Chapter 7

PHP

7.1 Basics of Scripting

A scripting or script language is a programming language that support programs written for special run time environment that automate the execution of tasks that could alternatively be executed one-by-one by a human operator. Most of the times scripting languages are often interpreted rather than compilation. The term scripting language is also used to refer to dynamic high level general purpose language such as Perl, TCL, Python, Php etc. Scripts often used for small programs (up to few thousand lines of code) in such languages, or in domain-specific languages such as the text processing language like: sed and AWK.

Scripting standards are types of:

1. BASH Shell Scripting using in Linux / Unix Operating system to perform some certain task.

2. For web-development we use: Php, Jsp, ASP etc.

7.1.1 PHP History and Introduction :

PHP as it is known today is actually the successor to a product named PHP/FI (Hypertext Interpreter / Form interpreter). Created in 1994 by Rasmus Lerdorf, the very first incarnation of PHP was a simple set of Common Gateway Interface (CGI) binaries written in the C programming language. Originally used for tracking visits to his online resume, he named the suite of scripts "Personal Home Page Tools," more frequently referenced as "PHP Tools." Over time, more functionality was desired, and Rasmus rewrote PHP tools and producing a much larger and richer implementation. This new model was capable of database interaction and more, providing a framework upon which users could develop simple dynamic web applications such as guest books. In June of 1995, Rasmus released the source code for PHP tools to the public, which allowed developers to use it as they saw fit.

PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML. Instead of lots of commands to output HTML (as seen in C or Perl), PHP pages contain HTML with embedded code that does "something" (in this case, output "Hello Students am in PHP script !"). The PHP code is enclosed in special start and end processing instructions **<?php //Your Code here ?>** that allow you to jump into and out of PHP mode.

```
<?php

    echo "Hello Students am in Php Script !";

?>
```

Example-1 an introductory example

7.1.2 Use of interpreter in PHP

Basically PHP language is interpreted and it is scripting type of language. The binary that lets you interpret PHP is compiled, but what you write is interpreted. The Apache web server used to interpret the code. Basically PHP code stored at server side (Under the apache web root directory) and interpreted by apache server into HTML and send back to client (here client term refers to web-browser like: Internet Explorer / Google Chrome / Firefox etc).

7.1.3 There are two main types of website

a) Static Website

b) Dynamic Website

a) Static Website: A static website is the simplest type of website you can design & developed easily. Mostly we code once a time and change very less times with such type of websites. Static websites are mostly written in HTML ,CSS (Cascading Style Sheet) and Java- Script (JS) code only. The only form of interactivity on a static website is hyperlinks (To jump / access on another webpage which given in hyperlinks).

If you want to create your own website to be a small one (3 pages or less), then a static website approach might be the easiest way to go. Static websites are easier to make than dynamic websites, because they require less coding and technical knowledge. To write static website code, we can use any editor to write them, like as Notepad, Dream viewer IDE etc. In this chapter, we will use Dream viewer editor for coding purpose. Static website also known as "client side scripting" because the web page which written in HTML, CSS and JS and they can be run / execute by just double click, so it open / run directly in the browser. Finally we can say that web browser only understand client side scripting like: HTML , CSS and JS.

b) Dynamic website: Mostly such types of websites are also kwon as web-applications, because the website you are which have with the database and written in server side script (like PHP), so only interpreter can execute / run such code and on the web browser only HTML page shown in the form of output. For example we code in HTML form tags, which gets information from the user and this information can be sent into table of database. (In the earlier chapters we have learnt about database and its tables). The table of database, store information of user in the form of rows and columns. Such kind of information can be use for the future aspects.

In this chapter we will learn how php connect with the mysql database, and how to code in HTML form tags (<form> </form>) and such form receive information. (The entered information in HTML form, may belongs to school / college / company, from user and this information can store in MySQL database).

We can divide user interface of website into three parts:

a) Type of webpage

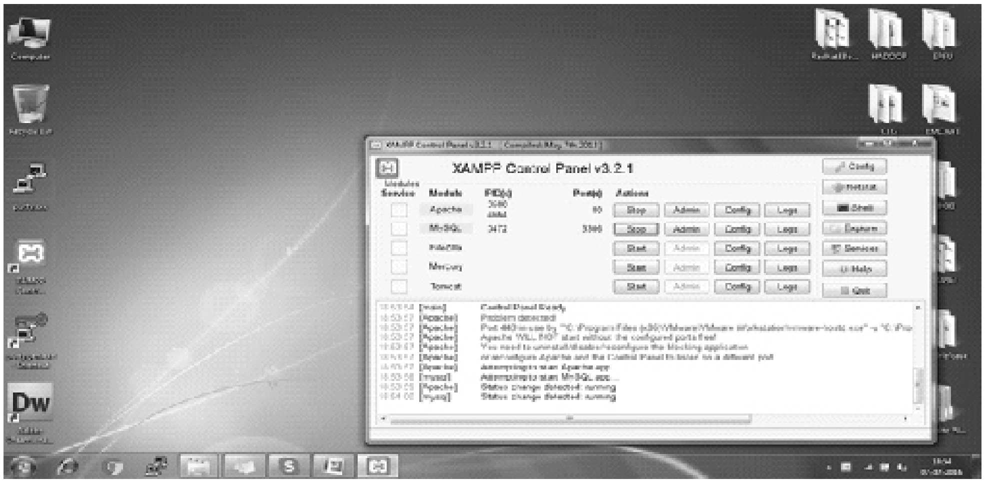
b) style of webpage

c) Behaviour of webpage

To create any website layout and its structure most of the times we use HTML and how the website look likes and how will be the interface look of website mostly we use: CSS. To validate the data entered by the user in HTML form, we use JS (Java Script).

7.1.4 Use of PHP in website development:

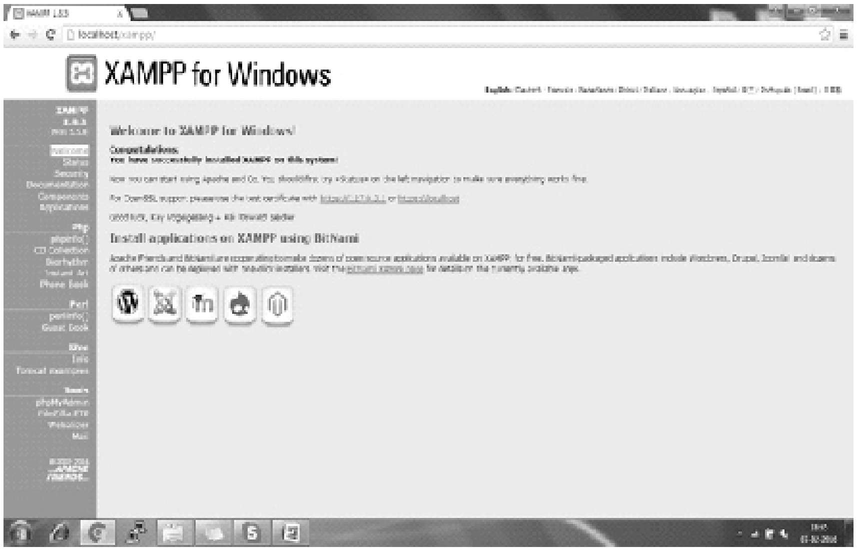
There is assumption that PHP is mainly focused on server-side scripting (dynamic web pages), It's not completely true. We can use PHP scripting in GTK(Genome Tool Kit) development to create desktop application for any kind of operating system. GTK based desktop applications can be execute independently without any dependency. We can run / execute php on command line interface (CLI) in Microsoft based operating system as well as open source OS. PHP works under various kind of web servers like: wamp, xampp, apache etc web-servers are available freely over the internet. In this chapter we will use XAMPP bundled software as server which consist of: apache webserver, PHP and mysql. The following figure shows you how to turn on / off the services of apache and mysql server.



```
[main]      Control Panel Ready
[Apache]    Problem detected!
[Apache]    Port 443 in use by "C:\Program Files (x86)\VMware\VMware Workstation\vmware-hostd.exe" ...
[Apache]    Apache WILL NOT start without the configured ports free!
[Apache]    You need to uninstall/disable/reconfigure the blocking application
[Apache]    or reconfigure Apache and the Control Panel to listen on a different port
[Apache]    Attempting to start Apache app...
[mysql]     Attempting to start MySQL app...
[Apache]    Status change detected: running
[mysql]     Status change detected: running
```

Figure 7.1 Control panel of XAMPP Software

After running the services of apache and mysql, we can test the web-server (apache) and data-base i.e. mysql by typing URL address in browser **http://localhost** , you will see there a welcome screen of XAMPP software. The apache web server works on TCP (Transmission control protocol) with port number is 80 and MySQL server also works under TCP service and having port number 3306. When we test the web services of XAMPP in browser, **http://** is representing as a protocol.



