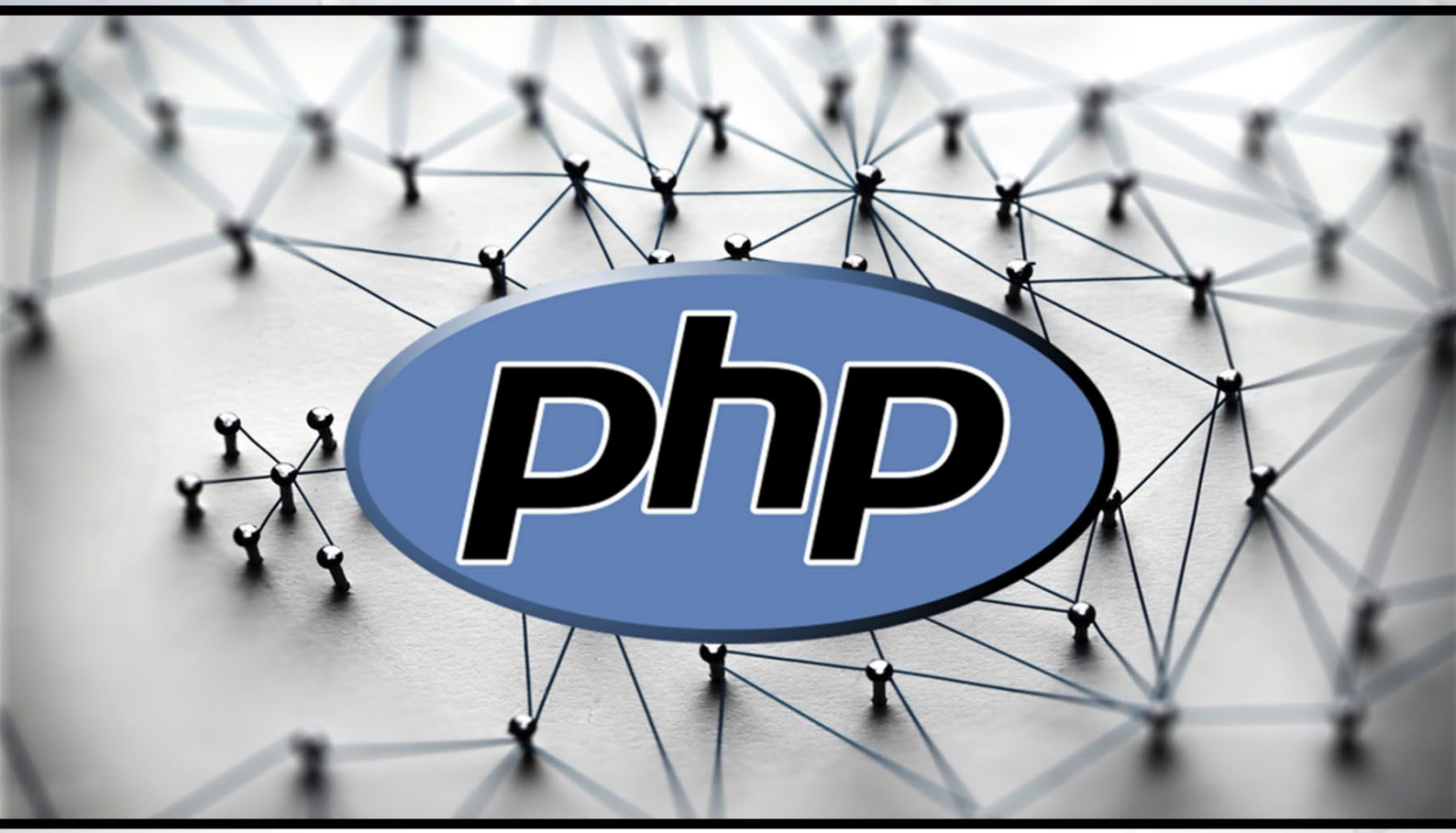
Beginners Guide



LEARN PHP

for Absolute Beginners

First Edition

PHPBOOTCAMP.COM

EASY TO PRACTISE CODE SAMPLES

About this Book

The PHP Hypertext Preprocessor (PHP) is a programming language that allows web developers to create dynamic content that interacts with databases. PHP is basically used for developing web-based software applications. This tutorial will help you understand the basics of PHP and how to put it in practice.

Audience

This tutorial has been designed to meet the requirements of all those readers who are keen to learn the basics of PHP.

Prerequisites

This book assumes you have no prior knowledge on Programming knowledge and assume you are at a beginner level.

How to use this Book

This book contains PHP Language Basics, Exercises and Examples which are part of the PHP Bootcamp Program. This bootcamp has helped many students to become PHP Full Stack Web Developer in Just 30 days.

>>>Check out more about this program here...

Copyright & Disclaimer

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. PHPBootcamp.com provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@phpbootcamp.com.

Found Typos & Broken Link

If you found any typos and broken links in the course or book, then please let me know using this email address.

contact@phpbootcamp.com

Support

You can reach me for technical discussions and other support related queries from here.

support@phpbootcamp.com

Thanks for your support!

Free Courses

	Learn HTML in 1 Hour
	Learn CSS in 1 Hour
	Learn JAVASCRIPT in 1 Hour

Become PHP Full Stack Web Developer in Just 30 Days

Table of Contents

1. PHP BASICS

1 PHP Basics

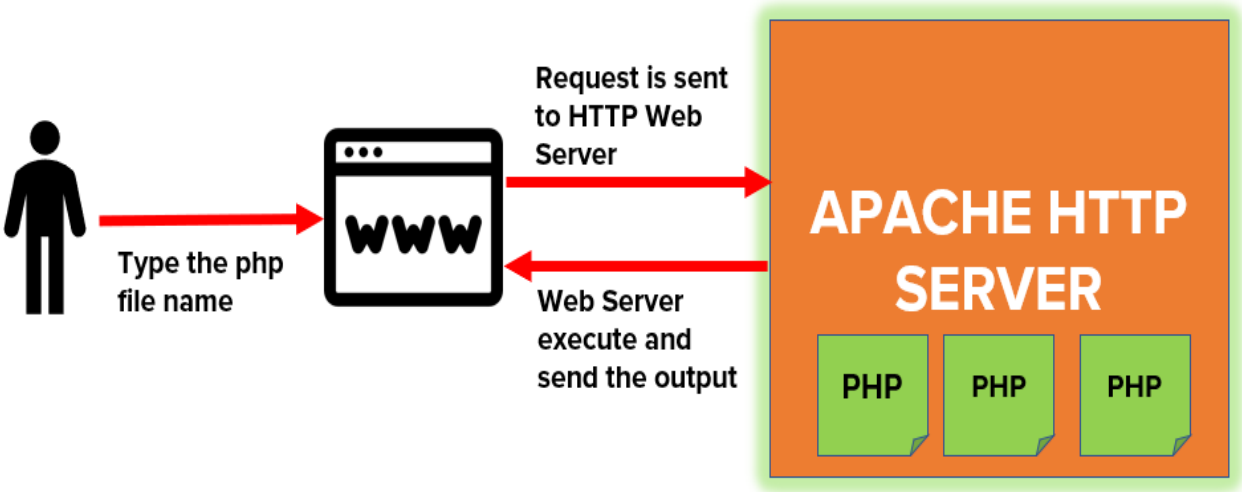
1.1 Installation of PHP

Installation of PHP

To run PHP on your local system you need to install HTTP Web Server.

PHP is not like Java or C where you can install the libraries and run PHP on command prompt.

PHP is a server side programming and used in web. It is designed to run from Web Server like IIS or Apache HTTP Server.



How PHP Works?

When user type the filename.php path in the browser url:

Browser will go to server where HTTP servers are running.

Web Server will listen to the request.

It will execute the PHP code on the server.

Take the output generated by the PHP Server.

Send it back to the browser.

Browser displays the output.

Things to Note:

You need a Web Server to run PHP code.

Server executes php on the server and returns the output of the code.

You won't be able to see PHP code on the browser.

Browser pass the user data from browser to server and fetch the data back from server.

Inserting into Database, Sending Email, Checking the login credentials is all done by php program at the server.

No one can see your php code.

Why need a Web Server?

Try this exercise.

Create a sample file name index.php

Add the following PHP code.

<?php echo "Hello PHP!"; ?>

Open the index.php file in the browser.

Watch how browser just show the PHP code.

Browser does not understand PHP nor it understand how to interpret it.

That's the job of a WebServer.

Webserver interpret the PHP code and send the output to the browser.

That is the reason you need a Web Server to execute PHP code.

Understand WAMP Server

You can choose install the following software individually:

Apache – https://httpd.apache.org/

PHP – http://php.net/software.php

MySQL – https://www.mysql.com/

phpMyAdmin – https://www.phpmyadmin.net/

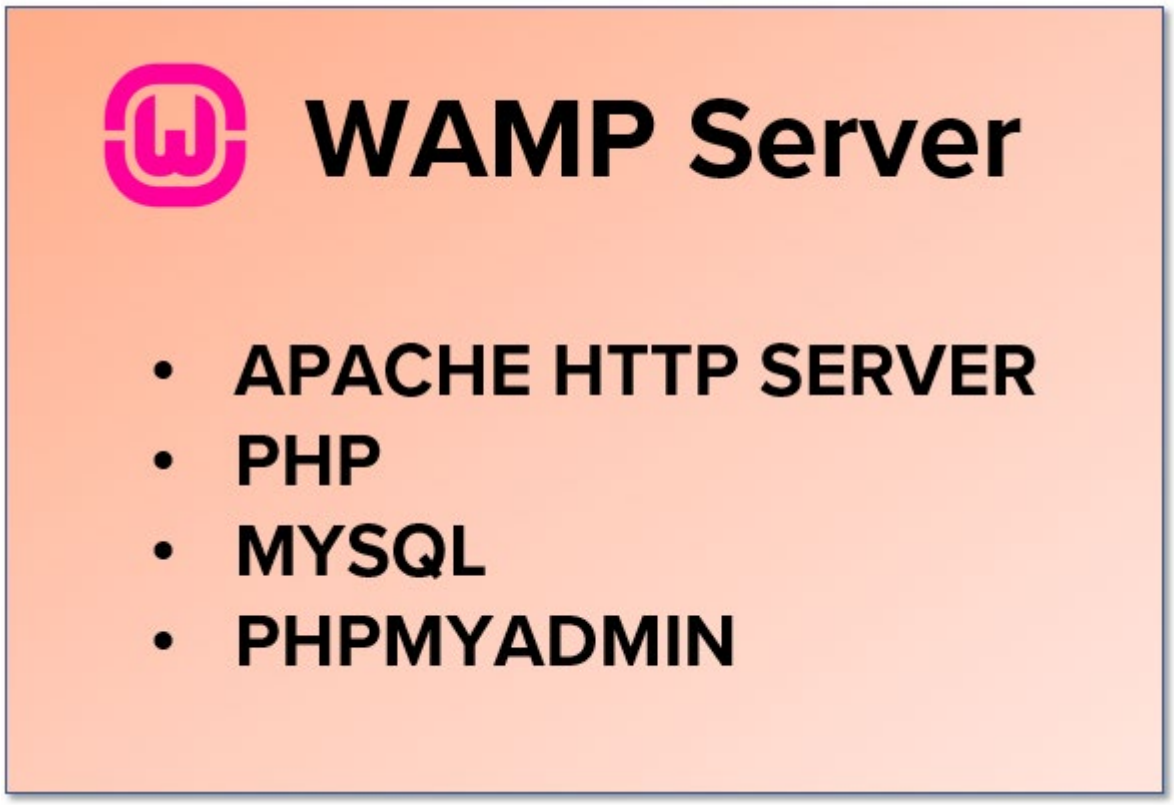
Apache is the Web Server

PHP is the libraries that helps to run the php code

MySQL is the Database to store the data.

phpMyAdmin is the admin client to access your database.

Instead of installing all the software there is a package which combines all the software together and give a Web Environment to build web applications.



That is WAMP. – Windows Web Development Environment

WampServer is a Windows web development environment. It allows you to create web applications with Apache2, PHP and a MySQL database. Alongside, PhpMyAdmin allows you to manage easily your databases.

For Mac use this Software – https://www.mamp.info/en/

When you install WAMP you get:

Apache2

PHP

MySQL

phpMyAdmin

All 4 software are installed on your local machine which provides an environment to develop web application.

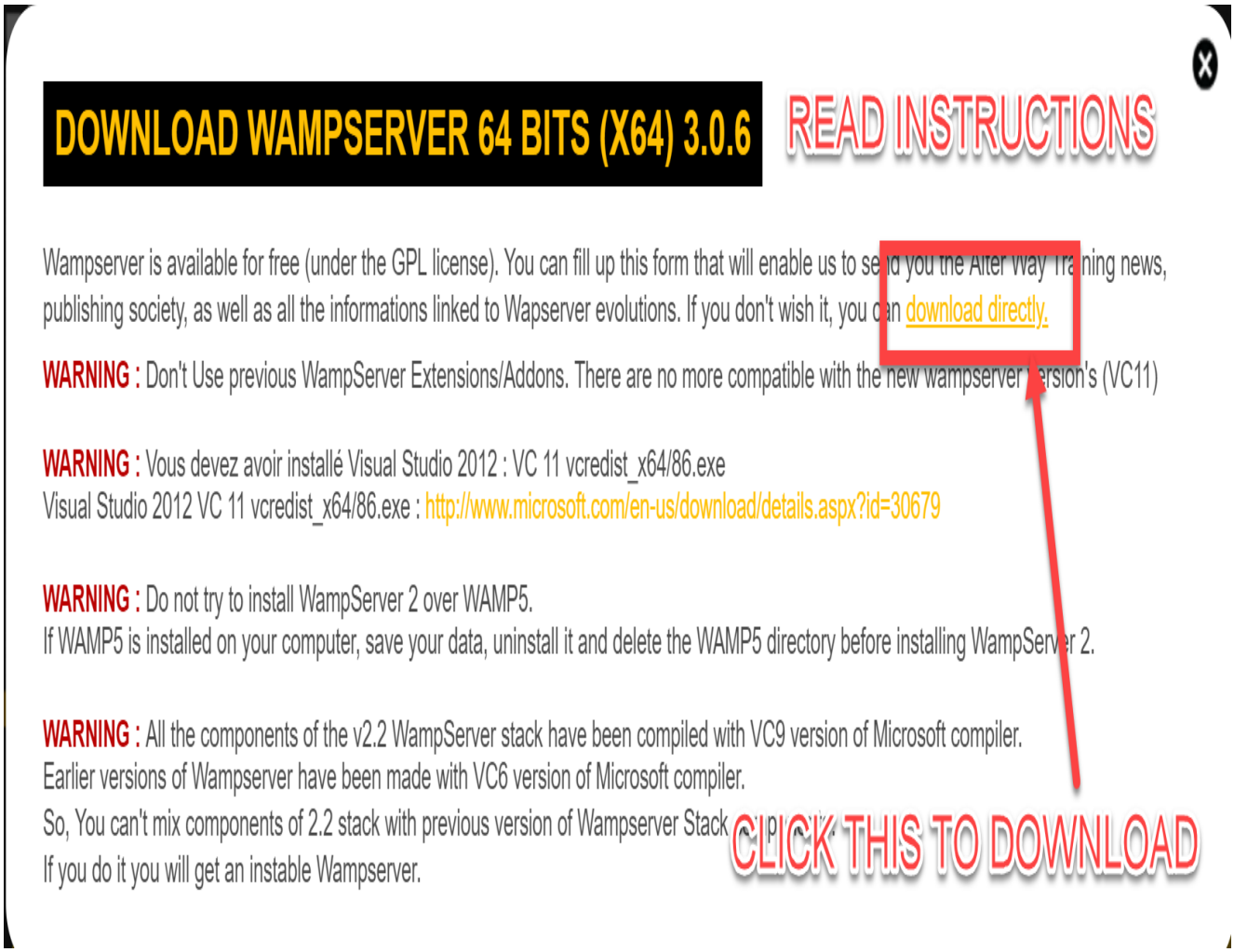
Visit & Download WAMP Server

Step 1: Visit WAMP Site – http://www.wampserver.com/en/

Step 2: Click on Download from Menu

Step 3: Install **64-bit** OR **32- bit** based on your Windows Machine.

Step 4: Instructions Page. Click on the Download Link to download the WAMP Server.



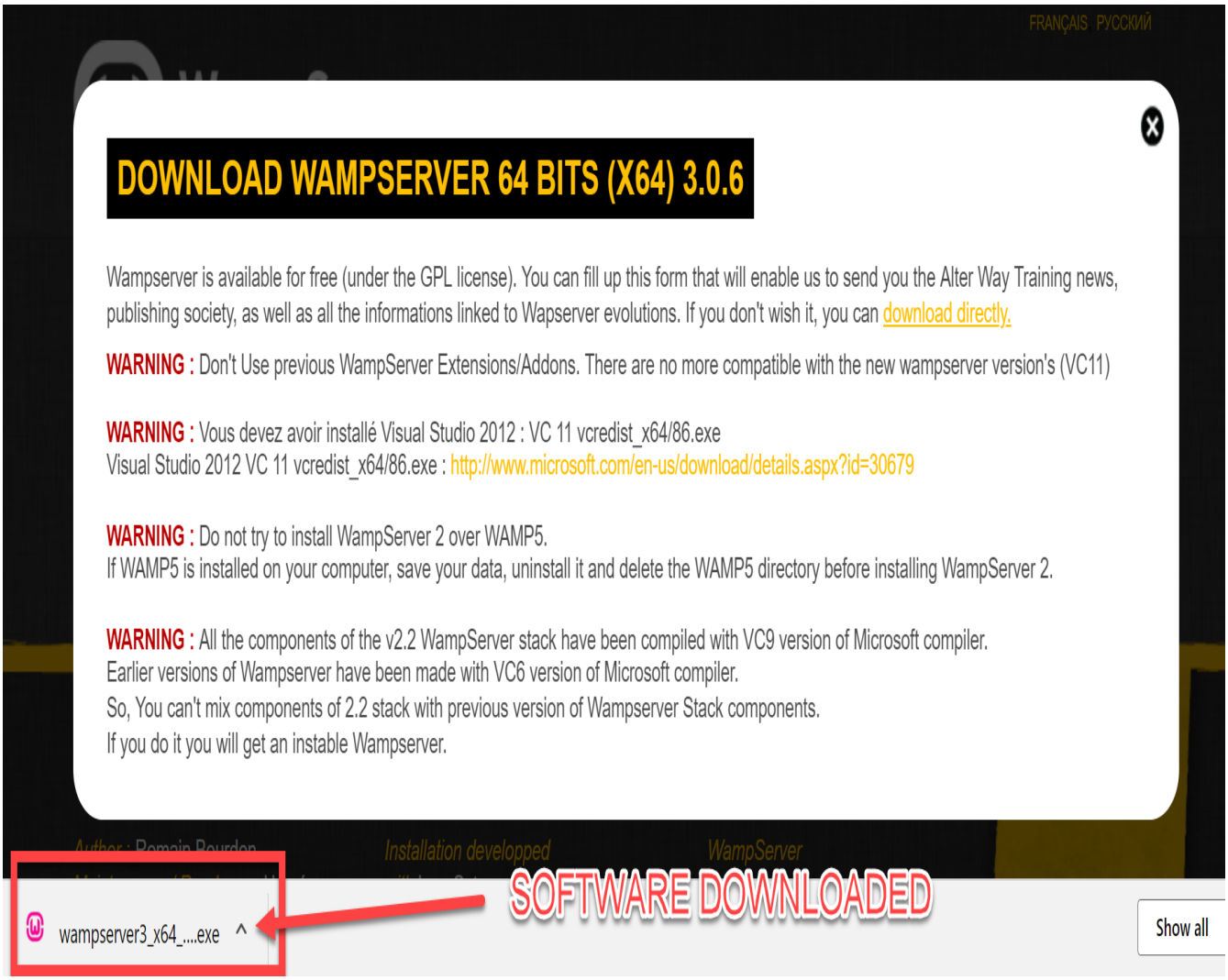
Step 5: Install the supported Libraries.

WARNING : Vous devez avoir installé Visual Studio 2012 : VC 11 vcredist_x64/86.exe
Visual Studio 2012 VC 11 vcredist_x64/86.exe : http://www.microsoft.com/en-us/download/details.aspx?id=30679

Step 6: It will redirect to sourceforge.net and download it from there.

File Name should be – **wampserver3.1.3_x64.exe**

Step 7: WAMP Server is downloaded.

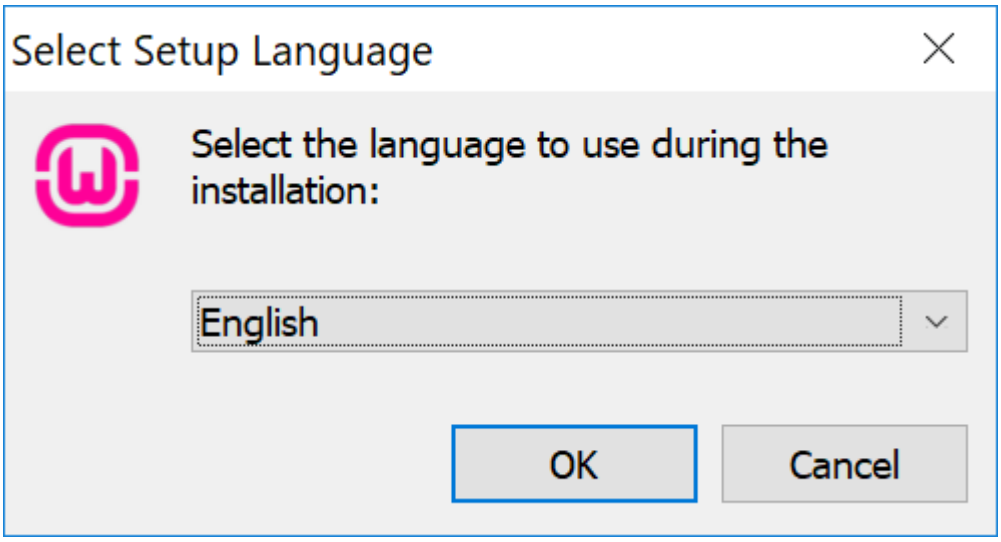


Install WAMP

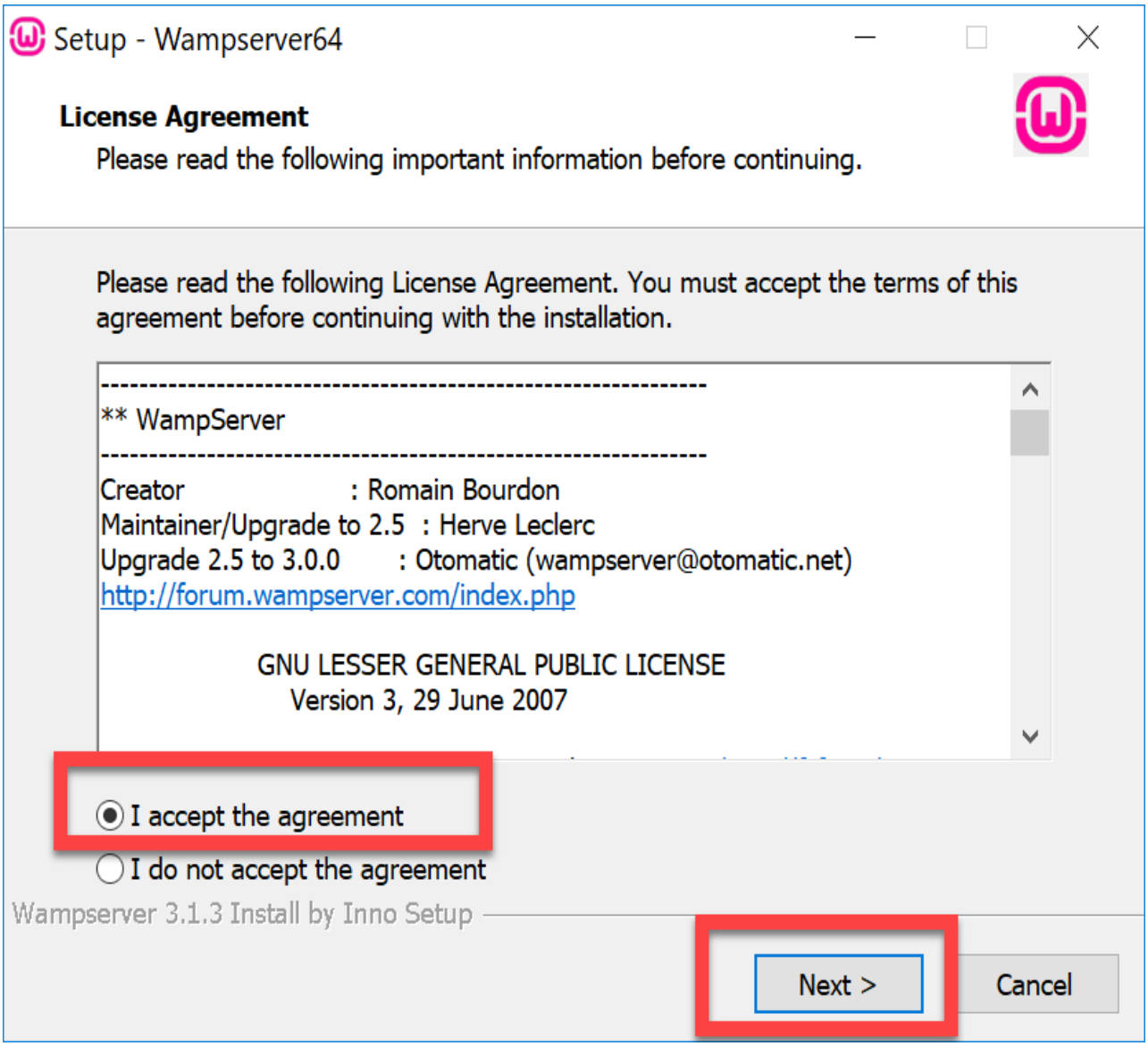
Step 1: Locate the downloaded Software “wampserver3.1.3_x64”

It will have this icon. Even you can find this icon in the system tray once the software is installed.

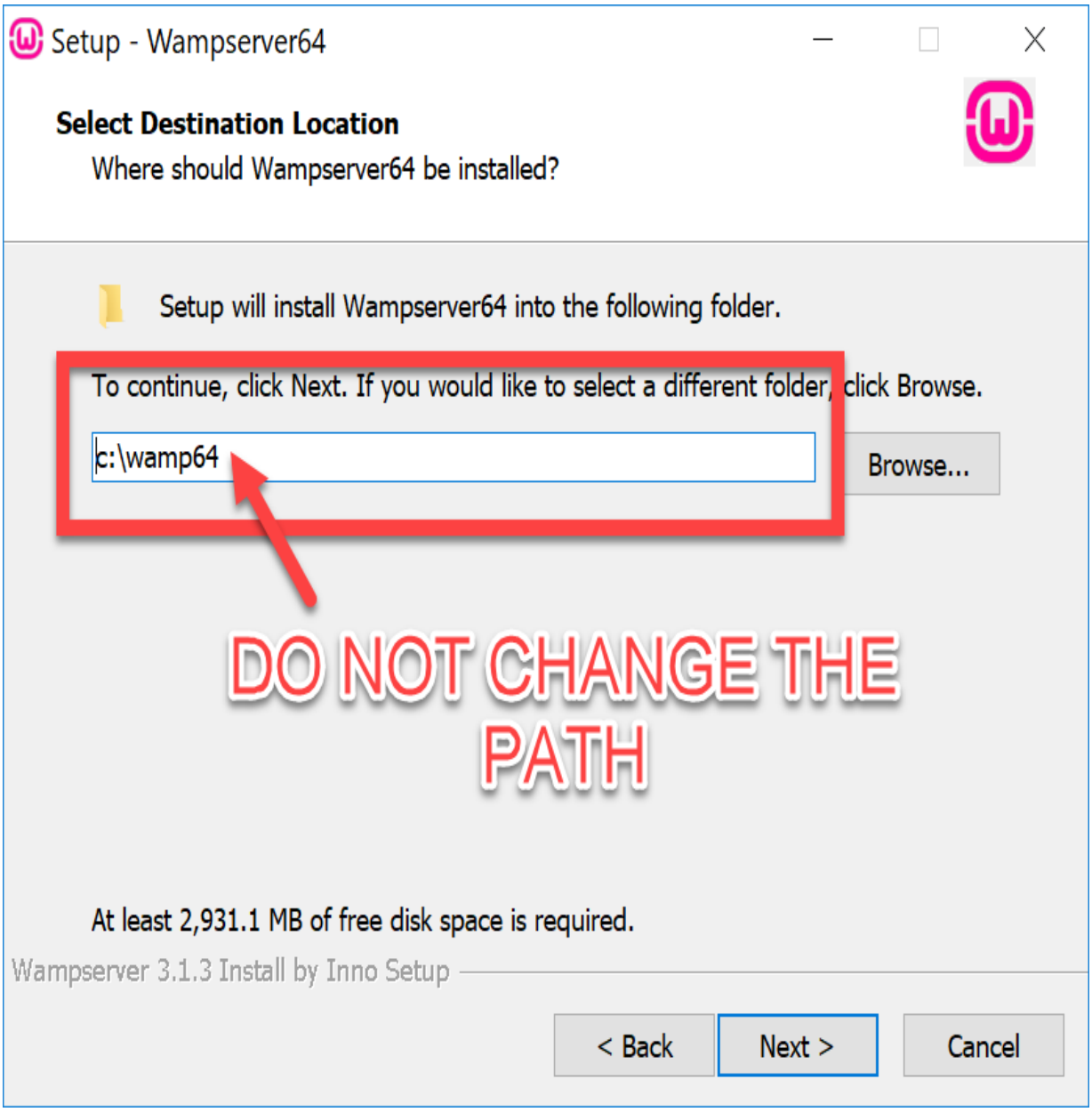
Step 2: Double Click to Start the Installation process.



