UNIT-IV

PHP (Hypertext Preprocessor):

- PHP Hypertext Preprocessor is a server scripting language, and a powerful tool for making dynamic and interactive Web pages.
- It allows web develop to create dynamic content that interacts with databases.
- **PHP** is Created in 1994 by Rasmus Lerdorf. Initially php it is used for track online visitors on web.

Basic Points about php

- PHP stands for HyperText Preprocessor.
- PHP files can contain text, HTML, CSS, JavaScript, and PHP code
- PHP files have extension ".php"
- PHP code are executed on the server so you need web browser to execute php code
- It is an interpreted language, i.e., there is no need for compilation.
- It is a server-side scripting language.
- It is faster than other scripting language e.g., asp and jsp.

Advantage of php

- It is open source.
- Widely used in all over the world
- Free to download

- It is executed on the server
- To execute php code no need compiler.

Features of php

- It is most popular and frequently used worldwide scripting language, the main reason of popularity is; It is open source and very simple.

1. Simple
2. Faster
3. Interpreted
4. Open Source
5. Case Sensitive
6. Simplicity
7. Efficiency
8. Platform Independent
9. Security
10. Flexibility
11. Familiarity
12. Error Reporting
13. Loosely Typed Language

Simple

- It is very simple and easy to use, compare to other scripting language it is very simple and easy, this is widely used all over the world.

Interpreted

- It is an interpreted language, i.e. there is no need for compilation.

Faster

- It is faster than other scripting language e.g. asp and jsp.

Open Source

- Open source means you no need to pay for use php, you can free download and use.

Platform Independent

- PHP code will be run on every platform, Linux, Unix, Mac OS X, Windows.

Case Sensitive

- PHP is case sensitive scripting language at time of variable declaration. In PHP, all keywords (e.g. if, else, while, echo, etc.), classes, functions, and user-defined functions are NOT case-sensitive.

Error Reporting

- PHP have some predefined error reporting constants to generate a warning or error notice.

Real-Time Access Monitoring

- PHP provides access logging by creating the summary of recent accesses for the user.

Loosely Typed Language

- PHP supports variable usage without declaring its data type. It will be taken at the time of the execution based on the type of data it has on its value.

Uses of php

- It is used for create dynamic website.
- To Interacting with web server (Apache etc)
- To interacting with any back-end / database server e.g. MySQL
- To interaction with the native file system of the OS
- To implement the business logical layers (one or more)
- It can Encrypt Data
- Access Cookies variable and set cookies
- Using php you can restrict user to access specific web page
- PHP usually used to output HTML code to the browser
- Used for connect web application with DataBase

- It is used for send and receive E-Mails.
- You can use PHP to find today's date, and then build a calendar for the month.
- If you host banner advertisements on your website, you can use PHP to rotate them randomly.
- Using php you can count your visitors on your website.
- You can use PHP to create a special area of your website for members.
- Using php you can create login page for your user.
  Using php you can add, delete, modify elements within your database thru PHP. Access cookies variables and set cookies.
- Using PHP, you can restrict users to access some pages of your website.
- It can encrypt data.
- PHP performs system functions, i.e. from files on a system it can create, open, read, write, and close them.
- It can handle forms, i.e. gather data from files, save data to a file.

Why use php

- It runs on different platforms such as Windows, Linux, Unix, etc.
- This language is very simple to learn and runs efficiently on the server side.
- It is compatible with almost all servers used today, such as Apache, IIS, etc.

- It supports many databases such as MySQL, Oracle, PostgreSQL etc.
- It is perfectly suited for Web development and can be embedded directly into the HTML code.
- PHP can also be used to create dynamic web pages.
- It is often used together with Apache (web server) on various operating systems. It can be also used with Microsoft's IIS on Windows.
- It is open source and it is free downloadable

How to Install PHP Server

To install PHP, we will suggest you to install AMP (Apache, MySQL, PHP) software stack. It is available for all operating systems. There are many AMP options available in the market that are given below:

- **WAMP** for Windows
- **LAMP** for Linux
- **MAMP** for Mac
- **SAMP** for Solaris
- **FAMP** for FreeBSD
- **XAMPP** (Cross, Apache, MySQL, PHP, Perl) for Cross Platform: It includes some other components too such as FileZilla, OpenSSL, Webalizer, OpenSSL, Mercury Mail etc.

Basic Syntax of PHP

You can run php code on any web browser. PHP script is executed on the server, and the plain HTML result is sent back to the browser.