Welcome to XAMPP for Windows!

Congratulations:
You have successfully installed XAMPP on this system!

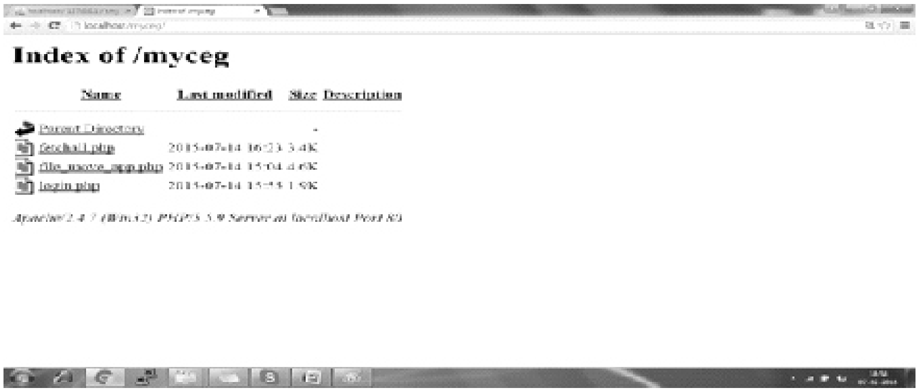
Now you can start using Apache and Co. You should first try »Status« on the left navigation to make sure everything works fine.

Install applications on XAMPP using BitNami

Figure 7.2 Welcome screen of xampp software in web browser

7.2 Web root directory in apache web-server:

The web root directory named as 'htdocs' located under the file system of XAMPP software. Under the **htdocs** directory all the websites are stored with their related own files and directory in windows the file system, and path is **c:\xampp\htdocs**. To create a new website project in xampp, we have to create a sub directory under the htdocs. For example we choose named as **myceg/** directory name. This directory behaves like website location which have path **c:\xampp\htdocs\myceg,** (example implemented in Microsoft Windows 7 OS). To create web-pages in PHP we may use editor like dream-viewer / Notepad / Eclipse. In following example we listed the web root directory by tying URL in the web browser as: **http://localhgost/myceg**



localhost/myceg/

Index of /myceg

Name	Last modified	Size	Description
Parent Directory		-	
fetchall.php	2015-07-14 16:23	3.4K	
file_move_app.php	2015-07-14 15:04	4.6K	
login.php	2015-07-14 15:53	1.9K	

Apache/2.4.7 (Win32) PHP/5.5.9 Server at localhost Port 80

Figure 7.3 Web root directory location in XAMPP software

7.2.1 Writing first PHP scripts and its tags

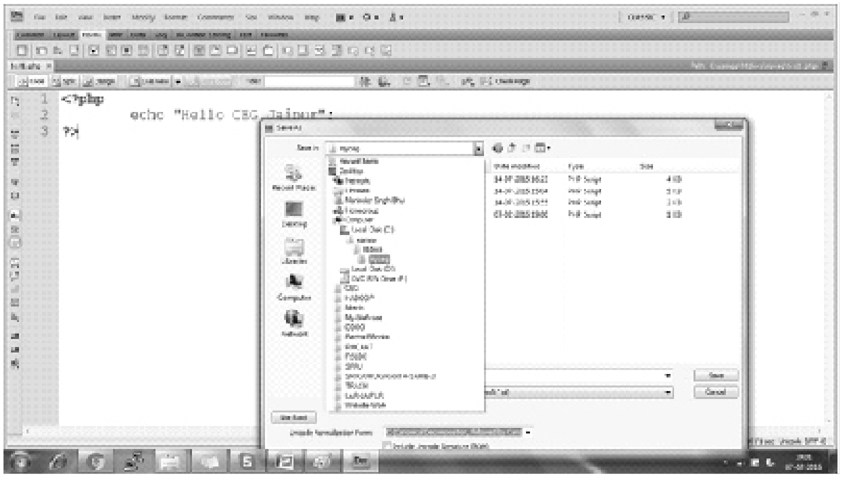
In Php we code all the statements in between following tags:

<?php

//—————Your code here———————

?>

Only above php tags are read by the interpreter of php, so whenever we create php web pages always write code in same tags. The example shown in the following figure 7.4 shows how to code and where to save the php document.



```
<?php
        echo "Hello CBG Jaipur";
?>
```

Figure 7.4 Php script saving

Remember that the extension of php page must be ***.php** , here * sign means name of web page and **.php** is extension. For example we have write " Hello World " program to execute this php script, we use here **echo** keyword for print purpose each and every statements are terminated by semicolon (;) only. The code is as follows:

```
<?php

        echo  "Hello World";

?>
```

7.2.2 Variable & Identifiers in Php script

In PHP, when we define a variable in PHP it starts with the **$** sign, followed by the name of the variable:

Identifiers rule:

- An identifier can consist of one or more characters and must begin with a letter or an underscore. It means, identifiers can consist of only letters, numbers, underscore characters.
- Identifiers are case sensitive. Therefore, a variable named $recipe is different from a variable named $Recipe, $rEciPe, or $recipE.
- Identifiers can be of any length. This is advantageous because it enables a programmer to accurately describe the identifier's purpose by using the identifier name.

- An identifier name can not be identical to any of PHP's predefined keywords. You can find a complete list of these keywords in the PHP manual appendix.

```
<?php

        $a  = "Hello World";

        $b  = 123;

        $c =   10.5;

?>
```

While using PHP script, when we want to store some values in the code, which vary time to time with its value, the identifier denotes the variable name. Such identifier kwon as variable identifier or variable. In above script $a , $b and $c are three variables and $ symbol represent them as an variables.

A variable can have a short name (like a or b) or a more descriptive name (age, carname, total_volume).

Rules to declare PHP variables:

- A variable can starts with the $ sign, followed by the name of the variable.
- A variable name must start with a letter or the underscore character.
- A variable name cannot start with a number.
- A variable name can only contain alpha-numeric characters and underscores like A-z, 0-9, and _.
- Variable names are case-sensitive ($age and $AGE are two different variables).

Output of Variables:

```
<?php

        $a  = "Hello World";

        $b  = 123;

        $c =   10.5;

        echo   $a . $b . $c;

?>
```

The output of above script is:

Hello World 123 10.5

NOTE: We use (.) Operator in above script between the variable names to concatenation the variable output of variables.

7.2.3 Output statements in Php

Without the output statements we can't proceed because each and every program that have three essential parts. In every programming we can define the life cycle of program in three phases:

a) INPUT

b) PROCESS

c) OUTPUT

For example:- Like as in C, C++ programming the inbuilt functions are used printf( ) and cout<< same in php the following two functions / keywords are used to get output.

i) **echo statement:** It is an type of statement which is mostly used for output in php script. It can also use as function , and even it does not have limit of arguments also accept mix data types of arguments. So that we can pass **n** number of arguments in the parenthesis ( ) of function name.

Important: The echo statement can be used with or without parentheses: echo or echo().

Syntax: void echo(String $arg, , , ,)

In above syntax there is no limit in passing arguments in the function / statement of echo. void is return type of function in syntax, it means no return data types are needed.

ii) **print() statement:** The print statement can be used with or without parentheses: print or print().

syntax: int print(string $argument)

In above syntax **int (integer)** is a return data type of print() function and only single arguments are used.