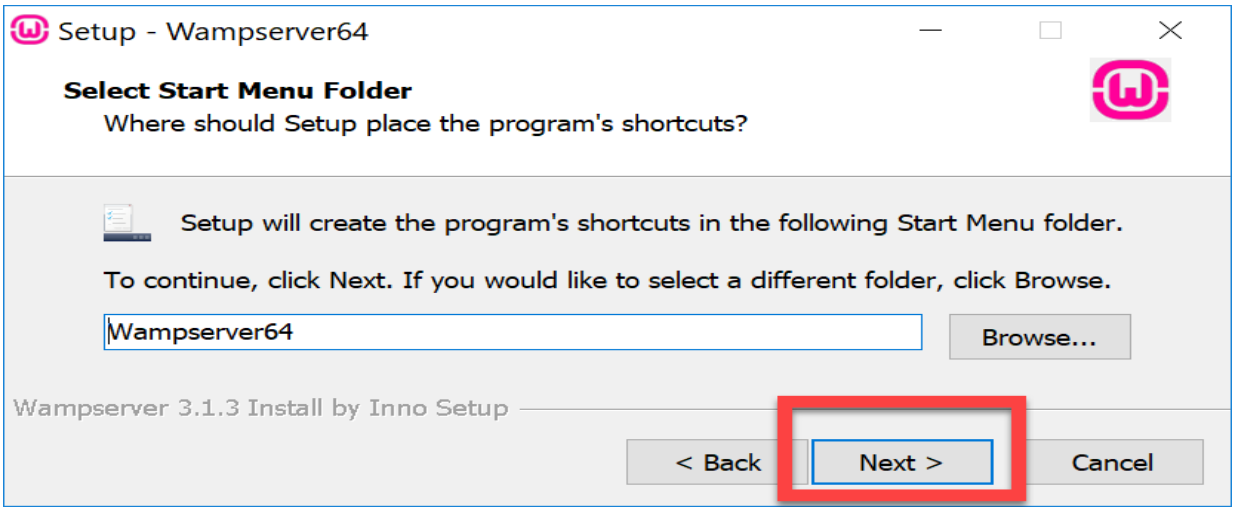
Step 3: Accept the Agreement. and Click Next.



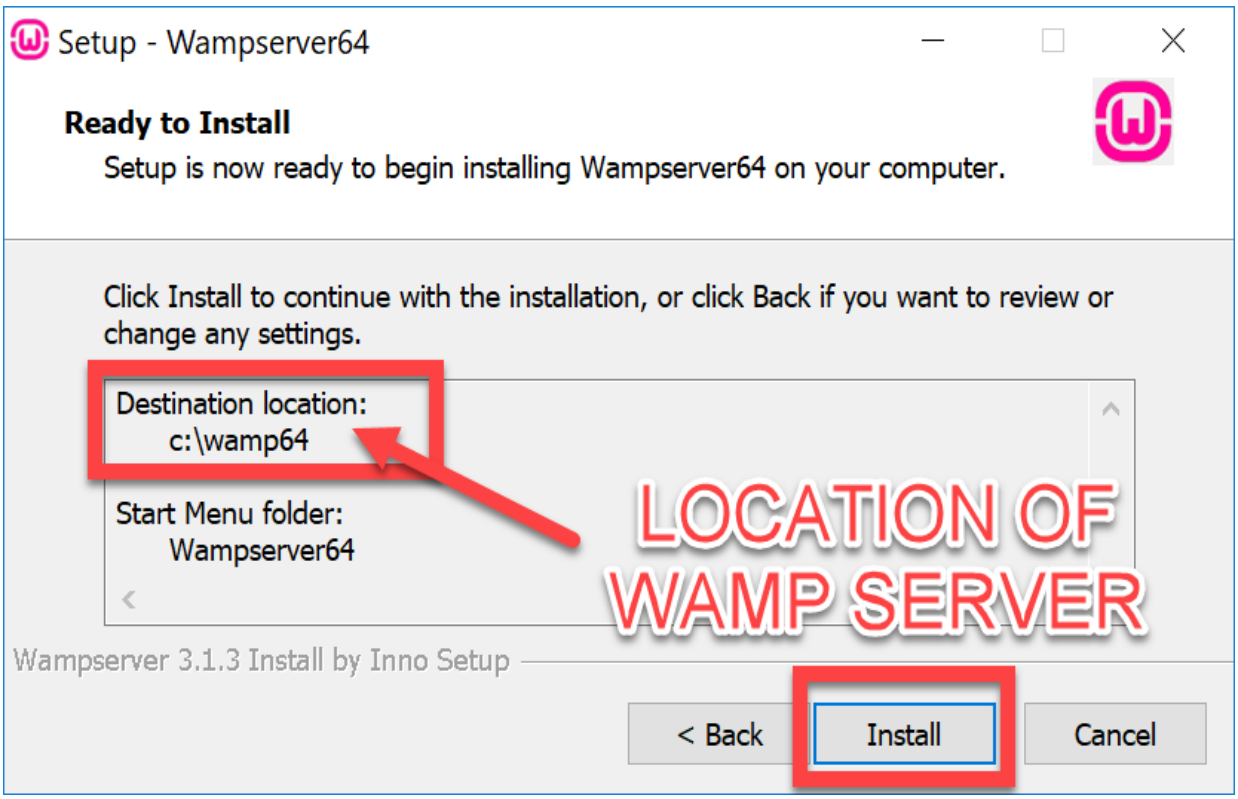
Step 4: Do not change the default path. It will be installed in C:\WAMP64



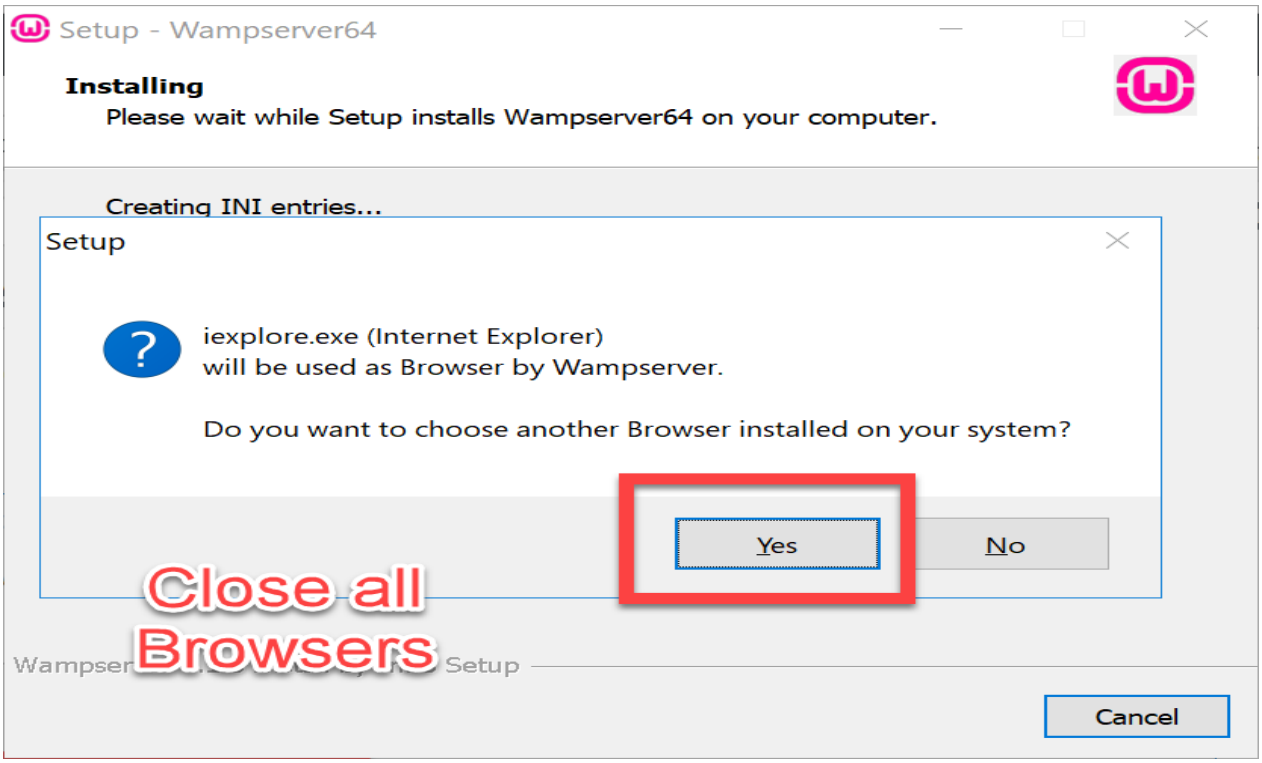
Step 5: Click Next



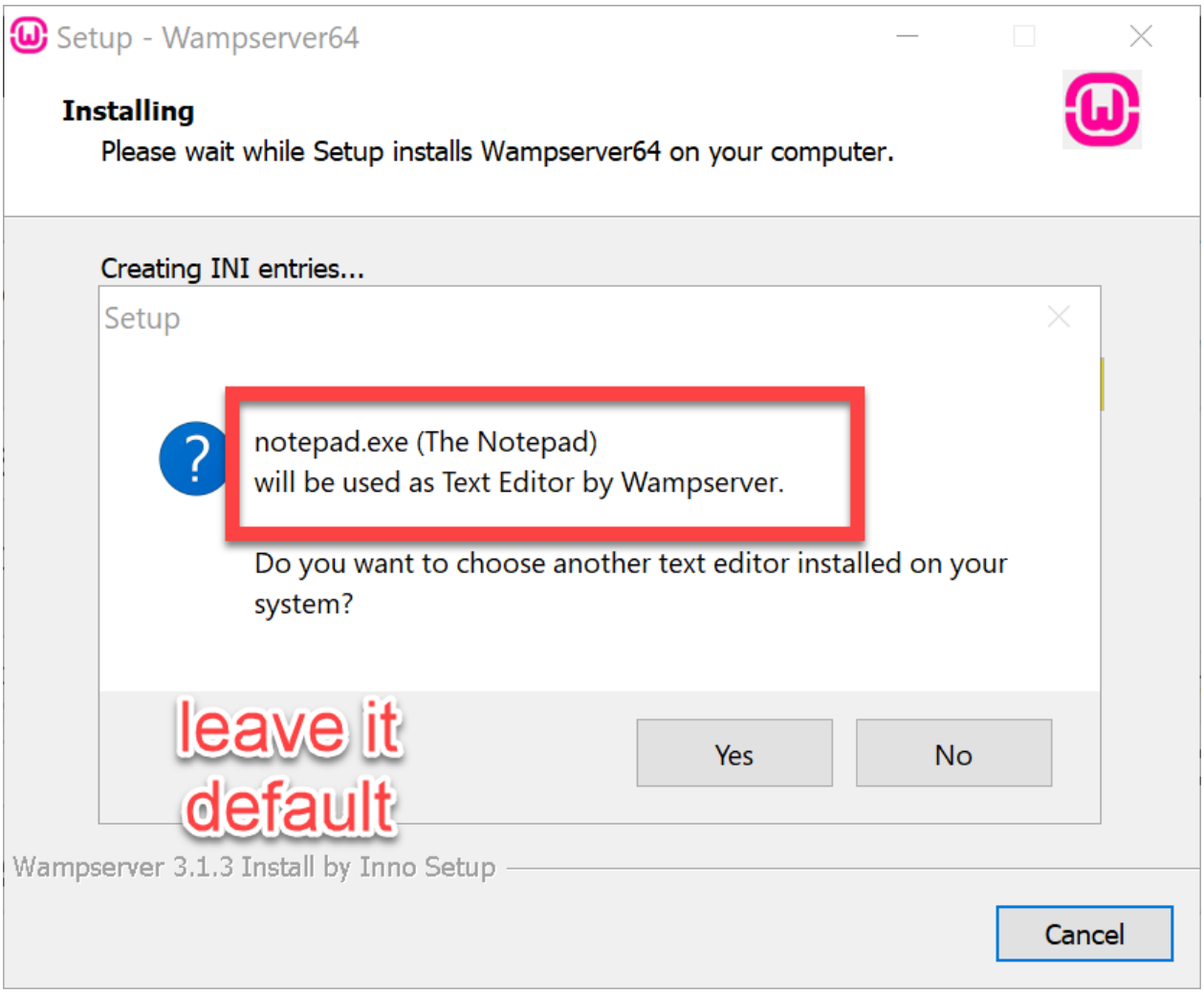
Step 6: Verify the path where the software is installed and click on Install.

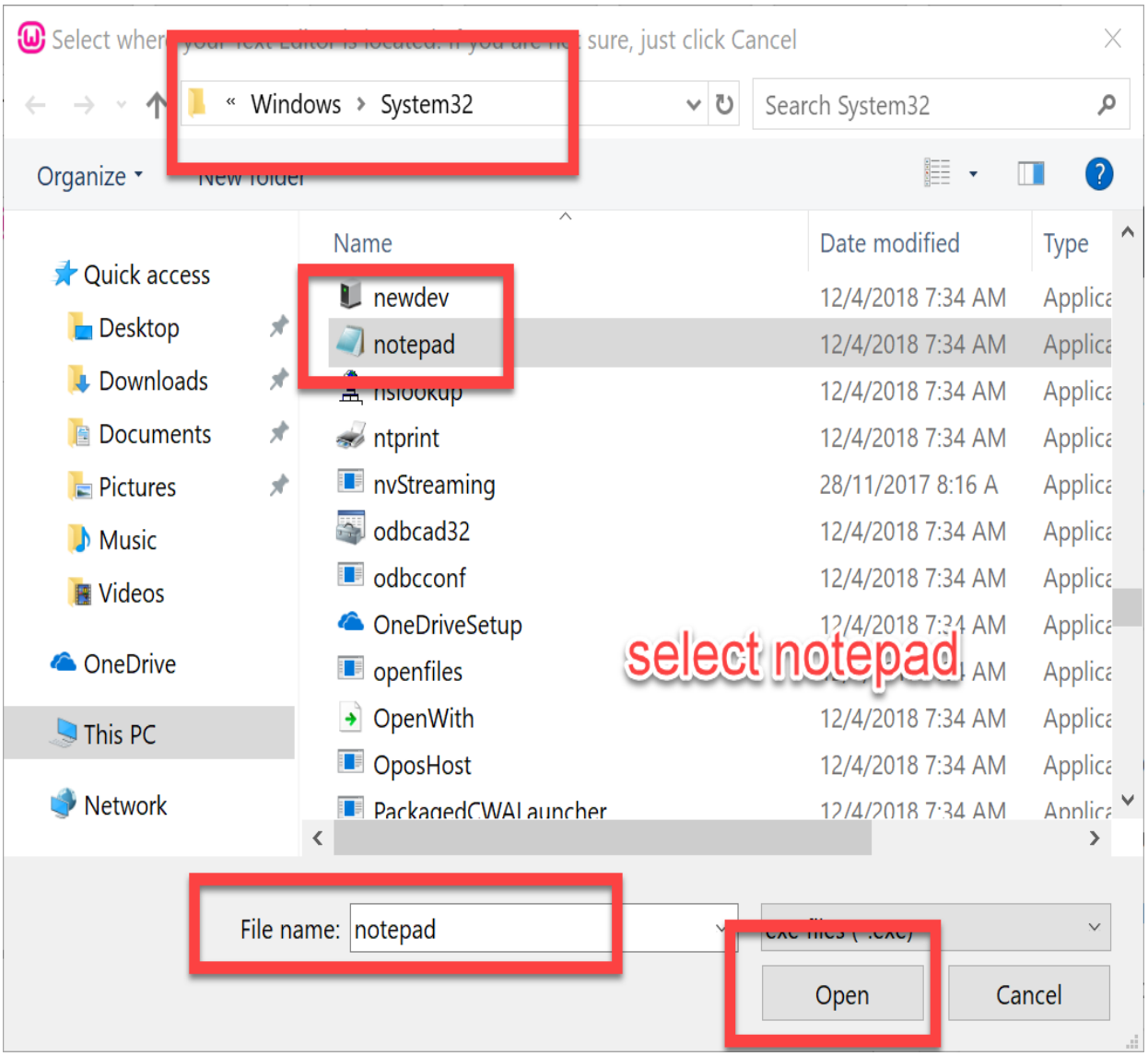


Step 7: Close all the Browsers and Select the Default Browser.

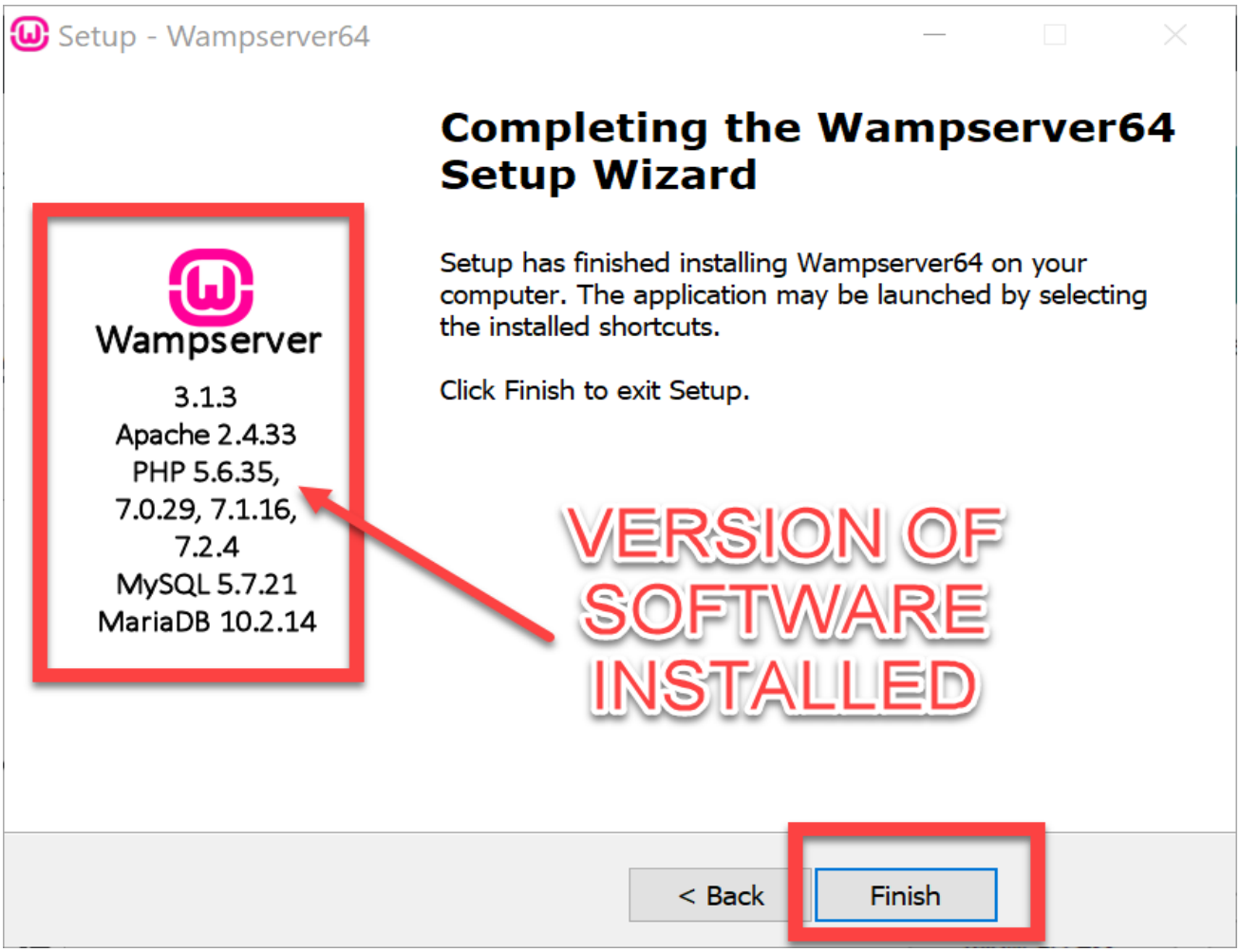


Step 7: Select the Notepad


Setup - Wampserver64
Installing
Please wait while Setup installs Wampserver64 on your computer.
Creating INI entries...
Setup
notepad.exe (The Notepad)
will be used as Text Editor by Wampserver.
Do you want to choose another text editor installed on your system?
leave it default
Yes
No
Wampserver 3.1.3 Install by Inno Setup
Cancel



Step 8: Click Finish and Note the Software Versions.



Install Microsoft VC Checker

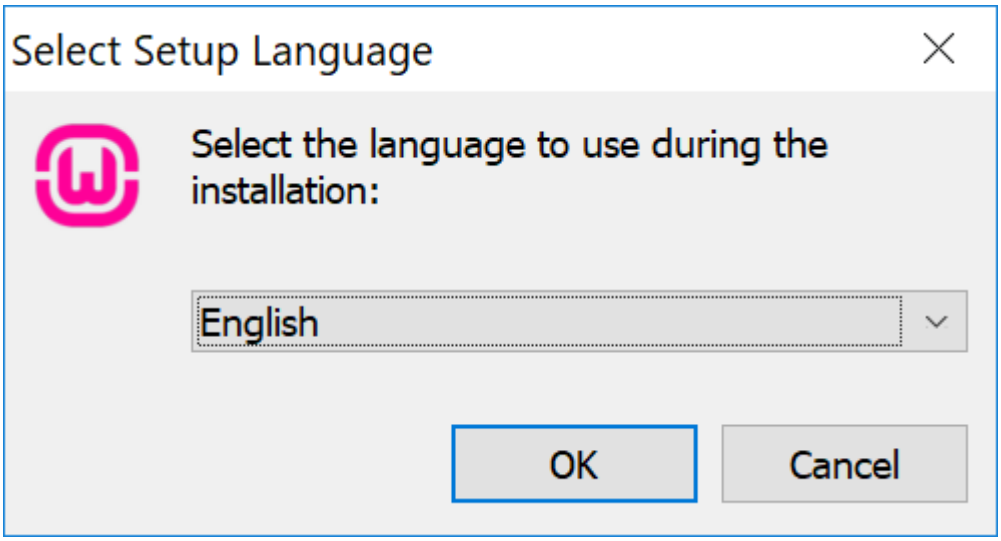
Next Step is to check if the Microsoft VC++ Package is missing from windows system.

Step 1: Download the VC Checker Software.

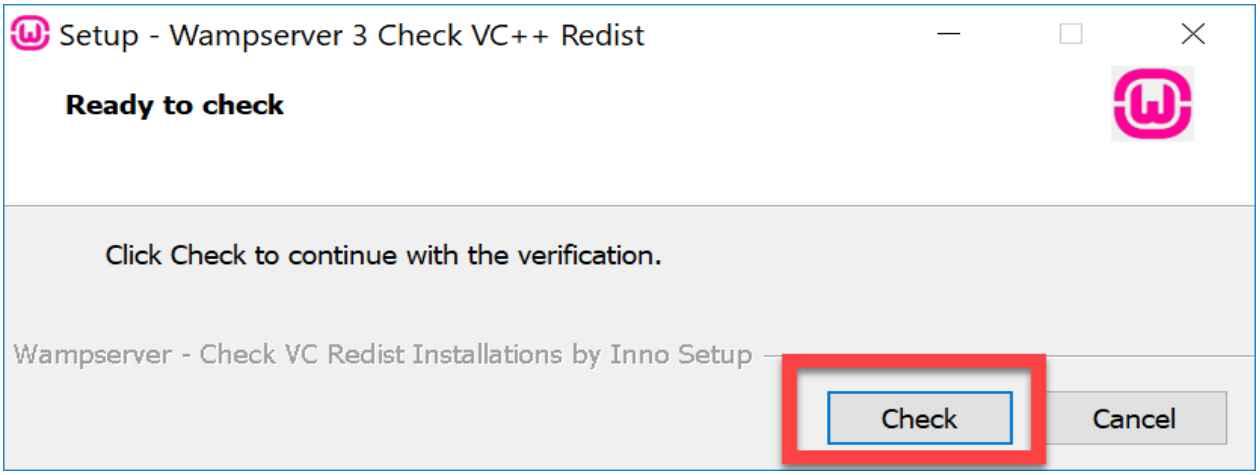
http://wampserver.aviatechno.net/files/tools/check_vcredist.exe

Step 2: It will download the **check_vcredist.exe** software

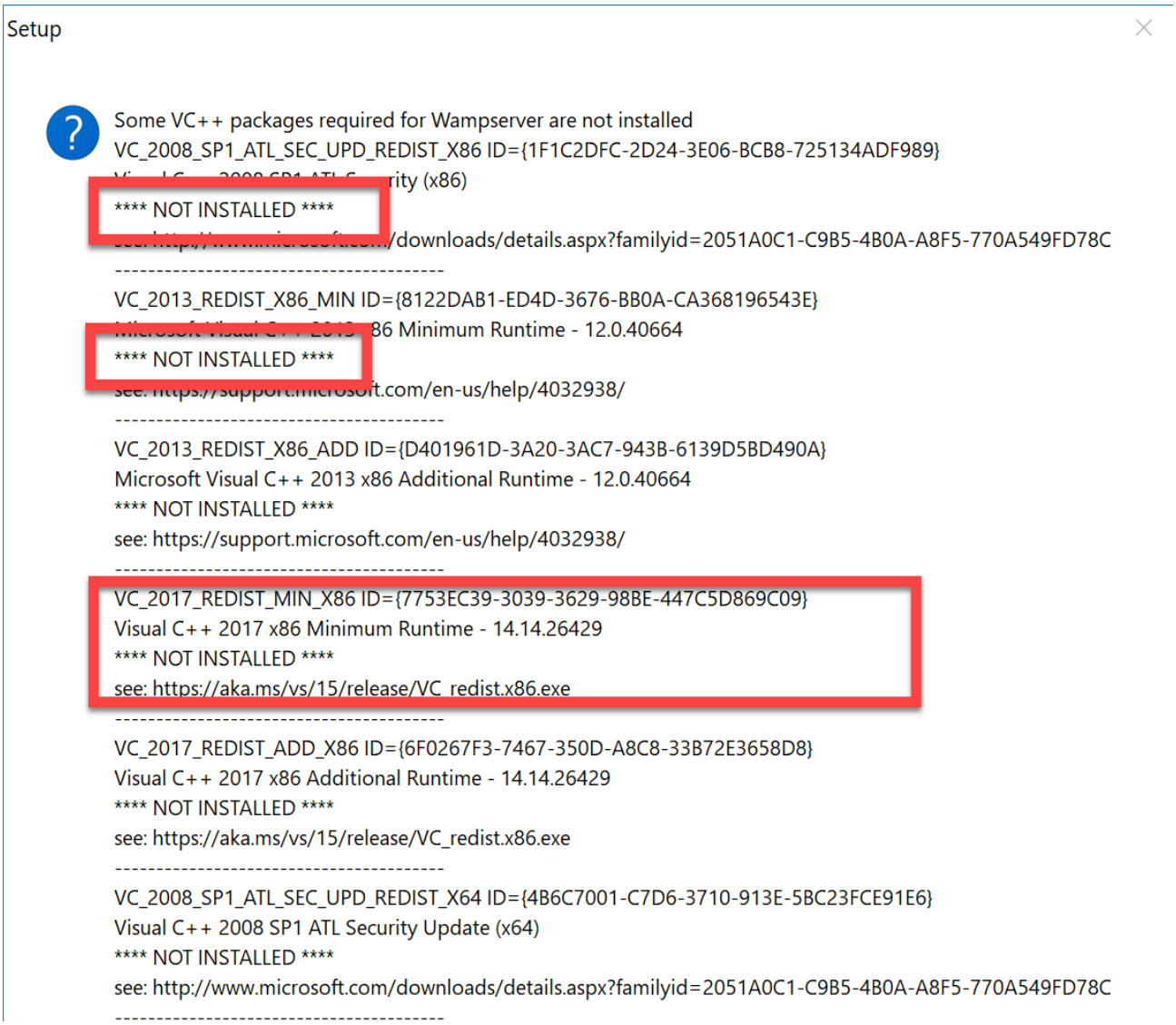
Step 3: Install **check_vcredist.exe** software.



Step 4: Click on Check



Step 5: Install the VC and then Run Again to make sure you all the supported Libraries.



Download from here:

https://support.microsoft.com/en-us/help/2977003/the-latest-supported-visual-c-downloads

Verify WAMP

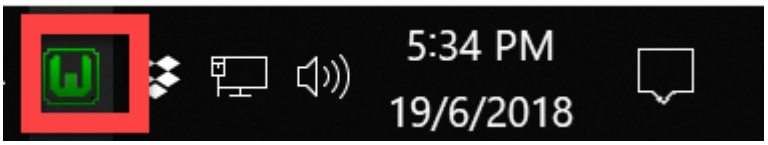
WAMP Server is installed in “C:\wamp64” folder.

