

CHAPTER - 4

Part two Server Side Programming using PHP

1

PHP Arrays

- An array is a data structure that stores one or more similar type of values in a single value.
- There are three different kind of arrays and each array value is accessed using an ID which is called array index.
- Arrays can store numbers, strings and any object but their index will be pre-presented by numbers. By default array index starts from zero.
- **Numeric array** - An array with a numeric index. Values are stored and accessed in linear fashion
- **Associative array** - An array with strings as index. This stores element values in association with key values rather than in a strict linear index order.
- **Multidimensional array** - An array containing one or more arrays and values are accessed using multiple indices

Cont...

- **Example 1: Numeric arrays**

```
<html>
```

```
<body>
```

```
<?php
```

```
/* First method to create  
array. */
```

```
$numbers = array( 1, 2, 3,  
4, 5);
```

```
foreach( $numbers as  
$value )
```

```
{  
    echo "Value is $value  
    <br />";  
}
```

```
// Second method to create  
array.
```

```
$numbers[0] = "one";
```

```
$numbers[1] = "two";
```

```
$numbers[2] = "three";
```

```
$numbers[3] = "four";
```

```
$numbers[4] = "five";
```

```
foreach( $numbers as  
$value )
```

```
{  
    echo "Value is $value  
    <br />";  
}
```

Cont...

▪ Example 2: Associative Arrays

```
<html><body>
<?php      /* First method to associate create
array. */
$salary = S("Mohammad"=> 2000,"Qadir"=>
    1000,"Zara"=> 50);
echo "Salary of mohammad is ". $S['Mohammad'].
    <br/>";
echo "Salary of qadir is ". $S['Qadir']. "<br />";
echo "Salary of zara is ". $S['Zara']. "<br />";
      /* Second method to create array. */
$S['mohammad']="high";  $S['qadir']="medium";
    $S['zara']="low";
echo "Salary of mohammad is ". $S['mohammad'] . "<br
/>";
echo "Salary of qadir is "  $S['qadir'] "<br />".
```

Cont...

- **Example 3: Multidimensional Arrays**

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

```
<?php
```

```
$marks = array("Abe" => array("physics" => 35,  
    "maths" => 30),  
    "qadir" => array("physics" => 30,  
    "maths" => 32),  
    "zara" => array("physics" => 31,"maths"  
=> 22,));
```

```
/* Accessing multi-dimensional array values */
```

```
echo "Marks for mohammad in physics : " ;
```

```
echo $marks['mohammad']['physics'] . "<br />";
```

```
echo "Marks for qadir in maths: ". $marks['qadir']  
['maths']. "<br />";
```

```
echo "Marks for zara in chemistry :". $marks['zara']
```

PHP Strings

- They are sequences of characters, like "PHP supports string operations".
- Example:
 - `$string_1 = "This is a string in double quotes";`
 - `$string_2 = "This is a somewhat longer, singly quoted string";`

✚ ***String Concatenation Operator:-*** to concatenate two string variables together, use the dot (.) operator

- Example

```
<?php
$string1="Hello World";
$string2="1234";
echo $string1 . " " . $string2;
?>
```

Cont...

- ✚ ***Using the strlen() function:-*** the strlen() function is used to find the length of a string.

- Let's find the length of our string "Hello world!":

```
<?php  
echo strlen("Hello world!");  
?>
```

- ***Using the strpos() function:-*** the strpos() function is used to search for a string or character within a string.
- If a match is found in the string, this function will return the position of the first match. If no match is found, it will return FALSE.
- Let's see if we can find the string "world" in our string:

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

PHP Function

- A function is just a name we give to a block of code that can be executed whenever we need it.
- A function is a piece of code which takes one more input in the form of parameter and does some processing and returns a value.

Creating PHP Function:- when you are creating a function, follow these simple guidelines:

- Always start your function with the keyword **function**
- Remember that your function's code must be between the "{" and the "}"
- When you are using your function, be sure you

Cont...

- **Example:**

```
<html>
<head><title>Writing PHP
Function</title>
</head>
<body>
<?php      /* Defining a PHP Function */
function writeMessage(){
    echo "You are really a nice person, Have
    a nice time!";
}
writeMessage(); /* Calling a PHP
Function */
?>
```

Cont...

Function with parameter

- PHP gives you option to pass your parameters inside a function which work like variables inside your function.
- The following function with parameter returns the sum of two numbers.

```
<?php
function mySum($numX, $numY){
    $total = $numX + $numY;
    return $total;
}
$myNumber = 0;
echo "Before the function, myNumber = ". $myNumber . "<br />";
$myNumber = mySum(3, 4); // Store the result of mySum in
    $myNumber
echo "After the function, myNumber = " . $myNumber . "<br />";
?>
```

Cont...