7.3 Commenting styles in PHP

Comment styles are very important part in any of the programming or scripting, where programmer can hide some line of code that does not require to be executed. By commenting one or more lines of code, the commented part never seen in output window. Comment can be use for describe the programme type, date of written code, author name and those line(s) of code which seems to be bad logic as output. It means the commented area cannot be executed by compiler / interpreter.

PHP uses following two commenting styles:

1) C / C++ Comment Style: Single or multiple lines of code can be hide like C and C++ programming.

(a) Single line comment: We can put double forward slashes (//) in front of line or code pattern. So that we can hide / comment the particular line.

Ex:

```
<?php

          $a = 1234;

          //     $b = “hello world”;

          $c  = 55.5;

          echo $a;

          echo $b;

          echo $c;

     ?>
```

Output is: 1234 55.5

Note: $b variable has been commented so it not showing in output.

(b) Multiple line comment: We can use /* —— */ pattern to hide / comment the multiple lines of code.

```
<?php

     for($i=0; $i<=5;$i++)

     {
```

```
                        echo $i ;
                /*      echo $i++;
                        echo $i+2;
                echo "Hi";   */
        }
        ?>
```

Output is: 0 1 2 3 4 5

Note: Remaining three lines of code has been commented so they are not showing in output.

2) Unix style comment: The "one-line" comment styles only comment to the end of the line or the current block of PHP code, whichever comes first.

```
        <?php
          echo 'This is a test'; // This is a one-line c++ style comment
              /* This is a multi line comment
                 yet another line of comment */
                echo 'This is yet another test';
           echo 'One Final Test';   # This is a one-line Unix shell-style
        comment
        ?>
```

7.4 Data types in Php:

Php have mainly three types of data type:

1) Scalar data type

2) Compound data type

3) Special data type

1) Scalar data type: Such types of data types are also known as scalar / single data types. It means such data types can store single value. The scalar variables are those which containing only integer, float, string or boolean data types.

a) Integer: An integer data type is a non-decimal number between -2,147,483,648 and 2,147,483,647. Rules for integers:

- An integer must have at least one digit
- An integer must not have a decimal point
- An integer can be either positive or negative
- Integers can be specified in three formats: decimal (10-based), hexadecimal (16-based - prefixed with 0x) or octal (8-based - prefixed with 0)

In the following example of code $a is an integer type of variable. The PHP function i.e. inbuilt function var_dump() function returns the data type and value:

```
<?php
     $a = 44785;
   var_dump($a);   # It will show the data type of variable.
?>
```

b) float / double: A float (floating point number) / double data types are numbers with a decimal point or a number in exponential form.

In the next example $b variable is a float / double type of variable . The PHP **var_dump()** inbuilt function returns the data type and value:

For Example:

```
<?php
     $b= 447.56;
   var_dump($b);   # It will show the data type of variable.
?>
```

c) string data type: A string is a sequence of characters, like “Hello Rajasthan “.

A string can be any text inside quotes. You can use single or double quotes:

```
        <?php
 $a = “Hello Rajasthan “;
$b = ‘Hello Jaipur ‘;
echo $a;
        echo “<br>”;    # It will insert a new line after print the value $a variable
echo $b;
?>
```

d) Boolean data type: A Boolean represents two possible states: true or false.

```
$x = true;
$y = false;
```

Booleans are often used in conditional testing.

2) Compound Data Type: Compound data can contain multiple values. PHP has 2 compound data types: array and object.

a) Array data type: An array stores multiple values in one single variable.

In the following example to print full name of student, variable name is $fullname type of an array. The PHP var_dump() function returns the data type and value:

```
        <?php
      $fullname = array(“Ekam”,”Jeet”,”Singh”);
      var_dump($fullname);   # Output is:   EkamJeet Singh
?>
```

b) Object data type: An object is a data type which stores data and information on how to process that data. In PHP, an object must be declared explicitly.

A class is a structure that can contain properties and methods. For example:

```
                <?php
                class Bike {
```

```
                function Bike() {
                   $this->model = "Welcome to Indian brand of bike";
                 }
             }
   // create an object
            $bajaj = new Bike();
// show object properties
            echo $bajaj->model;
            ?>
```

Output is : Welcome to Indian brand of bike

In above example $bajaj declared as object using **new** keyword.

3) Special Data Type: There are two special data types in PHP that have special meanings:

Resource: It holds a reference (link) to an external resource, e.g. file or database (mysql connectivity) connection.

Null: If a variable is null, it means the variable does not contain any value.

7.5 Constant declaration in PHP :

A constant is an identifier name for a simple value in any programming or scripting. As the name suggests, that value can't change during the execution of the script (except for magic constants, which are not actually constants). A constant is case-sensitive by default. By convention, constant identifiers are always defining in uppercase. We use define() function to declare constant in php.

The name of a constant follows the same rules as any label in PHP.

```
<?php
                         // Following are valid constant names
                                define("RAJ1",    "education");
```

```
                  define(“RAJ2”,    “is”);

                  define(“BOARD3”, “essential for everyone”);

              // Following are Invalid constant names

                  define(“2FOO”,    “something”);

              // Following is valid, but should be avoided:

                  define(“__RAJ4__”, “Hello World”);

    ?>
```

7.6 function in php:

PHP functions are similar to other programming languages like C , C++ and Java. A function is a piece of code which takes one more input in the form of parameter and does some processing and returns a value. Where in C programming explain with many functions like **fopen()** and **fread()** in file handling. Same in php built-in functions gives you option to create your own functions as well.

* **Creating a PHP Function**

In fact you hardly need to create your own PHP function because there are already more than 1000 of built-in library functions created for different area and you just need to call them according to your requirement.

It is very easy to create our own PHP function. Let’s take an example that we want to create a PHP function which will simply gives output as simple message “Welcome in Rajasthan” on your browser wall when you will call it. In Following example creates a function called **printMassage()** and then calls it just after creating it. Note that while creating a function its name should start with keyword function and all the PHP code should be put inside { and } braces as shown in the following example below:

```
    <?php

        /* Defining a PHP Function */

            function printMassage() {

              echo “Welcome in Rajasthan”;

              }

        /* Calling a PHP Function */
```

```
                    printMassage ();
    ?>
```

OUTPUT: Welcome in Rajasthan

Example of passing arguments in function:

```
<?php
function multiplex($x,$y)
{
            $z = ( $x * $y );
            return($z);
            }
            multiplex(5,4);    # calling the function with arguments
?>
```

OUTPUT is: 20

7.7 Control statements:

We can divide different types of control statements in four parts-

1) Sequential statements

2) Conditional statements

3) Loop Statements

4) Jumping statements

1) Sequential statements: Sequential statements are executed one after the another as they appear in the design from the top of the process body to the bottom sequentially.

2) Conditional statements: PHP scripting also allows you to write code that perform different actions based on the results of a logical or comparative test conditions at run time. This means, you can create test conditions in the form of expressions that evaluates to either true or false and based on these results you can perform certain actions.

There are several statements in PHP that you can use to make decisions:

*	The if statement

*	The if...else statement

*	The if...else, if....else statement

*	The switch...case statement

♦	**if statement:** this statement have very importance in programming because by using execution of statement we can control the flow of our program. The if construct is one of the most important features of many languages, PHP included. It allows for conditional execution of code fragments. PHP features an if structure that is similar to that of C.

Syntax:

```
if (expr) {

  code to be executed if condition is true;

}
```

For Example:

```
<?php

     if ($a > $b)

      echo “a is bigger than b”;

?>
```

♦	**if-else statement:** The if....else statement executes some code if a condition is true and another code if that condition is false.

Syntax:

```
if (condition) {

    code to be executed if condition is true;

} else {
 code to be executed if condition is false;
}
```

For example:

```
                         <?php
                                         $a = 40;
                                         $b = 50;
                                         if($a > $b)
                                                 {
                                                         echo “$a is greater than $b”;
                                                 }else{
                                                         echo “$b is greater than $a”;
                                         }
                         ?>
```

- **Nested if—else statement:** Nested if statements means an if block inside another if block. Shortly a control structure inside another control structure.