Step 1: Run the WAMP Server

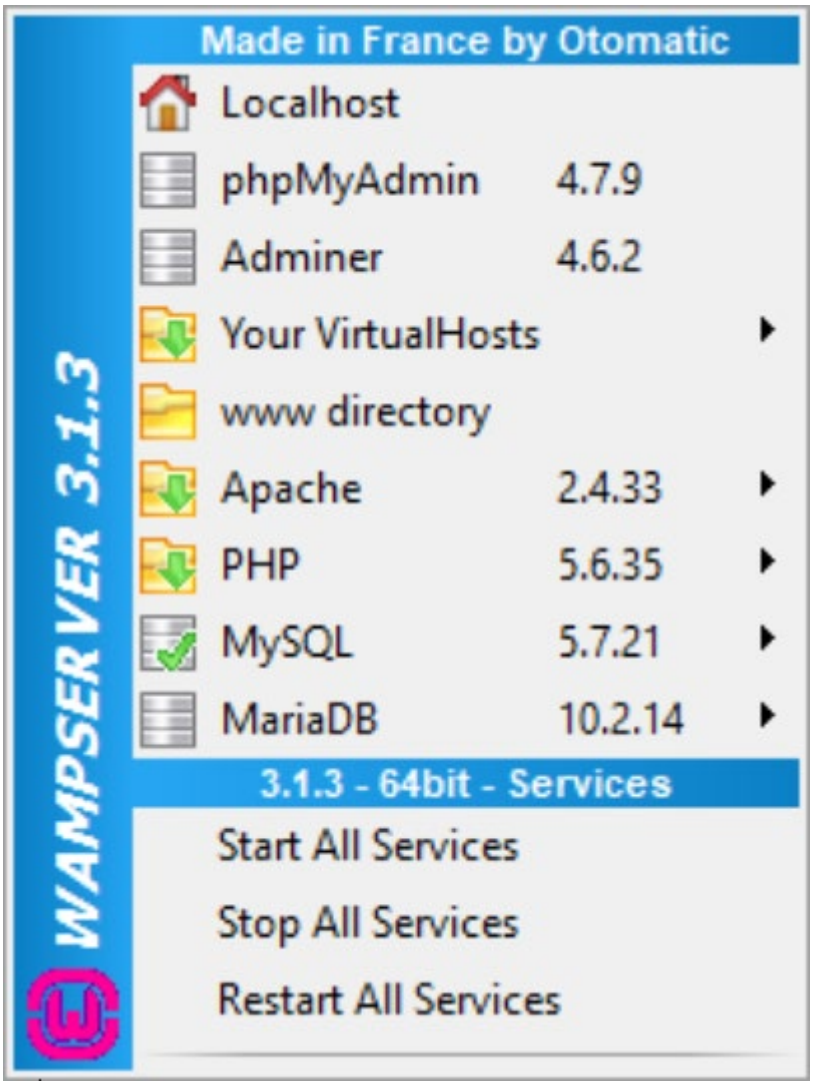
Run the file: C:\wamp64\wampmanager.exe

Goto C:\wamp64 folder and run the
WAMP Server – wampmanager.exe

See the Server running with GREEN color icon in the system tray.



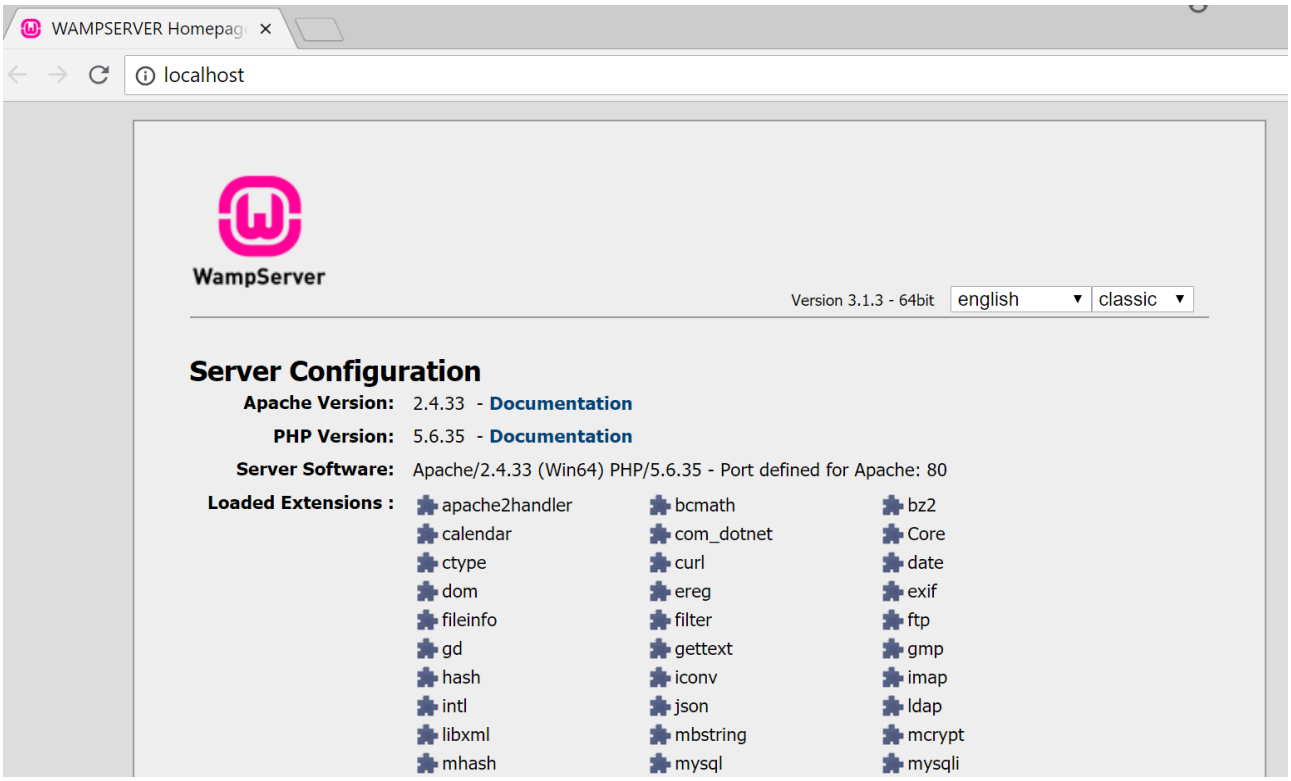
Click on the ICON and you should see this menu.



Step 2: Click on Localhost on the Menu

Browser will open the URL – localhost

You should see this page.



This confirms that WAMP server is installed properly.

Step 3: Recheck if you have the following things working:

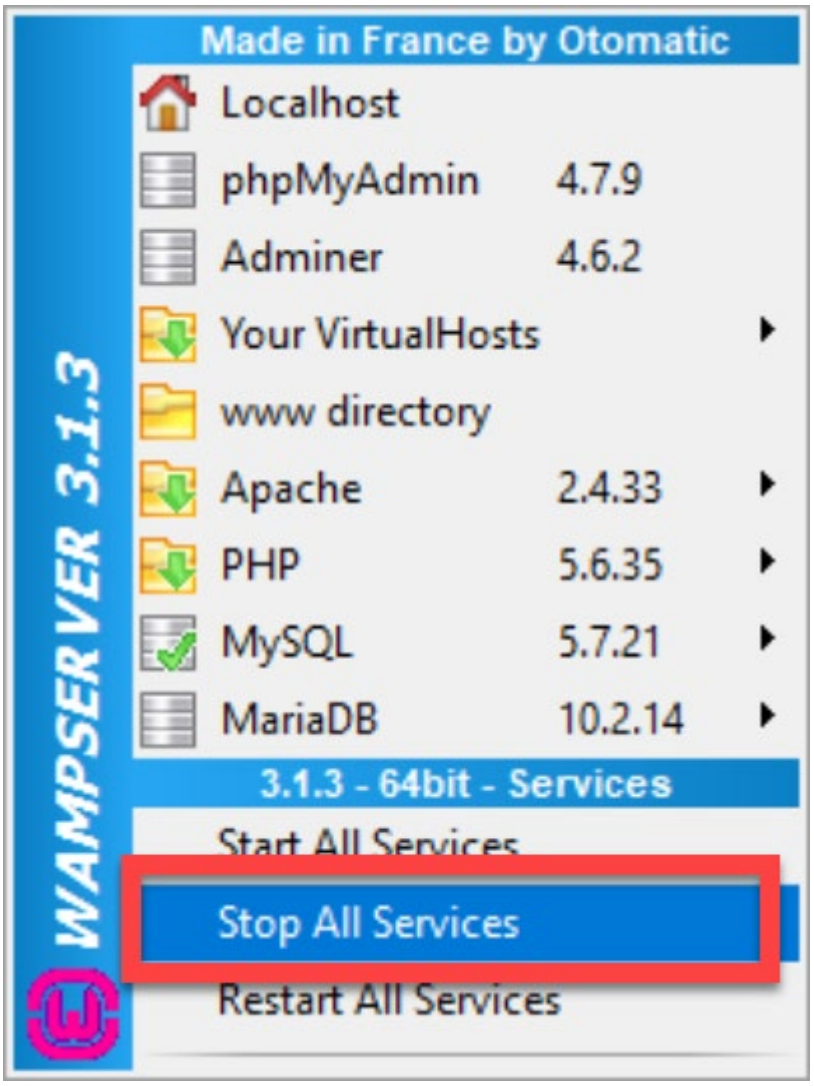
WAMP Icon in Green color in system tray

Click on Localhost in the WAMP menu

WAMP displays the localhost page correctly.

Stop WAMP

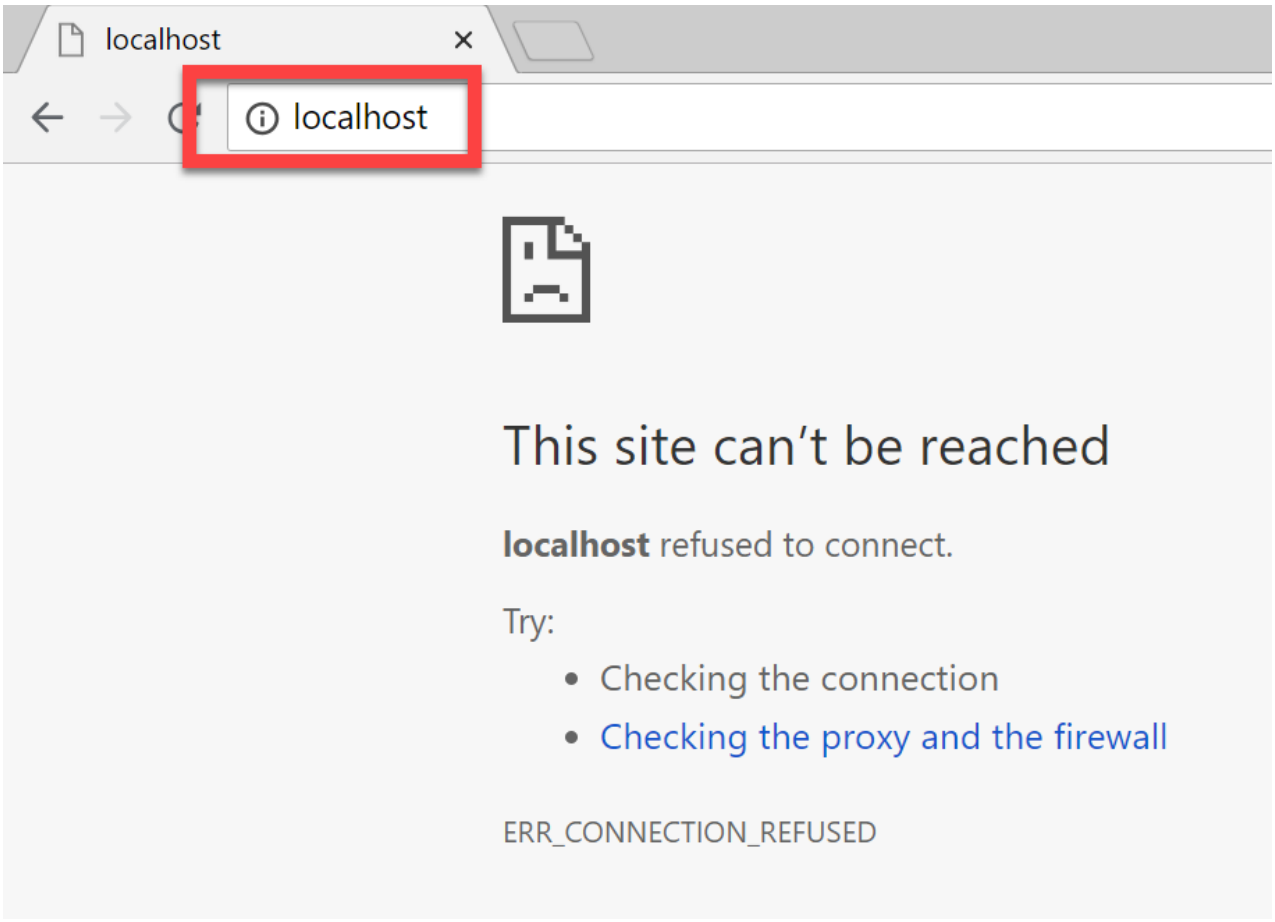
Step 1: Click on the System Tray WAMP Icon and Select Stop Services.



Step 2: WAMP Icon Turns to RED.

Step 3: Type “localhost” in the Browser URL

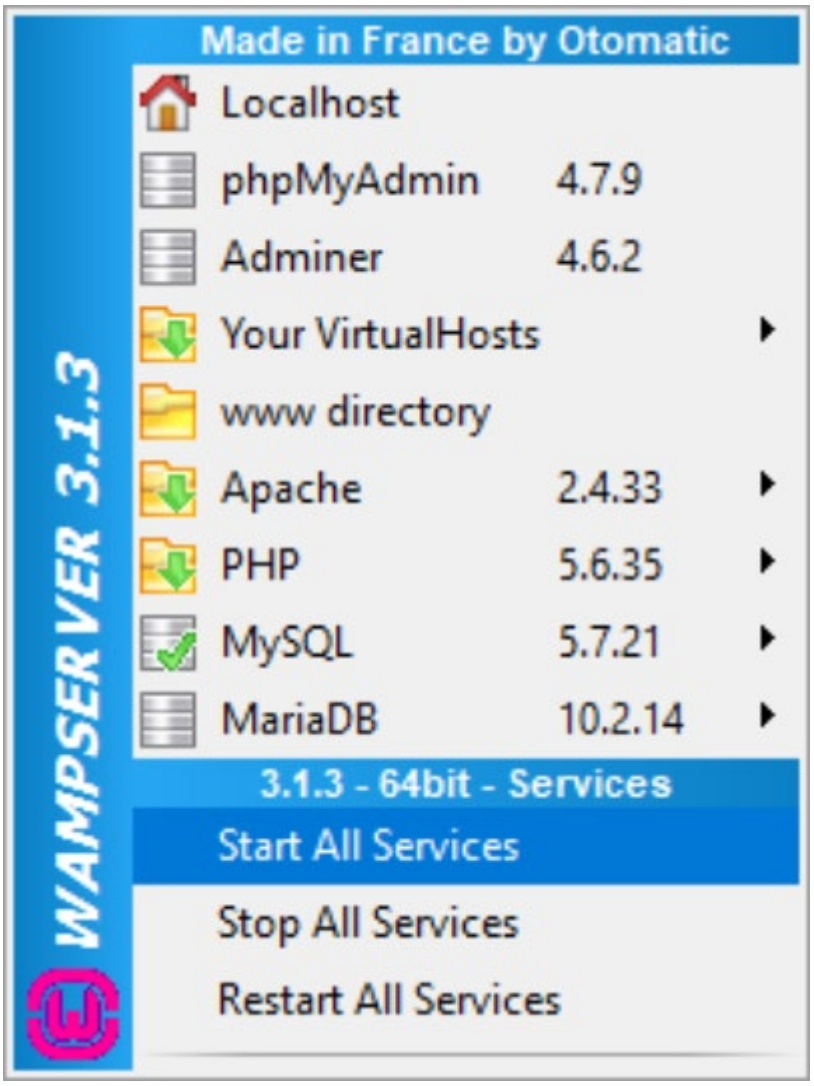
This should show error page in chrome.



This confirms that WAMP Server is Stopped.

Start WAMP

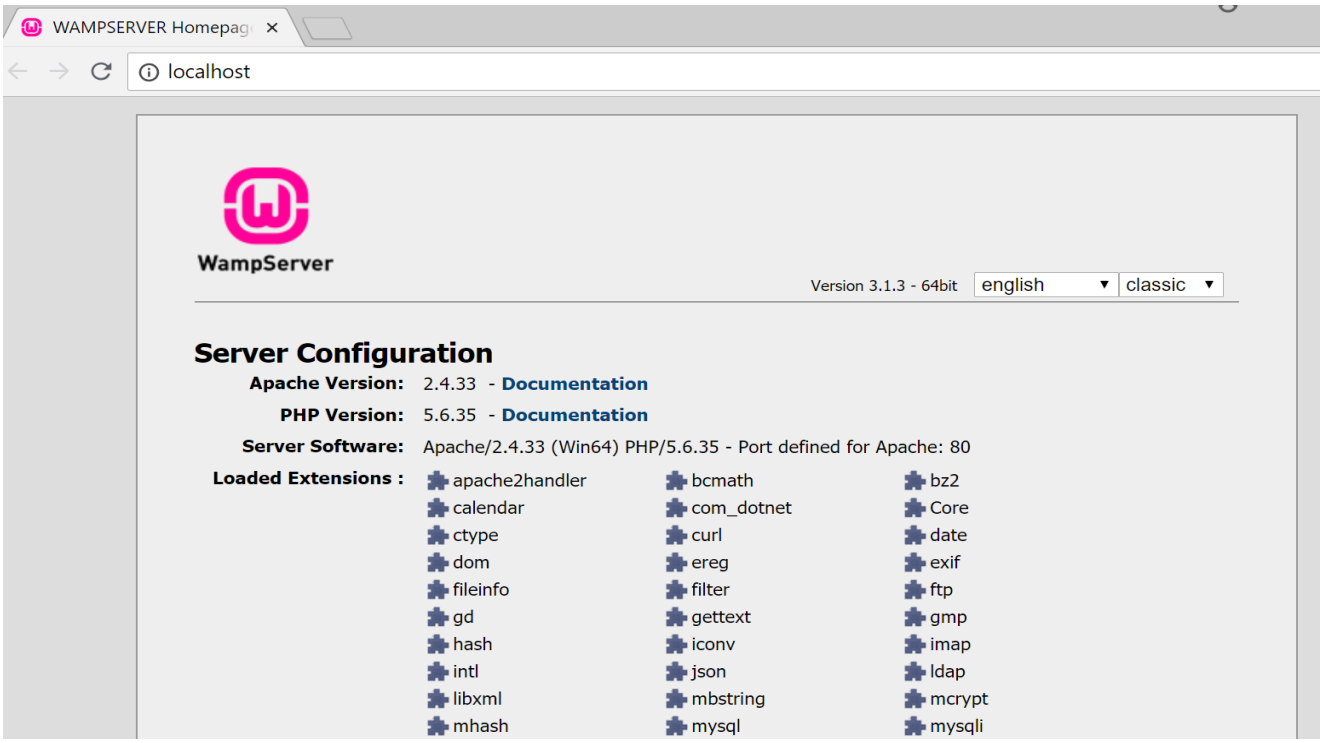
Step 1: Click on the System Tray WAMP Icon and Select Star Services.



Step 2: WAMP Icon Turns to GREEN

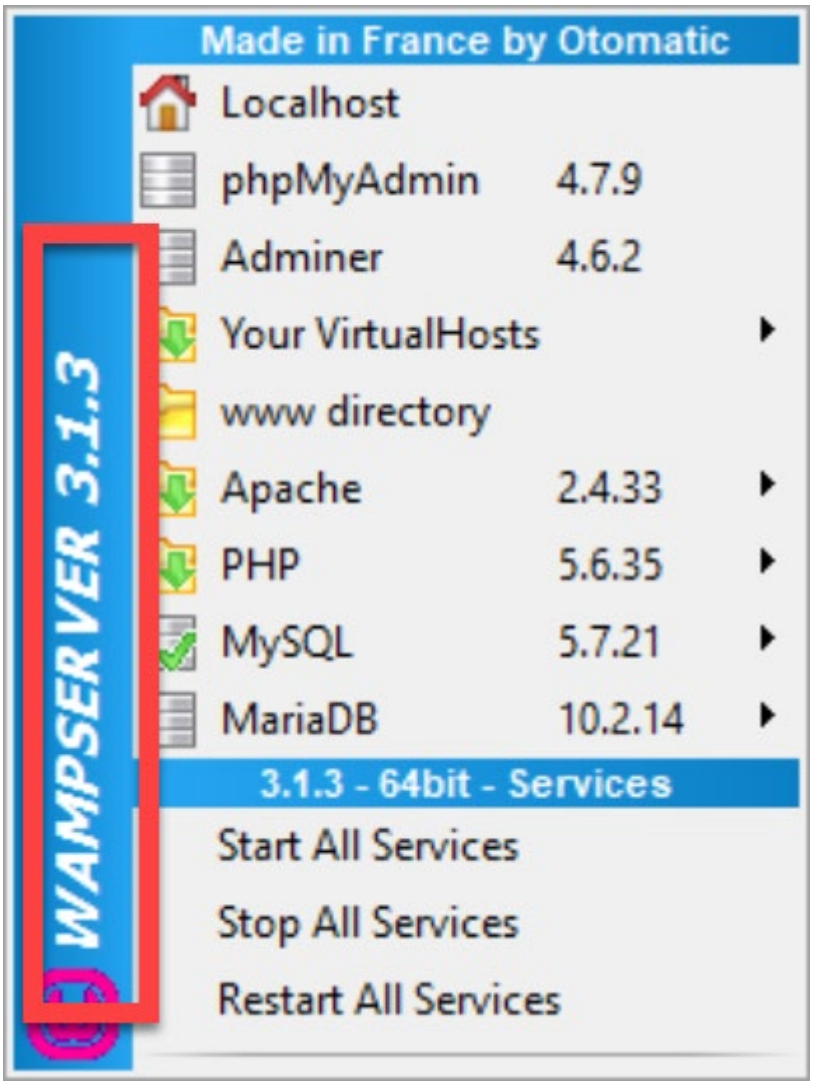
Step 3: Type “localhost” in the Browser URL

This should show the home page for localhost



Check WAMP Version

Open the WAMP Menu from System Tray and you can find the WAMP Version.



Verify the WAMP Folder

You should have this two folders in C:\

C:\wamp64

C:\wamp64\www

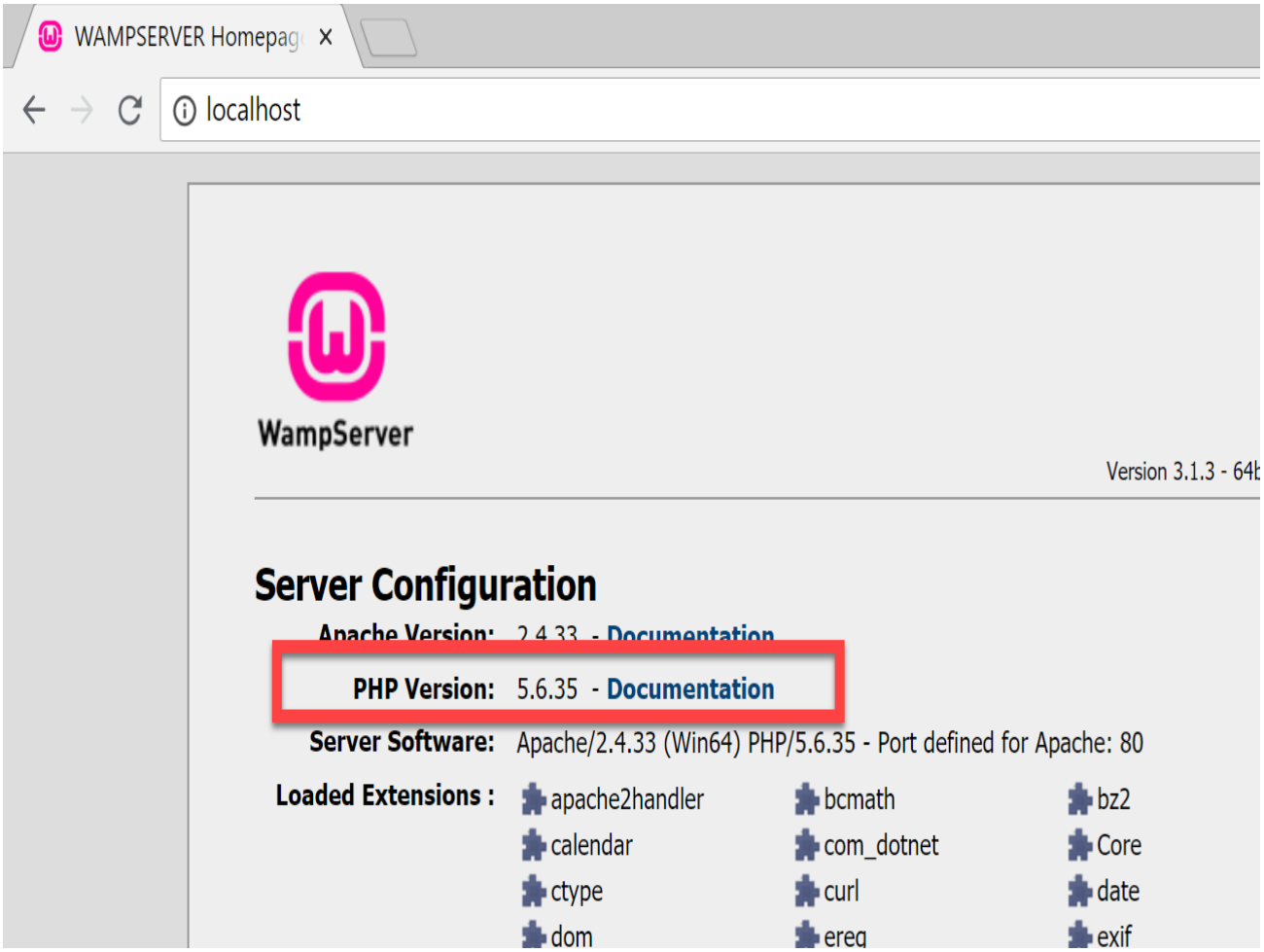
WAMP Log file is inside this folder:

C:\wamp64\logs

Check PHP Version

You can check the PHP version by opening the localhost when WAMP server is running.

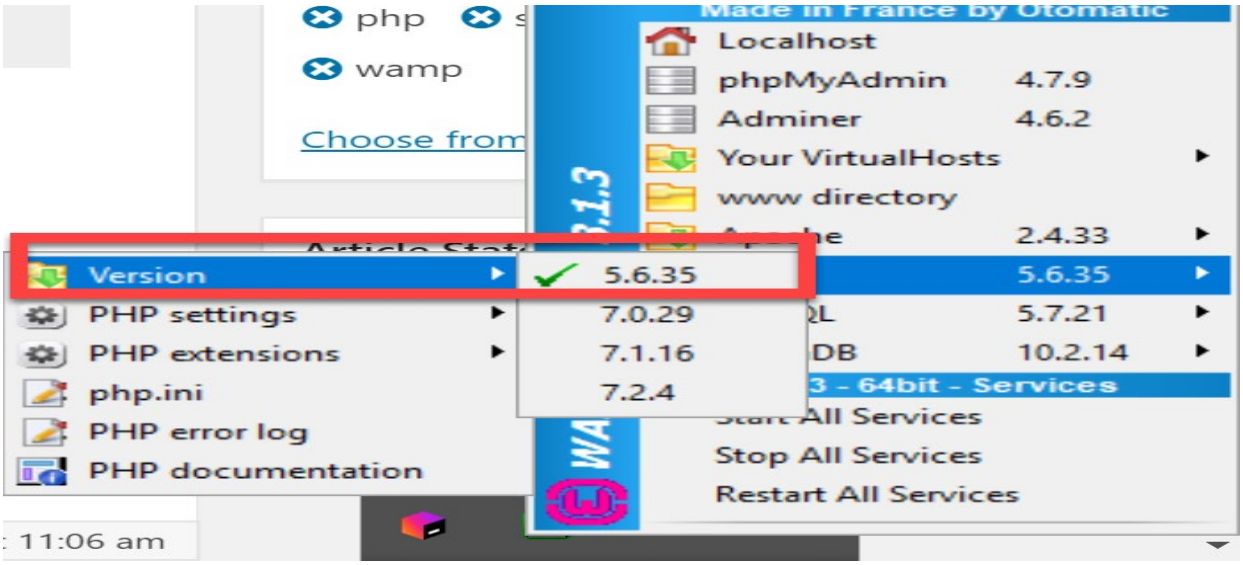
It will show the PHP Version.



Change the PHP Version

Open the WAMP Menu -> PHP -> Versions

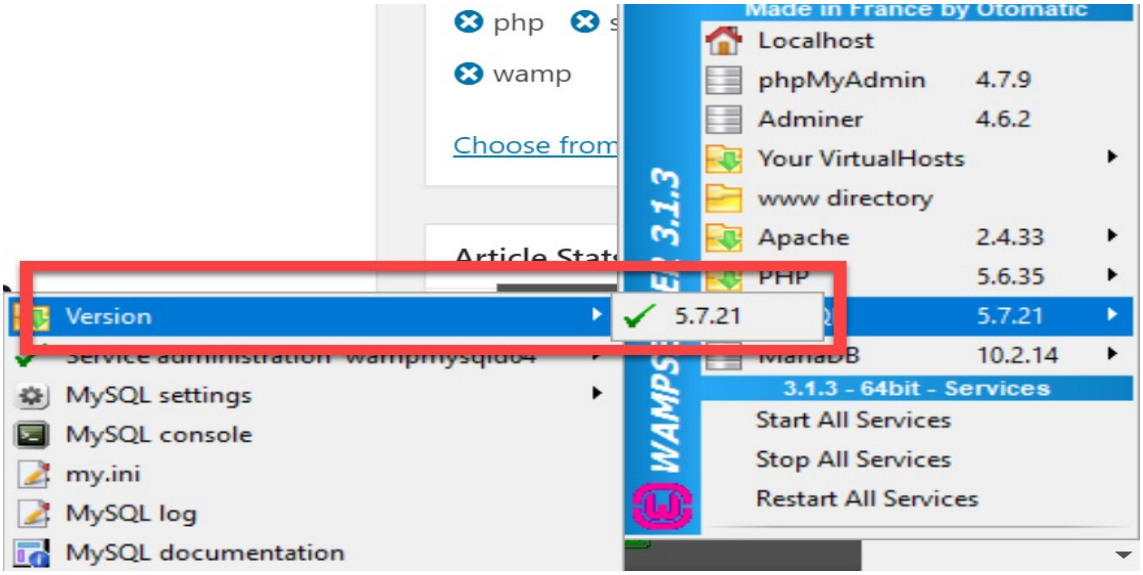
You can change the PHP version from here.