Basic Syntax of PHP

- PHP code is start with **<?php** and ends with **?>**
- Every PHP statements end with a semicolon **(;)**.
- PHP code save with .php extension.
- PHP contain some HTML tag and PHP code.
- You can place PHP code any where in your document.

PHP Syntax

```
<?php
    // PHP code goes here
?>
```

PHP files save with .php extension and it contain some HTML and PHP code.

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

<h1>This is my first PHP code</h1>

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

</body>
</html>
```

Output

This is my first PHP code

Simple Example of PHP

```
<!DOCTYPE>
<html>
<body>
<?php
echo "<h2>My First PHP Code</h2>";
?>
</body>
</html>
```

Output

My First PHP Code

Variable in PHP

Variable is an identifier which holds data or another one variable and whose value can be changed at the execution time of script. In PHP, a variable starts with the $ sign, followed by the name of the variable.

Syntax to declare variable in php

```
$variablename=value;
```

Example of variable in php

```
<?php
    $str="Hello world!";
    $a=5;
    $b=10.5;
    echo "String is: $str <br/>";
    echo "Integer is: $x <br/>";
    echo "Float is: $y <br/>";
?>
```

Output

```
String is: Hello world!
Integer is: 5
Float is: 10.5
```

- Variable $str will hold the string value **Hello world!**, variable $x will hold the interger value **5** and variable $y will hold float value **10.5**.

Rules to declare variable in PHP

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

Constant in PHP

Constants are name or identifier that can't be changed during the execution of the script. In php constants are define in two ways;

  - Using define() function
  - Using const keyword

- In php decalare constants follow same rule variable declaration. Constant start with letter or underscore only.

Create a PHP Constant

- Create constant in php by using define() function.

Syntax

define((name, value, case-insensitive)

name: Specifies the name of the constant

value: Specifies the value of the constant

case-insensitive: Specifies whether the constant name should be case-insensitive. Default is false

In below example we create constant with case-sensitive

Example of constant in php

```
<?php
    define("MSG","Hello world!");
    echo MSG;
?>
```

Output

Hello world!

- In below example we create constant with case-insensitive

Example of constant in php

```
<?php
        define("MSG","Hello world!", true);
        echo msg;
?>
```

Output

```
Hello world!
```

Define constant using cons keyword in php

- The const keyword defines constants at compile time. It is a language construct not a function. It is bit faster than define(). It is always case sensitive.

Example of constant in php

```
<?php
        cons MSG="Hello world!";
        echo MSG;
?>
```

Output

Hello world!

Datatype in PHP

PHP data types are used to hold different types of data or values. PHP supports the following data types:

- String
- Integer
- Float
- Boolean
- Array
- Object
- NULL
- Resource

String

- A string is a sequence of characters, for example "Hello world!".

Syntax

```
<?php

    $x = "Hello world!";
    $y = 'Hello world!';

echo $x;
echo "</br>";
echo $y;
?>
```

Output

Hello world! Hello world!

Control Statements

For Loop in PHP

The for loop is used when you know in advance how many times the script should run. In php for loops execute a block of code specified number of times.

Syntax

```
for (initilation; condition; increment/decrement)
{
   code to be executed;
}
```

Example of for Loop in PHP

```
<?php

    for ($i = 0; $i <= 10; $i++)
    {
       echo "$i <br>";
    }

?>
```

Output

0 1 2 3 4 5 6 7 8 9 10

While Loop in PHP

The while loop is used when you don't know how many times the script should run. while loops execute a block of code while the specified condition is true same like for loop.

Syntax

```
while(condition)
{
    //code to be executed
}
```

Example of while Loop in PHP

```
<?php
    $n=0;
    while($n<=10)
    {
        echo "$n<br/>";
        $n++;
    }

?>
```

Output

0 1 2 3 4 5 6 7 8 9 10

do while Loop in PHP

do while loop is used where you need to execute code at least once. The do...while loop will always execute the block of code once, it will then check the condition, and repeat the loop while the specified condition is true.

Syntax

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

Example of do while Loop in PHP

```
<?php
$x = 0;
do
{
    echo "$x <br>";
    $x++;
```

```
} while ($x <= 10);

?>
```

Output

0 1 2 3 4 5 6 7 8 9 10

if else in PHP

In php if else statement is used to test condition. If conditon is true execute the code other wise control goes outside.

PHP If Statement

PHP if statement is executed if condition is true.

Syntax

```
if(condition)
{
//code to be executed
}
```

if Statement Example in PHP

```
<?php
$age=20;
if($age<18)
{
        echo "$You are Adult";
}

?>
```

Output

You are Adult

PHP If-else Statement

- PHP if-else statement is executed whether condition is true or false.

Syntax