Syntax:

```
if (expression 1 )
{
if (expression 2 )
{
// statements 1
}
else
{
// Statements 2
}
}
else
{
if ( expression 2)
{
// Statements 3
}
else
{
// Statements 4
}
}
```

♦ **if—elseif—else ladder statement**: When PHP evaluates your If...elseif...else statement it will first see if the If statement is true. If that tests comes out false it will then check the first elseif statement. If that is false it will either check the next elseif statement, or if there are no more elseif statements, it will evaluate the else segment, if one exists.

Syntax:

```
        if(expression  or condition 1)
        {
                statement 1;
}elseif(expression or condition 2)
    {
        statement 2     ;
    }else{
```

```
                    statement 3;
          }
```

For example:

```
<?php
$employee = " Maninder singh";
if($employee == "Maninder singh"){
          echo "Hello Sir";
} elseif($employee == "kirti"){
          echo "Good Morning mam";
}else {
          echo "Morning ekam";
}
                                        ?>
```

- **switch—case statement**: Use the switch statement to select one of many blocks of code to be executed.

Syntax:

```
switch (n) {
                    case label1:
                       code to be executed if n=label1;
                         break;
          case label2:
                    code to be executed if n=label2;
                     break;
          case label3:
                    code to be executed if n=label3;
```

```
        break;
    ...
default:
     code to be executed if n is different from all labels;
 }
```

For example:

```
<?php
        $percentage = 55.85;
        $a = (int) $per;
        Switch($a)
        {
                Case ($a  =<85   &&   $a  >= 60):
                        Echo “ist division”;
                        Break;
                Case ($a  =<45   &&   $a  >= 59):
                        Echo “IInd division”;
                        Break;
                Case ($a  =<36   &&   $a  >= 44):
                        Echo “IIIrd division”;
                        Break;
                Default:
                        Echo “fail”;
                Break;
        }       ?>
```

3) **Iterative statements:** Often when we write code, and we need the same block of code to run over and over again in a row. Instead of this by adding several almost equal code-lines in a script, we can use iteration / loops to perform such task like this. It means Loops in PHP are used to execute the same block of code a specified number of times.

In PHP, we use the following looping statements:

* for - loops through a block of code a specified number of times.

* while - loops through a block of code if and as long as a specified condition is true.

* do -while - loops through a block of code once, and then repeats the loop as long as a special condition is true.

* foreach - loops through a block of code for each element in an array.

To use any loop / iterations in the programming having three essential parts:

i) **Initialization**: Initialize the loop counter value, for this most of the times we use = assignment operator to start the loop.

ii) **Condition:** Evaluated for each loop iteration. If it evaluates to TRUE, the loop continues. If it evaluates to FALSE, the loop ends.

iii) **size increment / decrement:** Increases or decrease the loop counter value

♦ **for loop:** In programming most of the times when we talk about loops, the name of for loop comes first of all. In this loop, we use three segments and separate by ; semi colon each other. We use such loop when we know how many time we need to repeat the code. It means when certainty occurs for loop come forward.

Syntax:

```
for(initialization ;  condition  ; step size part)

{

                    statements

}
```

For example:

```
<?php
```

```
                        for($i=1 ; $i<=20 ; $i++)
                            {
                                echo $i . "<br>";
                            }
                ?>
```

Output is :

```
1
2
3
```

♦ **while loop:** while loops are the simplest type of loop in PHP. They behave just like their C counterparts. The basic form of a while statement is:

Syntax:

```
while (expr)
        Statement
```

The meaning of a while statement is simple. It tells PHP to execute the nested statement(s) repeatedly, as long as the while expression evaluates to TRUE. The value of the expression is checked each time at the beginning of the loop, so even if this value changes during the execution of the nested statement(s), execution will not stop until the end of the iteration (each time PHP runs the statements in the loop is one iteration). Sometimes, if the while expression evaluates to FALSE from the very beginning, the nested statement(s) won't even be run once. Overall while loop when, condition is uncertain, means we cannot guess when will be the condition true or available, it continue till then.

For Example:

```
<?php
    /* example 1 */
    $i = 1;
    while ($i <= 10) {
      echo $i++;  /* the printed value would be
```

```
                    $i before the increment
                     (post-increment) */
        }
        /* example 2 */
        $i = 1;
        while ($i <= 10):
          echo $i;
          $i++;
        endwhile;
?>
```

- **do-while loop:** The PHP do-while loop statement allows you to execute a block of code repeatedly based on a condition. Unlike the while loop statement whose the condition is checked at the beginning of each iteration, the condition in the PHP do-while statement is checked at the end of each iteration. It means the loop will execute at least once even the condition is evaluated to false.

Syntax:

```
                    do {
 // code block to be executed
  } while (expression);
```

For example:

```
                                  <?php
$i = 0;
do {
 echo $i;
} while ($i > 0);
?>
```

♦ **foreach loop:** The foreach construct provides an easy way to iterate over arrays. foreach works only on arrays and objects, and will issue an error when you try to use it on a variable with a different data type or an uninitialized variable. There are two syntaxes:

```
foreach (array_expression as $value)

  statement

foreach (array_expression as $key => $value)

   statement
```

Note: The first form loops over the array is given by **array_expression**. On each iteration, the value of the current element is assigned to $value and the internal array pointer is advanced by one.

The second form will additionally assign the current element's key to the $key variable on each iteration.

4) Jumping statements: Sometimes we want to let the loops being without any condition, and allow the statements to go inside the brackets to decide when to exit the loop. There are two special statements that can be used inside the loops: **Break** and **Continue**.

Now we will try to understand a brief knowledge of break and continue keywords:

Break - ends execution of the current structure. Break accepts an optional numeric argument which tells it how much execution of nested structures to be interrupted.

Continue - is used to stop processing the current block of code in the loop and goes to the next iteration.

7.8 Array:

By the performance point of view array is an strongest data types in programming, because array is a special variable, which can hold more than one value at a time means An array stores multiple values in one single variable.

Syntax to create array in php:

```
array([$mixed data types ,$......]);
```

7.8.1 Types of Array:

In PHP, there are three types of array:

i. Indexed arrays - Arrays with a numeric index

ii. Associative arrays - Arrays with named keys

iii. Multidimensional arrays - Arrays containing one or more arrays

i. Indexed arrays:

There are two ways to create indexed arrays:

The index can be assigned automatically (index always starts at 0), like this:

```
<?php
        $cars = array(“Rajasthan”, “higher”, “education”);
        print_r($cars)
?>
```

Or

```
<?php
        $cars[0] = “Rajasthan”;
        $cars[1] = “higher”;
        $cars[2] = “education”;
        print_r($cars);
?>
```

ii. Associative arrays : Associative arrays are arrays that use named keys that you assign to them. There are two ways to create an associative array:

```
<?php
        $age = array(“maninder”=>”34", “guurpreet”=>”38", “jagpreet”=>”40");
        print_r($age);
?>
```

Or

```
  <?php
                $age[maninder] = “34”;
                $age[‘gurupreet’] = “38”;

                $age[‘jagpreet’] = “40”;

                print_r($age)

    ?>
```

iii. Multidimensional array: A multidimensional array is an array containing one or more arrays. PHP understands multidimensional arrays that are two, three, four, five, or more levels deep. However, arrays more than three levels deep are hard to manage for most people.

The dimension of an array indicates the number of indices you need to select an element.

* For a two-dimensional array you need two indices to select an element

* For a three-dimensional array you need three indices to select an element

```
<?php

                $cars = array(

                 array(“Volvo”,22,18),

                 array(“BMW”,15,13),

                 array(“Saab”,5,2),

                 array(“Land Rover”,17,15)  );

                echo $cars[0][0].”: In stock: “.$cars[0][1].”, sold:
                “.$cars[0][2].”.<br>”;

                echo $cars[1][0].”: In stock: “.$cars[1][1].”, sold:
                “.$cars[1][2].”.<br>”;

                echo $cars[2][0].”: In stock: “.$cars[2][1].”, sold:
                “.$cars[2][2].”.<br>”;

                echo $cars[3][0].”: In stock: “.$cars[3][1].”, sold:
                “.$cars[3][2].”.<br>”;

?>
```

7.8.2 Super global array in php:

Several predefined variables in PHP are known as “superglobals”, which means that they are always accessible, regardless of scope - and we can access them from any function, class or file without having to do anything special in the source code of php script.