Check MySQL Version

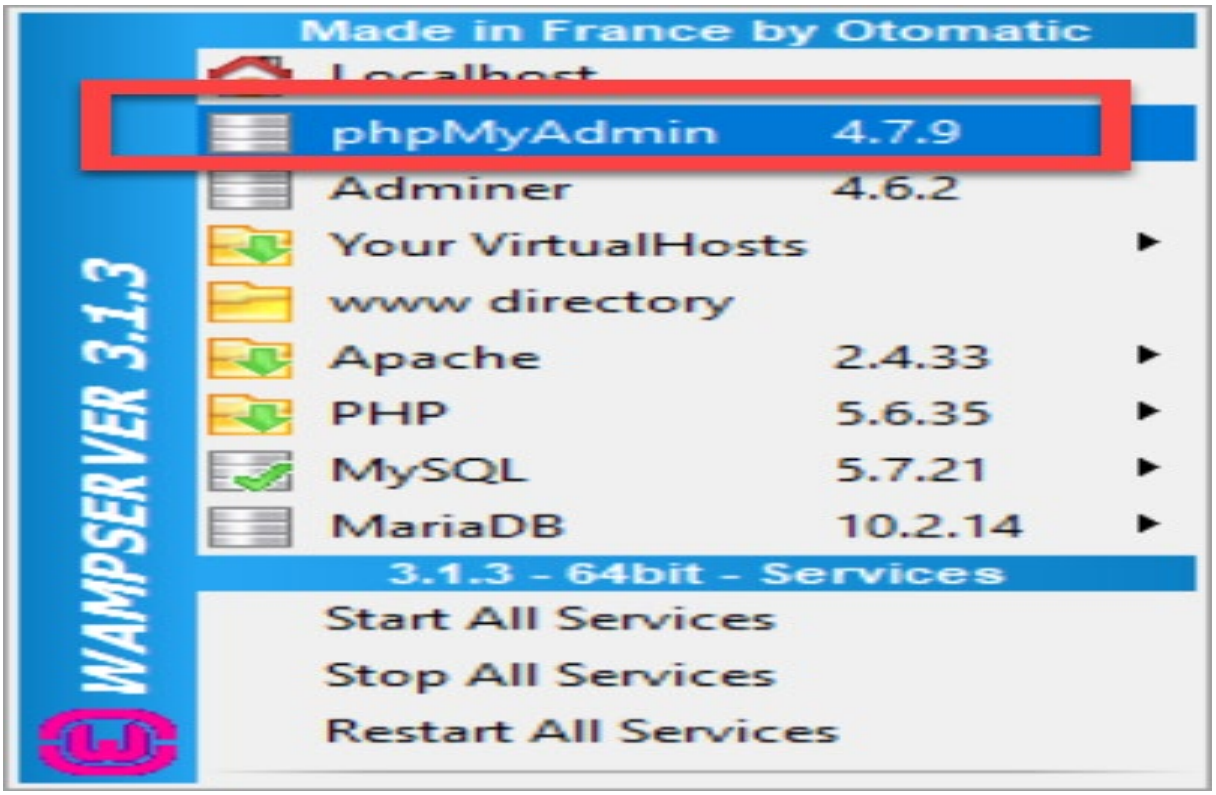
Open the WAMP Menu -> SQL -> Versions

You can change the SQL version from here.



Check phpMyAdmin Version

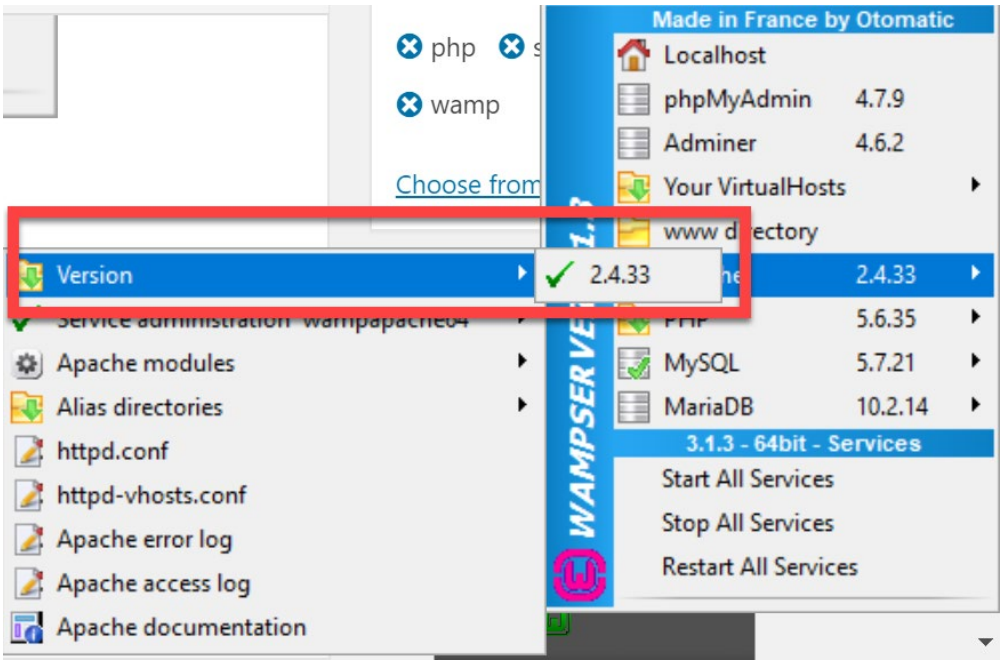
Open the WAMP Menu -> phpMyAdmin



Check Apache HTTP Version

Open the WAMP Menu -> Apache-> Versions

You can change the Apache version from here.



1.2 Echo

First PHP Program

php file name should end with "**.php**" extension

You should write the php code with in this starting and ending symbols.

<?php

?>

<?php – tells server to interpret the code from here

?> – server will stop interpreting the code.

You should write the PHP code inside this block

<?php

//PHP CODE

?>

SYNTAX:

<body>

<?php

?>

</body>

All the statements you write between the php block should end with ";"

Add semicolor ';' at the end of the line.

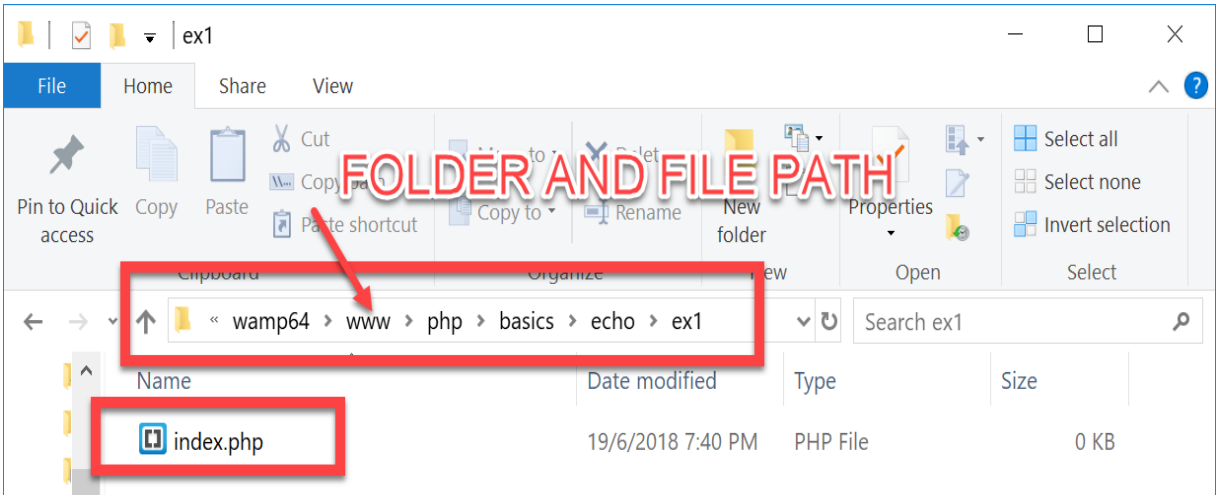
Step 1: Create the php file

Goto Folder: C:\wamp64\www

Create Folder: \php\basics\echo\ex1

Final Folder: C:\wamp64\www\php\basics\echo\ex1

Create file with name: index.php
Folder: C:\wamp64\www\php\basics\echo\ex1



Step 2: Add the code in index.php

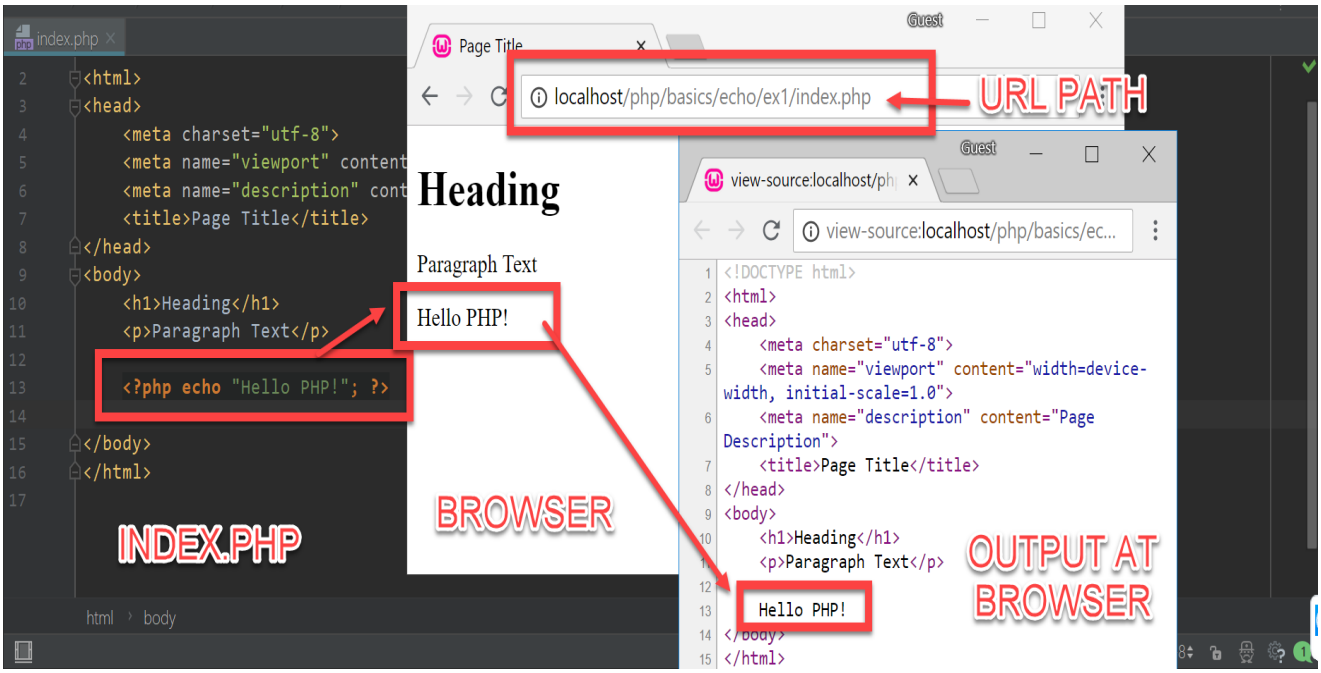
You can use the Brackets Software to open the file.

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="Page Description">
    <title>Page Title</title>
</head>
<body>
    <h1>Heading</h1>
    <p>Paragraph Text</p>
    <!-- PHP Code -->
```

```
    <?php echo "Hello PHP!"; ?>


</body>
</html>
```

Step 3: Access the index.php from browser

Type the URL: http://localhost/php/basics/echo/ex1/index.php



Step 4: Understanding echo

echo is a function to print the data on the browser.

There is no parenthesis () required to call the method and pass data to the method.

echo “some text to print”;

echo ‘some text to print’;

You use single quote or double quotes and statement should end with semi colon “;”.

You even use **print** function

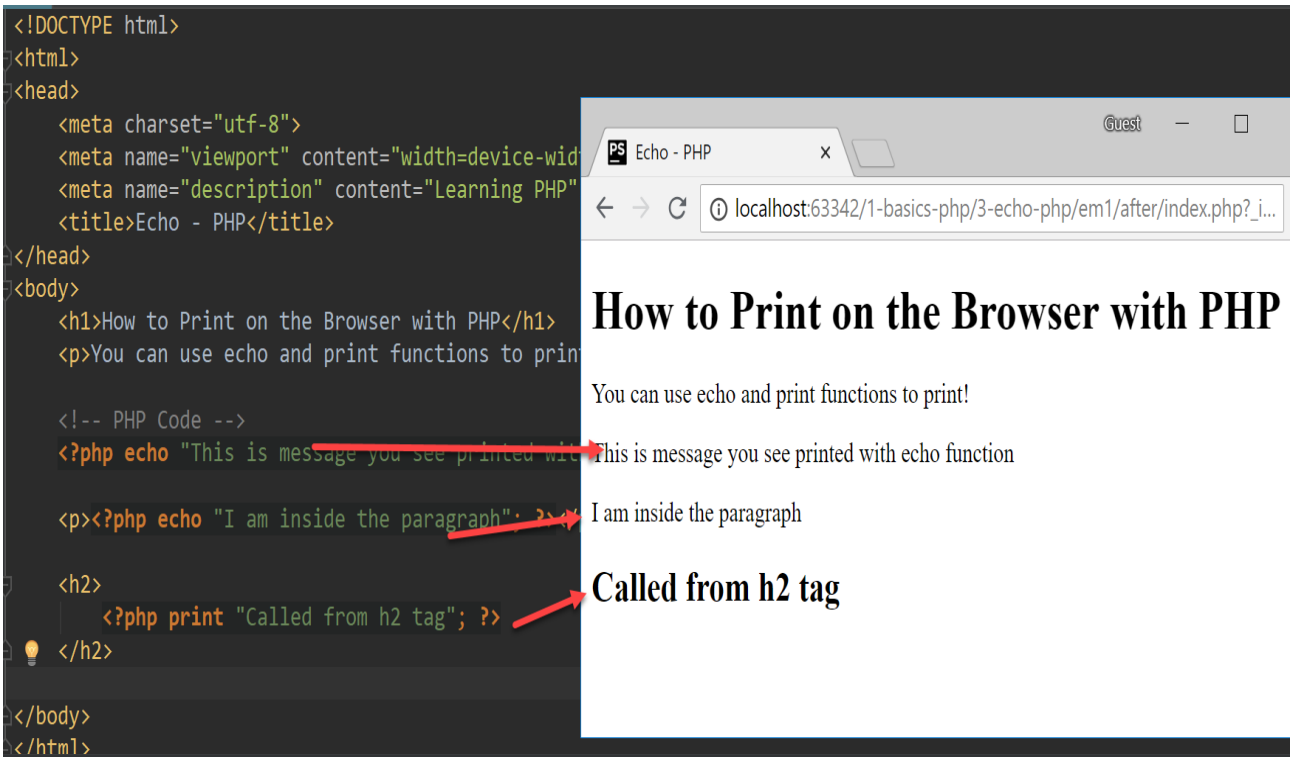
print “some text to print”;

print ‘some text to print’;

Sample Example

Download the Example

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="Learning PHP">
    <title>Echo - PHP</title>
</head>
<body>
    <h1>How to Print on the Browser with PHP</h1>
    <p>You can use echo and print functions to print!</p>

    <!-- PHP Code -->
    <?php echo "This is message you see printed with echo function"; ?>

    <p><?php echo "I am inside the paragraph"; ?></p>

    <h2>
        <?php print "Called from h2 tag"; ?>
    </h2>

</body>
</html>
```



Live Preview

Exercise 1

Download the Exercise 1

Exercise 1: Put the echo inside the <head> <script> tag and print alert('This is called from php');

See if the alert works?

Did you saw the alert?

1) PHP Code Executes First at server.

2) Javascript executes at Browser.

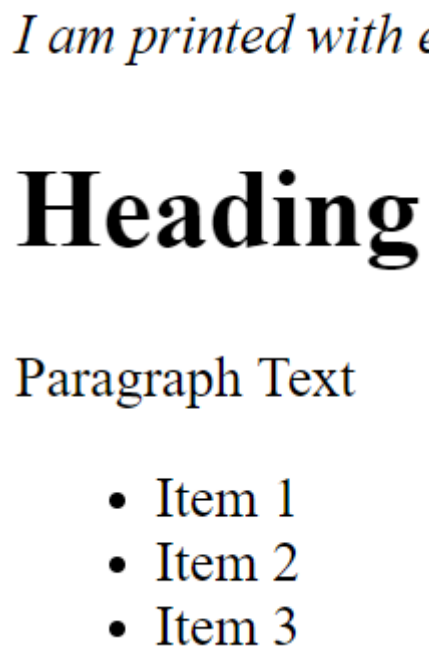
Live Preview

Exercise 2

Download the Exercise 2

Exercise 2: Write all the below HTML content using the php echo.

```
<h1>Heading</h1>
<p>Paragraph Text</p>
<ul>
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 3</li>
</ul>
```



Live Preview

1.3 Hello Program with PHP

Hello Program

You should know the following things:

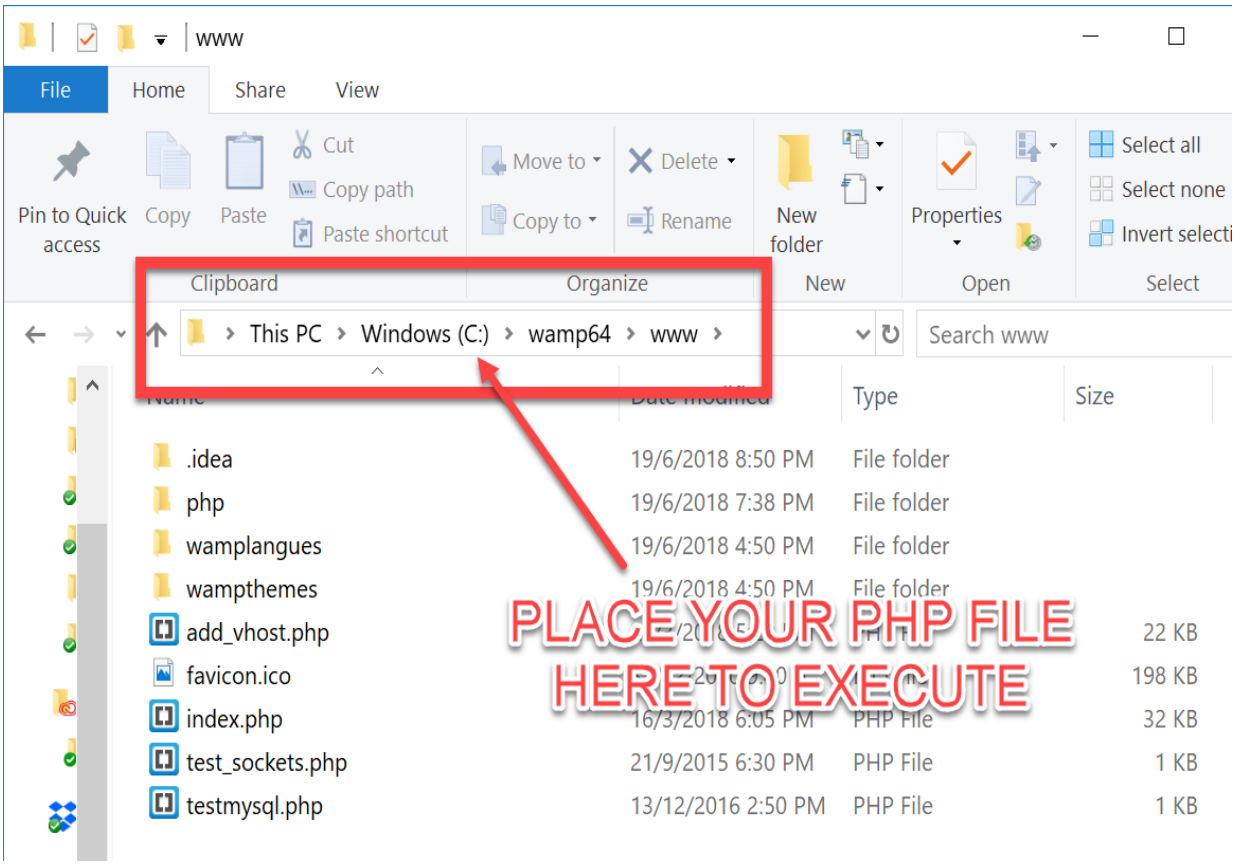
How to write and Execute the php Code.

How to start and stop WAMP Server.

How to view the php file from the browser.

Importance of 'www' folder in WAMP Folder

You can execute the php code only from the 'www' folder inside the c:\wamp64 folder.



Write a simple hello program with the following files:

index.php

css include

js include

Use css style file

Write one Function in the Javascript and use it.

Print some text with the php echo function.

Sample Example

Download the Example

index.php

```
<!DOCTYPE html>
```

```
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <meta name="description" content="Learning PHP">
    <title>Hello World Program!</title>

    <!-- Include CSS file -->
    <link rel="stylesheet" type="text/css" href="styles.css"
media="screen" />

    <!-- Include JS File -->
    <script type='text/javascript' src="scripts.js"></script>

</head>
<body>
    <h1>Welcome to PHP Application</h1>

    <div>
    <!-- on hover on this link to get a red line -->
    <a href="#" onclick="greet('Welcome to PHP'); ">Click Me!</a>
    </div>
    <hr>

    <div>
        <?php

            echo "Hello World PHP! This is printed
                  with the two and
```

```
                    multiple lines without any issues.";
        ?>
    </div>

</body>
</html>
```

---

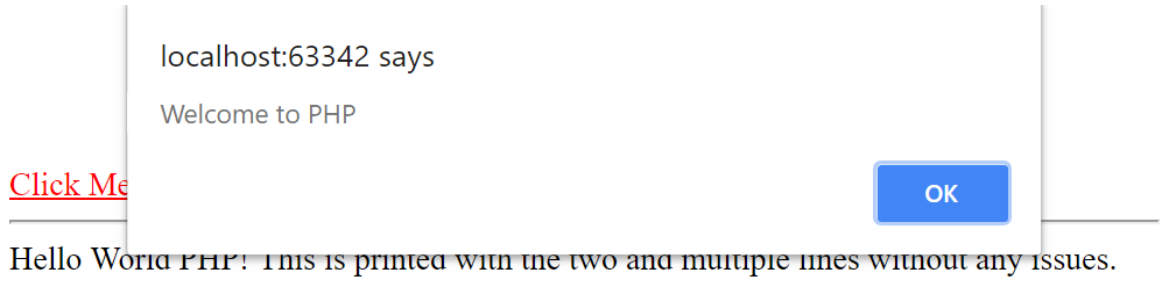
styles.css

```
a:hover{
    text-decoration: underline;
    text-color: red;
}

h1 {
    text-align: center;
}
```

---

scripts.js

```
function greet(message){
    alert(message);
}
```



Live Preview

Exercise 1

Download the Exercise 1

Exercise 1: Print the **<body>** tag with echo.

Live Preview

Exercise 2

Download the Exercise 2

Exercise 2: Write the complete HTML page with echo.

Understand that you can write anything on the page with echo.

Live Preview

1.4 Comments

Usage of Comments

You can use the special notation to comment the code inside the php.

Comments helps to document about the code with a single or multiple line.

Comments are ignored by the Web Server and it is not displayed and not sent to the browser.

You will not see php comments in the final HTML output.

There are two types of Comments in PHP:

Single Line Comments

Multi Line Comments

// – This is Single Line Comments. Use to comment one line.

 # – This is Single Line Comments. Use to comment one line.

/* – is used to indicate the comments are starting.

***/** – is used to indicate the comments are ended.

Anything between **/*** and ***/** will not be executed by the server.

Browser will never see php comments because it is ignored by the server.

SYNTAX:

<body>

<?php

// This is a Single Line Comments

This is also a Single Line Comments

/*

This is a multi line comments which

you cannot see on the browser

*/

?>

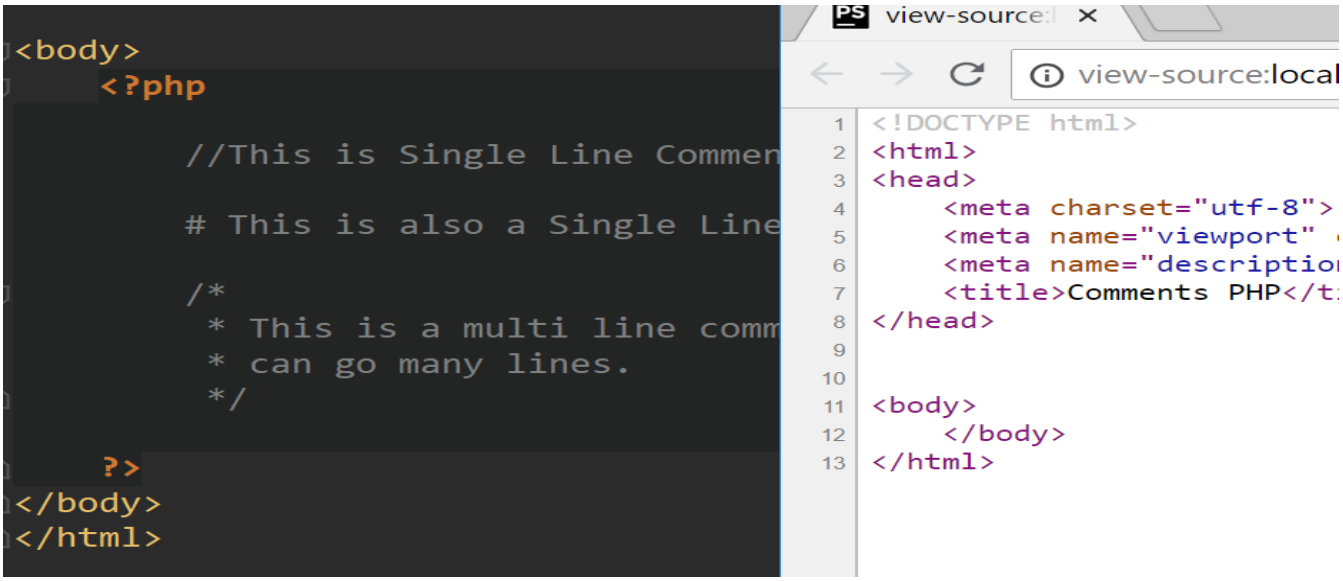
</body>

Sample Example

Download the Example

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="Learning PHP">
```

```
    <title>Comments PHP</title>
</head>



<body>
    <?php

        //This is Single Line Comments

        # This is also a Single Line Comments

        /*
         * This is a multi line comments
         * can go many lines.
         */

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



[Live Preview]

Exercise 1

[Download the Exercise 1]

Exercise 1: Put the php opening tag on top and start the multi line comments and then close it at the end of HTML.

Guess the output.

```
<?php /*

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
    scale=1.0">
    <meta name="description" content="Learning PHP">
    <title>Comments PHP</title>
</head>
<body>
</body>
</html>

*/?>
```

[Live Preview]

Exercise 2

Download the Exercise 2

Exercise 2: Start the opening /* at the start of the HTML page and do not close it. See if it raise any issue.

```
<?php /*
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="Learning PHP">
    <title>Comments PHP</title>
</head>


<body>

</body>
</html>
```

Live Preview

1.5 Functions

Usage of Fuctions

Functions is a block of statements that performs an action.

You can pass parameters to functions and it can return a value from the function using "**return**" keyword.

Syntax of Function Declaration:

function nameOfFunction(Parameters){

return someValue;

}

Example 1:

function area($width, $height){

return width * height;

}

echo “Area: ” . area(10,20);

. – Dot character is used to append two strings together.

Sample Example

Download the Example

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="Learning PHP">
    <title>Functions PHP</title>
</head>
<body>
    <?php

        // Print message using functions
        function printMessage(){
            echo "<h1>Welcome to PHP</h1>";
        }

        //Return value from functions
        function add($a, $b){
```

```
            return $a + $b;
        }

        // using "." to append data.
        echo "Addition of two value: " . add(5, 10 );

    ?>

    <?php printMessage(); ?>

</body>
</html>
```

Addition of two value: 15

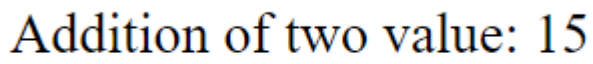
Welcome to PHP

Live Preview

Exercise 1

Download the Exercise 1

Exercise 1: Define a add function in <head> section and use them in the <body> section.

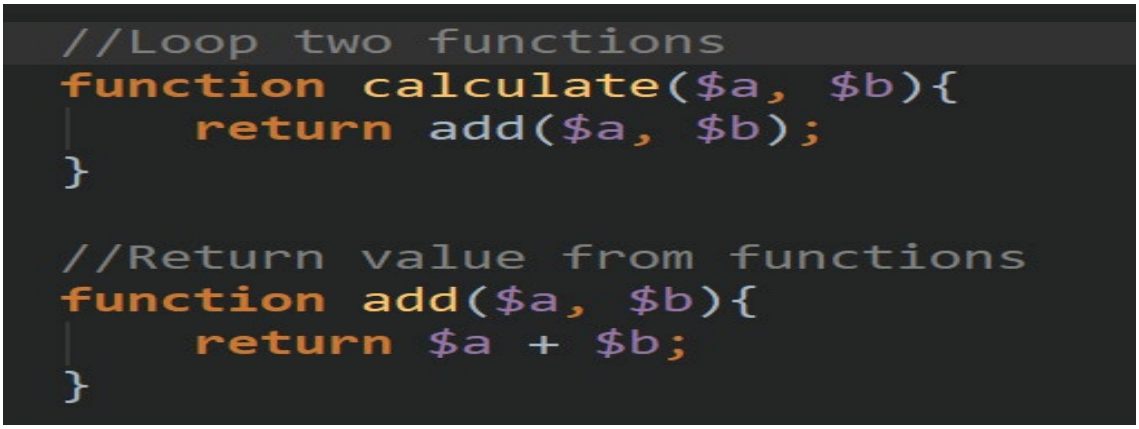


Live Preview

Exercise 2

Download the Exercise 2

Exercise 2: Create two functions and call one function from another function.