```
if(condition)
        {
                //code to be executed if true
        }
else
        {
                //code to be executed if false
        }
```

if else Example in PHP

```
<?php
$num=10;
if($num%2==0)
{
        echo "$num is even number";
}
else
```

```
{
        echo "$num is odd number";
}

?>
```

Output

10 is even number

Break Statement in PHP

PHP break statement breaks the execution of current for, while, do-while, switch and for-each loop. If you use break inside inner loop, it breaks the execution of inner loop only..

Syntax

```
jump statement;
break;
```

PHP Break inside loop

Example of break in php

```
<?php
	for($i=1;$i<=10;$i++)
	{
		echo "$i <br/>";
		if($i==5)
	{
		break;
	}
	}

?>
```

Output

1 2 3 4 5

Switch Case in PHP

In php switch statement is used to perform different actions based on different conditions. In others word PHP switch statement is used to execute one statement from multiple conditions. It works like PHP if-else-if statement.

Syntax

```
switch(expression)
{
        case value1:
         //code to be executed
         break;
        case value2:
         //code to be executed
         break;
        ......
        default:
         code to be executed if all cases are not matched;
}
```

Switch case Example in PHP

```
<?php
        $num=5;
        switch($num){
        case 1:
        echo("Monday");
        break;
        case 2:
        echo("Tuesday");
        break;
        case 3:
        echo("Wednasday");
        break;
        case 4:
        echo("Thrusday");
        break;
        case 5:
        echo("Friday");
        break;
        case 6:
        echo("Sateday");
        break;
        case 7:
        echo("Sunday");
```

```
        break;

        default:
        echo("Please enter number between 1 to 7");
        }

?>
```

Output

Friday

PHP Functions

- PHP function is a piece of code that can be reused many times.
- It can take input as argument list and return value.
- There are thousands of built-in functions in PHP.
- In PHP, we can define **Conditional function**, **Function within Function** and **Recursive function** also.

Advantage of PHP Functions

Code Reusability: PHP functions are defined only once and can be invoked many times, like in other programming languages.

Less Code: It saves a lot of code because you don't need to write the logic many times. By the use of function, you can write the logic only once and reuse it.

Easy to understand: PHP functions separate the programming logic. So it is easier to understand the flow of the application because every logic is divided in the form of functions.

PHP User-defined Functions

We can declare and call user-defined functions easily. Let's see the syntax to declare user-defined functions.

Syntax

```
function functionname(){
    //code to be executed
    }
```

PHP Functions Example

Save File: function1.php

```
<?php
        function sayHello()
        {
                echo "Hello PHP Function";
  }
        sayHello(); //calling function
  ?>
```

Output:

Hello PHP Function

PHP Function Arguments

- We can pass the information in PHP function through arguments which is separated by comma.
- PHP supports **Call by Value** (default), **Call by Reference**, **Default argument values** and **Variable-length argument list**.

Let's see the example to pass single argument in PHP function.

File: functionarg.php

```
<?php
function sayHello($name)
{
        echo "Hello $name<br/>";
}
        sayHello("Vikash");
        sayHello("Virat");
        sayHello("Vijay");
?>
```

Output:

Hello Vikash

Hello Virat

Hello Vijay

PHP Call By Reference

- Value passed to the function doesn't modify the actual value by default (call by value). But we can do so by passing value as a reference.
- By default, value passed to the function is call by value. To pass value as a reference, you need to use ampersand (&) symbol before the argument name.

Let's see a simple example of call by reference in PHP.

File: functionref.php

```
<?php
function adder(&$str2)
{
        $str2 .= 'Call By Reference';
}
        $str = 'Hello ';
        adder($str);
        echo $str;
?>
```

Output:

Hello Call By Reference

PHP Function: Default Argument Value

- We can specify a default argument value in function. While calling PHP function if you don't specify any argument, it will take the default argument. Let's see a simple example of using default argument value in PHP function.

File: functiondefaultarg.php

```
<?php
  function sayHello($name="Sonoo"){
                echo "Hello $name<br/>";
  }
                sayHello("Rajesh");
                sayHello();       //passing no value
                sayHello("John");
  ?>
```

Output:

```
Hello Rajesh
Hello Sonoo
Hello John
```

PHP Function: Returning Value

Let's see an example of PHP function that returns value.

File: functiondefaultarg.php

```
<?php
        function cube($n){
        return $n*$n*$n;
}
        echo "Cube of 3 is: ".cube(3);
?>
```

Output:

```
Cube of 3 is: 27
```

PHP File and Directory Functions

PHP Comes with a whole range of file and directory manipulation functions.

These Predefined Function allow you to check file attributes like copy, move delete etc.

Function	What it Does
file_exists()	• Tests if a file or directory exists or not. This function work for file and Directory Both
filesize()	• Returns the size of a file in bytes
realpath()	• Returns the absolute path of a file
pathinfo()	• Returns an array of information about a file and its path
stat()	• Provides information on file attributes and its permissions
is_readable()	• Tests if a file is readable
is_writable()	• Tests if a file is writable
is_executable	• Tests if a file is executable
fopen()	• Open a file(Write mode of file why want to open)
fread()	• Read a file(Files must be opened in read mode)
fgets()	• Read first line of a file(File must be opened in read mode)
fgetc()	• Read first Character of a file(File must be opened in read

	mode)
fwrite()	• Write Contents/information inside a file(File must be opened in write mode)
file_put_contents()	• Write Contents/information inside a file(it accept 2 parameters, file name and content)
file_get_contents()	• Read information from a file(it accpet 1 parameters file name only)
is_file()	• Tests if a directory is a file
copy()	• Copies a file
rename()	• Rename a file
unlink()	• Deletes a file
include()	• Reads an external file into the current PHP Script
require()	• Reads an external file into the current PHP Script
include_once()	• Reads an external file into the current PHP Script(One time only)
require_once()	• Reads an external file into the current PHP Script(One time only)
fclose()	• Close a open file
is_dir()	• Tests if a directory entry is a directory
is_dir()	• Tests if a directory entry is a directory

mkdir()	• Creates a directory
rmdir()	• Removes a directory
opendir()	• Open a directory
readdir()	• Read a directory(It reads a signle file at a time, so must use loop to read entire directory)
scandir()	• Read a directory(Return result in array format)

PHP create file and write contents

- PHP comes with a couple of different ways to do this as well.

  To **write** inside the file first file must be opened with **mode**.

How to Open a file in PHP

- fopen( ) is used to open a file. You have to pass two parameters inside this function. The first one is file name which you want to open and second is mode (purpose to open) of the file.

  Syntax

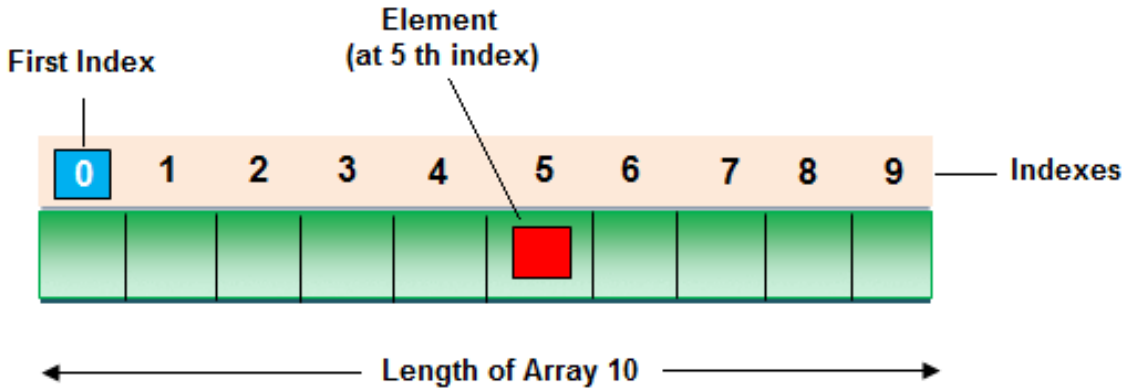
```
<?php

        $fo=fopen("filename","mode");

?>
```

Different types of File MODE

Modes	Description
w	• Write only. Opens and clears the contents of file; or creates a new file if it doesn't exist
w+	• Read/Write. Opens and clears the contents of file; or creates a new file if it doesn't exist
r	• Read only. Starts at the beginning of the file
r+	• Read/Write. Starts at the beginning of the file
a	• Append. Opens and writes to the end of the file or creates a new file if it doesn't exist
a+	• Read/Append. Preserves file content by writing to the end of the file
x	• Write only. Creates a new file. Returns FALSE and an error if file already exists
x+	• Read/Write. Creates a new file. Returns FALSE and an error if file already exists

Array in PHP

In PHP **Array** is used to store multiple values in single variable. An array is a special variable, which can hold more than one value at a time.



Create an Array in PHP

- In PHP, the **array()** function is used to create an array.

Array Syntax

```
array();
```

Types of Array in PHP

- There are three types of array in php, which are given below.
    1. Indexed arrays - Arrays with a numeric index
    2. Associative arrays - Arrays with named keys
    3. Multidimensional arrays - Arrays containing one or more arrays

Indexed Arrays

- The index can be assigned automatically (index always starts at 0), you can see in below example;

Array Example