The PHP superglobal variables are:

* $GLOBALS[]

* $_SERVER[]

* $_REQUEST[]

* $_POST[]

* $_GET[]

* $_FILES[]

* $_ENV[]

* $_COOKIE[]

* $_SESSION[]

This chapter we will emphasis some of the super global array variables, like : $GET[], $POST[] only. All above super global variables are associative types of array. As we mentioned we will discuss about only $_GET and $_POST, so let us see:

$_GET[] : PHP $_GET can also be used to collect form data after submitting an HTML form with method=”get”.

$_GET can also collect data sent in the URL.

Note: Assume we have an HTML page that contains a hyperlink with parameters:

```
<html>

<body>

<a href=”ceg_register.php?coursename=PHP&website=www.ceg.rajasthan.gov.in”>

        Click $GET

</a>
```

```
</body>

</html>
```

In above script the saved webpage name is : test1.php

Note: When someone click on Click hyperlink shown in above code of HTML, it will automatically redirect to ceg_register.php and print values like :

```
<?php

echo “Course Name You selected: “ . $_GET[‘coursename’] . “ at URL “ .
$_GET[‘website’];

?>
```

In above script the saved webpage name is: ceg_register.php

$_POST: In PHP, $_POST is widely used to collect form data after submitting an HTML form with method=”post”. $_POST is also widely used to pass variables. The data which passes through URL, are always hide. Let us take an example of $_POST, for this we create a webpage named as : test2.php and pass some argument in URL of browser. In following example as we use action as a same page for testing purpose.

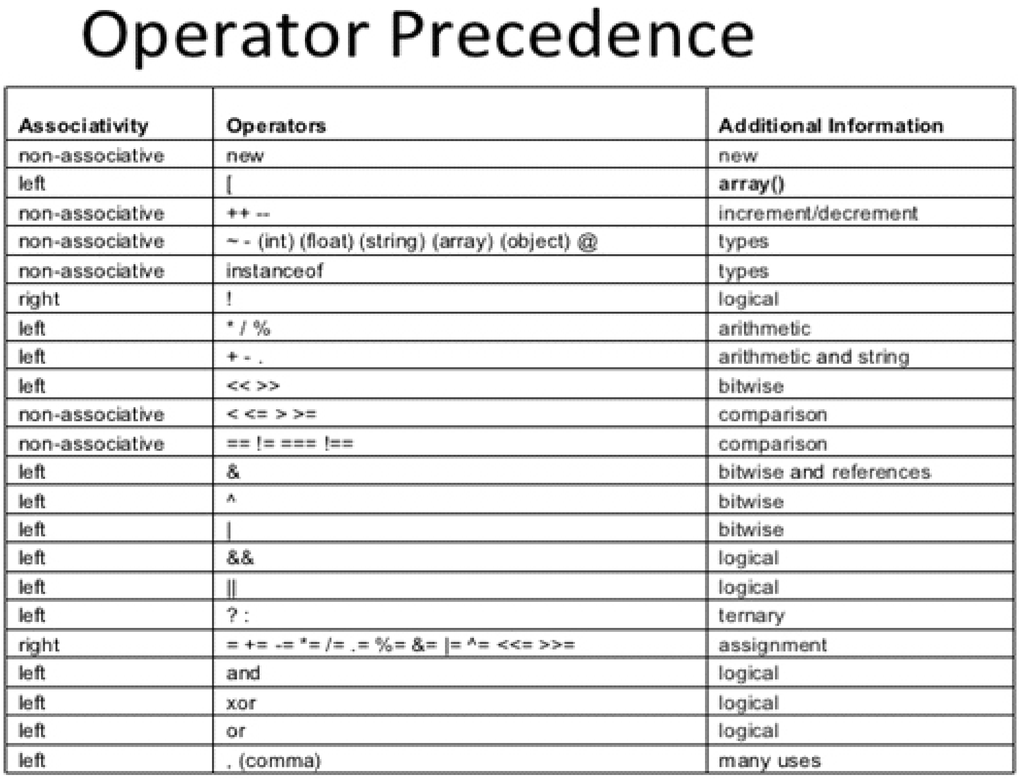
```
<html>

<body>

<form method=”post” action=”test2.php”>

 Student Name: <input type=”text” name=”sname”>

 <input type=”submit” value=”Save” name=”b1">

</form>

          <?php

               if(isset($_POST[‘b1’]))

               {

                    echo “Welcome “.$_POST[‘sname’] ;

          }
```

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

Note: When user execute this page in xampp server, it shows you a HTML webform, and when user enter the name and click on submit button which seen as Save name, for example you enter string as **ekam**, then you will get output: **Welcome ekam**.

7.9 Operators:

In any programming we try to obtain different types of results for any arithmetic / mathematical problem, so then we perform logical / mathematical operations in computer. To resolve all types of calculations we use symbolic notations with operands. These symbols are used to control the programming to solve the complex computations, it means these symbolic notation refer as Operators and both side of values in the calculation, are known as Operands. The associativity of operator in PHP are Right to Left, and when we solve a problem of expression in programming of PHP, the precedence are showing below (in figure) that which one of the operation will occur first and later respectively. To explain precedence of the operator, we need to understand following table:

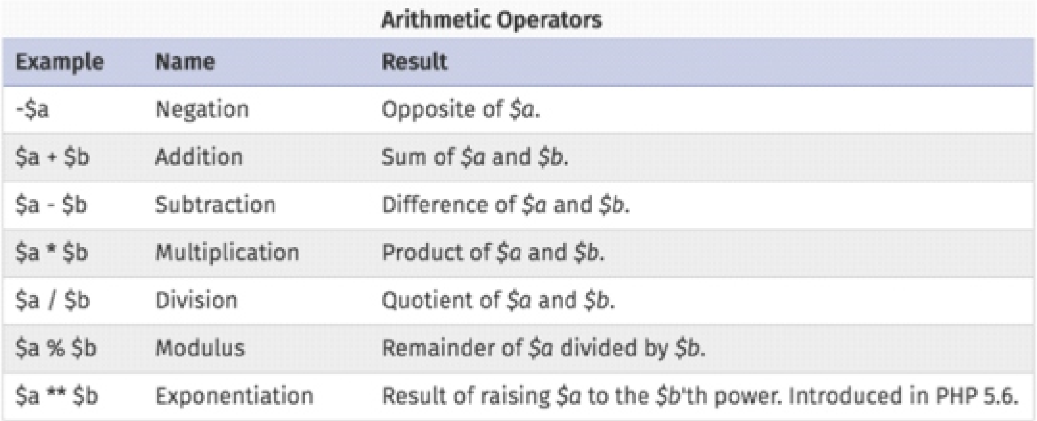


Operator Precedence

Associativity	Operators	Additional Information
non-associative	new	new
left	[	**array()**
non-associative	++ --	increment/decrement
non-associative	~ - (int) (float) (string) (array) (object) @	types
non-associative	instanceof	types
right	!	logical
left	* / %	arithmetic
left	+ - .	arithmetic and string
left	<< >>	bitwise
non-associative	< <= > >=	comparison
non-associative	== != === !==	comparison
left	&	bitwise and references
left	^	bitwise
left	|	bitwise
left	&&	logical
left	||	logical
left	? :	ternary
right	= += -= *= /= .= %= &= |= ^= <<= >>=	assignment
left	and	logical
left	xor	logical
left	or	logical
left	, (comma)	many uses

7.9.1 Arithmetic Operator: To solve different types of arithmetic expressions. We use five types of operator with their priority sequence:

* , % , + , - , /



Arithmetic Operators

Example	Name	Result
-$a	Negation	Opposite of *$a*.
$a + $b	Addition	Sum of *$a* and *$b*.
$a - $b	Subtraction	Difference of *$a* and *$b*.
$a * $b	Multiplication	Product of *$a* and *$b*.
$a / $b	Division	Quotient of *$a* and *$b*.
$a % $b	Modulus	Remainder of *$a* divided by *$b*.
$a ** $b	Exponentiation	Result of raising *$a* to the *$b*'th power. Introduced in PHP 5.6.

