart/form-data	No characters are encoded.
text/plain	Spaces are converted to "+" symbol but no special characters are encoded.

The target attribute is used to specify whether the submitted result will open in the current window or a new tab.

— self is default value, which results in the form submission in the same window.

— blank value opens the result in a new tab.

<form action="insert.php" method="get" target="self">

Above example shows the form submits to insert.php code using get method, the result will display in the same window

GET Method: Information sent from an HTML form with the GET method is displayed in the browser's address bar. In PHP super global variable `$_GET` is used to collect values from HTML form.

- It is restricted to send upto 2048 characters.
- It is not suitable to submit the sensitive information like password etc.
- It is not used to send binary data, like images or word documents to the server.

Ex:-

Form.html.

```
<html>
<body>
<form action = "registration.php" method = "get">
    Name : <input type = "text" name = "name">
    <input type = "submit">
</form> </body> </html>
```

registration.php

```
<html>
<body>
    welcome <?php echo $_GET["name"] ?>
</body> </html>
```

Output      Name: mohan.      Submit

when user click submit button it redirect the page into registration.php,

→ `[localhost /registration.php?name=mohan]` data displays in URL.  
welcome mohan

The POST method :- Information sent from a form with the POST method is invisible and has no limits on the amount of information to send.

- NOTE :- By default 8 MB is the maximum size, It can be changed by setting the POST\_max\_size.
- The POST method can be used to send ASCII as well as binary data.
  - The super global variable \$-POST is used to collect values from HTML using POST method.
  - The data sent by POST method goes through HTTP headers, so security depends on HTTP protocol.
  - Variables are not visible in the URL, so users can't bookmark your page.

Ex :-

```
<form action = "registration.php" method = "post">  
    Name <input type = "text" name = "name" >  
    <input type = "submit" >  
</form>
```

```
registration.php  
<html>  
<body>
```

```
Welcome <?php echo $_POST ["name"] ;  
?>
```

```
</body> </html>
```

O/P

```
Name : [ mohan ] [ Submit ]
```

NOTE :- The maximum size of POST can be increased by 8MB to whatever by setting POST\_max\_size

After submitting it redirect to registration page using POST method, The information not displays in URL

```
← → C | localhost / registration.php
```

```
welcome mohan
```

`$_REQUEST` variable contains the contents of both `$_GET` & `$_POST` and `$_COOKIE`.

`<body>`

`@ Welcome <?php echo $_REQUEST ["name"] ; ?>`

`</body>`

\* Form Validation : Data validation is the process of ensuring that user input is clean, correct and useful.

Form data validation is required to check whether user filled all required fields or not, entered a valid date, etc.

There are 2 types of validation.

1. Client side validation.

2. Server side validation.

1. Client side validation: It is performed on the client machine web browsers.

- It provides a better user experience by responding quickly at the browser level.
- All user input validates in user browser itself.
- It avoids the network traffic which will help server perform better.
- This validation is done using script languages such as Javascript, VBscript or HTML's attributes.

\* Server side Validation: After ~~B~~ submitted by data, the data has sent to a server and perform validation checks in server machine.

- validation is done using server side scripting languages such as ASP.NET, PHP etc.
- After the validation process on the server side, the feedback is sent back to the client by a new dynamically generated web page.
- It is more secure than client side validations, it can protect against the malicious users.

### Validation Rules

Field	Validation
Name	should required letters and whitespaces
Email	should required @ & .
website	should required a valid URL
Radio	Must be Selectable at least once.
checkbox	must be selectable at least once

### \* File Uploading and Downloading through Form.

A PHP script can be used with a HTML form to allow users to upload files to the Server.

Initially files are uploaded into a temporary directory and then relocated to a target destination by a PHP.

There are 6 important PHP configuration variables influencing POST file uploads.

1. file\_uploads: It returns Boolean, it indicates whether the file upload or not.

2. max\_execution\_time: This variable determines the number of seconds a PHP script requires to upload. If max-execution time exceeds the file terminated by the engine.
3. max\_input\_time: This variables controls the maximum amount of time a script has to receive input data.
4. upload\_max\_filesize: This variable sets the maximum size that file can upload.
5. post\_max\_size: This variable determines the maximum size of data that PHP can accept in a single POST request.
6. upload\_tmp\_dir: This variable determines the temporary directory used for uploaded file.  
\* To upload files the user will need to create a form using these 3 conditions.

1. The `<form>` must contain action attribute. It specifies the PHP script that will process the form.
2. The enctype attribute must specify that the data is multipart/form-data. It determines how the form data is encoded by the browser.
3. The method attribute should be "POST"

## File Download in PHP:

Using PHP we can create a webpage to download files using built-in "readfile()" function.

readfile() function reads a file and writes it to the output buffer, returning the size.