Passing Arguments by Reference:

- Reference to the variable is manipulated by the function rather than a copy of the variable's value.
- Any changes made to an argument in these cases will change the value of the original variable.
- You can pass an argument by reference by adding an ampersand to the variable name in either the function call or the function definition.
- Example:

```
<html>
<head>
<title>Passing Argument by Reference</title>
</head>
<body>
```

Cont...

```
<?php
```

```
function addFive($num)
```

```
{ $num += 5; }
```

```
function addSix(&$num)
```

```
{ $num += 6; }
```

```
$orignum = 10;
```

```
addFive( &$orignum );
```

```
echo "Original Value is $orignum<br />";
```

```
addSix( $orignum );
```

```
echo "Original Value is $orignum<br />";
```

```
?>
```

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

Cont...

PHP Functions returning value:

- A function can return a value using the return statement in conjunction with a value or object.
- Return stops the execution of the function and sends the value back to the calling code.
- You can return more than one value from a function using return array(1,2,3,4).

- Example:

```
<html>
```

```
<head>
```

```
<title>Writing PHP Function which returns  
value</title>
```

```
</head>
```

```
<body>
```

Cont...

```
<?php
function addFunction($num1, $num2)
{
    $sum = $num1 + $num2;
    return $sum;
}
$return_value = addFunction(10, 20);
echo "Returned value from the function :
    $return_value
?>

</body>
</html>
```

Cont...

Setting Default Values for Function Parameters:

- You can set a parameter to have a default value if the function's caller doesn't pass it.

```
<html>
```

```
<head><title>Writing PHP Function which returns  
value</title>
```

```
</head><body>
```

```
<?php
```

```
function printMe($param = NULL)
```

```
{  print $param;  }
```

```
printMe("This is test");
```

```
printMe();  ?>
```

Cont...

Dynamic Function Calls:

- It is possible to assign function names as **strings to variables** and then treat these variables exactly as you would the function name itself.
- Example: `<html><head><title>Dynamic Function Calls</title>`

```
</head><body>
```

```
<?php
```

```
function sayHello()
```

```
{ echo "Hello<br />"; }
```

```
$function_holder = "sayHello";
```

```
$function_holder();
```

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

Cont...

Functions and Variable scope

- Refers to where the value of a variable is accessible to read/modify.
- There are local and global scopes
 - **Local scope** is inside a function only
 - **Global scope** is everywhere **except a function** with the exception of when it is imported into that function.
 - **SuperGlobals** are available by default inside and outside functions i.e. `$_POST`, `$_COOKIE`, `$_GET`
- Constants also have superglobal scope
- The variable inside the function is totally separated

Cont...

➤ Example:

- Unless we make a special declaration in a function the function will not have access to global variables.

```
<?php
function customer(){
    global $age; $age=21;  $name="Abebe";
    print "wel come $name. Are You still $age years
        old?";
}
customer();
echo $age;// $age is global variable so 21 will be
    here too
echo $name;// $name is local variable, Abebe will
    not be displayed here
?>
```

Cont...

PHP - Predefined Variables

- PHP provides a large number of predefined variables to any script which it runs.
- Also it provides an additional set of predefined arrays containing variables from the web server the environment, and user input which is called superglobals:
- **\$GLOBALS:** Contains a reference to every variable which is currently available within the global scope of the script. The keys of this array are the names of the global variables.
- **\$_SERVER:** This is an array containing information such as headers, paths, and script locations. The entries in this array are created by the web server. There is no guarantee that every web server will

Cont...

- **\$_GET:**
 - Associative array of variables passed to the current script via the HTTP GET method.
- **\$_POST:**
 - Associative array of variables passed to the current script via the HTTP POST method.
- **\$_FILES:**
 - Associative array of items uploaded to the current script via the HTTP POST method.
- **\$_REQUEST:**
 - An associative array consisting of the contents of \$_GET, \$_POST, and \$_COOKIE.

Cont...

- **\$_COOKIE:**

- An associative array of variables passed to the current script via HTTP cookies.

- **\$_SESSION:**

- An associative array containing session variables available to the current script.

- **\$_PHP_SELF:**

- A string containing PHP script file name in which it is called.

- **\$php_errormsg:**

- A variable containing the text of the last error message generated by PHP.