7.9.2 String Operator: In PHP mainly, The string may be define with two PHP operators. The first one is period / dot (.) operator, and second one is used to demonstrate decimal types of values. For example 15.6 is floating point number and we want to print this float type number with spaces in php script so that we will try with the following script:

```
<?php
        $a = 15.6;
        echo "Hello world  ". $a.  " <br> ".  "Thanks a lot";
?>
```

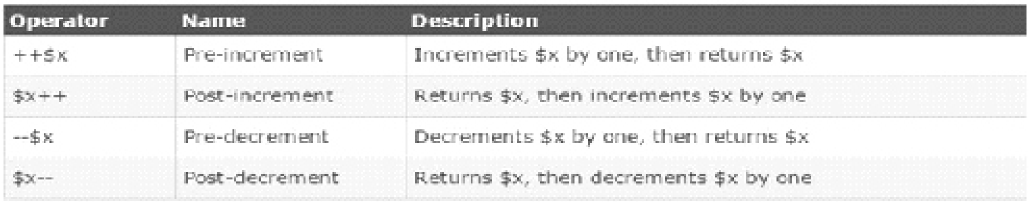
Output is : **Hello world**

15.6

Thanks a lot

Note: In above example we use dot operator in floating point number and also use it for concatenation purpose to concat different types of operands like string , break line code and then string again.

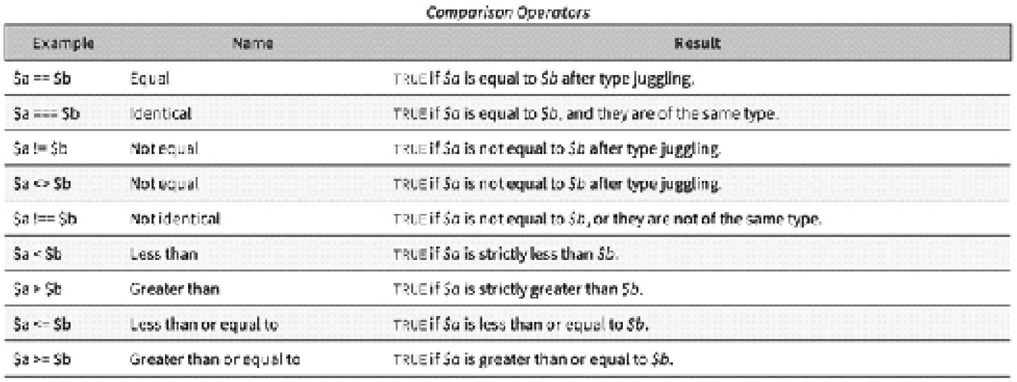
7.9.3 Increment and Decrement Operator: These are same as other operators, we denote increment operator as ++ and — for decrement operator. The importance of operator is they only use with single Operand. The operator listed in the following table format showing along with their description.



Operator	Name	Description
++$x	Pre-increment	Increments $x by one, then returns $x
$x++	Post-increment	Returns $x, then increments $x by one
--$x	Pre-decrement	Decrements $x by one, then returns $x
$x--	Post-decrement	Returns $x, then decrements $x by one

```
<?php
        echo “<h3>Postincrement</h3>”;
          $a = 5;
         echo “Should be 5: “ . $a++ . “<br />\n”;
         echo “Should be 6: “ . $a . “<br />\n”;
        echo “<h3>Preincrement</h3>”;
          $a = 5;
         echo “Should be 6: “ . ++$a . “<br />\n”;
         echo “Should be 6: “ . $a . “<br />\n”;
         echo “<h3>Postdecrement</h3>”;
            $a = 5;
          echo “Should be 5: “ . $a— . “<br />\n”;
          echo “Should be 4: “ . $a . “<br />\n”;
          echo “<h3>Predecrement</h3>”;
            $a = 5;
          echo “Should be 4: “ . —$a . “<br />\n”;
          echo “Should be 4: “ . $a . “<br />\n”;
?>
```

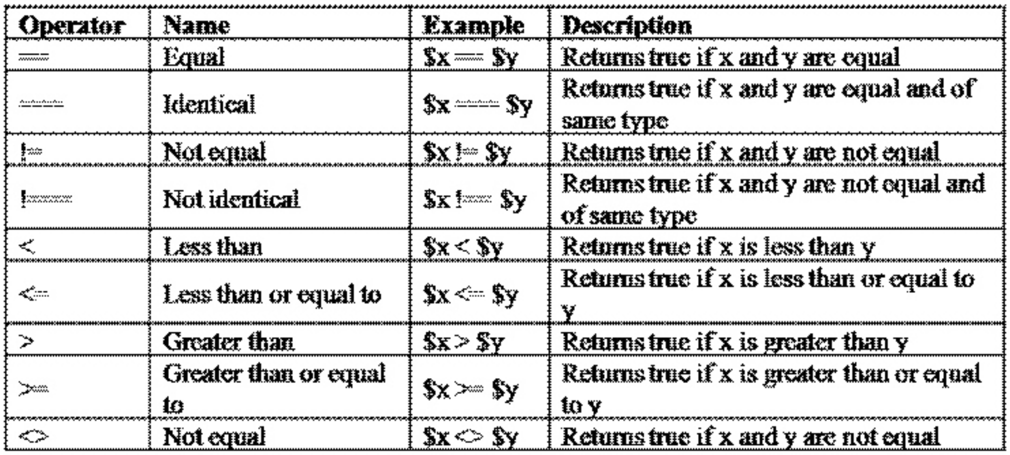
7.9.4 Equal operator: Such types of operators are used to compare two values in scripting. The description given in the following table



Comparison Operators

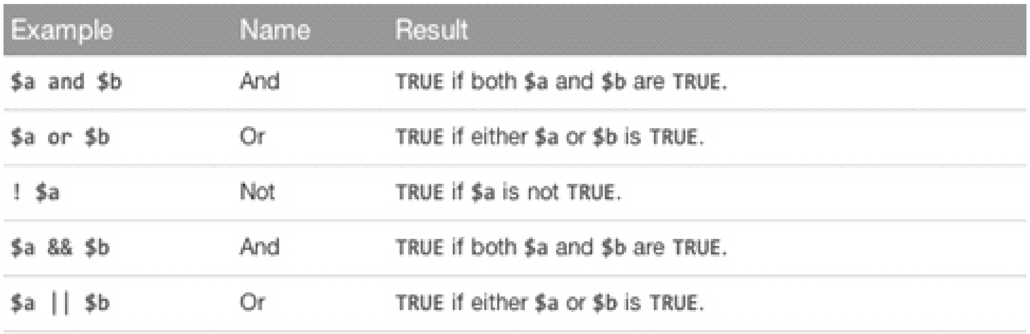
Example	Name	Result
$a == $b	Equal	TRUE if $a is equal to $b after type juggling.
$a === $b	Identical	TRUE if $a is equal to $b, and they are of the same type.
$a != $b	Not equal	TRUE if $a is not equal to $b after type juggling.
$a <> $b	Not equal	TRUE if $a is not equal to $b after type juggling.
$a !== $b	Not identical	TRUE if $a is not equal to $b, or they are not of the same type.
$a < $b	Less than	TRUE if $a is strictly less than $b.
$a > $b	Greater than	TRUE if $a is strictly greater than $b.
$a <= $b	Less than or equal to	TRUE if $a is less than or equal to $b.
$a >= $b	Greater than or equal to	TRUE if $a is greater than or equal to $b.

7.9.5 Relational Operator: In php and other scripting, a relational operator is a programing language construct or operator that tests or defines some kind of relation between two entities as listed following:



Operator	Name	Example	Description
==	Equal	$x == $y	Returns true if x and y are equal
===	Identical	$x === $y	Returns true if x and y are equal and of same type
!=	Not equal	$x != $y	Returns true if x and y are not equal
!==	Not identical	$x !== $y	Returns true if x and y are not equal and of same type
<	Less than	$x < $y	Returns true if x is less than y
<=	Less than or equal to	$x <= $y	Returns true if x is less than or equal to y
>	Greater than	$x > $y	Returns true if x is greater than y
>=	Greater than or equal to	$x >= $y	Returns true if x is greater than or equal to y
<>	Not equal	$x <> $y	Returns true if x and y are not equal

7.9.6 Logical Operator: Such operator is used to check logical calculations and its results, whether they are equal or not. They also work with two operands, and return true or false value respectively.



