Basic of PHP

version 1.0

Last generated: August 31, 2022

Table of Contents

Getting started with Php

Summary: This is brief instruction about the requirments that you need to have before you start writing codes on Php.

Installations

You need to have following requirments to run php on your local machine

1. Download and Install Xampp

XAMPP stands for Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purposes

You can download xampp here

2. Download and Install a code editor

You can have one of these editors to write/edit your code:-

- Visual code studio
- Atom
- Sublime-Text

You can also use other tools as per your prefrence.

Git & Github

1. Github account

GitHub is a web-based hosting service for software development projects that use the Git revision control system.

For beginners it can be a best place to store your code. Learning about version control system in the earlier phase of programming can give you a head-start later phase of coding and developement.

2. Gitbash for windows

Git-Bash is an application for Microsoft Windows environments which provides an emulation layer for a Git command line experience. Bash is a popular default shell on Linux and macOS.

> ℹ **Note:** You don’t have to download git bash if you use Mac or Linux as your main operating system.

[]:

Introduction

Overview

This site provides documentation, training, and other notes for the php programming language. The documentation is designed for people who have no prior knowledge about php.

What is PHP?

- php stands for Hypertext Preprocessor.
- PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages.
- PHP is a widely-used, free, and efficient alternative to

What you should know?

Before you get started you should have basic knowledge about following topic:-

- HTML
- CSS
- JS

What is a PHP file ?

PHP files have extension “.php”. The code runs on the server and the result is returned to the browser as plain HTML.

A PHP file can contain HTML, CSS, JavaScript and PHP code.

> **Important information**: PHP is the core of wordpress, the biggest blogging platform. It is also used by facebook, the largest social media site. More than anything it is very easy to learn.

Getting started

To get started, see Getting Started (page 3).

[]:

Letter From the Author

Hi,

This documentation is designed as a teaching material for php class. You can feel free to use this material as your learning guide.

If you have any queries or feedbacks I would love to know them, you can comment your queries in the respective post or find my email address on the top navigation bar of this website to write me personally.

Thanks,

-Riwaj Chalise (Author)

[]:

Introdution to HTML

Summary: HTML one of the major prerequisite you need to have in order to understand php.

What is HTML?

- HTML stands for Hypertext Markup Language.
- HTML is a the markup language designed for creating the webpage.
- HTML describes the structure of the web page.
- HTML elements are represented by tags.
- Browser uses HTML tags to render HTML content.

Note: The file extension for a HTML file is .html

A Basic HTML Document?

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>Heading</h1>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, se
d do eiusmod tempor incididunt ut labore et dolore magna aliqu
a. Ut enim ad minim veniam, quis nostrud exercitation ullamco l
aboris nisi ut aliquip ex ea commodo consequat. </p>

</body>
</html>
```

DEMO

Understanding the tags

- The `<!DOCTYPE html>` declaration defines this document to be HTML5
- The `<html>` element is the root element of an HTML page
- The `<head>` element contains meta information about the document
- The `<title>` element specifies a title for the document
- The `<body>` element contains the visible page content
- The `<h1>` element defines a large heading
- The `<p>` element defines a paragraph

⚠ **Important:** Some of the important in html are `<table>`, `<form>`, `<img>` etc.

If you want to know more about HTML you can check the HTML_tutorial by w3 school.

Introdution to CSS

Summary: CSS is another major prerequisite you need to have in order to understand php.

What is a CSS?

- CSS stands for Cascading Style Sheets
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media
- External stylesheets are stored in CSS files

Note: The file extension for an external CSS file is .CSS

Three basic ways of implementing CSS

1. Inline CSS

The inline CSS is written to uniquly style a single element.

Inline styles are defined within the “style” attribute of the relevant element:

Example of inline CSS

```
<!DOCTYPE html>
<html>
<body>

<h1 style="color:blue;text-align:center;">This is a heading</h
1>
<p style="color:red;">This is a paragraph.</p>

</body>
</html>
```

DEMO

2. Internal CSS

The internal style is defined inside the `<style>` element, inside the head section.

Exmaple of Internal CSS

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: linen;
}

h1 {
  color: maroon;
  margin-left: 40px;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

DEMO

3. External CSS

External styles are defined within the `<link>` element, inside the `<head>` section of an HTML page:

Example of External CSS

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Here is how the “mystyle.css” file looks like:

```
body {
  background-color: lightblue;
}

h1 {
  color: navy;
  margin-left: 20px;
}
```

DEMO

⚠ **Important:** A CSS file should not contain html tags

> If you want to know more about CSS you can check the CSS_tutorial by w3 school.

[]:

HTML tables

Summary: This post will give you the brief introduction about HTML tables. However, It won't cover everything about HTML tables but just enough to use it for you PHP lesson.

- A HTML table is defined by `<table>` tag.
- The head is defined by `<th>` tag.
- The row is defined by `<tr>` tag.
- The data is defined with `<td>` tag.

Here is what a simple HTML table looks like:-

Example of simple table

```
<table>
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Charls</td>
    <td>Doe</td>
    <td>22</td>
  </tr>
  <tr>
    <td>Dudly</td>
    <td>Jons</td>
    <td>44</td>
  </tr>
</table>
```

DEMO

However, we can add css properties to enhance the design of the table

Example of a styled table.

```
<head>
 <style>
table, th, td {
  border: 1px solid black;
}
</style>
</head>
<body>
<table style="width:100%">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Charls</td>
    <td>Doe</td>
    <td>22</td>
  </tr>
  <tr>
    <td>Dudly</td>
    <td>Jons</td>
    <td>44</td>
  </tr>
</table>
```

DEMO

HTML Forms

Summary: This post will give you the brief introduction about HTML forms. The resources available here is just enough to understand this course.

The <form> Tag

The html form is defined by `<form>` tag.

Syntax of Form Element

```
<form>
...
Various input tags
...
</form>
```

The <input> Tag

The `<input>` element is most important element to make forms in html.

The input element can have various forms that depens on text attribute:-

Some common types of input elements.

1. **text** :- Used to define text field.
2. **radio** :- Used to select one of many choices.
3. **checkbox** :- Used to select Multiple choice.
4. **submit** :- Used to define submitt button for the forms data.
5. **password** :- Used to define password field.
6. **date**:- used to define date field.

Example of Simple form

```
<form action="">
  First name:<br>
  <input type="text" name="firstname"><br>
  Last name:<br>
  <input type="text" name="lastname"><br>
  email:<br>
  <input type="email" name="email"><br>
  password:<br>
  <input type="password" name="password"><br>
  <input type="submit" value="Submit">
</form>
```

DEMO

The <fieldset> and <legend> tags.

The `<fieldset>` is used to group related date in a form and the `<legend>` tag is used to define the csption fo the fieldset.

Examples of forms using `<fieldset>` and `<legend>`

```
<form action="">
  <fieldset>
    <legend>Fill the Form:</legend>
<form action="/action_page.php">
  First name:<br>
  <input type="text" name="firstname"><br>
  Last name:<br>
  <input type="text" name="lastname"><br>
  email:<br>
  <input type="email" name="email"><br>
  password:<br>
  <input type="password" name="password"><br>
  <input type="submit" value="Submit">
</form>
  </fieldset>
</form>
```

DEMO

Designed Forms in CSS

You can use various CSS properties to design a form.

Examples of CSS Designed Form

```
<style>
input{
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  border-radius: 4px;
  box-sizing: border-box;
}

input[type=submit] {
  width: 100%;
  background-color: red;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  border-radius: 4px;
  cursor: pointer;
}

</style>


<h3>Using CSS to style an HTML Form</h3>

<form action="">
  <fieldset>
    <legend>Fill the Form:</legend>
<form action="/action_page.php">
  First name:<br>
  <input type="text" name="firstname"><br>
  Last name:<br>
  <input type="text" name="lastname"><br>
  email:<br>
  <input type="email" name="email"><br>
  password:<br>
  <input type="password" name="password"><br>
  <input type="submit" value="Submit">
</form>
  </fieldset>
</form>
```

DEMO

HTML and CSS practice questions

Summary: Here you can find questions regarding HTML and CSS.

Question 1

Make a simple HTML page. The page must contain your picture information about your hobbies and education. Use External CSS to style your page.

Question 2

Make a HTML table with 5 entities(columns) and 6 records(rows). Use external CSS to style the table.

Question 3

Make a HTML form to that takes input as following entities Name(text), Email(email), Phone-number(tel), Gender(radio), Message(textarea), Submit Button(submit). Use appropriate input types. Also, use External CSS to style your page.

echo

Summary: This post will help you understand how to use echo() function on php.

echo is a php function that can be used to print one or many string.

Syntax of echo