```
<?php
    $student = array("Harry", "Varsha", "Gaurav");
    echo "Class 10th Students " . $student[0] . ", " . $student[1] . " and " .
    $student[2] . ".";
?>
```

Output

Class 10th Students Harry, Varsha and Gaurav

Find Length of an Array in PHP

Using **count()** function you can find length of an array in php.

Array Example

```
<?php

        $student = array("Harry", "Varsha", "Gaurav");
        echo "Length of Array: ";
        echo count($cars);

?>
```

Output

```
Length of Array: 3
```

Array Example using for Loop

Array Example

```
<?php

        $student = array("Harry", "Varsha", "Gaurav");
        $arrlength = count($student);

for($i = 0; $i < $arrlength; $i++)
{
        echo $student[$i];
        echo "<br>";
}

?>
```

Output

Harry

Varsha

Gaurav

Associative Arrays in PHP

- In this type of array; arrays use named keys that you assign to them.

Syntax

```
$age = array("Harry"=>"10", "Varsha"=>"20", "Gaurav"=>"30");
```

or

```
$age['Harry'] = "10";
$age['Varsha'] = "20";
$age['Gaurav'] = "30";
```

Multidimensional Arrays in PHP

- A multidimensional array is an array containing one or more arrays. For a two-dimensional array you need two indices to select an element

Array Example

```
<?php
        $student = array
         (
         array("Harry",300,11),
         array("Varsha",400,10),
         array("Gaurav",200,8),
         array("Hitesh",220,8)

 );

        $student = array("Harry", "Varsha", "Gaurav");
        echo $student[0][0].": Marks: ".$student[0][1].", Class:
        ".$student[0][2].".<br>";
        echo $student[1][0].": Marks: ".$student[1][1].", Class:
        ".$student[1][2].".<br>";
        echo $student[2][0].": Marks: ".$student[2][1].", Class:
        ".$student[2][2].".<br>";
        echo $student[3][0].": Marks: ".$student[3][1].", Class:
        ".$student[3][2].".<br>";
?>
```

How to Get data From Html Form using PHP

- For get data from from Html form you need $_POST function. First create one variable and pass filename in $_POST[] as a argument.

Syntax

```
$variable_name = $_POST["field_name"];
```

Example

```
<form action="" method="post">
        Username: <input type="text" name="username"
        placeholder="username"><br>
        Password: <input type="password" name="password"
        placeholder="password"><br>
        <button type="submit" value="submit">Submit</button>
</form>
```

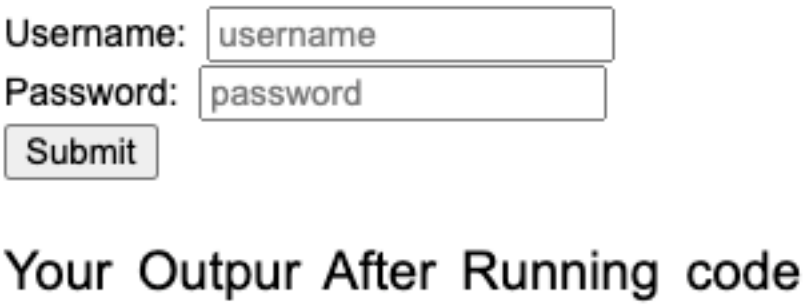
- Using above Html page you can fill your data in textfile, this is html file after doing this process we create new php file and get Html form data on php page using $_POST function and store this data in $name and $pass variable. because on Html page we are using post method so use $_POST function.
- At last print value on web page using **echo " ";**

Get Html textfield data using PHP

```
<?php
        $name = $_POST["username"];
        $pass = $_POST["password"];

        echo $name."</br>";
        echo $pass."</br>";

?>
```

Output

```
VIKASH
1288
```



Your Outpur After Running code

PHP Create Database MySQL

- Using below syntax we can create database in MYSQL.

Syntax

```
CREATE Database database_name;
```

connect-db.php

Connect to database

```
<?php
    $server = 'localhost';
    $user = 'user_name';
    $pass = 'password';
    $db = 'database_name';

// Connect to Database

    $connection = mysql_connect($server, $user, $pass)
    or die("Could not connect to server ... \n" . mysql_error ());

// select database
```

```
        mysql_select_db($db)
        or die("Could not connect to database ... \n" . mysql_error ());

?>
```

create-database.php

Create Database in database

```
<?php
        include 'connect-db.php';