Syntax:

```
int readfile( string $filename [, bool $use_include_path  
= false [, resource $context ] ] )
```

where

\$filename represents the filename to download.

\$use\_include\_path: It is used to search the file.

\$context: It represents the content stream resource.

int : It returns the number of bytes read from the file.

\* PHP include or require functions

The include & require statements are used to include other files into a PHP file.

These are helpful to developing a website that contains the same navigation menu across all the pages. User can create a common header then includes it in every page using include or require statements.

Syntax:

```
<?php  
include 'file-name';  
?>
```

```
<?php  
required 'file-name';  
?>
```

file-name is the name of the file to be included.

Ex :-

```

<?php
header("Content-Type: text/html");
echo "<a href = \"index.php\"> Home </a><br/>
<a href = \"aboutus.php\"> about us </a>
<a href = \"services.php\"> services </a><br/>
<a href = \"contactus.php\"> contactus </a>" ;
?>
```

Index.php

```

<?php
include "header.php";
echo "<a href = \"index.php\"> Home </a><br/>
<a href = \"aboutus.php\"> about us </a>
<a href = \"services.php\"> services </a><br/>
<a href = \"contactus.php\"> contactus </a>" ;
?>
```

Output of simple file write

Index.php at 192.168.1.4 - [2017-07-11 11:20:47]

home aboutus services contactus

If we run this script it contains all errors

### Include

It issues a warning when an error occurs

Execution of the script

continues when an error occurs

### Require

Does not issue a warning

Execution of the script

stops when an error occurs

## \* Working with files in PHP

File is a collection of different kinds of data and also contains information about themselves.

Directories are a special kind of file made for storing other files.

PHP makes working with the file system easily by using inbuilt file functions.

1. opening a file :- fopen() function is used to open a file. It returns a file handle associated with the file.

When a file is opened, a file handle is created. A file handle is a pointer associated with the open file that you can then use to access the file contents.

Syntax : fopen (filename, mode)

filename indicates the name of the file you want to open.

mode indicates in which mode the file should be opened.

Exa : \$handle = fopen ("file1.txt", "r").

It opens the file file1.txt with read mode.

## \* Modes of files.

mode	Description.
r	open the file for reading only
r+	open the file for reading and writing
w	open the file for writing <del>overwriting</del> <sup>only</sup> and clears the contents of file. If the file does not exist, PHP will create that file
w+	open the file for reading & writing, if file does not exist it will create
a	Append. opens the file for writing only. It preserves the file content by writing to the end of the file. If file not exist it will create.
a+	Read / Append, same as a
x	open the file for writing only. If the file already exist it generates error. If file not exist it will create
x+	open the file for reading / writing. Same as x

2. Closing a file: The `fclose()` function is used to close the file.

Syntax: `fclose($handle)`

PHP will close all open files automatically when script terminates, But it is good practice to close file using `fclose()` function.

3. Writing the files: `fwrite()` function is used to write data to a file or append to an existing file.

Syntax: `fwrite($handle, text)`

`$handle` represents the filename to write the text  
`text` represents the data to write filename.

<?php

```
$handle = fopen("file1.txt", "w");
$data = "PHP is a programming language";
fwrite($handle, $data);
fclose($handle);
echo "Data written to the file successfully";
```

Above PHP code creates the file `file1.txt` with write mode & write data "PHP is a programming language" in to that `file1.txt`

4. Reading the contents of file: `fread()` function is used to read the entire file at once. It is conjunction with `filesize()` function to read file.  
`fread($handle, filesize($file))`

5. Renaming files : Using rename() function, we can rename a file or directory.

```
rename ("file1.txt", "file2.txt");
```

6. copying files : copy() function is used to copy file from one location to another file.

```
copy ('sample.txt', 'sample.new.txt');
```

7. Removing files : we can delete files or directories using unlink() function.

8. fgetc() : Reads a single character at a time

9. feof() : checks if the end of the file is reached or not.

10. fgets() ; Reads single line at a time.

11. file() ; Reads an entire file into an array.

12. fseek() ; Repositions the file pointer to a specified point in the file.

13. ftell() ; Returns the current position of the file pointer.

14. file\_exists(); To check whether a file exist or not.

15. file\_get\_contents(); Reads an entire file into a string without needing to open it.

16. file\_put\_contents(); It writes the data into the file.

17. filesize(); Function to determine the size of a file on the hard disk.

## Directory functions.

1. Creating new directories :- `mkdir()` function is used to create a directory.

`mkdir ( directory-name , mode ) ;`

2. Deleting a directory :- `rmdir()` function is used to remove the directory. It works only on empty directories.

`rmdir ( directory-name )`

3. `opendir ( &directory-name )` :- It opens the directory to read or write

4. `closedir ( &dir-name )` :- It closes the opened directory.

5. `chdir()` :- It changes the current directory.