```
echo (string);
```

Example-1

A simple echo statement

```
<?php
echo "Welcome to php class.";
?>
```

DEMO

You can also print variables

Example-2 (Variables)

You can also print variables.

> ⚠ **Important:** PHP variables will be discussed later in the documentation. For now just know that a php variable starts with a $ sign.

```
<?php
$str = "Welcome to PHP Class";
echo $str;
?>
```

DEMO

Example-3 (HTML tags)

You can also embed HTML tags in the echo statement.

```
<?php
echo "<h1>This is h1 tag</h1>";
echo "<p>This is p tag</p>";
?>
```

DEMO

Example-4 (Joining)

You can join the two variables together by using `.` as a joining operator:-

```
<?php
$str1 = "Welcome to";
$str2 = "PHP class";
echo $str1 . " " . $str2;
?>
```

DEMO

[]:

Comments

Summary: This post will help you understand how to comment your codes in php.

A comment is the part of the program that is not executed. The comments are only visible to the people who read source code.

Why are comments used ?

- Comments can help readers to understand code.
- Comments can be used to Describe the codes.
- Comments can be used to make codes dormant i.e. making codes unexecutable.

Example-1 (Single line comments)

This example shhows how single line comment is written in php.

```
<?php
// single-line comment by double back slash

# single-line comment by using a hash
?>
```

DEMO

Example-2 (Multiple line comments)

This example shows how multiple line comments are written in php

```
<?php
/* This is a multiple like comment
and can be written in
more than one line.
*/
?>
```

DEMO

Variables

Summary: This post will help you understand what are variables and how to make variables in php.

Variable is a name that has a value associated with it. Variables are used to store values such as strings, numeric values, characters or memory address.

Declaring variables in PHP

Variables in PHP start with a `$` sign followed by the name of the variable.

For a variable to be valid the variable name must start with alphabet (a-z or A-Z) or underscore (_), followed by numbers, letters or underscore.

There cannot be white space in variable so if the variable has more than oneo word it is seperated by commas (for example `$two_words` , `deer_walk`)

Some Valid PHP Variables

Here are some valid php variables:

Example-1

```
<?php
$xyz = 'welcome';
$XYZ = 'welcome';
$Xyz = 'welcome';
$xyz1231 = 'welcome';
$xyZ = 'welcome';
$XYZ = 'welcome';
$_XYZ = 'welcome';
$XYZ_xyz = 'welcome';
?>
```

DEMO

> ⚠ **Important:** PHP variables are case sensitive.

Example-2(Operation)

```php
<?php
$x = 20;
$y = 30;
echo $x + $y;
?>
```

DEMO

PHP: A loosly typed language

PHP automatically detects the datatype of the variable by the type of value we give the variable.

The data type is not set in strict manner, that means you can do things like adding strings to intiger without causing an error.

However, PHP 7 allows user to assigne datatype by enabling strict. This gives an option to declare datatype while assigning a variable.

```php
<?php
$a = 2;
$b = 2.33;

echo $a + $b;
?>
```

DEMO

Data Types

Summary: This post will help you understand datatypes in php.

Variables store different datatypes like string, intiger, float, boolean, array.

Note: var_dump() is a function that returns datatype and value.

String

A string is a group of characters put together, like “deer walk”:

A string is always assigned inside a duble or a single quote:-

```
<?php
$name = "Class of 2022";
var_dump($name)
?>
```

DEMO

Integer

Integer is a datatype with non-decimal numbers. A integer can be either Positive or negetive. Arthmatic opreations can be performed on intigers.

```
<?php
$x = 5985;
$y = 2722;
$z = $x + $y;
var_dump($z);
?>
```

DEMO

Float

A float is a number with decimal or exponential value.