PHP FILE INCLUSION

- You can include the content of a PHP file into another PHP file before the server executes it.
- This helps in creating functions, headers, footers, or elements that can be reused on multiple pages.
- Used to type up a **common header** or **menu file** that you want all your web pages to include
- This will help developers to make it easy to change the layout of complete website with minimal effort. If there is any change required then instead of changing thousand of files just change included file.
- **Example:** changing or Updating the header/any content of your website which is common for most of the webpage of your site

Cont...

- PHP has four commands that can be used to include code **from an external file**, namely:
include() include_once()
require() require_once()
- Takes a file name and simply inserts that file's contents into the script that calls used the include function.
- **Example:** The following example includes scope.php as a part of the page

```
<html><body>  
<?php include("scope.php"); ?>  
<p>This is my home page that uses a common  
menu to save me time when I add  
new pages to my website!</p>  
</body></html>
```

Cont...

- **include()**

- Takes all the text in a specified file and copies it into the file that uses the include function
- If there is any problem in loading a file then the include() function generates a warning but the script will continue execution even if the external file is missing

- **require()**

- Takes all the text in a specified file and copies it into the file that uses the include function is used.
- If there is any problem in loading a file then the require() function generates a fatal error and halt the execution of the script.

- No difference between the function; except they handle error conditions. It is recommended to use the require() function instead of include(), because scripts should not continue executing if files are

Cont...

- The other two commands, **include_once()** and **require_once()**, work the same way, but they prevent the same file from being included more than once in a page.

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

```
<?php include_once("scope.php"); ?>
```

```
<p>scope.php will not be included in this age  
any more</p>
```

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

More Example:

- Assume you want to create a common menu for your website. Then create a file **menu.php** with the following content.

```
<a
```

```
href="http://www.tutorialspoint.com/index.htm">Ho
```

Cont...

- Now create as many pages as you like and include this file to create header. For example now your test.php file can have following content.

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

```
<?php include("menu.php"); ?>
```

```
<p>This is an example to show how to include  
PHP file!</p>
```

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

- **Example for including wrong file**

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

```
<?php require("xmenu.php"); ?>
```

```
<p>This example to show how to include  
wrong file!</p>
```

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

- **Note:** You may get plain warning messages or fatal

Cont...

- **Error Message for wrong file inclusion using `include()`:**

Warning:

main(xmenu.php):

failed to open stream: No such file or directory
in....

Hello World!

- **Error Message for wrong file inclusion using `require()`:**

Warning:

main(xmenu.php):

failed to open stream: No such file or directory
in....

Fatal error: main():

- Use **`include()`**:- if your page would remain usable

PHP WITH HTML FORMS

POST AND GET METHODS

- Before the browser sends the information, it encodes it using a scheme called URL encoding. In this scheme,
 - name/value pairs are joined with equal signs and
 - different pairs are separated by the ampersand. `name1=value1&name2=value2&name3=value3`
 - Spaces are removed and replaced with the + character and
 - any other nonalphanumeric characters are replaced with a hexadecimal values.
- After the information is encoded it is sent to the server.
- There are two ways the browser can send information to web server

Cont...

A) POST Method

- Transfers information via HTTP headers.
- The information is encoded as described above and put into a header called QUERY_STRING.
- Does not have any restriction on data size to be sent.
- It can be used to send ASCII as well as binary data.
- The data sent by POST method goes through HTTP header so security depends on HTTP protocol. By using Secure HTTP you can make sure that your information is secure.
- Syntax:- **<form action="process.php" method="POST">**
- HTML code specifies that the form data will be

Cont...

- The form names are used as the keys in the associative array.
- Example: the following html page

```
<html>
<body>
<form action="process.php" method="post">
    Name <input name="quantity" type="text" />
    <input type="submit" value="submit">
</form>
```
- Then, process.php can retrieve data as follows,
\$name=\$_post["name"];
- Then u can use the name of the user for any purpose what u want.
E.g. I can say, echo "Wel come \$name";

Cont...

More Example:

```
<?php
    if( $_POST["name"] || $_POST["age"] ) {
        echo "Welcome ". $_POST['name']. "<br />";
        echo "You are ". $_POST['age']. " years old.";
        exit(); }
?>
<html><body>
    <form action="<?php $_PHP_SELF ?>"
    method="POST">
    Name: <input type="text" name="name" />
    Age: <input type="text" name="age" />
    <input type="submit" />
    </form>
</body>
</html>
```

Cont...

B) GET Method

- It sends the encoded user information appended to the page request.
- The page and encoded information are separated by ? Character.

<http://www.test.com/index.htm?name1=value1&name2=value2>

- GET method is restricted to send up to 1024 characters only.
- Never use GET method if you have password or other sensitive information to be sent to the server.
- GET can't be used to send binary data, like images or word documents, to the server.

