



Chapter Two

Working with Forms

2.1. Working with Forms

- We use one of two methods to submit form information.
- The methods pass the form data differently and have different advantages and disadvantages
- ***GET method:***
- The form data is passed by adding it to the URL that calls the form-processing script.
- For example, the URL may look like this:
`processform.php?lname=Smith&fname=Goliath`
- The advantages of this method are simplicity and speed.
- The disadvantages are that
 - less data can be passed and
 - the information is displayed in the browser, which can be a security problem in some situations.

2.1. Working with Forms...

- ***POST method:***
- The form data is passed as a package in a separate communication with the processing script.
- The advantages of this method are unlimited information passing and security of the data.
- The disadvantages are the additional overhead and slower speed.
- The form data is available in the processing script in the PHP built-in arrays.
- Information from forms that use the POST method is available in the built-in array called `$_POST`.
- If your form uses the GET method, the information is available in the array `$_GET`.
- Both types of form information are also stored in an array called `$_REQUEST`.
- You get information from the array by using the form field name as the array key.

2.1. Working with Forms...

- The syntax is:
`$_POST["form element name"]`
`$_GET["form element name"]`
- For example, suppose that you echo the following field in your form that uses the POST method:
`<input type='text' name='firstName'>;`
- The value entered in this textbox can be accessed by
`$_POST['firstName']`
- This contains the text the user typed into the field.
- If the form uses GET method, the above textbox can be accessed as
`$_GET["firstName"]`

2.1. Working with Forms...

Try out the following `$_GET` method example by putting the first source code in **first.php** file and the second source code in **sec.php** file.

```
<?php
echo "Welcome ". $_GET['name']. "<br />";
echo "You are ". $_GET['age']. " years old.";
?>

<html> <body>
<form action="sec.php" method="GET">
Name: <input type="text" name="name" />
Age: <input type="text" name="age" />
<input type="submit" />
</form>
</body> </html>
```

2.1. Working with Forms...

Try out the following `$_POST` method example by putting the first source code in **first.php** file and the second source code in **sec.php** file.

```
<?php
echo "Welcome ". $_POST['name']. "<br />";
echo "You are ". $_POST['age']. " years old.";
?>

<html> <body>
<form action="sec.php" method="POST">
Name: <input type="text" name="name" />
Age: <input type="text" name="age" />
<input type="submit" />
</form>
</body> </html>
```

2.1. Working with Forms...

Example: Calculator using switch statement.
Make the file name **calc.php**

```
<html>
<body>
<form action="3.php"
method="GET">
<input type="number"
name="x" placeholder="Enter
X"/> <br>
<input type="text" name="op"
placeholder="Enter
Operator"/> <br>
<input type="number"
name="y" placeholder="Enter
Y"/>
<input type="submit" />
```

2.1. Working with Forms...

Make the file name **calc2.php**

```
<?php
$x= $_GET["x"];
$y= $_GET["y"];
$op= $_GET["op"];
echo $x. " $op ".$y. " = ";
switch($op){
    case '+': echo $y+
    $x;
        break;
    case '*': echo
    $y*$x;
        break;
    case '/': echo $x/$y;
        break;
    case '-': echo $x-$y;
        break;
    case '%': echo $x%$y;
        break;
    default;
        echo "Wrong Operator";
}
?>
```

Exercise

- Write a php program that accept two number and determine which is the largest.
- Write a php program that accept a number and determine whether it is odd or even.
- Write a php program that accept a number and determine whether it is positive, negative or zero.
- Write a php program that incrementally display only odd number starting from

2.2. File Inclusion

- You can include the content of a PHP file into another PHP file before the server executes it.
- This is a strong point of PHP which helps in creating functions, headers, footers, or elements that can be reused on multiple pages.
- This will help developers to make it easy to change the layout of complete website with minimal effort.

2.2. File Inclusion...

- If there is any change required then instead of changing thousands of files just change included file.
- There are two PHP functions which can be used to include one PHP file into another PHP file.
 - The include() Function
 - The require() Function

2.2. File Inclusion...

□ The **include()** Function

- The **include()** function takes all the text in a specified file and copies it into the file that uses the include function.
- If there is any problem in loading a file then the **include()** function generates a warning but the script will continue execution.

```
<html>
<body>
<?php include("menu.php"); ?>
<p>This is an example to show how to include
PHP file!</p>
</body>
</html>
```

2.2. File Inclusion...

□ **The require() Function**

- The **require()** function takes all the text in a specified file and copies it into the file that uses the include function.
- If there is any problem in loading a file then the **require()** function generates a fatal error and halt the execution of the script.
- So there is no difference in **require()** and **include()** except they handle error conditions.

2.3. PHP Date and Time

- The PHP `date()` function is used to format a date and/or a time.
- **The PHP `date()` Function**
 - The PHP `date()` function formats a timestamp to a more readable date and time.
 - ***Syntax: `date(format,timestamp)`***
 - A **timestamp** is a sequence of characters,

Parameter	Description
format	Required. Specifies the format of the timestamp
timestamp	Optional. Specifies a timestamp. Default is the current date and time

2.3. PHP Date and Time...

□ **Get a Simple Date**

- The required format parameter of the date() function specifies how to format the date (or time).
- Here are some characters that are commonly used for dates:
 - d - Represents the day of the month (01 to 31)
 - m - Represents a month (01 to 12)
 - Y - Represents a year (in four digits)
 - l (lowercase 'l') - Represents the day of the week
 - Other characters, like "/", ".", or "-" can also be inserted between the characters to add additional formatting.

2.3. PHP Date and Time...

- The example below formats today's date in three different ways:

```
<?php
echo "Today is " . date("Y/m/d") .
"<br>";
echo "Today is " . date("Y.m.d") .
"<br>";
echo "Today is " . date("Y-m-d") .
"<br>";
echo "Today is " . date("|");
?>
```

2.3. PHP Date and Time...

□ Get a Simple Time

- Here are some characters that are commonly used for times:
 - h - 12-hour format of an hour with leading zeros (01 to 12)
 - i - Minutes with leading zeros (00 to 59)
 - s - Seconds with leading zeros (00 to 59)
 - a - Lowercase Ante meridiem and Post meridiem (am or pm)

```
<?php echo "The time is " . date("h:i:sa"); ?>
```

□ Automatic Copyright Year

- Use the **date()** function to automatically update the copyright year on your website:

```
<html> <body>&copy; 2010- <?php echo date("Y"); ?>
```

- </body></html>



End!