```php
<?php
$x = 12.124;
var_dump($x);
?>
```

DEMO

Boolean

Boolean is used in conditional testing to test if the statement is True or False.

```php
<?php
$true = true;
$false = false;

var_dump($true);
var_dump($false);
?>
```

Demo

Array

Array is used to store multiple valued in a single variable.

```php
<?php
$company = array("google","facebook","twitter");
var_dump($company);
var_dump($company[0]);
var_dump($company[1]);
var_dump($company[2]);
?>
```

DEMO

There are more datatypes. For more information on datatypes check out w3school .

Operators in PHP

Summary: This post can help you undersand about operators in PHP.

Operators are used to perform operations. The types of operators in php are:

- Arithmetic Operators
- Assignment Operators
- Comparison Operators
- Increament/Decrement Operators
- Logical Operators
- String Operators
- Array Operators
- Conditional Assignment Operators

Arthmetic Operators

Arthmatic operators are used to perform common arithmetical operations such as addition, subtraction, division.

Name (Operators)	Discription	Demo
Addition (+)	Performs Addition	DEMO
Subtraction (-)	Performs Subtraction	DEMO
Multiplication (*)	Performs Multiplication	DEMO
Division(/)	Performs Division	DEMO
Modulus (%)	Finds Reminder	DEMO
Exponentiation (**)	Raise to the power	DEMO

Assignment Operator

Php assignment operators are used with numerical value to write a value to a variable.

Expression	Equivalent	Discription	Demo
a = b	a = b	The value on the left hand side get assigned to the value on the right hand side	DEMO
a += b	a = a + b	Addition	DEMO
a-=b	a = a - b	Subtraction	DEMO

The multiplication, division and modulus will go on in the same way.

Comparision Operators

The comparision operator is used to compare two valves.

Operator	Discription	Demo
==	True if two variables are equal	DEMO
===	True if two variables are identical i.e. both value and datatype	DEMO
!=	True if two variables are not equal	DEMO
<>	True if are not equal	DEMO
!==	True if two variables are not identical	DEMO

Operator	Discription	Demo
>	Greater than	DEMO
<	Less than	DEMO
>=	Greater or equal to	DEMO
<=	Less than or equal to	DEMO

Please refere W3school for further information on PHP operators

Conditional Statement

Summary: This post will help you understand conditional logic in PHP

Conditional statements are used to perform actions based on conditions.

Types of conditional statements in PHP:-

1. if statement
2. if...else statement
3. if...elseif...else statement
4. switch statement

if Statement

The `if` statement executes codes if one condition is true.

Syntax

```
if (condition) {
    //code;
}
```

Example

```
<?php
$t = 20;

if ($t == "20") {
    echo "I have Rs. 20";
}
?>
```

DEMO

if...else Statement

The `if...else` statement executes some code if the condition is true and another code if the condition is false.

Syntax

```
if (condition) {
   // code1
} else {
   // code2;
}
```

Example

```
<?php
$t = 10;

if ($t == "20") {
    echo "I have Rs. 20";
} else {
    echo "I don't have Rs. 20";
}
?>
```

DEMO

if...elseif...else Statement

If there are more than one condition then we can use `if...elseif...else` condition.

Syntax

```
if (condition) {
    //code1
} elseif (condition) {
    //code2
} else {
    //code3
}
```

Example

```php
<?php
$t = 10;

if ($t == "20") {
    echo "I have Rs. 20";
} elseif($t < 20) {
    echo "I have less than Rs. 20";
} else{
    echo "I have more than Rs. 20";
}
?>
```

DEMO

Switch

Summary: This post will describe about Switch case in php.

The `switch` statement in PHP can be used to perform differnet action based on different cases. If can be used as a substitute for if…elseif

Syntax

```
switch (a) {
    case value1:
        code to be executed if a = value1;
        break;
    case var2:
        code to be executed if a=var2;
        break;
    ...
    default:
        code to be executed if a is different from all var;
}
```

Understading The Syntax

- There are a number of expressions (variables).
- There are number of cases.
- The case and expression is checked, if there is a match the block of code associated with the case is executed.
- The `break` statement prevent the code from running into the next case automatically.
- The code segment `default` case is executed if no match is found.

Example