        $sql = 'CREATE Database mydb';
        $result mysqli_query($sql, $connection);
if($result))
{
        echo "Database mydb created successfully.....";
}
else
{
        echo "Sorry, database creation failed ".mysqli_error($conn);
}

        mysqli_close($connection);
?>
```

Output

Database mydb created successfully.....

Create Table Mysql PHP

Table is the collection of row and column, using below SQL query we can create own table in database.

Create Table in database

```
create table table_name
(
    id int(5) NOT NULL auto_increment,,
    f_name varchar(20)
    l_name varchar(20)
)
```

- For create table in database using PHP webpage; first conncet to database then use below code.

Connect-db.php

Connect to database

```
<?php

        $server = 'localhost';
        $user = 'user_name';
        $pass = 'password';
        $db = 'database_name';

// Connect to Database

    $connection = mysqli_connect($server, $user, $pass)
    or die("Could not connect to server ... \n" . mysqli_error ());

// select database
    mysqli_select_db($db)
    or die("Could not connect to database ... \n" . mysqli_error ());
?>
```

create-table.php

Create Table in database

```
<?php
        include 'connect-db.php';

        $sql = "id int(5) NOT NULL auto_increment, f_name varchar(20), l_name
        varchar(20)";
        $result = mysqli_query($conn, $sql);
if($result)
{
         echo "Table user created successfully.....";
}
else
{
        echo "Could not create table: ". mysqli_error($conn);
}
        mysqli_close($conn);
?>
```

Output

Table user created successfully.....

How to Insert Record in Database using PHP

- For insert data in database using PHP first we create Html form page and get data from user, now connect database and run insert command.

index.html

Html page

```
<form action="insert-data.php" method="post" id="nameform">
First Name: <input type="text" name="fname"></br>
Last Name: <input type="text" name="lname"></br>
<button type="submit" form="nameform"
value="submit">Submit</button>
<button type="reset" value="Reset">Reset</button>
</form>
```

connect-db.php

Connect to database

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

// Create connection
    $conn = mysqli_connect($servername, $username, $password);

// Check connection

if (!$conn)
{
     die("Connection failed: " . mysqli_connect_error());
}

    echo "Connected successfully";

    mysqli_close($conn);

?>
```

insert-data.php

Example to Insert Record in Database using PHP

```
<?php
        include 'db-connect.php';

        $f_name = $_POST["fname"];
        $l_name = $_POST["fname"];

        $sql = "insert into user(f_name, l_name)
            VALUES('$f_name', '$l_name'";

        $result = mysql_query($sql);
if($result)
{
    header("Location: add-record-successfully");
}
else {
    echo mysql_error();
}

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

Delete Data from Database in PHP

To delete any record from database on webpage first you need to create user interface so here i am going to create Html page and search box. First search record from database if it present then record will delete. To delete any record from database we use below MYSQL query.

Syntax

```
delete from table_name where id=2;
```

connect-db.php

Connect to database

```
<?php

        $server = 'localhost';
        $user = 'user_name';
        $pass = 'password';
        $db = 'database_name';
```

// Connect to Database

```
$connection = mysqli_connect($server, $user, $pass)
or die("Could not connect to server … \n" . mysqli_error ());
```

// select database

```
mysqli_select_db($db)
or die("Could not connect to database … \n" . mysqli_error ());
```

```
?>
```

delete.html

Delete page

```
<?php
include 'db-connect.php';

mysqli_select_db("tutorohx_rw", $conncection);

$valueToSearch = $_POST['valueToSearch'];

$sql = "SELECT * FROM letter_dispatch WHERE s_no=$valueToSearch";
$result = mysqli_query($sql, $connection);

?>
```

```
<?php

if ($result)
 {
          while($row = mysqli_fetch_array($result))
 {

?>
```

// Show Record

```
          <h3>You Want to Delere Record:</h3>
          <table class="table" border='0'>
          <form method="post" action='delete-record-successfully.php'
          id="deleteform">
<tr><th>Serial no</th><td><span style="color:red; font-weight: bold; font-
size:20px;"><?php echo $valueToSearch;?></span></td></tr>
<tr><th></th><td><input type=hidden name=s_no value=<?php echo
$row['s_no']; ?>></td></tr>
<tr><td><input type="submit" form="deleteform" value="Delete"></td></tr>
</form>
```

```
<?php
 }
}
?>
</table>
```

delete-record-successfully.php

Record Deleted

```
<?php

        include 'db-connect.php';

        mysqli_select_db("tutorohx_rw", $conncection);

        $s_no = $_POST['s_no'];

        $sql = "DELETE FROM letter_dispatch where s_no='$s_no'";