```php
//Loop two functions
function calculate($a, $b){
    return add($a, $b);
}

//Return value from functions
function add($a, $b){
    return $a + $b;
}
```

Live Preview

2. DATA BASICS

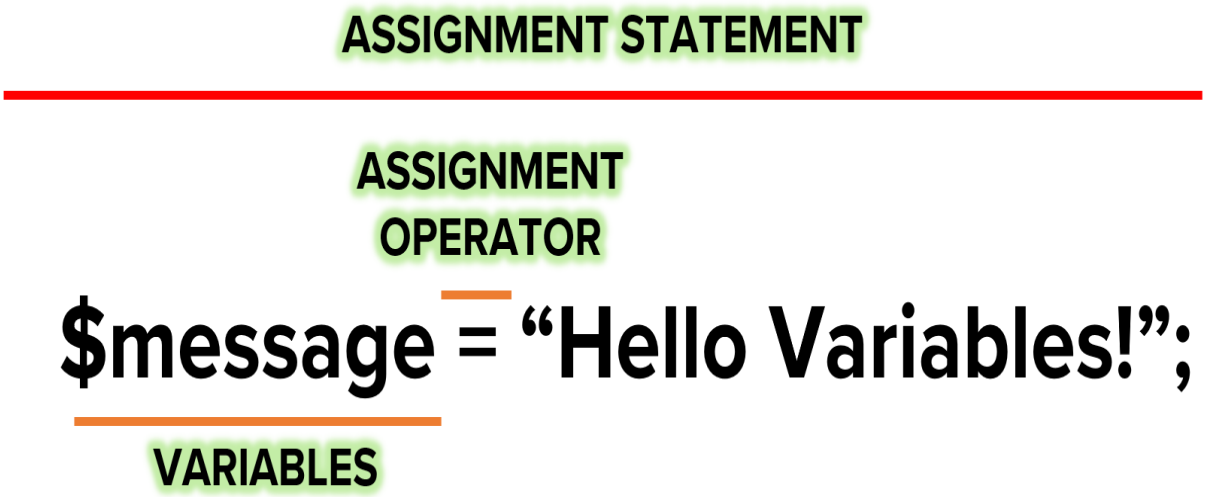
2 Data Basics

2.1 Variables

Usage of Variables

Variables are used to store information which are used inside the program.

Variables in php are defined with dollar ($) sign in front of it.



Here are the rules for defining the Variables:

Variable should start with dollar($) sign. Example : $message.

Letters, numbers and underscore is allowed inthe variable name.

After $ it should not be a number. Example: $123name is wrong.

Variables are case sensitive. $message and $Message are not same.

Examples of Variables:

$index = 0;

$firstName = 'WPFreelancer';

$firstName = "WPFreelancer";

$price = 10;

$price = 10.50;

$result = true;

$fullName = $firstName;

$ is used to indicate it is a variable.

= is the assignment operator used to assign the value to the variable.

; semicolor is used to end the assignment statement.

Showing variable with echo

$siteName = "WPFreelancer";

echo "This is the $siteName for WordPress";

echo "This is the site name" . $siteName;

You can add the variables inside the **double quotes only** with the variable names.

"." DOT symbol can be used to add the variable to the strings.

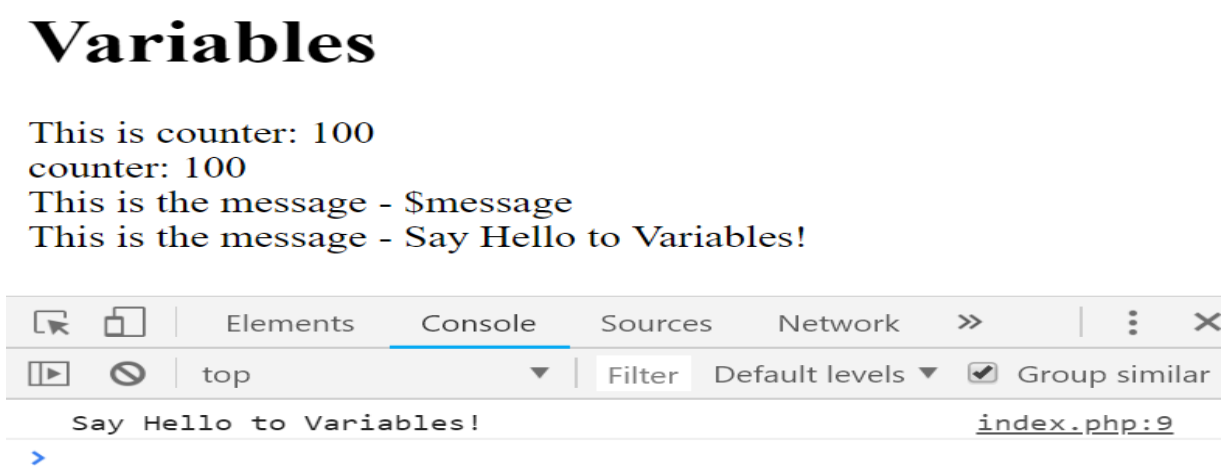
Sample Example

Download the Example

Data Basics

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="Page Description">
    <title>Variables</title>
    <?php

        //Declaring & Initalizing a Variable
        $message = "";

        //Assigning value to the Variables
        $message = "Say Hello to Variables!";

        //Displaying the Variables via the Javascript
        echo "<script>console.log('" . $message . "');</script>";

    ?>

</head>

<body>
<h1>Variables</h1>
<?php
    $counter = 100;

    echo "This is counter: $counter <br>";
    echo "counter: ". $counter . "<br>";
```

```
    echo 'This is the message - $message <br>';
    echo "This is the message - $message <br>";

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

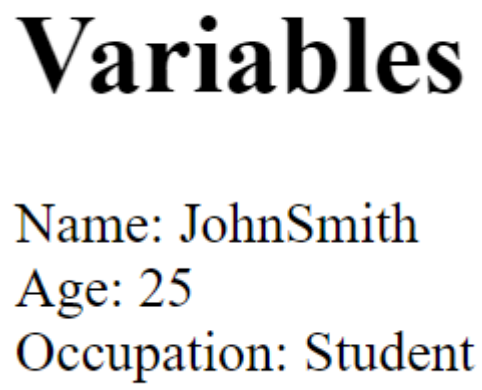


Live Preview

Exercise 1

Download the Exercise 1

Exercise 1: Define all the Variables and Initialize them in the head section and then display them in the body section.

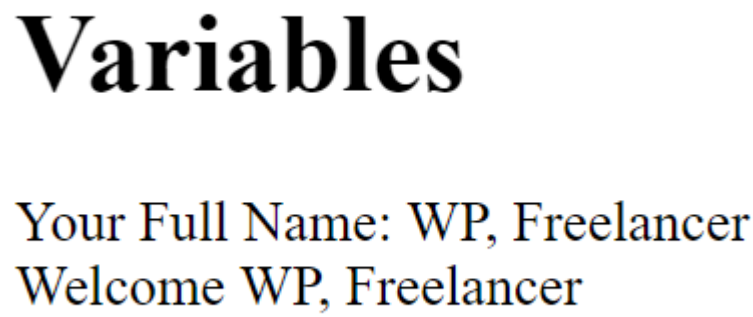


Live Preview

Exercise 2

Download the Exercise 2

Exercise 2: Write a function to add two strings with a "," and return the new string. call this function from body.



Live Preview

2.2 Strings

Usage of String

Strings in PHP can be enclosed with Single Quote or Double Quotes.

You can use Single quotes inside the double quotes and vice-versa.

. – DOT Symbol is used to concatenate two strings together.

Variable when used inside the Double Quote then it will resolve into the variable value.

This is called as *Interpolation*.

Interpolation of Variables only happens with Double Quotes and it does not work with Single Quotes.

That's why you must always use Single Quotes and only use Double Double quotes when you need interpolation feature.

Define a Variable with NULL

$message = NULL;

or

$message = null;

NULL or null is a special keyword that is used to define a empty variable.

String Examples

$firstName = 'WPFreelancer';

$firstName = "WPFreelancer";

$fullName = $first . " – " . $last;

$fullName = "$first – $last";

$places = ""; //Empty String

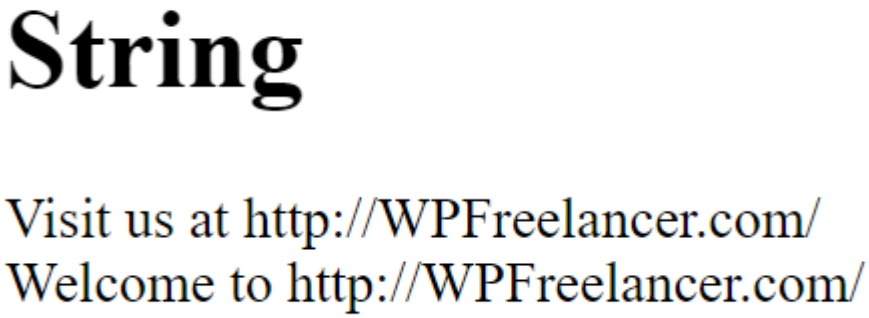
$numPrice = 10.20;

$SubTotal = "Price: " . $numPrice;

Sample Example

Download the Example

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="Page Description">
    <title>String</title>

</head>

<body>
<h1>String</h1>
<?php
    $fullName = 'WPFreelancer';
    $site = "http://$fullName.com/";
    echo "Visit us at $site <br>";

    $message = "Welcome to";
    echo $message . " " . $site;
?>
</body>
</html>
```



Live Preview

Exercise 1

Download the Exercise 1

Exercise 1: Convert a string to uppercase and lowercase. Display the text on body.

Lowercase Function: **strtolower**($variable);
Uppercase Function: **strtoupper**($variable);
Example: $site = strtolower($site);

Live Preview

Exercise 2

Download the Exercise 2

Exercise 2: Substring “WP” from “WPFreelancer” word using the String function.

substr($variable, startIndex, length);

Example: $newValue = substr($fullName, 0, 2);

Live Preview

2.3 Numbers

Usage of Numbers

Numbers can also be called as Integers.

Integers are positive or negative numbers.

Floating number is represented with number separated with “.” as decimals.

So, there are two types of numbers in PHP:

Integers

100

-200

Floating Point

10.34

Numbers are not wrapped with quotes or they do not include “,”

You can append the – (minus) symbol in front of the number to indicate it is negative number.

Most common methods used with Integer:

round() – This will round the decimals

intval() – This will convert string integer to integer.

Examples:

$price = 10;

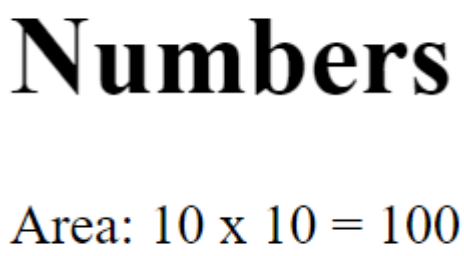
$total = 10.20;

Sample Example

Download the Example

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="Page Description">
    <title>Numbers</title>

</head>

<body>
<h1>Numbers</h1>
<?php
    $length = 10;
    $breath = 10;
```

```
    $area = $length * $breath;
    echo "Area: $length x $breath = $area";
?>
</body>
</html>
```



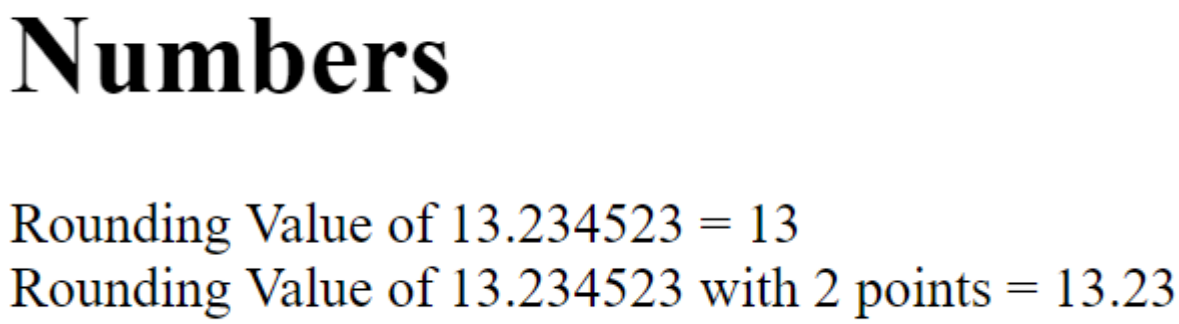
Live Preview

Exercise 1

Download the Exercise 1

Exercise 1: Write a function to round a given decimal number and display the output.

Tips: Use the round(value)

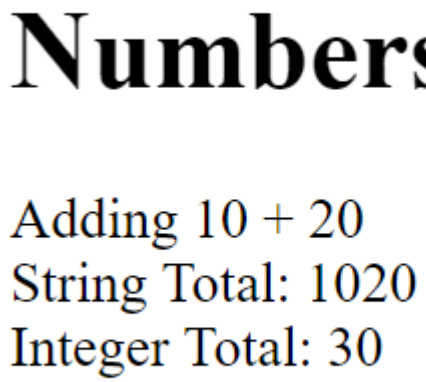


Live Preview

Exercise 2

Download the Exercise 2

Exercise 2: Convert String to Integer and add two string numbers and add two integers numbers.



Live Preview

2.4 Arrays

Usage of Arrays

Arrays is a data type that holds one or more items called as elements.

Each element could be combination of data types.

length is used to indicate the number of elements in the array.

Every element of the array can be accessed with an index number.

First element of Array starts with 0 index.

Define an Array

$arrayName = array($value1, $value2);

Examples:

$colorName = array('red', 'white', 'yellow');

```
$newColors = array();
$newColors[0] = $colorName[0];
$newColors[1] = $colorName[1];
```

```
$age = array(10, 40, 34);
$studentage[0] = $age[0];
$studentage[1] = 50;
```

Print Arrays

print_r($arrayName);

Sample Example

Download the Example

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="Page Description">
    <title>Arrays</title>

</head>

<body>
<h1>Arrays</h1>
<?php
    //Define an Array
    $colorNames = array('red', 'green', 'white');

    //Print an Array
    print_r($colorNames);
    echo "<br>";

    //Loop thru the arrays
    foreach ($colorNames as &$value) {
        echo $value . "<br>";
    }
```

```
    //Define an Empty Array
    $age = array();
    $age[0] = 10;
    $age[1] = 20;
    $age[3] = 30;

    //Print Array
    print_r($age);

?>

</body>
</html>
```

Arrays

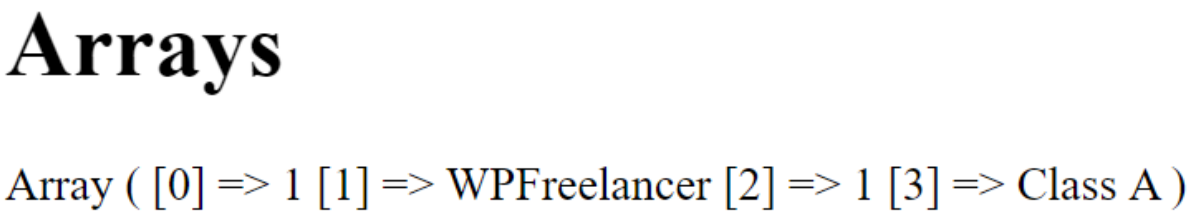
Array ([0] => red [1] => green [2] => white)
red
green
white
Array ([0] => 10 [1] => 20 [3] => 30)

Live Preview

Exercise 1

Download the Exercise 1

Exercise 1: Create an Array with String, Number and Boolean and display them.



Live Preview

Exercise 2

Download the Exercise 2

Exercise 2: Do the following exercise with Arrays

Define an Empty Array

Add 1 Element

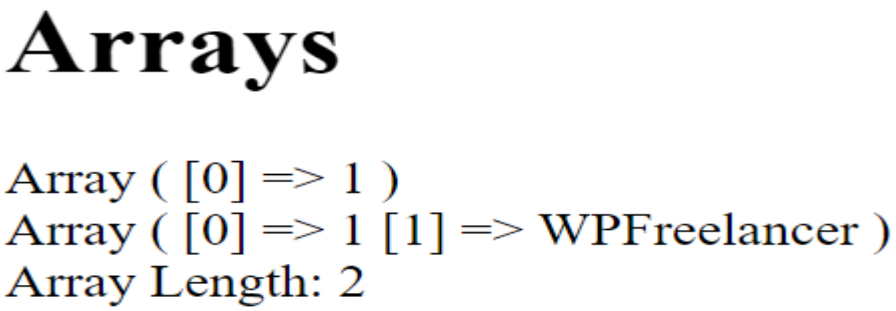
Display the Length

Add 2 Element

Show the two Elements

Show the Length of the Array

Use count($arrayName) method to show the length of an array.



Live Preview

2.5 Objects

Usage of Classes and Objects

Class is a collection of Variables and Functions together.

Object is an instance of a Class used to store values in the class variables and access them via the functions.

Instead of defining methods and variables separately you can create a class and store them.

You need to create an instance of class to access it.

-> symbol is used to access the variable or methods in the object.

$this is a special object that will help to access the existing object of a class.

Class Syntax:

class Student{

private $id, $name, $age;

```
public function __construct($id, $name, $age){
$this->id = $id;
$this->name = $name;
$this->age = $age;
}

public function getName(){
return $this->name;
}

public function setName($name){
$this->name = $name;
}

public function getStudentDetails(){
$details = “ID: ” . $this->id . ” NAME: ” . $this->name . ” AGE:” . $this->age;
return $details;
}

}
```

Create Instance of Class

```
$studentObj = new Student(1, “WP”, 20);
echo $studentObj->getStudentDetails();
```

Sample Example

Download the Example

```
<!DOCTYPE html>
<html>
<head>
```

```
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="Page Description">
    <title>Classes and Objects</title>
    <?php
            class Student{

                private $id, $name, $age;

                public function __construct($id, $name, $age){
                    $this->id = $id;
                    $this->name = $name;
                    $this->age = $age;
                }

                public function getName(){
                    return $this->name;
                }

                public function setName($name){
                    $this->name = $name;
                }

                public function getStudentDetails(){
                    $details = "ID: " . $this->id . " NAME: " . $this->name . " AGE:" . $this->age;
                    return $details;
```

```
                }

            }
    ?>

</head>

<body>
<h1>Classes and Objects</h1>
<?php
    $studentObj = new Student(1, "WP", 20);
    echo $studentObj->getStudentDetails();
?>

</body>
</html>
```

Classes and Objects

ID: 1 NAME: WP AGE:20

Live Preview

Exercise 1

Download the Exercise 1

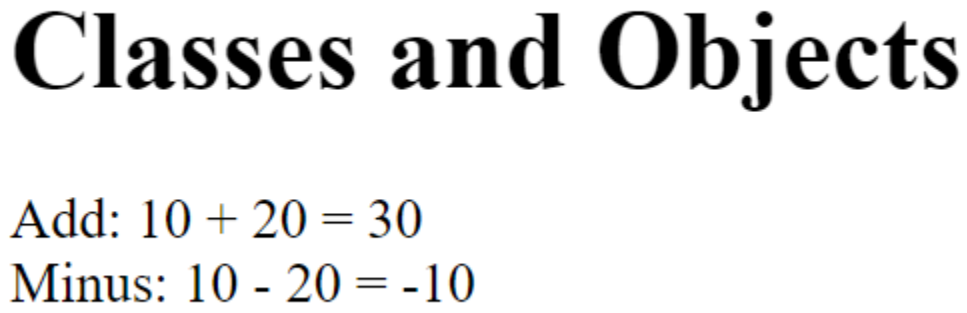
Exercise 1: Create a Product Class with productName and price variables and create one function to access the price.

Live Preview

Exercise 2

Download the Exercise 2

Exercise 2: Create a Calculator class and pass two values and create add and minus function.

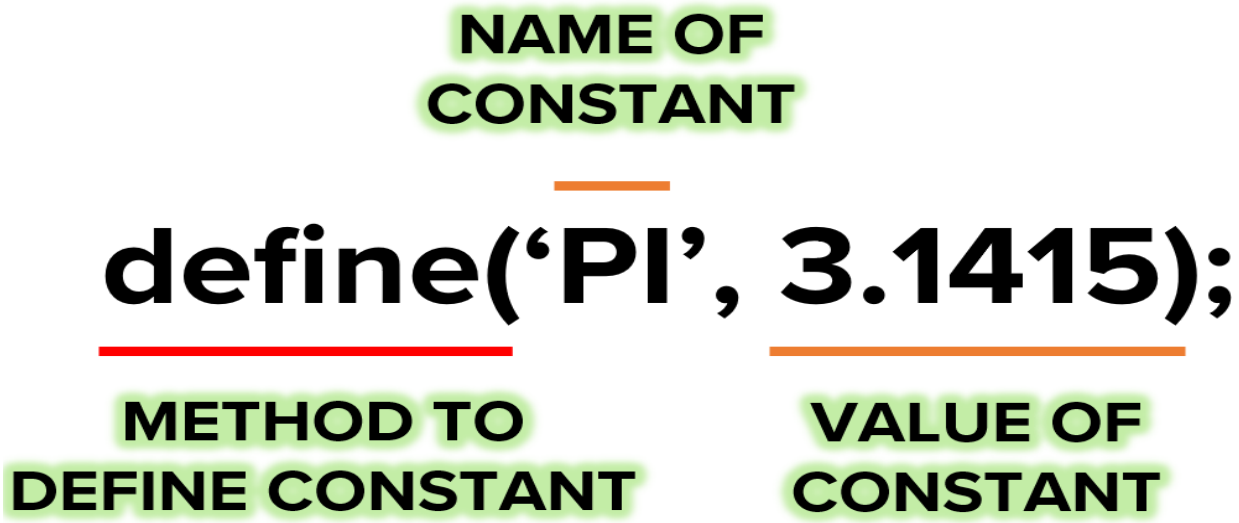


Live Preview

2.6 Constants

Usage of Constant

Constant are like variables but once you define the constant with a fixed value you cannot change it later.



Constant variables values are fixed and cannot be changed later.

define() is a method used to define a constant.

constant does not need $ dollar because it is not like a variable.

Example:

define('AGE', 20);

echo AGE;

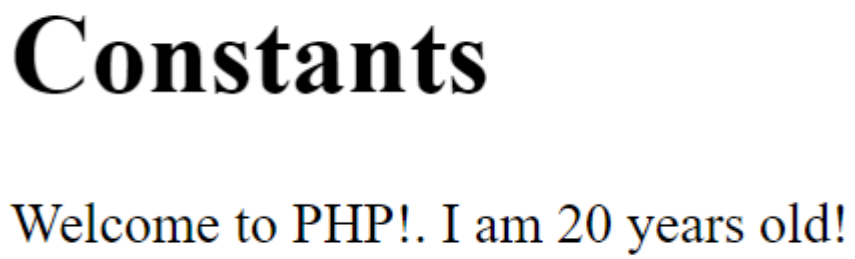
You don't need $ to access it.

Typically, constant is defined in UPPERCASE to differentiate easy between variables and constants.

Sample Example

Download the Example

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="Page Description">
    <title>Constants</title>
    <?php

        //Define a Constant
         define('MESSAGE', "Welcome to PHP!");
         define('AGE', 20);

    ?>

</head>

<body>
<h1>Constants</h1>
<?php
    echo MESSAGE . ". I am " . AGE . " years old!";
?>
</body>
</html>
```

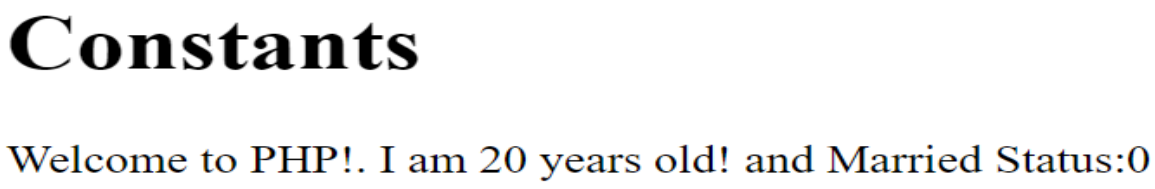


Live Preview

Exercise 1

Download the Exercise 1

Exercise 1: Define String, Number and Boolean Constant and display them.

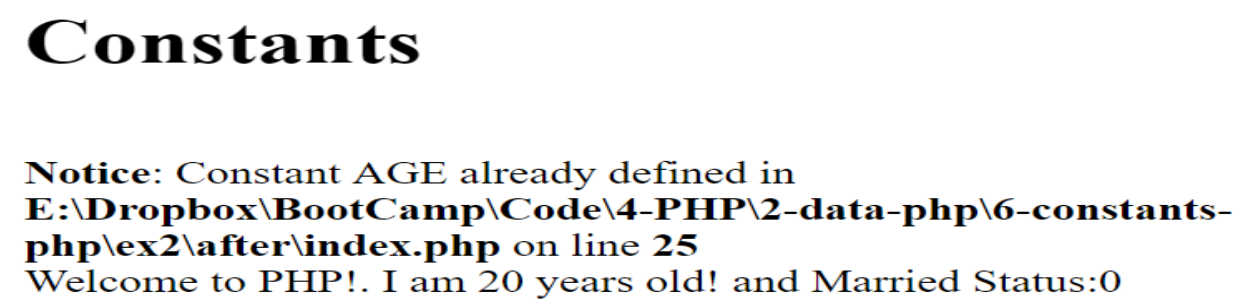


Live Preview

Exercise 2

Download the Exercise 2

Exercise 2: Try to change the value of Constant and observe the error.



Live Preview

2.7 Boolean

Usage of Boolean

Boolean Variables helps to make decisions or store a decision based on an expression.

Boolean values can be true or false.

You can use boolean variable as a condition to check if the value is true or false.

$result = 2 > 1;
$message = ($result) ? “CORRECT”: “WRONG”;

Sample Example

Download the Example

```
<!DOCTYPE html>
<html>
<head>
```

```
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="Page Description">
    <title>Constants</title>

</head>

<body>
<h1>Constants</h1>
<?php
    $result = 2 > 1;
    $message = ($result) ? "CORRECT" : "INCORRECT";
    echo "Is 2 > 1? - $message";
?>
</body>
</html>
```

Live Preview

Exercise 1

Download the Exercise 1

Exercise 1: Guess the Output.

```php
<?php
    $strText = "100";
    $message = ($strText > 10) ? "CORRECT" : "INCORRECT";
    echo "$message";
?>
```

Live Preview

Exercise 2

Download the Exercise 2

Exercise 2: Create a Constant and use the constant to check the condition.

Live Preview

2.8 Date and Time

Usage of Date and Time

date() is a class library available to access the date.

We need to pass the format of the date to get the system date.

$today = date('Y-m-d'); //2018-01-01

$today = date('l, F, d, Y'); //Thusday, Jun 21, 2018

By default, $today will have user system date.

TimeStamp:

$todaytime = time();

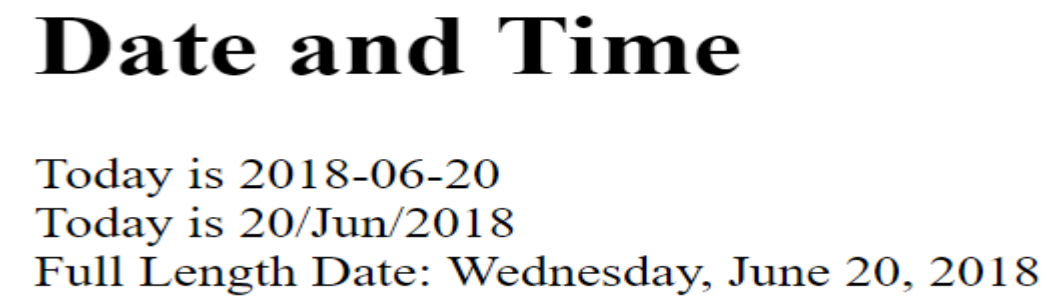
echo $todaytime();

Sample Example

Download the Example

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
    <meta name="description" content="Page Description">
    <title>Date and Time</title>
</head>
<body>
<h1>Date and Time</h1>
<?php
    echo "Today is " . date('Y-m-d');
    echo "<br>";
    echo "Today is " . date('d/M/Y');
    echo "<br>";
    $formatdate = date('l, F d, Y');
    echo "Full Length Date: " . $formatdate;
?>
</body>
</html>
```



Live Preview

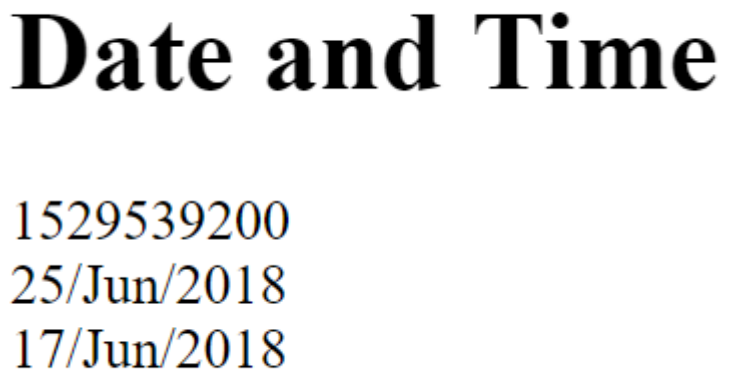
Exercise 1

Download the Exercise 1

Exercise 1: Print tomorrow date with this function strtotime('tomorrow');

Format the Date with date function

```
echo date( 'd/M/Y', strtotime('+5 days'));
```



Live Preview

Exercise 2

Download the Exercise 2

Exercise 2: Print the '**first day of next month**' and format with date function.

Tip:

```
$futuredate = strtotime('first day of next month');
```

Live Preview

3. EXPRESSIONS

3 Expressions

3.1 Assignments

Usage of Assignment Expressions

Operators are used to perform some operation on the variables and they are represented with some symbols.

Expressions are evaluated into a result value or final value or single value.