```php
<?php
$num = 3;

switch ($num) {
    case 1:
        echo "Your favorite number is 1";
        break;
    case 2:
        echo "Your favorite number is 2";
        break;
    case 3:
        echo "Your favorite number is 3";
        break;
    default:
        echo "Your favorite number is not mentioned";
}
?>
```

DEMO

For Loop

Summary: This post describes the conceptof for loop in php

What is a Loop?

In computer science, a loop is a programming structure that repeats a sequence of instructions until a specific condition is met.

For loop

For loop is used to run a specific set of code until the specified number of times.

Syntax

```
for (initialization; condition; increment){
   code to be executed;
}
```

Example1

A simple for loop program to print number from 0 to 10

```
<?php
for ($x = 0; $x <= 10; $x++) {
    echo "The number is: $x <br>";
}
?>
```

DEMO

Here in this example:

- $x = 0 initilization
- $x <= 10 Condition
- $x++ Increases the loop value counter by 1

Example2

A simple program to print the table of 2;

```
<?php
for ($x = 1; $x <= 10; $x++) {
    $b = 2 * $x;
    echo "2 * " .  $x . "= ". $b. "\n";
}
?>
```

DEMO

While Loop

Summary: This post explains the concept of while loop in PHP.

While Loop

The `while` loop executes a block of code as long as the specified condition is true.

Syntax

```
while (true) {
    #code1;
}
```

Example1

A program to print number from 1 to 10 using while loop.

```
<?php
$x = 1;

while($x <= 10) {
    $x++;
}
?>
```

DEMO

Do While loop

Summary: Tis post describes do while loop in PHP

D0...While

The `do...while` loop will always execute the block of code once, then check the condition, and repeat the loop until the specified condition is true.

```
do {
    #code;
} while (condition);
```

Example

A program to print number from 1 to 10 using do_while.

```
<?php
$x = 1;

do {
    echo "$x";
    $x++;
} while ($x <= 10);
?>
```

DEMO

Difference between While and DO...While loop.

While	Do...While
In ‘while’ loop the controlling condition appears at the start of the loop.	In ‘do-while’ loop the controlling condition appears at the end of the loop.

While	Do…While
The iterations do not occur if, the condition at the first iteration, appears false.	The iteration occurs at least once even if the condition is false at the first iteration.
Semicolon is not used.	Semicolon is used at the end of the loop.

DUMMY

Summary: This post will help to understand array in php.

Array

Array is a variable that can store multiple values. In php `array()` function is used to make an array.

Example

```
<?php
$bikes = array("RC", "CRF", "DUKE");
echo "I like " . $bikes[0] . ", " . $bikes[1] . " and " . $bike
s[2] . ".";
?>
```

DEMO

Types of Array

The types of array are:-

- Indexed Arrays
- Associative Arrays
- Multidimentional Arrays

Indexed Arrays

An index in an array is a unique value that is used to identify a value of an array. The index always starts from 0.

Index can be assigned automatically:-

```
$bikes = array("RC", "CRF", "DUKE");
```

Index can be assgned manually:-

```
$cars[0] = "Volvo";
$cars[1] = "BMW";
$cars[2] = "Toyota";
```

Looping in Indexed array

Here we use for loop to loof through ndexed array.

```php
<?php
$bikes = array("RC", "CRF", "DUKE");
$arrlength = count($bikes);

for($x = 0; $x < $arrlength; $x++) {
    echo $bikes[$x];
    echo "<br>";
}
?>
```

DEMO

Associative Arrays

Associative arrays are arrays that use named keys that you assign to them.

```
$bikes = array("KTM"=>"RC", "Honda"=>"CRF", "KTM2"=>"DUKE");
```

Looping in Associative Array

```php
<?php
$bikes = array("KTM"=>"RC", "Honda"=>"CRF", "KTM2"=>"DUKE");

foreach($bikes as $x => $x_value) {
    echo "Key=" . $x . ", Value=" . $x_value;
    echo "<br>";
}
?>
```

DEMO

Multidimensional Arrays

A multidimensional array is an array containing one or more arrays.