Example	Name	Result
$a and $b	And	TRUE if both $a and $b are TRUE.
$a or $b	Or	TRUE if either $a or $b is TRUE.
! $a	Not	TRUE if $a is not TRUE.
$a && $b	And	TRUE if both $a and $b are TRUE.
$a || $b	Or	TRUE if either $a or $b is TRUE.

7.9.7 Bitwise Operator: Such operators work on binary digit means bit. Rest of all the operators are work on value of the byte.

Bitwise Operators

Example	Name	Result
$a & $b	And	Bits that are set in both $a and $b are set.
$a | $b	Or (inclusive or)	Bits that are set in either $a or $b are set.
$a ^ $b	Xor (exclusive or)	Bits that are set in $a or $b but not both are set.
~ $a	Not	Bits that are set in $a are not set, and vice versa.
$a << $b	Shift left	Shift the bits of $a $b steps to the left (each step means "multiply by two")
$a >> $b	Shift right	Shift the bits of $a $b steps to the right (each step means "divide by two")

7.9.8 Assignment operator: In Php, we use = operator to denote as assignment operator. In any programming, the calculation is being solved in the expression from right to left according to precedence rule and solve the expression and save into a variable. For example:

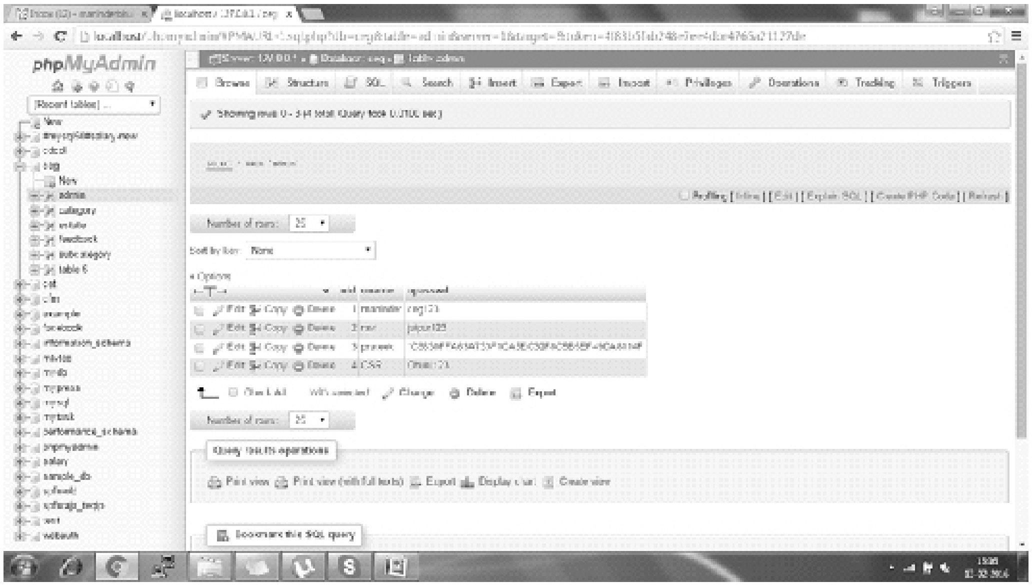
```
<?php

        $a = $b + $c ;

   # In above = operator use for assign the addition result in
$a variable

?>
```

7.10 MySQL Database

As describe in previous chapters we have learnt about database such as MySQL / MS-SQL in details and have been practices on command line. We also have learnt how to code on command line in SQL. In this chapter we will use the GUI (Graphical User Interface) of MySQL and also try to create database and its table as graphically. Once again we define the MySQL database "A database is a collection of information that is organized so that it can be easily accessed, managed and updated". Data is organized into rows, columns and tables, and it is indexed to make it easier to find relevant information. Data gets updated, expanded and deleted as new information is added. Databases process workloads to create and update themselves, querying the data they contain and running applications against it.

In the tables of database, all the columns are also known as fields of the table. Remember that all the records of the tables are denotes the common feature of the database table.

The database and its table stucture are as dipict as following:

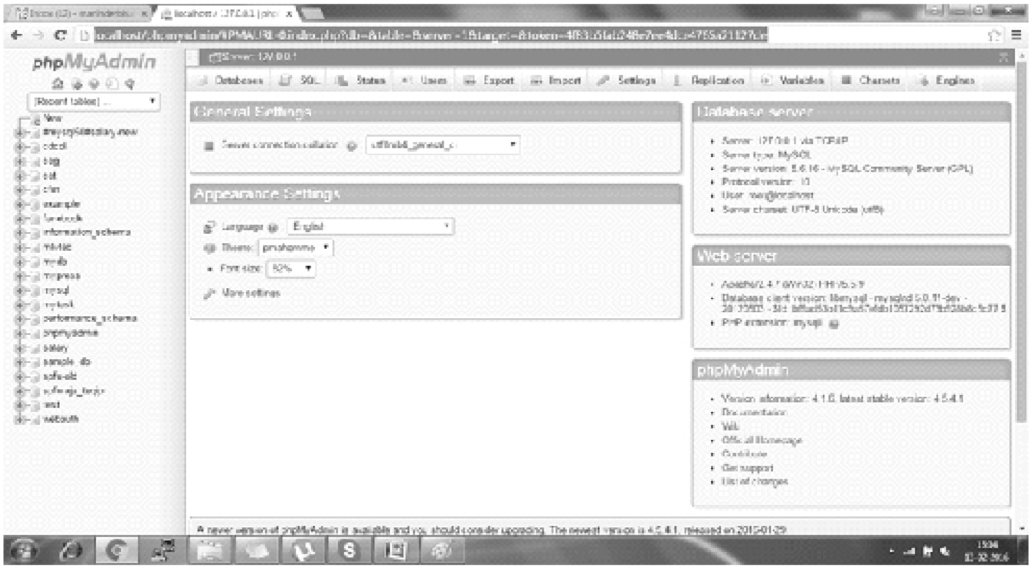


	aid	uname	upassed
Edit Copy Delete	1	maninder	reg123
Edit Copy Delete	2	ravi	pIpur123
Edit Copy Delete	3	praveen	[illegible]
Edit Copy Delete	4	CSR	Oman123

Figure 7.5 Sample MySQL Database and listing of tables

For example, we would like to take example of any school and its students. When we want to store the information of the students in the database and its table, for this we will use SQL code to insert the records in it. So it means, in future when we require to print data on HTML page in bulk as output. At that time we will use Php to fetch data from database and MyQL as back end.

7.10.1 Phpmyadmin: phpmyadmin is free and open source software for mysql access. Actually mysql come with command line, but when we use phpmyadmin it means we can access mysql plateform as GUI also. The phpmyadmin project comes along with XAMPP software which included: Apache, MySQL and Php. In this bundled software the MySQL comes in PhpMyadmin project which can be access over the browser with **http://** protocol. The method is showing in following snapshot:



Server connection collation: utf8mb4_general_ci

Language: English

Theme: pmahomme

Font size: 82%

More settings

- Server: 127.0.0.1 via TCP/IP
- Server type: MySQL
- Server version: 5.6.16 - MySQL Community Server (GPL)
- Protocol version: 10
- User: root@localhost
- Server charset: UTF-8 Unicode (utf8)

- Apache/2.4.7 (Win32) PHP/5.5.9
- Database client version: libmysql - mysqlnd 5.0.11-dev - 20120503 - $Id: [illegible] $
- PHP extension: mysqli

- Version information: 4.1.6, latest stable version: 4.5.4.1
- Documentation
- Wiki
- Official Homepage
- Contribute
- Get support
- List of changes

A newer version of phpMyAdmin is available and you should consider upgrading. The newest version is 4.5.4.1, released on 2016-01-29.

Figure 7.6 Welcome screen of GUI MySQL (i.e. phpmyadmin screen for MySQL access)

We can access GUI mysql, in web browser by typing URL section as http://localhost/phpmyadmin and just open it. Remember it the XAMPP software Apache and MySQL services are running with the TCP port number 80 and 3306 respectively, if they are not running then try to run / execute by click on start the services.