        $result = mysqli_query($sql, $connection);
if($result)
{
         echo"Record Deleted Successfully.....";
}
else
```

```
{
        echo"Try Again.....";
}

?>
```

Servlet

- Servlet is a java program, exist and executes in j2ee servers, used to received the http protocol request, process it and send response to client.
- Using Servlet, we can collect input from users through web page forms, present records or another source, and create web pages dynamically.
- The Servlet specification was created by Sun Microsystems, with version 1.0 finalized in June 1997. The version 2.3, developed under the Java Community Process.

Why Use Servlet

- **Servlet** is a java program, exist and executes in j2ee servers, used to received the http protocol request, process and send response to client.
- Using Servlets, we can collect input from users through web page forms, present records from a database or another source, and create web pages dynamically.

Advantage of Servlet

- Better performance:
- Portability
- Robust
- Secure

Better performance:

- Because it creates a thread for each request not process (like CGI).

Portability:

- Because it uses java language and java is robust language.

Robust:

- Servlet are managed by JVM so no need to worry about memory leak, garbage collection etc.

Secure:

- Because it uses java language and java is a secure language.

Features of Servlet

- Servlet is a java program, exist and executes in j2ee servers, used to received the http protocol request, process and send response to client.
- Using Servlets, we can collect input from users through web page forms, present records from a database or another source, and create web pages dynamically. Some important features of servlet are; More secure than normal java code, portability, robust, secure, servlet code have better performance etc.

Features of Servlet

- Better performance:
- Portability
- Robust
- Secure
- Inexpensive
- Extensibility
- Efficiency
- Integration
- Persistent
- Server Independent
- Protocol Independent

- **Better performance:** Because it creates a thread for each request not process (like CGI).
- **Portability:** Because it uses java language and java is robust language.
- **Robust:** Servlet are managed by JVM so no need to worry about memory leak, garbage collection etc.
- **Secure:** Because it uses java language and java is a secure language. Java have automatic garbage collection mechanism and a lack of pointers protect the servlets from memory management problems.
- **Inexpensive** There are number of free web servers available for personal use or for commercial purpose. Mostly web server are very costly. So by using free web server you can reduce project development price.
- **Extensibility** The servlet API is designed in such a way that it can be easily extensible. Servlets being written in Java, can be extended and polymorphed into the objects that suits the user requirement.
- **Efficiency** Servlets invocation is highly efficient as compared to any CGI programs.

- **Integration** Servlets are tightly integrated with the server. Servlet can use the server to translate the file paths, check authorization, perform logging and MIME type mapping etc.
- **Persistent:** Servlets remain in memory until explicitly destroyed. This helps in serving several incoming requests. Servlets establishes connection only once with the database and can handle several requests on the same database.
- **Server Independent:** Servlets are compatible with any web server available today.
- **Protocol Independent:** Servlets can be created to support any protocols like FTP commands, Telnet sessions, NNTP newsgroups, etc. It also provides extended support for the functionality of HTTP protocol.
- **Fast:** Since servlets are compiled into bytecodes, they can execute more quickly as compared to other scripting languages. The bytecode compilation feature helps servlets to give much better performance. In addition, it also provides advantage of strong error and type checking.

What is Servlet API

Servlet API provides Classes and Interface to develop web based applications.

Package

- Servlet API contains two java packages are used to developed the servlet programs, they are:

1. javax.servlet
2. javax.servlet.http

javax.servlet

- **javax.servlet** package contains list of interfaces and classes that are used by the servlet or web container. These classes and interface are not specific to any protocol.