There are couple of Expressions:

Assignment Expressions

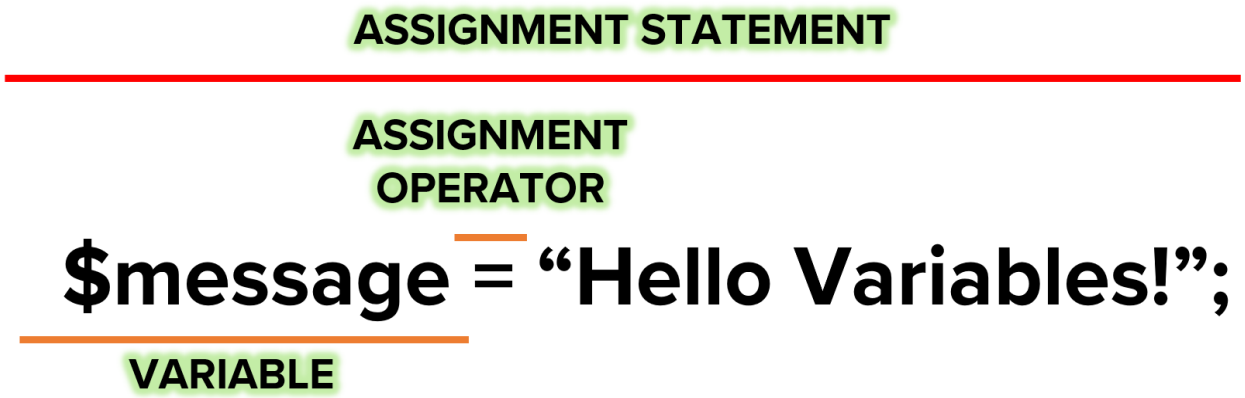
Arithmetic Expressions

Comparison Expressions

Logical Expressions

We will look at Assignment Expressions in this topic.

Expressions uses Operators to perform the activity.



= is the **assignment operator** which assigns the value to the variable.

message = “something”; is a assignment expression that assign value to the message variable.

Compound Assignment Operators

Compound assignment operators helps to do more than one operators job. It combines two operators together to perform an action.

Compound Assignment Operators are:

.=

+=

-=

*=

/=

%=

$message = ‘WP’. ‘Freelancer’;

OR

$message = 'WP';
$message .= 'Freelancer';

counter **+=** 1;
counter = counter + 1;

counter **-=** 1;
counter = counter – 1;

counter ***=** 1;
counter = counter * 1;

Sample Example

Download the Example

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="Page Description">
    <title>Assignment Expressions</title>
</head>

<body>
<h1>Assignment Expressions</h1>
<?php
    $firstName = "WP";
```

```
    $lastName = "Freelancer";
    $fullName = $firstName . ", " . $lastName;
    echo "Name $fullName <br>";

    $counter = 10;
    echo "Counter: $counter <br>";
    $counter += $counter;
    echo "Counter+=: $counter <br>";
    $counter -= 10;
    echo "Counter-=: $counter <br>";

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

Assignment Expressions

Name WP, Freelancer
Counter: 10
Counter+=: 20
Counter-=: 10

Live Preview

Exercise 1

Download the Exercise 1

Exercise 1: Guess the difference between the two assignments.

$counter++ and $counter += counter;

```php
$counter = 10;

echo "Counter: $counter <br>";
$counter += $counter;
echo "Counter: $counter <br>";

$counter = 10;
echo "Counter: $counter <br>";
$counter++;
echo "Counter: $counter <br>";
```

Assignment Expressions

Counter: 10
Counter: 20
Counter: 10
Counter: 11

Live Preview

Exercise 2

Download the Exercise 2

Exercise 2: What is the result of $counter -= $counter and $counter -= 10; when $counter = 10;

```php
$counter = 10;

echo "Counter: $counter <br>";
$counter -= $counter;
echo "Counter: $counter <br>";

$counter = 10;
echo "Counter: $counter <br>";
$counter -= 10;
echo "Counter: $counter <br>";
```

Assignment Expressions

Counter: 10
Counter: 0
Counter: 10
Counter: 0

Live Preview

3.2 Arithmetic

Usage of Arithmetic Expressions

Expressions

Expressions are evaluated into a result value or final value or single value.

There are couple of Expressions:

Assignment Expressions

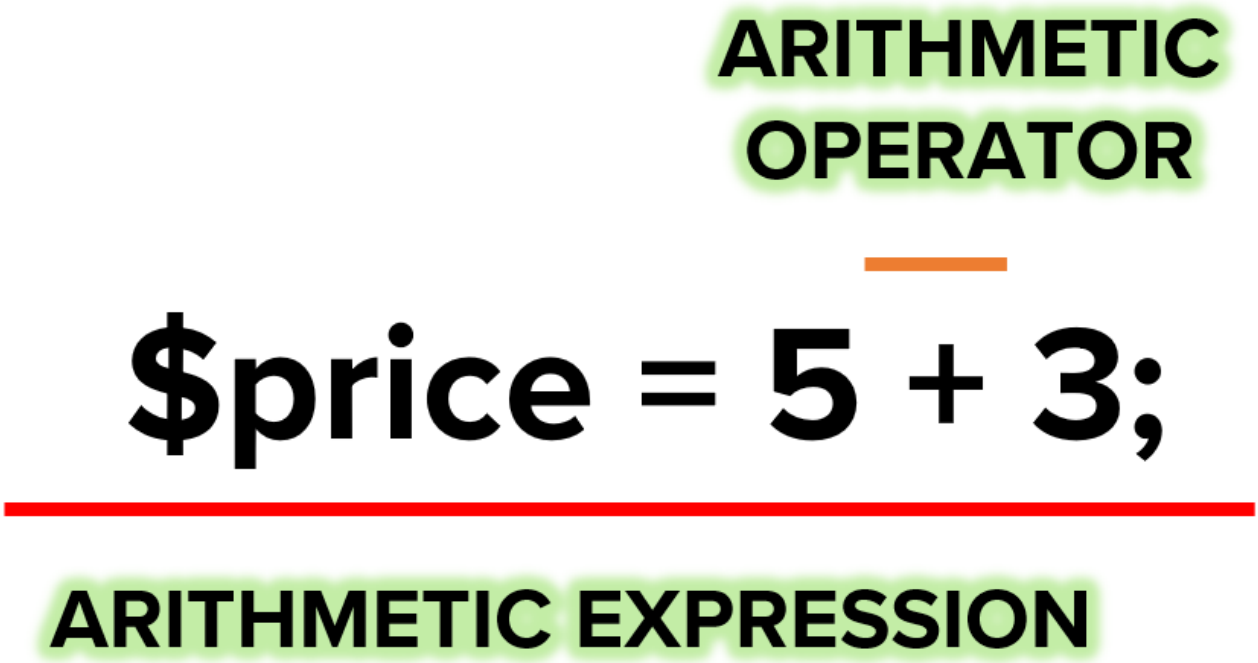
Arithmetic Expressions

Comparison Expressions

Logical Expressions

We will look at Assignment Expressions in this topic.

Expressions uses Operators to perform the activity.



+ is the **arithmetic operator** which is used to perform mathematics calculations.

Arithmetic Operators are:

+

–

*

/

%

++

—

//Increment by 1
counter**++**;
counter = counter + 1;

//Decrement by 1
counter**—**;
counter = counter – 1;

Order of Precedence:

++

—

* / %

+ –

Anything mentioned in () will get higher precedence over anything.

(5 + 2) * 2; //Result – 14

Sample Example

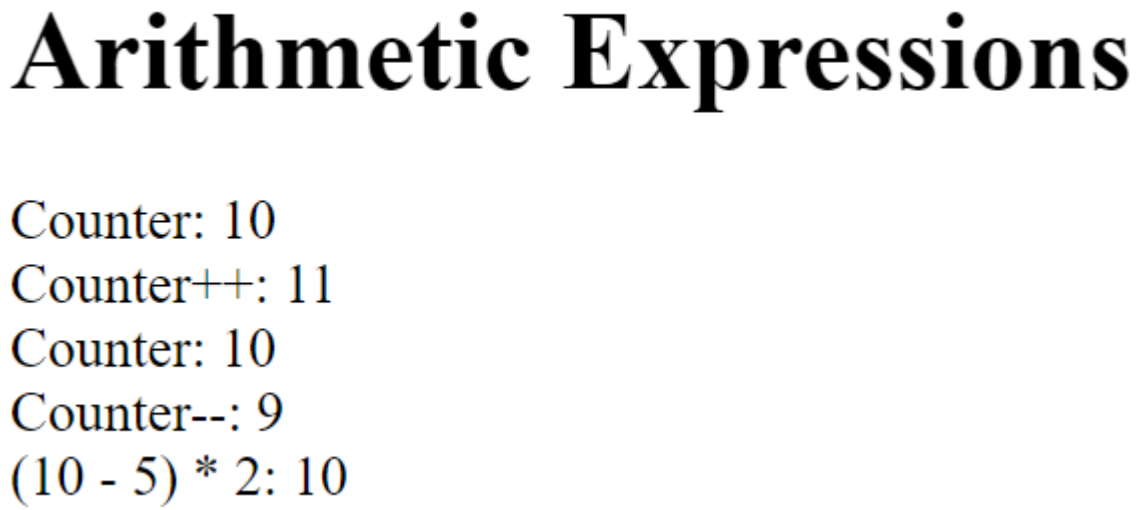
Download the Example

```
<!DOCTYPE html>
```

Expressions

```
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="Page Description">
    <title>Arithmetic Expressions</title>
</head>

<body>
<h1>Arithmetic Expressions</h1>
<?php



    $counter = 10;

    echo "Counter: $counter <br>";
    $counter++;
    echo "Counter++: $counter <br>";

    $counter = 10;
    echo "Counter: $counter <br>";
    $counter--;
    echo "Counter--: $counter <br>";

    $counter = (10 - 5) * 2;
    echo "(10 - 5) * 2: $counter <br>";
```

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



Live Preview

Exercise 1

Download the Exercise 1

Exercise 1: Find a 15 is even or odd using Modulus (%) symbol.

```
$counter = 15%2;
```

Live Preview

Exercise 2

Download the Exercise 2

Exercise 2: Try to apply the ++ to the following things:

1++;

echo “$counter++”;

++1;

$message = “Hello”;

$message++;

$result = true;

$result++;

Some of the things won’t work so comment them and run the code.

Live Preview

3.3 Comparison

Usage of Comparison Expressions

Expressions are evaluated into a result value or final value or single value.

There are couple of Expressions:

Assignment Expressions

Arithmetic Expressions

Comparison Expressions

Logical Expressions

We will look at Comparison Expressions in this topic.

Expressions uses Operators to perform the activity.

Comparison expression always evaluate into a true or false value.

> is the **comparison operator** which compares the two values.

message variable will be have a true or false.

Comparison Operations:

< – Less than

> – Greater than

== – Equal to

=== – Equal value and Equal Data Type

!== – Not Equal Value and Equal Data Type

!= – Not Equal

>= – Greater than and Equal

<= – Less than and Equal

Conditional (Ternary) Operator

PHP has a conditional operator that assigns a value to a variable based on some condition.

$variablename = ($condition) ? value1:value2
$counter = (10<=10) ? 10 : 0;

Sample Example

Download the Example

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="Page Description">
    <title>Comparison Expressions</title>
</head>

<body>
<h1>Comparison Expressions</h1>
<?php


    $counter = 10 == 10;
    echo $counter; //1 = true and 0 = false

    echo "<br>";
    $result = 15 <= 21;
    $message = ($result)? "YES": "NO";
```

```
    echo $message;


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

Comparison Expressions

1
YES

Live Preview

Exercise 1

Download the Exercise 1

Exercise 1: Learn how to compare NULL in the string with === operator.

Comparison Expressions

TRUE
FALSE

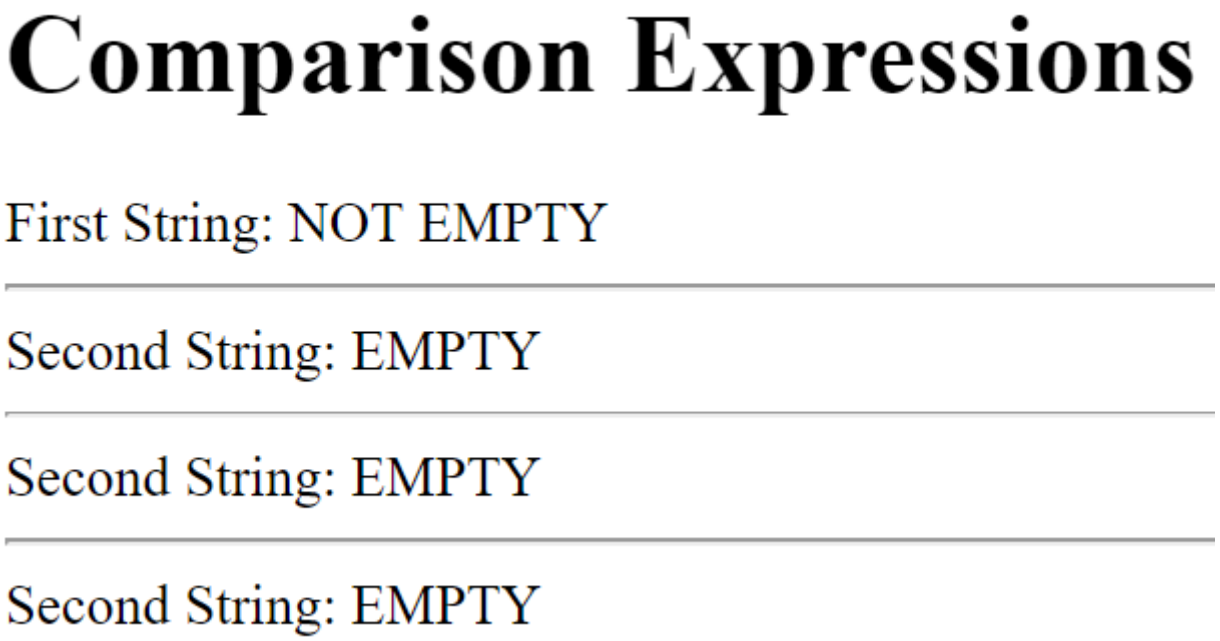
Live Preview

Exercise 2

Download the Exercise 2

Exercise 2: Find empty string from 2 strings.

```
$result = empty($emptyString1) ? "EMPTY": "NOT EMPTY";
echo "First String: " . $result;
echo "<br><hr>";
$result = empty($emptyString2) ? "EMPTY": "NOT EMPTY";
echo "Second String: " . $result;
echo "<br><hr>";
$result = ($emptyString2 == "") ? "EMPTY": "NOT EMPTY";
echo "Second String: " . $result;
echo "<br><hr>";
```



Live Preview

3.4 Logical

Usage of Logical Expressions

Expressions are evaluated into a result value or final value or single value.

There are couple of Expressions:

Assignment Expressions

Arithmetic Expressions

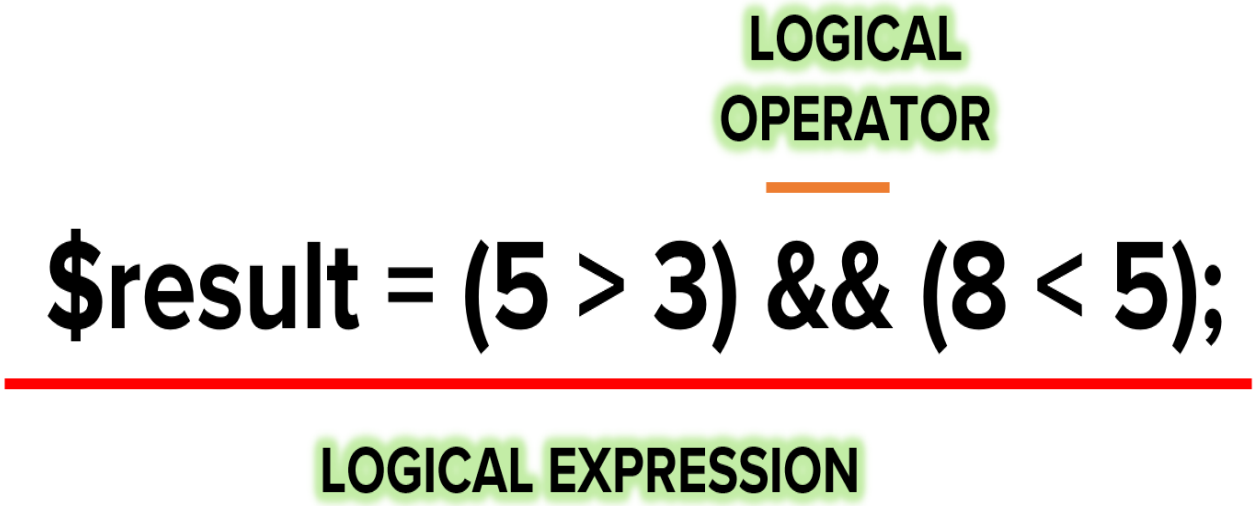
Comparison Expressions

Logical Expressions

We will look at Logical Expressions in this topic.

Expressions uses Operators to perform the activity.

Logical Operators are used to check the if the condition is true or false based on many conditions.



&& is the **logical operator** which checks left side and right side value and decides if the condition is true or false.

(5 > 3) – true
(8 < 5) – false
true && false = false

result variable will have **false** boolean value.

Logical Operations:

&& – AND

|| – OR

! – NOT

Order of Precedence

Order of precedence decides which operates evaluates first.

NOT

AND

OR

Sample Example

Download the Example

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="Page Description">
    <title>Logical Expressions</title>
</head>

<body>
<h1>Logical Expressions</h1>
<?php

    $input1 = 10;
    $input2 = 20;
    $result = ($input1 < $input2) || ($input1 == $input2);

    echo $input1 . " <= " . $input2 . " is " . $result;
```

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

Logical Expressions

10 <= 20 is 1

Live Preview

Exercise 1

Download the Exercise 1

Exercise 1: Guess the Output from below code.

```php
$input1 = 10;
$input2 = 10;
$result = ($input1 <= $input2) && ($input1 >= $input2);
echo $input1 . " <= " . $input2 . " is " . $result;
```

Live Preview

Exercise 2

Download the Exercise 2

Exercise 2: Guess the output from the below code.

```php
$input1 = 10;
$input2 = "20";
$result = ($input1 === $input2) || ($input1 == $input2);
echo $input1 . " <= " . $input2 . " is " . $result;
```

Live Preview

4. STATEMENTS

4 Statements

4.1 If Statements

Usage of if Statements

if statement are used to check a condition and make a decision based on the result of the condition.

It can choose some action when the condition is true and also take some action when it false.

You can nest multiple conditions together and decide to choose one action based on multiple conditions.

Syntax:

```
if ( condition1 ){

// Statements

} else if( condition1 || condition2 ){

// Statements

} else {

// Statements
```

}

Example 1:

if($marks > 35){

echo “Your’re Passed!”;

}else{

echo “Better Luck Next Time!”;

}

Example 2:

if($marks > 35 && $marks < 60){

echo “You are Passed with Grade C”;

}else if ($marks > 60 && $marks < 80){

echo “You are Passed with Grade B”;

}else if ($marks > 80){

echo “You are Passed with Grade A”;

}else{

echo “Try Again!”;

}

Download the Example

```
<!DOCTYPE html>
<html>
<head>
```

```
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="Page Description">
    <title>If Statements</title>
    <?php
        $dayOfWeek = "Wednesday";

        $result = "";
        $dayOfWeek = strtolower($dayOfWeek);

        if( $dayOfWeek == 'monday' || $dayOfWeek == 'mon'){
            $result = "First Day of Week";
        }else if( $dayOfWeek == 'tuesday' || $dayOfWeek == 'tue' ){
            $result = "Second Days of Week";
        }else if( $dayOfWeek == 'wednesday' || $dayOfWeek == 'wed' ){
            $result = "Mid Week";
        }else if( $dayOfWeek == 'thursday' || $dayOfWeek == 'thurs'){
            $result = "Preparing for Weekend";
        }else if( $dayOfWeek == 'friday' || $dayOfWeek == 'fri'){
            $result = "It's Friday!";
        }else if( $dayOfWeek == 'saturday' || $dayOfWeek == 'sat' ){
            $result = "Enjoying Day!";
```

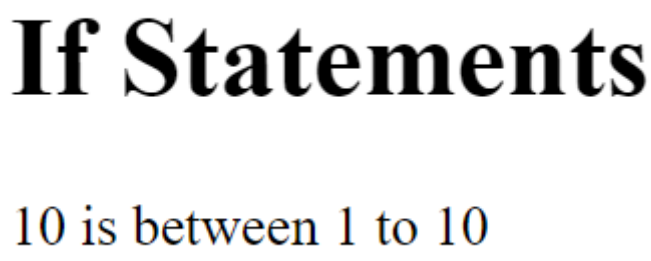
```
        }else if( $dayOfWeek == 'sunday' || $dayOfWeek == 'sun'){
            $result = "Resting Day!";
        }else{
            $result = "Cannot find that Value!";
        }
    ?>

</head>

<body>
<h1>If Statements</h1>
<?php
    echo "This is the result: " . $result;

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

Live Preview

Exercise 1

Download the Exercise 1

Exercise 1: Set the inputVariable number between 1 to 10 and check if that variable is between 1 to 10 with if Statement.

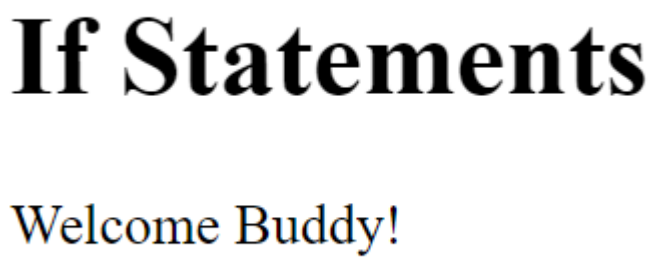


Live Preview

Exercise 2

Download the Exercise 2

Exercise 2: Enter the age of the user in the InputVariable and decides if he is kid, man (age > 21) or senior citizen (age > 55).



Live Preview

4.2 Switch Statements

Usage of Switch Statements

switch statement are used to check a value and make a decision based on the result of the value matching.

Syntax:

switch($variable){

case value:

//Statement

break;

case value:

//Statement

break;

default:

//Statement

break;

}

Example 1:

switch ($dayOfWeek){

case ‘Mon’:

echo “Welcome Monday”;

break;

case ‘Tuesday’:

echo “Welcome Tuesday”;

break;

default:

echo “Try Again!”;

break;

}

Download the Example

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="Page Description">
    <title>Switch Statements</title>
    <?php
        $dayOfWeek = "Friday";

        $result = "";
        $dayOfWeek = strtolower($dayOfWeek);
```

```
        switch($dayOfWeek){
            case 'monday':
                $result = "First Day of Week";
                break;
            case 'tuesday':
                $result = "Second Days of Week";
                break;
            case 'wednesday':
                $result = "Mid Week";
                break;
            case 'thursday':
                $result = "Preparing for Weekend";
                break;
            case 'friday':
                $result = "It's Friday!";
                break;
            case 'saturday':
                $result = "Enjoying Day!";
                break;
            case 'sunday':
                $result = "Resting Day!";
                break;
            default:
                $result = "Cannot find that Value!";
        }
    ?>
```

```
</head>

<body>
<h1>Switch Statements</h1>
<?php
    echo "This is the result: " . $result;
?>
</body>
</html>
```

Live Preview

Exercise 1

Download the Exercise 1

Exercise 1: Set InputVariable number between 1 to 10 and check if the variable is between 1 to 10 as requested with the switch Statement.

Live Preview

Exercise 2

Download the Exercise 2

Exercise 2: Set a Variable to user age and decides if he is kid, man (age > 21) or senior citizen (age > 55). Use the Switch Statement.

Tips:

```
switch(true){
case ($input > 21):
//Statements
break;
}
```

Live Preview

4.3 While Statements

Usage of While Loop

While statement are used to loop a block code and run it until a condition is met.

Running the same block of code until the condition is satisfied.

Syntax:

```
while( condition ) {
//Statements
}
```

Example 1:

```
$counter = 0;
while ( $counter < = 10 ){
$counter++;
}
```

Syntax:

```
do{
//Statement
}while ( condition );
```

Example 1:

$counter = 0;

do{

$counter++;

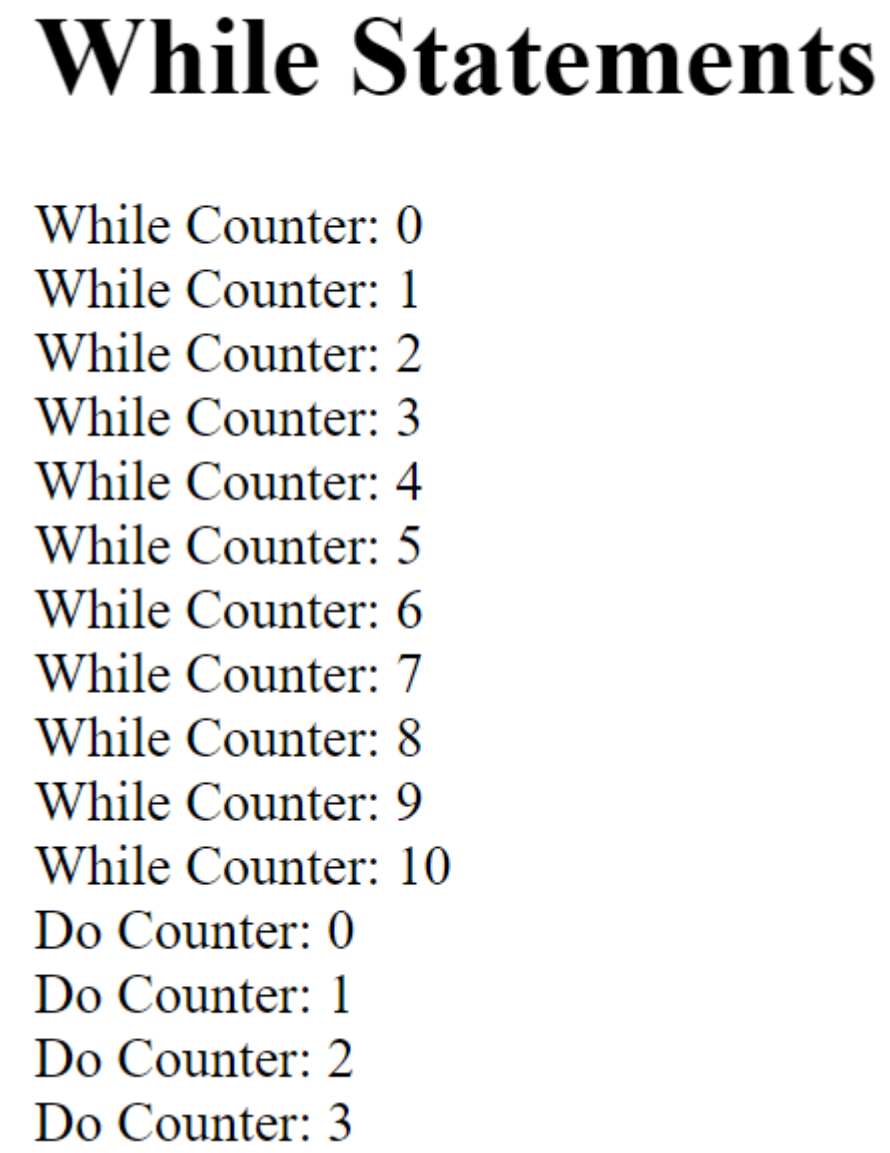
} while ($counter < = 10);

break; – break is the keyword used to break the loop and come out of the loop and execute statements after the while loop.

Download the Example

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="Page Description">
    <title>Boolean Expressions</title>
    <script>

        var result = 2 > 1;
        var message = (result) ? "CORRECT": "WRONG";

    </script>

</head>
```

```
<body>
<h1>Boolean Expression</h1>
    <script type="text/javascript">
        document.write(message);
    </script>
</body>
</html>
```



Live Preview

Exercise 1

Download the Exercise 1

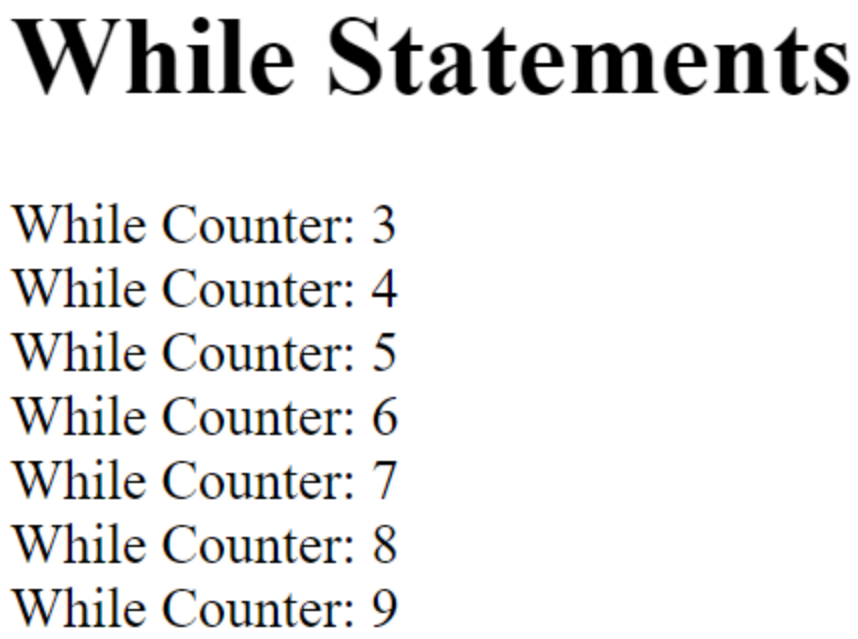
Exercise 1: Define CONSTANTS INPUT1 and INPUT2 and print all the numbers between them. Print only maximum of 10 numbers.

break; – is a keyword that can be used in the loop to break from the loop.

continue; – is a keyword that can be used in the loop to continue the loop by skipping the below statements.