The following diagram illustrates how the constraints are apply on columns of the mysql table. We use different-different types of SQL for applying constraint, by which we can store unique data in columns, so that no one can insert record as repeatedly. For this we use following types of constraints (Criteria) :

1) primary key

2) unique key

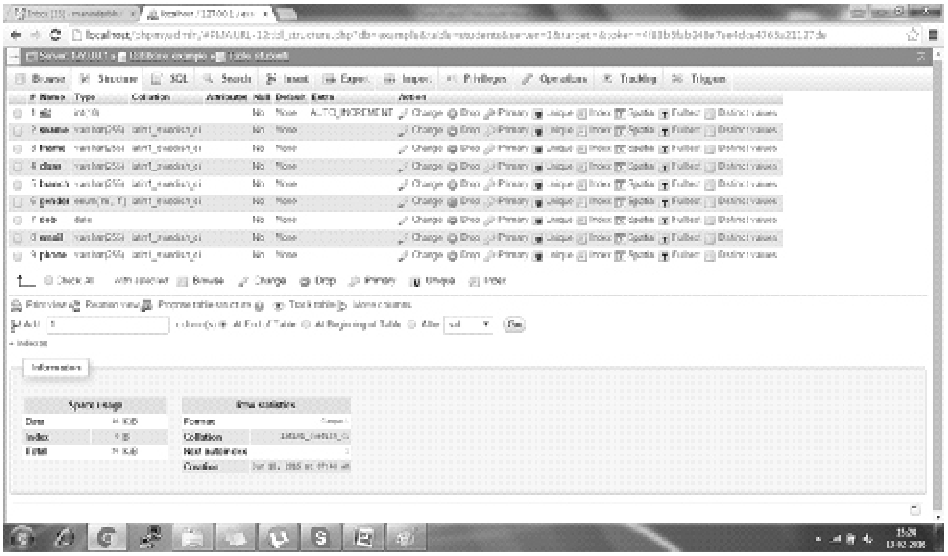
3) foreign key

4) null / not null

5) check

In the next diagram the database in form of tables showing record (data) of the students. The table and its data gives us information how it was stored in row and column format. For example, name of the student, parents name and his / her belonging information can be store for the future aspect.

For Example:



#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	int(10)			No	None	AUTO_INCREMENT	Change Drop Primary Unique Index Spatial Fulltext Distinct values
2	sname	varchar(255)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
3	fname	varchar(255)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
4	class	varchar(255)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
5	branch	varchar(255)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
6	gender	enum('m', 'f')	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
7	dob	date			No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
8	email	varchar(255)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values
9	phone	varchar(255)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext Distinct values

Figure 7.7 constraints and data types for the fields of mysql table

In above diagram the tables shows us every features of the database table which keeps the information also the tables gives us information how the constraint(Criteria) and data types are used simultaneously.

7.10.2 Connectivity code for Php with MySQL: When we want to connect php script and mysql database for getting stored data in the database for many sql operations like: Fetch , Insert, Delete , Update and Search etc. That’s why we need to connect php webpage with mysql database. Every time while connecting php with mysql, remember that mysql user name and password must be secure and not shared with anyone. The following code is an example for connectivity of php script and mysql database:

```
<?php

        Mysql_connect(“localhost”,”root”,””) or die(mysql_erro());

        Mysql_select_db(“ceg”)  or die(mysql_error());

?>
```

Note: in above php script we create an database named as **ceg** and its table name is student. Now when we want to connect php page with myql for fetching students record, we must use such code for the connectivity purpose, and save above script as connect.php and save this under the file system, **c:\xampp\htdocs\myceg**.

In above location of the project is cegjaipur and we need to create a directory under the file system c:\xampp\htdocs\ and then we need to save connect.php code there. After this we can run this script by typing in the URL section of the browser as follows:

http://localhost/myceg/connect.php

When a another php page require an connection establishment with database, then we can call the same connect.php code by using include() or require() function in php script. It means no need to create connectivity code again and again.

For example:

```
<?php

        incldue("connect.php");

        $a = mysql_query("select * from students");

        echo "Hello Students how are you ";

?>
```

In above example the mysql_query() function will execute the query, when the connect.php is working fine, It means connectivity working properly otherwise we get an error message.

7.10.3 Data types in mysql: In the previous chapter we learnt about the data types in details. So we are listing some of the basic data types available in mysql :

1) date & time

2) timestamp

3) boolean

4) varchar

5) char

6) enum

7) int

8) float

9) double

11) text

7.10.4 Database attribute: There are in-built types of attribute which are available in the mysql database, such as **auto_increment,** which increment the value of integer type of data type in the table of mysql. For example when we store data in the table of mysql, just imagine about students records, in which we want to identified the row number and that particular number never repeat or null / blank type, so that we use auto increment with constraint of primary key. Always remember that the particular key must be integer type and being auto incremental. This feature also allow us to count all the number of records have been inserted in the table.

7.10.5 Index attributes: As finding one word from the dictionary which contains thousand of words, similarly in database when we trying to find a particular data out of thousands sets of records in database, we mostly used index attribute by implement this feature on a particular table. Many times the database administrator apply such attribute manually on table, but it implement automatically while we use primary and unique key on fileds of the table. Always remember that when a newly tabled created in mysql database, it always have constraint of unique and not null types.

Important Points

1 Always install XAMPP / WAMP software, to use apache, myql and php.

2 Always create web projects under the file system: c:\xampp\htdocs\

3 Php script can be write in any of the IDE (Interface development environment)

4 Always write the php code in between **<?php ?>** tags.

5 **echo** keyword can be used for output statement.

6 Php was developed by Rasmus Lerdorf.

7 We comment single line of code in php by using // and by using #.

8 In php there are four types of control statements: Sequential, conditional, iterative, jumping.

9 To show logical expression, we use operator and operands.

10 Mysql is a Relational database management system.

11 By connectivity code in php we may perform many operations like: DQL (Data Query Language), DML(Data Manipulation Language), DDL(Data Definition Language), and many more.

12 The TCP port number for access apache webserver normally (http://) and securely (https://) is 80 and 443 respectively. The mysql use TCP port number 3306.

Exercises

Objective Type Questions

1. Php was developed by:

a) Radarford b) Bill gates

c) Rasmus Lerdorf d) Jems Robert

2. How to initialize a variable in php:

a) # b) @

c) $ d) *

3. Which one of the Open Source bundled software is used for php and mysql :

a) JDK b) Microsoft .NET

c) Eclips d) XAMPP

4. Which port number is used by the Apache webserer:

a) 82 b) 85

c) 90 d) 80

5. Which port number is used by the mysql server

a) 3122 b) 8080

c) 3000 d) 3306

6. To start php script which one of the scripttlet tags are being used ?

a) <?php ———————— ?> b) <!php ———————— !>

c) <%php ———————— %> d) <%php ———————— ?>

7. Which one is example of loop / iteration in php script ?

a) while b) float

c) int d) long

8. Which one is the conditional statement type ?

a) if —else b) while

c) for d) foreach

9. Which one statement is used for the collection of similar data types ?

a) array b) union

c) constructor d) class

s10. Which one is the type of associative array of type ?

a) $_GET b) array()

c) for each d) class

Short Type Questions

1. How to write a php script.?
2. How to declare and create a function in php?
3. How to create array in php?
4. Create an expression in php ?
5. What is order of operators in php?
6. Prepare the list of operators.
7. Explain **echo** and print() function.
8. Differentiate between array and class of php.
9. Brief about php and mysql connectivity.
10. Explain the inbuilt function mysql_connect()

Essay Type Questions

1. Create a php script to declare and initialize the variable.
2. How to connect php and mysql and also explain inbuilt function: mysql_connect() and mysql_select_db()
3. Create a php script to perform all the arithmetic operations along with the function.
4. What types keys are available in mysql ?
5. Explain in details of all operators available in PHP.

Answer Key

1.	c	2.	c	3.	d	4.	d
5.	d	6.	a	7.	a	8.	a
9.	a	10.	a				