```
$bikes = array
  (
  array("RC",56,34),
  array("DUKE",12,6),
  array("CRF",4,2),
  array("Pulser",13,11)
  );
```

Presentation of Multidimensional Array

```
<?php
$bikes = array
  (
  array("RC",56,34),
  array("DUKE",12,6),
  array("CRF",4,2),
  array("Pulser",13,11)
  );

echo $bikes[0][0].": In stock: ".$bikes[0][1].", sold: ".$bike
s[0][2].".<br>";
echo $bikes[1][0].": In stock: ".$bikes[1][1].", sold: ".$bike
s[1][2].".<br>";
echo $bikes[2][0].": In stock: ".$bikes[2][1].", sold: ".$bike
s[2][2].".<br>";
echo $bikes[3][0].": In stock: ".$bikes[3][1].", sold: ".$bike
s[3][2].".<br>";
?>
```

DEMO

Operators in PHP

Summary: Questions related to php.

Theoratical Questions

Question 1

Write the syntax if...else and switch statement. Write the difference between if...else and switch statement.

Question 2

Write the syntax while and do...while loop. Write the difference between while and do...while loop.

Question 3

Write the syntax of for loop. Explain the for loop in detail.

Question 4

Define opereator. What are the different types of operator.

Practical Questions

Question 1

Given two variable, write a program using if else statement to find the sum, difference and multiplication of the two variables.

Question 2

Do the first question using switch statement.

Question 3

Write a program to print all even numbers between 1 to 1000 using for loop.

Question 4

Using While loop, Write a program to print the multiplication table of 731.

Question 5

Writea program to demonstate an example of Do…While loop.

Intro To Database

Summary: This post provides brief introducction on database.

What is Data ?

Data is fact related to any object into consideration. Example: Name, Age, Class etc.

What is Database ?

Database is the place where data are stored in table. Database makes the management of data easy.

What is table?

A table is made of rows and columns and each row is called a record and each column represent an entity.

What is SQL?

SQL stands for Structured Query Language. A query language is a kind of programming language that’s designed to facilitate retrieving specific information from databases.

Database query

Summary: This post discusses basic SQL Query used in this lesson.

A database query is a request for data from a database. The request is to retrive and manipulate data from the database.

Select

Select statement is used to select data from database. It returns the stored data from the database.

Syntax

```
SELECT column1, column2, ...
FROM table_name;
```

To select all field available in the table we use * insted of column name.

```
SELECT * FROM table_name;
```

Insert into

The insert into statement is used to insert new record in the table.

Syntax

```
INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);
```

Update

The Update statement is used to modify the existing record in the table.

Syntax

```
UPDATE table_name
SET column1 = value1, column2 = value2, ...
WHERE condition;
```

Example

```
UPDATE Customers
SET ContactName = 'Somthing', City= 'iland'
WHERE CustomerID = 1;
```

Delete

The delete statement is used to delete existing record in the table.

Syntax

```
DELETE FROM table_name WHERE condition;
```

Example

```
DELETE FROM Customers WHERE CustomerName='something';
```

Database Connection

Summary: This post describes how mysqli database is connected in PHP.

MYSQLi

The MySQLi Extension (MySQL Improved) is a relational database driver used in the PHP programming language to provide an interface with MySQL databases. In this tutorial we will use MySQLi driver since it provides procedual and OOP interface.

Databasse Connection

```php
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "test";

// Create database connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
?>
```

> when you install XAMPP in your windows machine, the root password for the MySQL is set to empty. But this is not recommended, as the MySQL database without a password will be accessible to everyone. To avoid this, a proper/secure password must be set to the user root

Introduction to CRUD Operation

Summary: This post describes CRUD operation in PHP and MySql.

CRUD stands for create read update and delete of data in database.

DB connection

It is very important to connect to database for each of crud operation. So first we make a database file named `config.php` and add the object everytime we perform one of the CRUD operations. `include_once()` method is used to add the database object.

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "test";

// Create database connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
?>
```

Create

Execute the followig code to create a table named `message` in the database.

```php
<?php

include_once("db.php");

$sql = "Create table message
        (Name varchar(30),
        Email varchar(30),
        Message varchar(500)
        )";

if ($conn->query($sql) === TRUE) {
    echo "New table created";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}

$conn->close();
?>
```