- The data sent by GET method can be accessed using QUERY STRING environment variable.

Cont...

- **Syntax:-** **<form action="process.php" method="GET">**
 <html> <body>
 <form action="process.php" method="post">
 <select
name="item">option>Sugar</option>... </select>
 <input name="quantity" type="text" />
 <input type="submit"
value="submit"></form>
- It passes the variables along to the "process.php" web page by appending them onto the end of the URL
- After clicking submit, The URL would have this added on to the end of it: "?"
 item=##&quantity=##"

Cont...

More Example:

```
<?php
    if( $_GET["name"] || $_GET["age"] ) {
        echo "Welcome ". $_GET['name']. "<br />";
        echo "You are ". $_GET['age']. " years old.";
        exit();
    }?>
<html><body>
    <form action="<?php $_PHP_SELF ?>"
    method="GET">
    Name: <input type="text" name="name" />
    Age: <input type="text" name="age" />
    <input type="submit" />
    </form>
</body>
</html>
```

PHP FORM PROCESSOR

- An HTML page form data can be processed using PHP codes externally as follows,

Order.Html

```
<html><body>
<h4>Tizag Art Supply Order Form</h4>
<form action="process.php"  method="post">
<select name="item">
<option>Paint</option>
<option>Brushes</option>
<option>Erasers</option></select>
Quantity: <input name="quantity" type="text" />
<input type="submit" />
</form>
</body>
</html>
```

Cont...

Process.php

```
<html><body>
<?php
$quantity = $_POST['quantity'];
$item = $_POST['item'];
echo "You ordered ". $quantity . " " . $item .
".<br />";
echo "Thank you for ordering from Tizag Art
Supplies!";
?>
</body>
</html>
```

Cont...

The \$_REQUEST variable

- It contains the contents of both \$_GET, \$_POST, and \$_COOKIE and It can be used to get the result from form data sent with both the GET and POST methods.

- Example:**

```
<?php
    if( $_REQUEST["name"] || $_REQUEST["age"]
    )
        { echo "Welcome ". $_REQUEST['name'].
" <br />";
            echo "You are ". $_REQUEST['age'].
" years old.";
            exit(); } ?>
```

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

```
<form action="<?php $_PHP_SELF ?>"
```

Cont...

Linking HTML with PHP

- This command tells the browser to send all data to registration.php when the submit button is clicked on HTML page.

***<form name="reg_frm"
action="registration.php"
method="POST">***

- We can also build HTML pages using PHP as follows

```
<?php echo"<html><body>";  
echo"<form method='POST'>";  
echo"What is your name<input type='text'  
name='name'>";  
echo"<input type='submit' value='submit'>";  
$nm=$POST["name"];  
echo"welcome $nm";
```

VALIDATION

1) Retrieving Form Data - Setting up Variables

- PHP provides a function called **IsSet("\$var")** that tests a variable to see whether it has been assigned a value.
- And there is also **empty("\$var")** function to check the value of the variable whether it has a value or not.

Example: From the above example we can check the variable \$name value

```
if(!IsSet($name))  
    {    echo"Error";    }
```

OR

```
if(empty($name)){  
    echo'<script type="text/javascript">'
```

Cont...

Accessing ComboBox, CheckBox and radio button value

- **Example:** After filling the following registration form the data will be processed in the server and display all the data back to the user.

<form name="regfrm"

action
method

Name

Dept

sex: ☐ Female
☒ Male

User Name

Password

Favorite courses

☒ Internet Programming

☐ Distributed Sysytem

☒ Advanced Programming

Cont...

- Before retrieving the data Remember that All form elements has their own value. `<option value="Maths">Mathematics</option>`
- In the HTML page the checkbox name must be array like this,

```
<html>
```

```
<body>
```

```
<form name="regfrm" action="registration.php"  
method="POST">
```

```
<input type="checkbox" name="course[]"  
value="InternetProgramming"><br>
```

```
<input type="checkbox" name="course[]"  
value="DistributedSystem"><br>
```

```
<input type="checkbox" name="course[]"  
value="AdvancedProgramming"><br>
```

```
</form>
```

Cont...

- Registration.php will have posted data as follows

```
<?php
$name = $_POST["name"];
$dept = $_POST["dept"];
$gender = $_POST["gender"];
$username = $_POST["userName"];
$password = $_POST["password"];
$course = $_POST["course"];
```
- We can validate whether they have value or not using the built in validation PHP functions

Cont...