javax.servlet.http

- **javax.servlet.http** package contains list of classes and interfaces to define http servlet programs. This package are used to interact with browser using http protocol. It is only responsible for http requests.

Interfaces in javax.servlet package

- Servlet
- ServletRequest
- ServletResponse
- RequestDispatcher
- ServletConfig
- ServletContext
- SingleThreadModel
- Filter
- FilterConfig
- FilterChain
- ServletRequestListener
- ServletRequestAttributeListener
- ServletContextListener
- ServletContextAttributeListener

Classes in javax.servlet package

- GenericServlet
- ServletInputStream
- ServletOutputStream
- ServletRequestWrapper
- ServletResponseWrapper
- ServletRequestEvent
- ServletContextEvent
- ServletRequestAttributeEvent
- ServletContextAttributeEvent
- ServletException
- UnavailableException

Interfaces in javax.servlet.http package

- HttpServletRequest
- HttpServletResponse
- HttpSession
- HttpSessionListener
- HttpSessionAttributeListener
- HttpSessionBindingListener
- HttpSessionActivationListener
- HttpSessionContext (deprecated now)

Classes in javax.servlet.http package

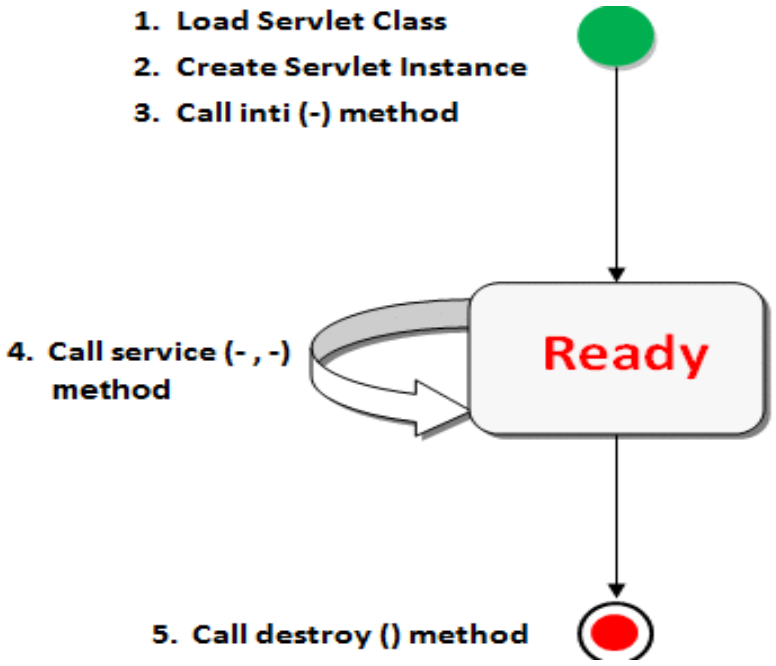
- HttpServlet
- Cookie
- HttpServletRequestWrapper
- HttpServletResponseWrapper
- HttpSessionEvent
- HttpSessionBindingEvent
- HttpUtils (deprecated now)

Life Cycle of Servlet

- The web container maintains the life cycle of a Servlet instance or object.

Life cycle of Servlet

- Loading (Servlet class is loaded)
- Installation (Servlet instance is created)
- Initialization (init method is invoked)
- Service providing (service method is invoked)
- Destroying (destroy method is invoked)
-



- As displayed in the above diagram, there are three states of a Servlet: new, ready and end.
- The Servlet is in new state if Servlet instance is created. After invoking the init() method, Servlet comes in the ready state.
- In the ready state, Servlet performs all the tasks. When the web container invokes the destroy() method, it shifts to the end state.

1. Servlet class is loaded

- The classloader is responsible to load the Servlet class. The Servlet class is loaded when the first request for the Servlet is received by the web container.

2. Servlet instance is created

- The web container creates the instance of a Servlet after loading the Servlet class. The Servlet instance is created only once in the Servlet life cycle.

3. init method is invoked

- The web container calls the init method only once after creating the Servlet instance. The init method is used to initialize the Servlet. It is the life cycle method of the javax.servlet.Servlet interface. Syntax of the init method is given below:

Syntax

- public void init(ServletConfig config) throws ServletException

4. service method is invoked

- The web container calls the service method each time when request for the Servlet is received. If Servlet is not initialized, it follows the first three steps as described above then calls the service method. If Servlet is initialized, it calls the service method. Notice that Servlet is initialized only once. The syntax of the service method of the Servlet interface is given below:

Syntax

```
public void service(ServletRequest request, ServletResponse response)
  throws ServletException, IOException
```

5. destroy method is invoked

- The web container calls the destroy method before removing the Servlet instance from the service. It gives the Servlet an opportunity to clean up any resource for example memory, thread etc. The syntax of the destroy method of the Servlet interface is given below:

Syntax

```
public void destroy()
```

Servlet Life Cycle Example

```
import javax.servlet.*;
import java.io.*;

public class myservlet extends GenericServlet
{
 public void init(ServletConfig sc)
{
  System.out.println("init executed...");
}
public void service(ServletRequest req, ServletResponse resp)throws IOException,
ServletException
{
System.out.println("service executed...");
PrintWriter out=resp.getWriter();
resp.setContentType("text/html");
out.println("plz observe output on server console window");
}
public void destroy()
{
 System.out.println("Distroy executed...");
}
}
```

web.xml

```
<web-app>
<servlet>
<servlet-name>srv</servlet-name>
<servlet-class>myservlet</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>srv</servlet-name>
<url-pattern>/ms</url-pattern>
</servlet-mapping>
</web-app>
```