Insert

Executing the code below will insert a record in the table created.

```php
<?php

include_once("config.php");

$sql = "INSERT INTO message (Name, Email, Message)
VALUES ('John', 'john@example.com', 'this is a message')";

if ($conn->query($sql) === TRUE) {
    echo "New record created successfully";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}

$conn->close();
?>
```

Read

Executing the code below will show/retrive the record.

```
 <?php

include_once("db.php");

$sql = "SELECT Name, Email, Message from message";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        echo "<br> Name: ". $row["Name"]. " - Email: ". $row["E
mail"]. " " .  " -Message:" . $row["Message"] . "<br>";
    }
} else {
    echo "no results";
}

$conn->close();
?>
```

Update

Executing the code below will update the email of the records name John.

```php
<?php

include_once("db.php");

$sql = "UPDATE message SET Email = 'wa@php.m' where Name = 'Joh
n' ";

if ($conn->query($sql) === TRUE) {
    echo "Updated successfully";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}

$conn->close();
?>
```

Delete

Executing thec code below will delete the record.

```php
<?php

include_once("db.php");

$sql = "DELETE FROM message WHERE Name = 'John'";

if ($conn->query($sql) === TRUE) {
    echo "Deleated Successfully";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}

$conn->close();
?>
```

In the next article we will perform these operation by creating a user interface.

HTTP Method (GET and POST)

Summary: This post describes the HTTP method (GET and POST) method.

HTTP Method

HTTP defines methods to indicate the desired action to be performed on the identified resource. We have two methods for the shakke of understanding this tutorial.

- GET
- POST

GET Method

Get method is used to request data from specific resource. In get method string value pairs are sent in the URL of a get request.

```
/test/demo_form.php?name=Raj&id=2
```

POST method

POST is used to send data to a server to create/update a resource. The data sent to the server with POST is stored in the request body of the HTTP request:

```
POST /test/demo_form.php HTTP/1.1
Host: charlsphp.com
name=Raj&id=2
```

Adding Data Using Form.

Summary: Adding data on database using form

Database Connection

Database connection is very important while playing with any of the CRUD operation. Name the Database file `config.php`.

Download and import the database.

Download Database

config.php

```
<?php


$databaseHost = 'localhost';
$databaseName = 'test';
$databaseUsername = 'root';
$databasePassword = '';

$mysqli = mysqli_connect($databaseHost, $databaseUsername, $dat
abasePassword, $databaseName);

?>
```

HTML Form

Add the following HTML form. The form has Http method post and the action as add.php. This form contain three different inputs:-

- Name
- Age
- Message

add.html

```
<html>
  <head>
    <title>Add Data</title>

  </head>

  <body>
    <a href="index.php">Home</a>
    <br /><br />

    <form action="add.php" method="post" name="form1">
      <table width="25%" border="0">
        <tr>
          <td>Name</td>
          <td><input type="text" name="name" /></td>
        </tr>
        <tr>
          <td>Age</td>
          <td><input type="text" name="age" /></td>
        </tr>
        <tr>
          <td>Message</td>
          <td>
            <textarea type="text" name="message" rows="10" col
s="50"></textarea>
          </td>
        </tr>
        <tr>
          <td></td>
          <td><input type="submit" name="Submit" value="Add"
/></td>
        </tr>
      </table>
    </form>

  </body>
</html>
```

PHP Code

This file handles the data entered by the users.

add.php

```php
<?php
//including the database connection file
include_once("config.php");

if (isset($_POST['Submit'])) {
        $name = mysqli_real_escape_string($mysqli, $_POST['nam
e']);
        $age = mysqli_real_escape_string($mysqli, $_POST['ag
e']);
        $message = mysqli_real_escape_string($mysqli, $_POST['m
essage']);



        $result = mysqli_query($mysqli, "INSERT INTO users(nam
e,age,message) VALUES('$name','$age','$message')");

        //display success message
        echo "<font color='green'>Data added successfully.";
        echo "<br/><a href='index.php'>View Result</a>";
}
?>
```

In the above code:

- First the data from the foms are captured.
- If the fields are not empty the data are inserted in database.
- Then display screen is shown.

Retrive Data From the Database

Summary: In the last post we inserted data in the database. This post discusses about the retrival of data from the database.