Tip:

```
if( $counter >= 10){
   break;
}
```

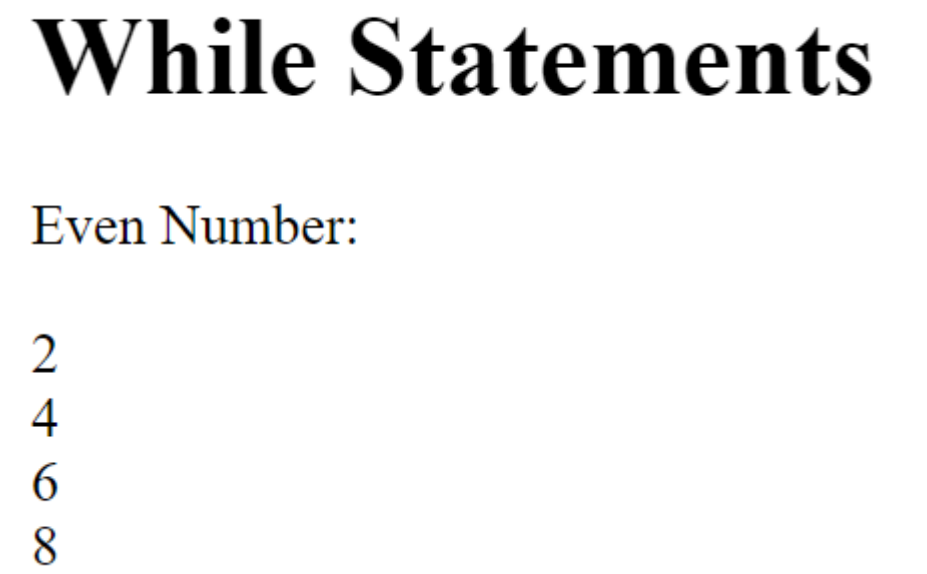


Live Preview

Exercise 2

Download the Exercise 2

Exercise 2: Define TWO CONSTANT and print **EVEN** numbers only between them. Print only maximum of 10 numbers.



Live Preview

4.4 For Statements

Usage of For Loop

for Loop are used to loop a block code and run it until a condition is met.

Running the same block of code until the condition is satisfied.

Syntax:

```
for( counter initialization; condition; increments ) {
//Statements
}
```

Example 1:

```
for($counter = 0; $counter <= 10; $counter++){

echo $counter;

}


for($counter = 0, $input1 = 10; $counter <= 10; $counter++,
$input1++){

echo $input1;

}
```

Download the Example

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <meta name="description" content="Page Description">
    <title>For Statements</title>
</head>

<body>
<h1>For Statements</h1>
<?php
```

```
    for($counter=0; $counter<=10; $counter++){
        echo "For Counter: " . $counter . "<br>" ;
    }

?>

</body>
</html>
```

For Statements

For Counter: 0
For Counter: 1
For Counter: 2
For Counter: 3
For Counter: 4
For Counter: 5
For Counter: 6
For Counter: 7
For Counter: 8
For Counter: 9
For Counter: 10

Live Preview

Exercise 1

Download the Exercise 1

Exercise 1: Define two constant number with default value and print all the numbers between them. Print only maximum of 10 numbers.

break; – is a keyword that can be used in the loop to break from the loop.

continue; – is a keyword that can be used in the loop to continue the loop by skipping the below statements.

For Statements

Starts from: 3 and Ends At: 9

Input0 : 3
Input1 : 4
Input2 : 5
Input3 : 6
Input4 : 7
Input5 : 8
Input6 : 9

Live Preview

Exercise 2

Download the Exercise 2

Exercise 2: Define two constant with default value and print **EVEN** numbers only between them. Print only maximum of 10 numbers.

Use the Loop like this:

```
for($counter = 0; $counter<=10 && ($input3 < $input4); $counter++, $input3++){
```

For Statements

Starts from: 3 and Ends At: 9

Input1 : 4
Input3 : 6
Input5 : 8

Live Preview

5. GENERAL

5 General

5.1 Exceptions

Usage of Try Catch Blocks

There are 3 types of error you can find in PHP:

Syntax Errors

Runtime Errors

Logic Errors

Syntax errors is where you forgot to follow the rules of PHP. It will cause error when you execute the program.

Runtime Errors could be when it is running the program the PHP interpreter could not understand how to proceed and throws and error.

Logic Errors are logically error that are cause because the program instructions are not logically correct.

Exceptions in programming are referred as Runtime Errors.

When Runtime Errors happens there is a way to catch those errors and pass it to the application and make a clean exit from the program.

This is called as Exception Handling.

The process of making sure the code will not break and if it does it know the reason for it and make a clean exit.

To handle Runtime errors in the PHP we have try-catch blocks. Using this we can catch the errors and decide what to do next.

SYNTAX:

try{

//Statements

}catch(Exception $exceptionObj) {

//Statments

}

Example:

try{

$firstName = “”;

if($firstName == “”) **throw new Exception(** “Name is empty”);

}**catch**(Exception $e){

echo “Message: ” . $e->getMessage();

}**finally**{

echo “This is from the Finally Block”;

}

try {} block will have all the statements

throw is a keyword to throw an error from the program. Once the program throw an error it will stop processing the next steps. It will jump to catch {} block.

catch {} block will be executed once the throw is called. The program will execute out from the catch block safely.

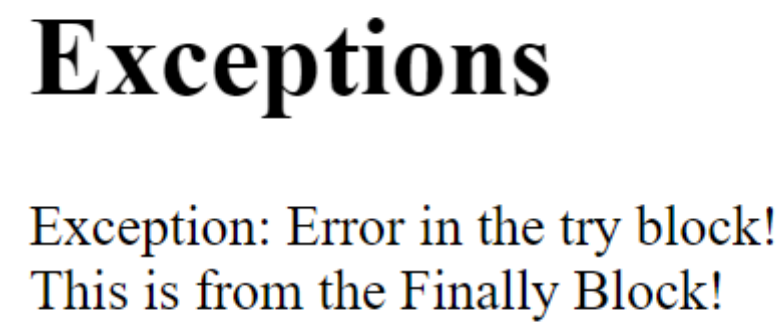
finally{} block will execute every time irrespective of error or not. It is good place to close all open connections. Handle clean exit.

```php
try{
    throw new Exception( message: "Error in the try block!");
}catch(Exception $e){
    echo "Exception: ". $e ->getMessage();
}finally{
    echo" <br> This is from the Finally Block!";
}
```

Download the Example

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="Page Description">
```

```
    <title>Exceptions</title>
</head>

<body>
<h1>Exceptions</h1>
<?php
    try{

        throw new Exception("Error in the try block!");

    }catch(Exception $e){

        echo "Exception: ". $e ->getMessage();

    }finally{

        echo" <br> This is from the Finally Block!";

    }

?>

</body>
</html>
```



Live Preview

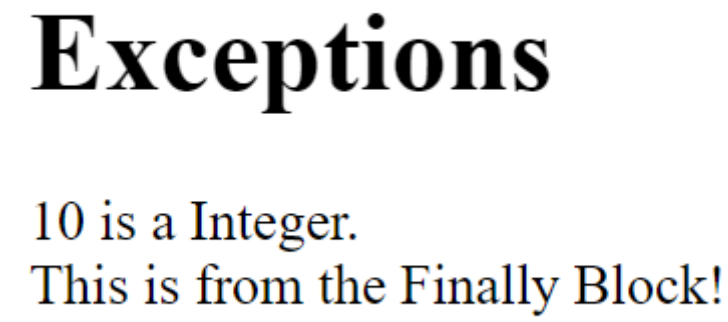
Exercise 1

Download the Exercise 1

Exercise 1: Define one constant number and raise error if the number is not integer.

Tip:

```
if( !is_numeric(INPUTVALUE ) )
```



Live Preview

Exercise 2

Download the Exercise 2

Exercise 2: Define a number between 1 to 100.

Show the following message on the page:

If the input is between 1 to 100 – less than 100.

If the input is greater than 100 – greater than 100.

If the input is not number – throw an error

```php
define('INPUTVALUE', "10");

if( !is_numeric( var: INPUTVALUE ) ){
    throw new Exception( message: "Input Value is not an Integer.");
}else if ( INPUTVALUE <= 100){
    echo INPUTVALUE . " is less than 100.";
}
else{
    echo INPUTVALUE . " is greater than 100.";
}
```

Live Preview

5.2 Debug

Step 1: Write PHP Code with Errors

Try the following Errors:

$inputValue = 10;echo inputValue;

echo “This is a String” + “another String”;

Download the Exercise 1

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="Page Description">
    <title>Debugging</title>
</head>

<body>
<h1>Debugging</h1>
<?php

    $inputValue = 10;
    echo inputValue;

?>

</body>
</html>
```

Debugging

Warning: Use of undefined constant inputValue - assumed 'inputValue' (this will throw an Error in a future version of PHP) in **E:\Dropbox\BootCamp\Code\4-PHP\5-general-php\2-debug-php\em1\after\index.php** on line **16**
inputValue

Live Preview 1

Exercise 1:

Add Sum() method in the php file and do not create that function. This will cause the program to fail.

Exercise 2:

Write echo statement before the sum() method and echo after the sum() method.

Find out the error by seeing the echo messages.

Step 2: Debug

Using echo you need to write the trace logs.

See which the last echo that was executed.

5.3 Files

Usage of Files

General

You can read files on the server using the PHP libraries.

Modes used during opening the file:

rb – Open the file for reading purpose.

wb – Create a new file and if already exists then it overwrite it.

ab – Create the file and append the data is already existed.

xb – Create a new file and if already exists then it does not create.

Here are the methods to open and close a file

fopen($path, $mode)

fclose($file)

feof($file) – To check if the file is at the end.

Method to read and write into the file:

fread($file, $length) – length is used to specify number of bytes to read.

fgets($file) – Read a line

fwrite($file, $data) – Write the $data value to the file.

Sample Example

Download the Example

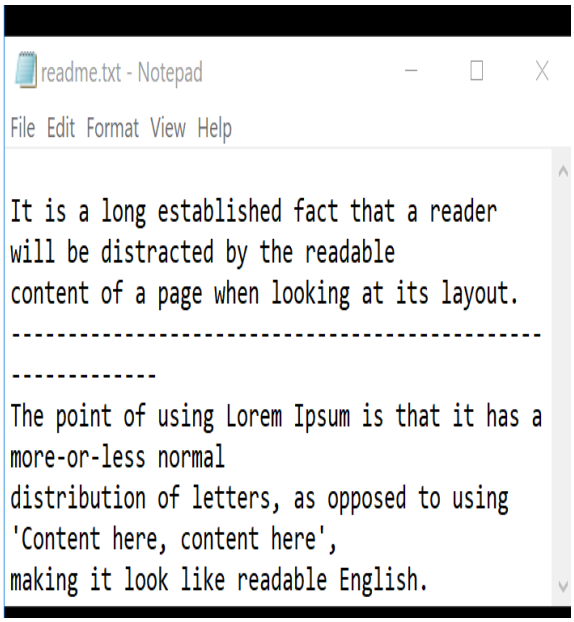
```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="Page Description">
    <title>Files</title>
</head>

<body>
<h1>Files</h1>
<?php

    //Create a local file and name it "readme.txt"

    /*
     * READ A FILE - rb mode
     *
     */
    $file = fopen('readme.txt', 'rb');
    $line = "";
    while( !feof($file) ){
        $line = fgets($file);
        echo $line;
    }
    fclose($file);

    /*
     * WRITE A FILE - wb mode
     *
```

```
     */
    $file = fopen("newfile.txt", "wb");
    fwrite( $file,  "<br><h1>It is a long established fact that
a reader
                      will be distracted by the readable
                    content of a page when looking at its
layout.</h1>");
    fclose( $file );

?>

</body>
</html>
```

Files

It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. ---------- ---------- The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using 'Content here, content here', making it look like readable English.



newfile.txt - Notepad

File Edit Format View Help

```
<br><h1>It is a long established fact that a reader
                    will be distracted by the readable
                   content of a page when looking at its layout.</h1>
```

Live Preview

Exercise 1

Download the Exercise 1

Exercise 1: Write a new file newfile.txt with some content. Read the same file and display the output on the browser.

Files

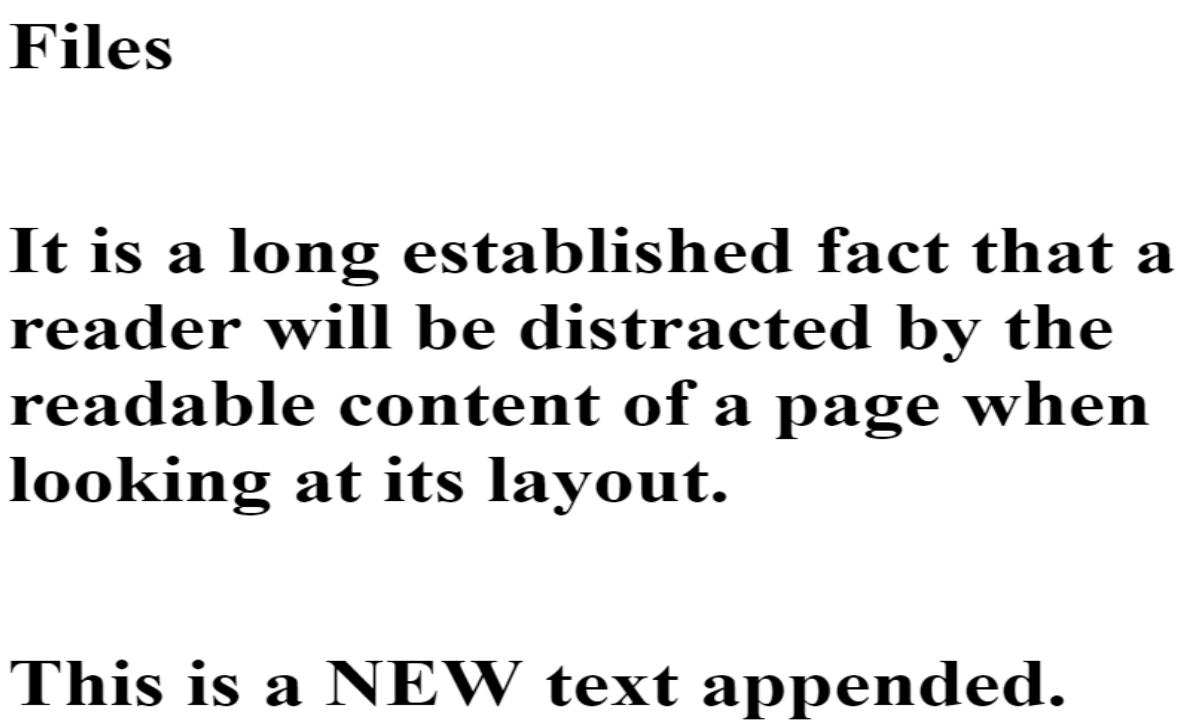
It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout.

Live Preview

Exercise 2

Download the Exercise 2

Exercise 2: Append some text to an existing file with '**ab**' mode.



Live Preview

5.4 Includes & Requires

Usage of Includes and Requires

PHP allows to break the code into small pieces of file and then include then in the main page.

These are statements that can be used:

include

include_once

require

require_once

include and require both help to include the file into another file.

If the included file is not available then **include** statement will ignore and continue to execute the other part of the code.

if the included file is not available then **require** statement will stop the execution of the program.

require_once or include_once will not import if the file is already included.

Sample Example

Download the Example

File: index.php

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="Page Description">
    <title>Includes</title>
</head>

<body>
<h1>Includes</h1>
<?php
```

```
    include 'functions.php';
    echo add(1, 2);

    require 'display.php';
?>

</body>
</html>
```

File: functions.php

```
<?php
    function add($a, $b){
        return $a + $b;
    }
?>
```

File: display.php

```
<?php
    echo '<br><h1>This is displayed from display.php</h1>';
?>
```

Live Preview

Exercise 1

Download the Exercise 1

Exercise 1: Use the same code from above and delete the display.php and functions.php in this exercise and see the error.

Observe the Warning and Fatal Error. Warning is coming from include and Fatal Error from require.

Includes

Warning: include(functions.php): failed to open stream: No such file or directory in **E:\Dropbox\BootCamp\Code\4-PHP\5-general-php\4-include-php\ex1\after\index.php** on line **15**

Warning: include(): Failed opening 'functions.php' for inclusion (include_path='.;C:\php\pear') in **E:\Dropbox\BootCamp\Code\4-PHP\5-general-php\4-include-php\ex1\after\index.php** on line **15**

Fatal error: Uncaught Error: Call to undefined function add() in E:\Dropbox\BootCamp\Code\4-PHP\5-general-php\4-include-php\ex1\after\index.php:16 Stack trace: #0 {main} thrown in **E:\Dropbox\BootCamp\Code\4-PHP\5-general-php\4-include-php\ex1\after\index.php** on line **16**

Live Preview

Exercise 2

Download the Exercise 2

Exercise 2: Put all the HTML code in index.html and include that file in the index.php

Live Preview

5.5 Libraries

Usage of Libraries

You can create a library file and put all the functions that you commonly use in this library file.

This is the common practice for any web development where you break the main program into smaller chunks of code and then include them in the main program.

You can use ***include*** statement to import this functions file in your page so that you can access those functions.

Sample Example

Download the Example

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="Page Description">
    <title>Libraries</title>
</head>

<body>
<h1>Libraries</h1>
<?php

    include 'calculator.php';

    define('INPUTVALUE1', 50);
    define('INPUTVALUE2', 23);

    echo 'Calculator : '. INPUTVALUE1 . ' AND ' . INPUTVALUE2 . '<br>';
    echo 'Addition: '. add(INPUTVALUE1, INPUTVALUE2) . '<br>';
    echo 'Minus: '. minus(INPUTVALUE1, INPUTVALUE2) . '<br>';
    echo 'Multiply: '. multiply(INPUTVALUE1, INPUTVALUE2) . '<br>';

?>
```

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

FileName: functions.php

```
<?php

    function add($a, $b){
        return $a + $b;
    }

    function minus($a, $b){
        return $a - $b;
    }

    function multiply($a, $b){
        return $a * $b;
    }

?>
```

Libraries

Calculator : 50 AND 23
Addition: 73
Minus: 27
Multiply: 1150

Live Preview

Exercise 1

Exercise 1: Create your own Library and use it. Continue using it for other projects as well.

Follow this practice to split the project into small files and include them.

6. FORMS

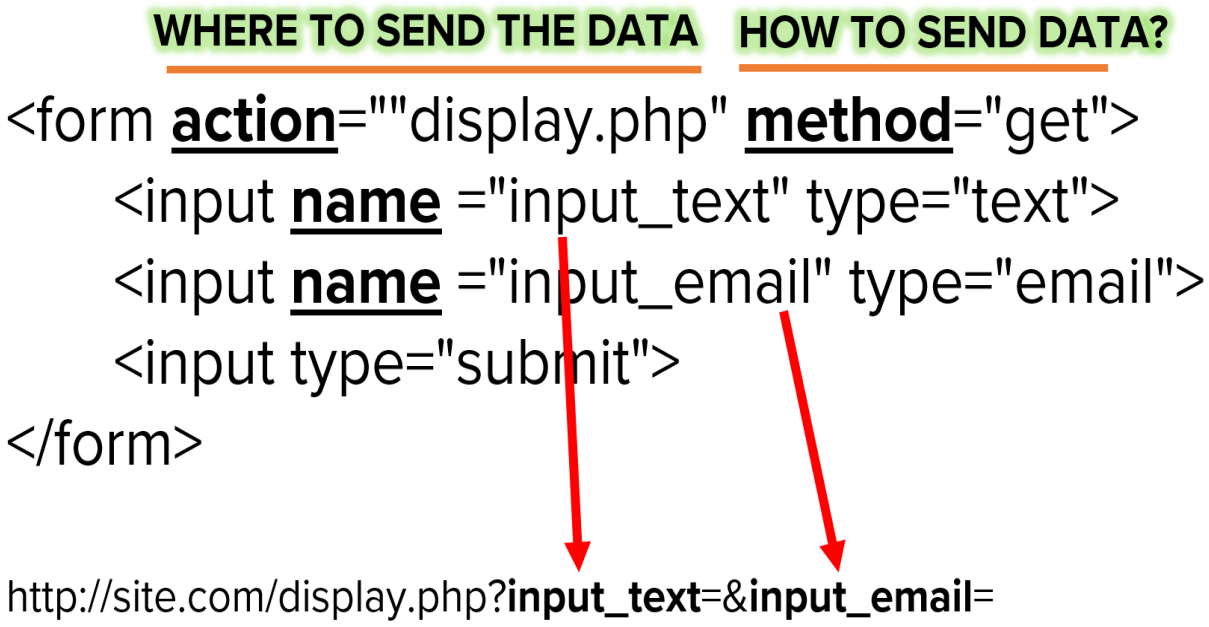
6 Forms

6.1 GET

Usage of Form GET Method

GET is type of method used by the form to pass the form data to the page that is mentioned in the action of the form.

GET method will send the data in the url.



When you define a form here are the important things:

action attribute – This define to which file this form data has to be sent to.

method attribute are of two types – GET and POST.

GET means data is visible in the URL bar.

POST means data is hidden.

Using GET we can send limited data and using POST we can send huge data.

<input> type has a **name** attribute which helps to define the name of the element. This name is like a variable that holds the data what user enters.

name attribute for form element is posted to the next page.

<input **type="submit"**> will show a submit button when clicked the form will be submit the data the file. This action is taken care by the browser.

How to Read the GET Variables from the URL

If the URL is something like:

http://site.com/display.php?**input_text**=hello&**input_email**=test@test.com

$text = $_GET['**input_text**'];
$emailid = $_GET['**input_email**'];

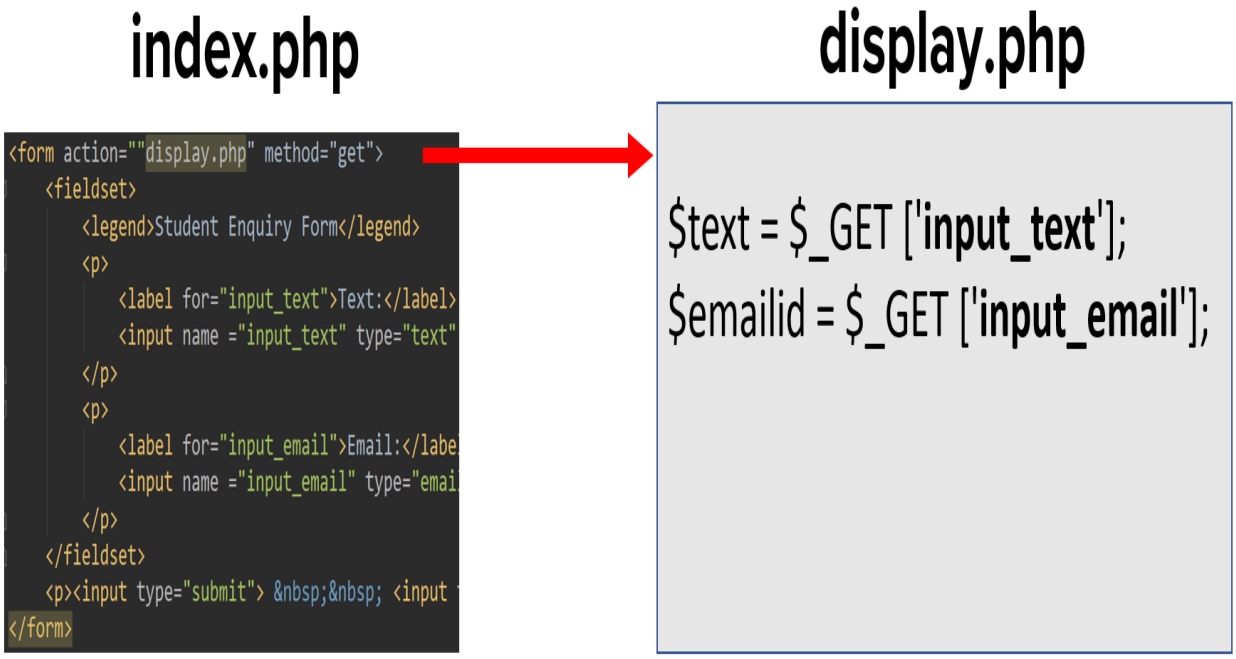
URL will have the name=value pairs separated with &.

$_GET[] is an array that gets created automatically with the GET parameters which you can access it and check if any value is there or not.

$_GET[] is super global variable which is always available even it is empty it is available.

When you click the submit button the form data is passed to display.php which read the parameters and display the output.

Browser will also go from index.php page to display.php page.



Sample Example

Download the Example

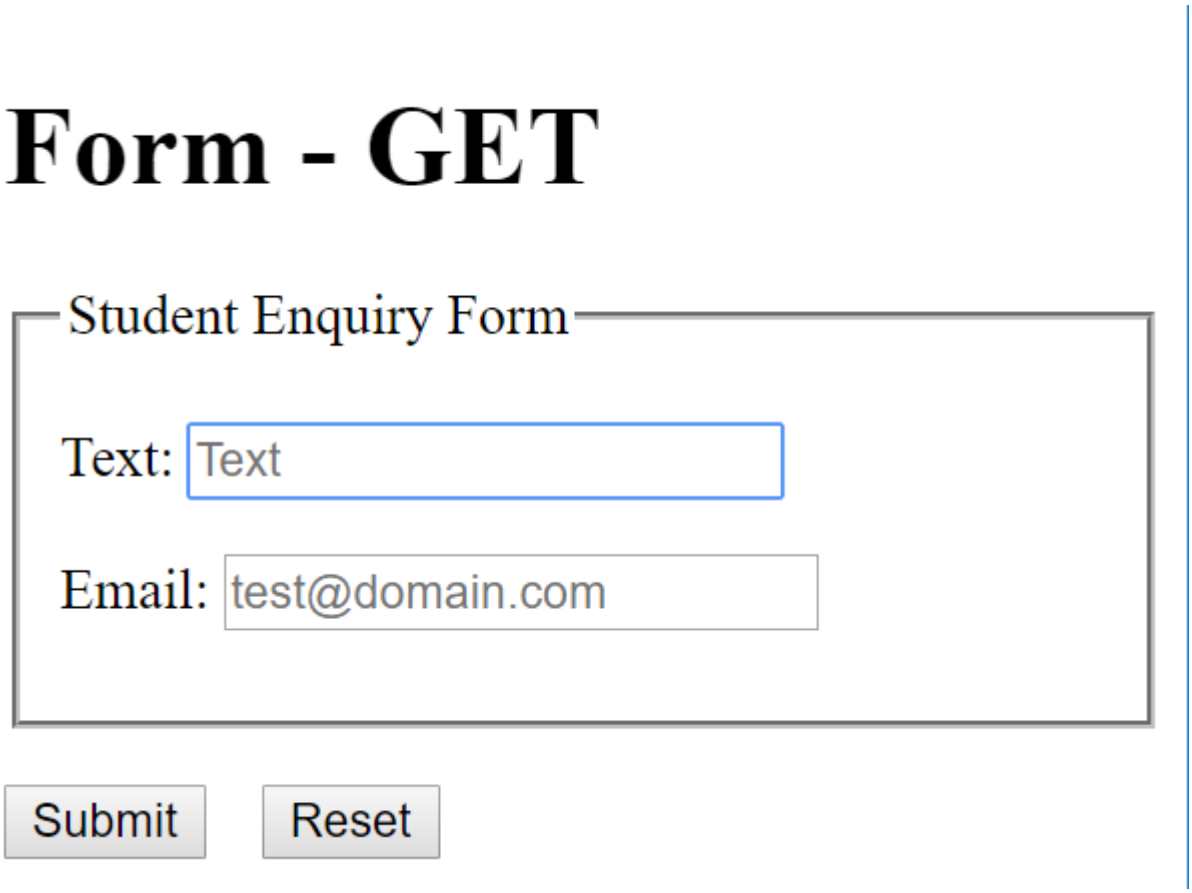
```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="Page Description">
    <title>Forms - GET</title>
```

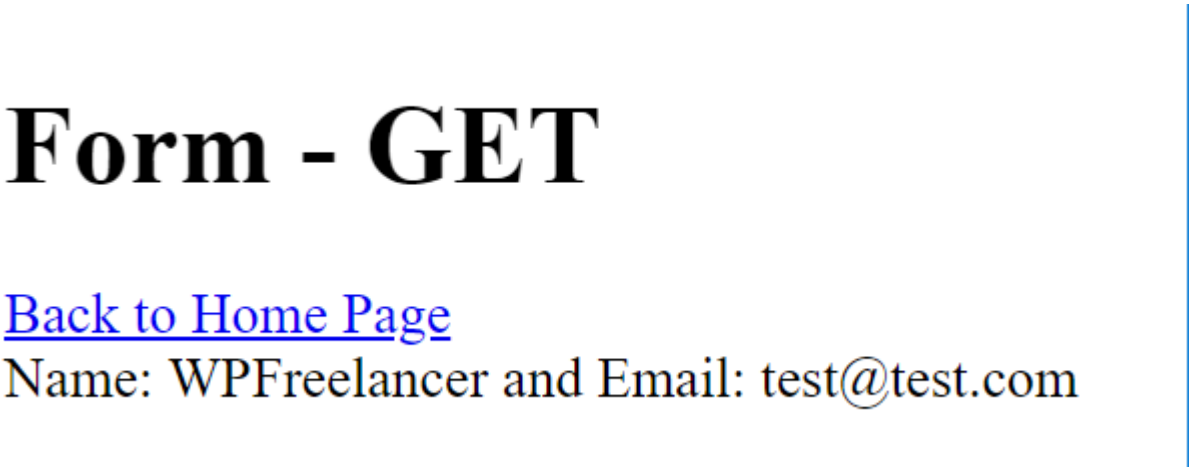
```
</head>


<body>
<h1>Form - GET</h1>
<form action="display.php" method="get">
    <fieldset>
        <legend>Student Enquiry Form</legend>
        <p>
            <label for="input_text">Text:</label>
            <input name ="input_text" type="text"
placeholder="Text">
        </p>
        <p>
            <label for="input_email">Email:</label>
            <input name ="input_email" type="email"
placeholder="test@domain.com">
        </p>
    </fieldset>
    <p><input type="submit">    <input type="reset"></p>
</form>
</body>
</html>
```

FileName: display.php

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