Retrival(Select)

The Following code shows the data in table format. It will fetch all the data from the database.

index.php

```
<?php
//including the database connection file
include_once("config.php");

//fetching data in descending order (lastest entry first)
$result = mysqli_query($mysqli, "SELECT * FROM users ORDER BY i
d DESC"); // using mysqli_query
?>

<html>
<head>
        <title>Homepage</title>
</head>

<body>
<a href="add.html">Add New Data</a><br/><br/>

    <table width='80%' border=1px>

    <tr bgcolor='#CCCCCC'>
            <td>Name</td>
            <td>Age</td>
            <td>Message</td>
            <td>Delete</td>
    </tr>
    <?php
        //Using while loop to pick each row until last row
    while($res = mysqli_fetch_array($result)) {
            echo "<tr>";
            echo "<td>".$res['name']."</td>";
            echo "<td>".$res['age']."</td>";
            echo "<td>".$res['message']."</td>";
            echo "<td><a href=\"edit.php?id=$res[id]\">Edi
t</a> | <a href=\"delete.php?id=$res[id]\" onClick=\"return con
firm('Are you sure you want to delete?')\">Delete</a></td>";
    }
    ?>
    </table>

</body>
</html>
```

In above code:-

- Data are fetched in decending order.

- While loop is used to display the record until the last record.
- The record is displayed in table format.

Data Types

Summary: This post will help you understand how to delete data from database.

Delete

In the previous data we selected data from the database. In refrence to previous post we will now understand how to delete data from database.

delete.php

```
<?php
//including the database connection file
include("config.php");

//getting id of the data from url
$id = $_GET['id'];

//deleting the row from table
$result = mysqli_query($mysqli, "DELETE FROM users WHERE id=$i
d");

//redirecting to the display page (index.php in our case)
header("Location:index.php");
?>
```

In above code:-

- We first include the database file.
- Get the id of row to be deleted.
- Apply the delete query to delete the data.
- redirect to index.php

Update(Edit) Data from Database

Summary: This post will help you understand how to edit data from database.

Edit

In refrence to previous post (retrive) we will now understand how to edit data from database.

edit.php

```php
<?php
// including the database connection file
include_once("config.php");

if(isset($_POST['update']))
{

        $id = mysqli_real_escape_string($mysqli, $_POST['id']);

        $name = mysqli_real_escape_string($mysqli, $_POST['nam
e']);
        $age = mysqli_real_escape_string($mysqli, $_POST['ag
e']);
        $message = mysqli_real_escape_string($mysqli, $_POST['m
essage']);


                //updating the table
                $result = mysqli_query($mysqli, "UPDATE users S
ET name='$name',age='$age',message='$message' WHERE id=$id");

                //redirectig to the display page. In our case,
it is index.php
                header("Location: index.php");

}
?>

<?php
//getting id from url
$id = $_GET['id'];

//selecting data associated with this particular id
$result = mysqli_query($mysqli, "SELECT * FROM users WHERE i
d=$id");

while($res = mysqli_fetch_array($result))
{
        $name = $res['name'];
        $age = $res['age'];
        $message = $res['message'];
}
?>
<html>
<head>
```

```
        <title>Edit Data</title>
</head>

<body>
        <a href="index.php">Home</a>
        <br/><br/>

        <form name="form1" method="post" action="edit.php">
                <table border="0">
                        <tr>
                                <td>Name</td>
                                <td><input type="text" name="na
me" value="<?php echo $name;?>"></td>
                        </tr>
                        <tr>
                                <td>Age</td>
                                <td><input type="text" name="ag
e" value="<?php echo $age;?>"></td>
                        </tr>
                        <tr>
                                <td>Email</td>
                                <td><input type="text" name="me
ssage" value="<?php echo $message;?>"></td>
                        </tr>
                        <tr>
                                <td><input type="hidden" nam
e="id" value=<?php echo $_GET['id'];?>></td>
                                <td><input type="submit" nam
e="update" value="Update"></td>
                        </tr>
                </table>
        </form>
</body>
</html>
```

- First the database file is included once.
- Then we retrive the data from database and display in the form.
- Then the form value can be edited and updated.