```
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="Page Description">
    <title>Forms - GET</title>
</head>

<body>

<h1>Form - GET</h1>
<a href="index.php">Back to Home Page</a><br>
<?php

    $name = $_GET['input_text'];
    $email = $_GET['input_email'];

    echo "Name: $name and Email: $email";
?>
</body>
</html>
```



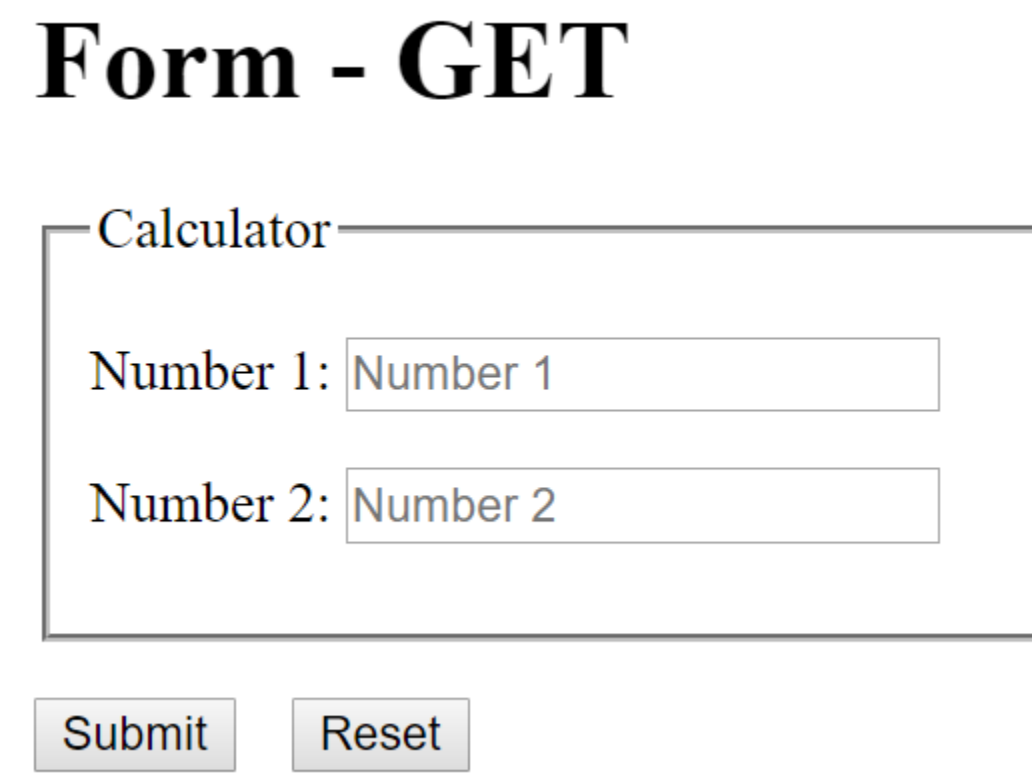


Live Preview

Exercise 1

Download the Exercise 1

Exercise 1: Create a Form that accepts two numbers and show the calculation of those two numbers on another page.



Live Preview

Exercise 2

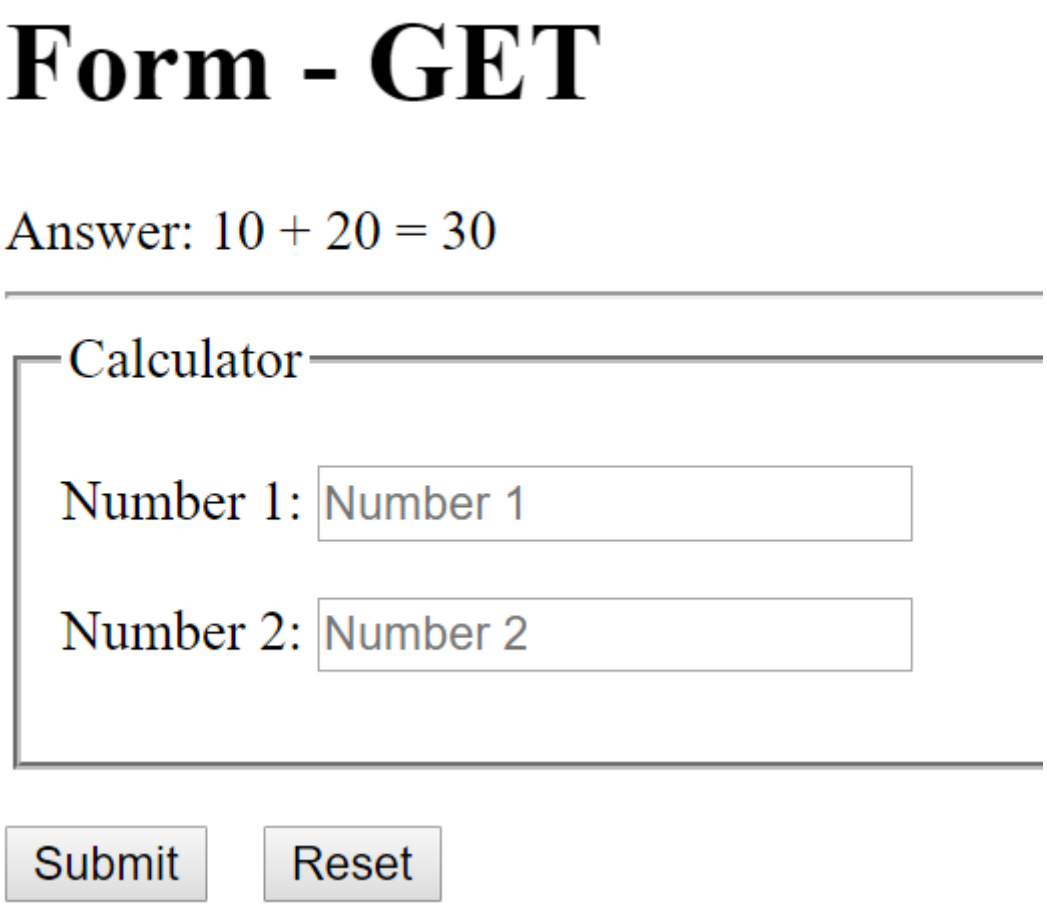
Download the Exercise 2

Exercise 2: Show the result on the same page and redirect to same page.

Tips:

Check if the values are not empty

```
if( !empty( $_GET['input_text1'] ) && !empty(
$_GET['input_text2'] ) ){
```



Live Preview

6.2 POST

Usage of Form POST Method

POST is type of method used by the form to pass the form data to the page that is mentioned in the action of the form.

POST method will send the data as an attachment. It is not visible in the URL.

WHERE TO SEND THE DATA HOW TO SEND DATA?

```
<form action=""display.php" method="post">
    <input name ="input_text" type="text">
    <input name ="input_email" type="email">
    <input type="submit">
</form>
```

Data is hidden inside the message sent to server.

When you define a form here are the important things:

action attribute – This define to which file this form data has to be sent to.

method attribute are of two types – GET and POST.

GET means data is visible in the URL bar.

POST means data is hidden.

Using GET we can send limited data and using POST we can send huge data.

<input> type has a **name** attribute which helps to define the name of the element. This name is like a variable that holds the data what user enters.

name attribute for form element is posted to the next page.

<input **type="submit"**> will show a submit button when clicked the form will be submit the data the file. This action is taken care by the browser.

How to Read the POST Variables

POST variable are not visible in the URL.

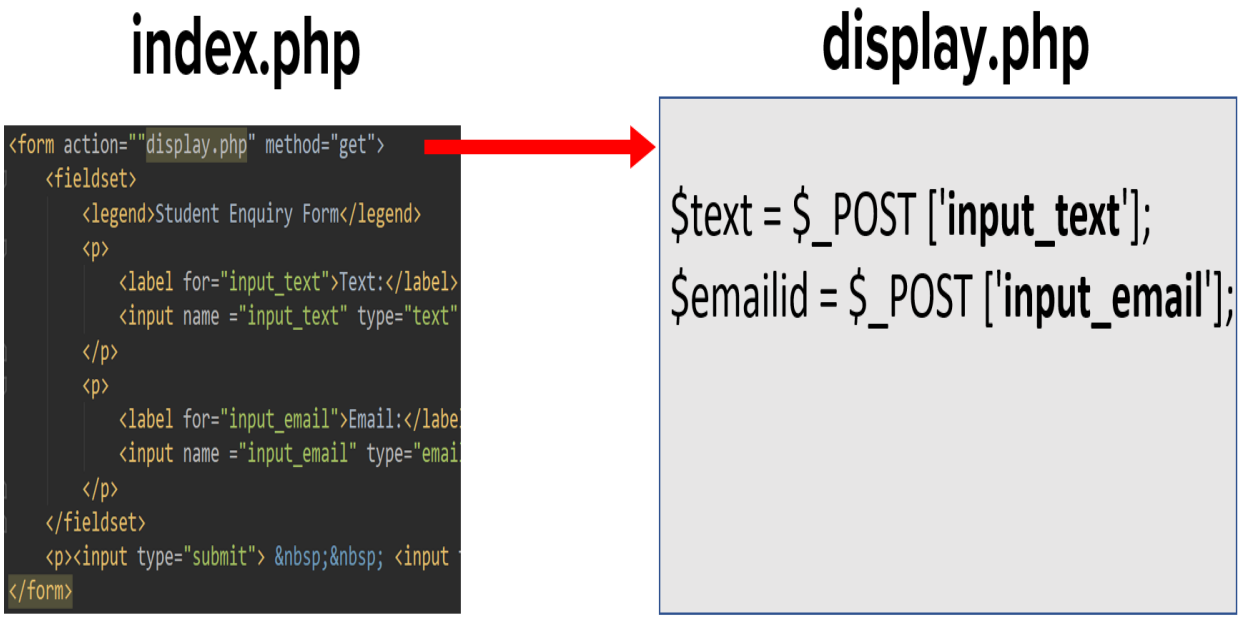
$text = $_POST['**input_text**'];
$emailid = $_POST['**input_email**'];

$_POST[] is an array that gets created automatically with the POST parameters which you can access it and check if any value is there or not.

$_POST[] is super global variable which is always available even it is empty it is available.

When you click the submit button the form data is passed to display.php which read the parameters and display the output.

Browser will also go from index.php page to display.php page.



Sample Example

Download the Example

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="Page Description">
    <title>Forms - POST</title>
</head>

<body>
<h1>Form - POST</h1>
<form action="display.php" method="post">
    <fieldset>
        <legend>Student Enquiry Form</legend>
        <p>
            <label for="input_text">Text:</label>
            <input name ="input_text" type="text" placeholder="Text">
        </p>
        <p>
            <label for="input_email">Email:</label>
```

```
            <input name ="input_email" type="email"
placeholder="test@domain.com">
        </p>
    </fieldset>
    <p><input type="submit">    <input type="reset"></p>
</form>
</body>
</html>
```

FileName: display.php

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <meta name="description" content="Page Description">
    <title>Forms - POST</title>
</head>

<body>

<h1>Form - POST</h1>
<a href="index.php">Back to Home Page</a><br>
<?php

    $name = $_POST['input_text'];
```

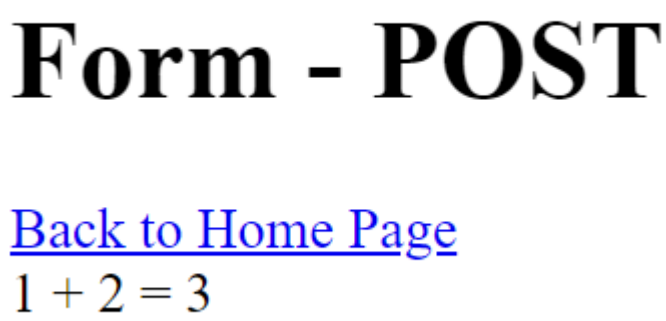
```
    $email = $_POST['input_email'];

    echo "Name: $name and Email: $email";
?>
</body>
</html>
```

Live Preview

Exercise 1

Download the Exercise 1

Exercise 1: Create a Form that accepts two numbers and show the calculation of those two numbers on another page. Use POST Method.



Live Preview

Exercise 2

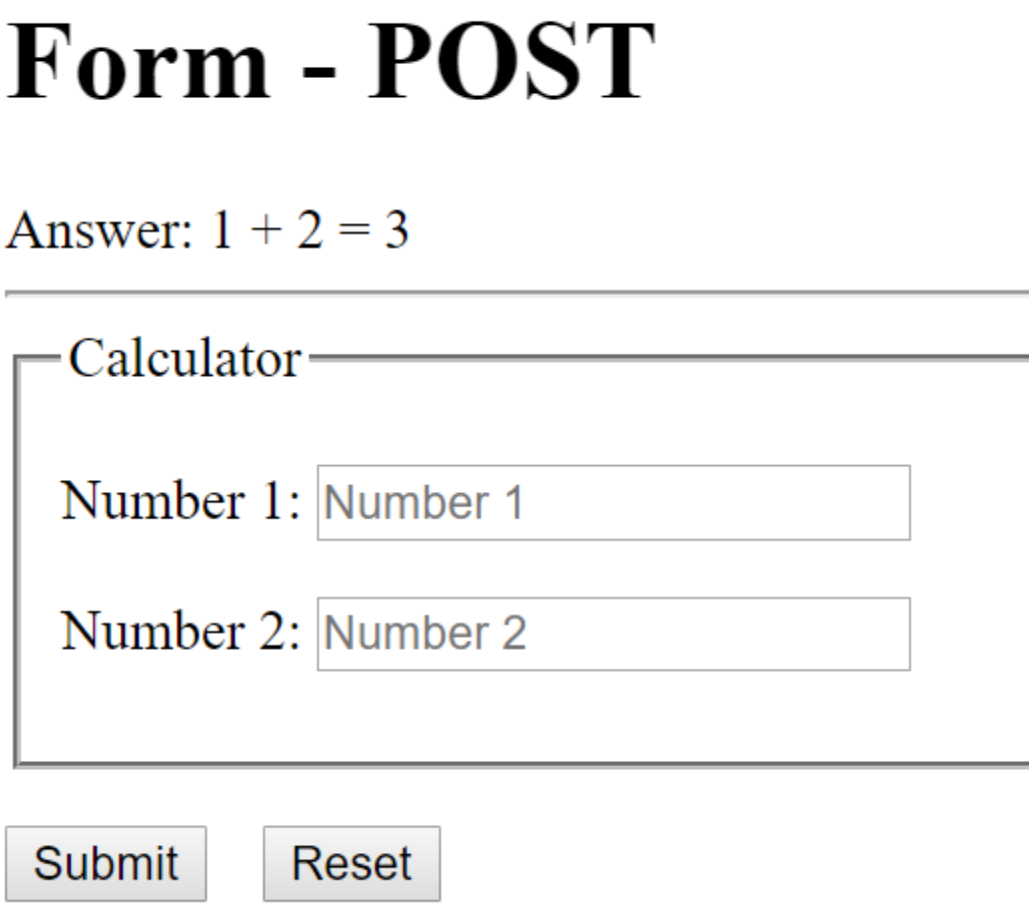
Download the Exercise 2

Exercise 2: Show the result on the same page and redirect to same page. Use POST Method.

Tips:

Check if the values are not empty

```
if( !empty( $_POST['input_text1'] ) && !empty(
$_POST['input_text2'] ) ){
```



Live Preview

6.3 Cookies

Usage of Cookies

Cookies are information that you can store at clients browser.

Cookies are stored in a file at the client system in name=value pair format.

Cookies helps to track what user is doing on the web page and send that information to server so that server knows what client did on the web page.

For every request, browser sends the cookies to server and if there are any changes to cookies then that information is also sent to the server.

Cookies helps to store information and capture the user actions on the web page inside it. This information is then sent to server.

Cookies last until the browser is closed. We can also manually set the expiration time for any cookie.

Some of the browser disable cookies in that case cookies will not work and also user can choose to change browser setting to not store cookies.

Cookies can help to change the view of the page based on the user actions.

How to Set a Cookie

$name = 'WPFreelancer.com';
$value = 20;
$expire = strtotime('+1 year');
$path = '/';
//Its a name=value pair.
setcookie($name, $value, $expire, $path);

How to Get a Cookie

```
$_COOKIE[$cookie_name]
```

How to Delete a Cookie

By setting the time last year all the cookies with that name are deleted.

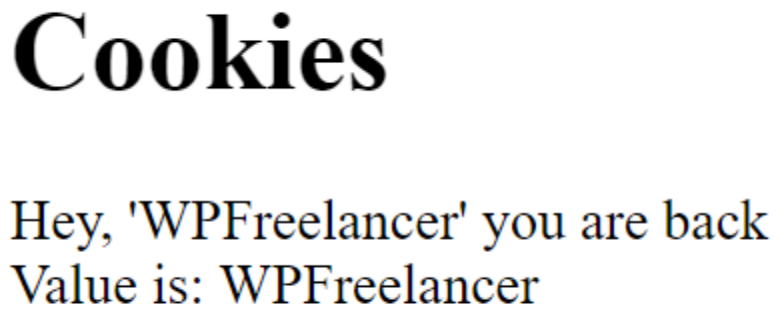
$expire = strtotime('-1 year');
setcookie('WPFreelancer.com', '', $expire, '/');

Sample Example

Download the Example

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="Page Description">
    <title>Cookies</title>
</head>

<body>
<h1>Cookies</h1>
<?php

    $cookie_name = "user";
    $cookie_value = "WPFreelancer";
    setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/");


    if(!isset($_COOKIE[$cookie_name])) {
```

```
        echo "Welcome '" . $cookie_value . "'. Nice to meet
you!";

    } else {

        echo "Hey, '" . $cookie_value . "' you are back<br>";
        echo "Value is: " . $_COOKIE[$cookie_name];

    }


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

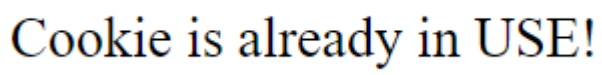


Live Preview

Exercise 1

Download the Exercise 1

Exercise 1: Ask user to enter the cookie name on one page and check on the other page if the cookie is new or old.



Live Preview

Exercise 2

Download the Exercise 2

Exercise 2: Add and check the cookie from same page.

Live Preview

6.4 Session

Usage of Sessions

Sessions is an array that is stored at the server based on the session id.

session id are generated when the user first time visit the site and this session id are stored in the cookie.

So, every time user make request to the server, this cookie is passed to the server with the sessionid and based this session id server is able to maintain an active session of the user.

If a session is already created for that specific user then PHP will not create a duplicate session.

By default, PHP uses a cookie to store a session ID in each browser. Then, the browser passes the cookie to the server with each request.

Start Session

session_start();

is used to start a session.

Create a Session Variable

```
session_start();
$_SESSION[“firstname”] = “WPFreelancer”;
```

Read a Session Variable

```
session_start();
echo $_SESSION[“firstname”];
```

Delete a Session Variable

```
session_start();
if(isset($_SESSION[“firstname“])){
unset($_SESSION[“firstname“]);
}
```

Sample Example

Download the Example

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="Page Description">
    <title>Sessions</title>
</head>

<body>
<h1>Sessions</h1>
<?php

    session_start();

    //Create a Session
    $_SESSION["firstname"] = "WPFreelancer";

    echo $_SESSION["firstname"];


?>
</body>
```

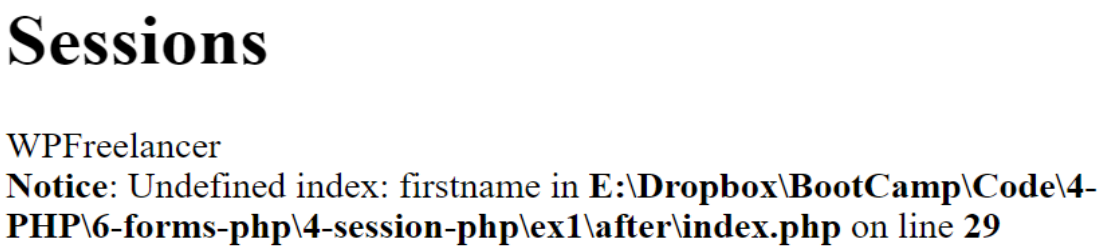
```
</html>
```

Live Preview

Exercise 1

Download the Exercise 1

Exercise 1: Delete the session and print it again.



Live Preview

Exercise 2

Download the Exercise 2

Exercise 2: Create a form and accept a session value and store it and display it.

Sessions

WPFreelancer Submit

Live Preview

7. Snippets

7 Snippets

7.1 Regex

Usage of Regex

Regular Expression are special searching pattern that is very powerful to search for matching patter in text strings.

To start using the Regular expression we need to get the data between forward slash and then use the method preg_match() on the data and this pattern to search.

You create a pattern and then use method preg_match() to match on the data.

preg_match() method will return true or false.

SYNTAX:

preg_match($pattern, $data);

$pattern = “/WP/’;

$sitename = “WP Freelancer”;

$found = preg_match($pattern, $sitename);

$found will have true or false.

Some patters:

. – Any Single Character

\w – Any Letter, number or underscore

\W – Any character

\d – Any Digit

\s – White Space Character

\S – Any Character that is not whitespace.

Sample Example

Download the Example

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="Page Description">
    <title>Regular Expressions</title>
</head>

<body>
<h1>Regular Expressions</h1>
<?php

    $pattern = '/WP/';
    $sitename = "WPFreelancer";
    $found = preg_match($pattern, $sitename);
    echo $found;
```

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

Live Preview

Exercise 1

Download the Exercise 1

Exercise 1: Search incase sensitive search for the same string.

Regular Expression: $pattern = ‘/wp/i’;

Live Preview

Exercise 2

Download the Exercise 2

Exercise 2: Match a pincode p-104848 in the string.

Regular Expression: $pattern = '/p-\d/';

Live Preview

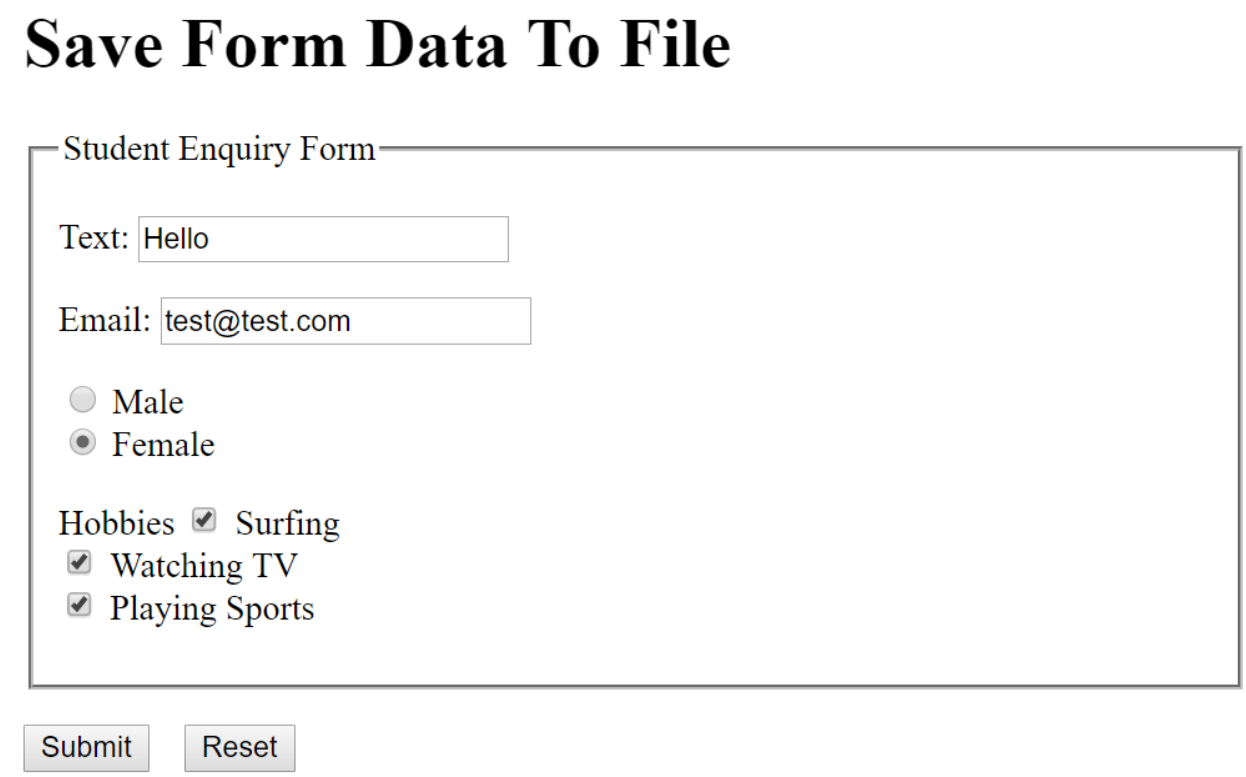
8. PROJECTS

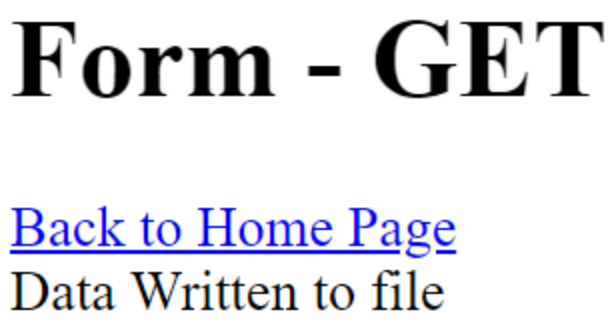
8 Projects

8.1 Save Student Registration Form Data to File

In this Project, you will write a simple application where user will his details and when submitted the data is save in a text file in append mode.

Download the Exercise 1





Live Preview

EXERCISE 1:

Become PHP Full Stack Web Developer in Just 30 Days

The above program will throw error when checkbox is not checked. This is on purpose.

You need find out first if the parameter exists in that array $_GET and then assign the value to variable.

Use the empty method to check if the variable is empty before inserting.

EXERCISE 2:

Make a function to save the content and save it in functions.php

Include the functions.php file and call this function.

8.2 Online Test

In this Project, you will write a simple application online test with some question. Once user provide the answer he immediately verify the result.

Download the Exercise 1

Online Test

Result: 30

Online Test

What is JS Stands for?

- (•) JavaScript
- ( ) JavaSight

What is HTML stands for?

- (•) HyperText Transport Protocol
- ( ) Hyper Text Transfer Protocol

What is CSS stands for?

- (•) Cascaded Style sheet
- ( ) Cascaded Single Sheet!

Submit Reset

Live Preview

EXERCISE 1:

Change the $_GET to $_POST and perform the same operation.

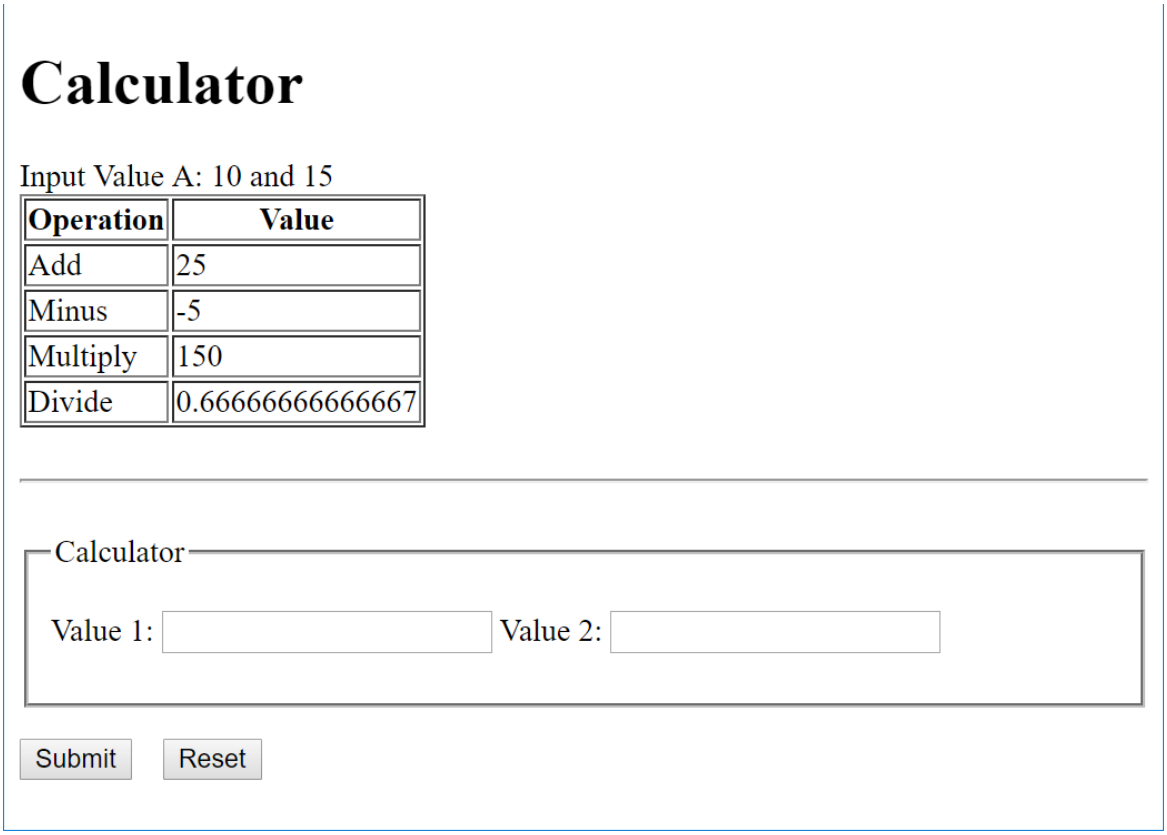
EXERCISE 2:

Display the result on the next page.

8.3 Online Calculator

In this Project, you will write a simple calculator application where user will enter the values and our program will calculate various math formulas.

Download the Exercise 1



[Live Preview]

EXERCISE 1:

Add some Math formulas and add to the table.

EXERCISE 2:

Add some styling to the table.

Become PHP Full Stack Web Developer in Just 30 Days