- Then, the following page must be appear when the user clicks Register button

....

```
echo "Wel come $name";  
echo "Here is your informa";  
echo "Department: $dept";  
echo "User Name: $userNa";  
echo "Password: $password";  
echo "Favourite Courses are";  
$x=count($course);  
For($a=0; $a<$x; $a++)  
{   echo $course[$a] ."<br>"; }  
include("Login.html");  
?>
```

Wel come **Abebe**
Here is your Information:
Department: Maths
User name: **Abe**
Password: **123**
Favorite Courses are:
InternetProgramming
AdvancedProgramming
[Login](#)

Cookies and Session in PHP

- Both are used in terms of data storage in programming

1) Cookies

- Are text files stored on the client computer and they are kept for use tracking purposes.
- Cookies are usually set in an HTTP header (although JavaScript can also set a cookie directly on a browser).
- If the browser is configured to store cookies, it will then keep information such as name value pair, a GMT date, a path and a domain until the expiry date.
- If the user points the browser at any page that matches the path and domain of the cookie, it will resend the cookie to the server.
- A PHP script will then have access to the cookie in

Cont...

Setting Cookies with PHP:

- PHP provided ***setcookie()*** function to set a cookie.
- This function requires up to six arguments
setcookie(name, value, expire, path, domain, security);
- Setcookie() should be called before <html> tag and for each cookie this function has to be called separately.
- You can use ***isset()*** function to check if a cookie is set or not.
- **Name** - This sets the name of the cookie and is stored in an environment variable called HTTP_COOKIE_VARS which used while accessing cookies.
- **Expiry** - This specifies a future time in seconds since

Cont...

- **Value** -This sets the value of the named variable and the content that you actually want to store.
- **Path** -This specifies the directories for which the cookie is valid. A single forward slash character permits the cookie to be valid for all directories.
- **Domain** - This can be used to specify the domain name in very large domains and must contain at least two periods to be valid. All cookies are only valid for the host and domain which created them.
- **Security** - This can be set to 1 to specify that the cookie should only be sent by secure transmission using HTTPS otherwise set to 0 which mean cookie can be sent by regular HTTP.

Deleting Cookie with PHP

- Officially, to delete a cookie you should call `setcookie()` with the name argument only but this does not always work well, however, and should not

Cont...

Example: Creating Cookie (two cookies name and age these cookies will be expired after one hour.)

```
<?php
    setcookie("name", "John Watkin", time()+3600,
"/", "", 0);
    setcookie("age", "36", time()+3600, "/", "", 0);
?>
<html>
<head>
<title>Setting Cookies with PHP</title>
</head>
<body>
<?php echo "Set Cookies"?>
</body>
</html>
```

Cont...

Example: Accessing Cookies with PHP

```
<html>
<head><title>Accessing Cookies with
  PHP</title>
</head><body>
<?php
if( isset($_COOKIE["name"]) &&
    ( isset($_COOKIE["age"])))
    echo $_COOKIE["name"]. "<br />";    /* is
    equivalent to */
    echo $HTTP_COOKIE_VARS["name"]. "<br
    />";
    echo $_COOKIE["age"] . "<br />";    /* is
    equivalent to */
    echo $HTTP_COOKIE_VARS["name"] . "<br
    />";
```

Cont...

Example: Deleting Cookie

```
<?php
    setcookie( "name", "", time()- 60, "/", "", 0);
    setcookie( "age", "", time()- 60, "/", "", 0);
?>
<html>
<head>
<title>Deleting Cookies with PHP</title>
</head>
<body>
<?php echo "Deleted Cookies" ?>
</body>
</html>
```

Session

- An alternative way to make data accessible across the various pages of an entire website.
- Creates a file in a temporary directory on the server where registered session variables and their values are stored.
- The location of the temporary file is determined by a setting in the **php.ini** file called ***session.save_path***
- When a session is started following things happen:
 - PHP first creates a unique identifier for that particular session which is a random string of 32 hexadecimal numbers such as 3c7foj34c3jj973hjkop2fc937e3443.
 - A cookie called PHPSESSID is automatically sent to the user's computer to store unique session identification string.
 - A file is automatically created on the server in the

Cont...

- visitors are not able to edit Sessions since it stored on the server

Starting a PHP Session:

- Easily started by making a call to the **session_start()** function which first checks if a session is already started or not.
- It is recommended to put the call to **session_start()** at the beginning of the page(before the `<html>` tag).
- Session variables are stored in associative array called `$_SESSION[]` which can be accessed during lifetime of a session.
- Make use of **isset()** function to check if session variable is already set or not.
- If you wish to delete session data, use the **unset()** or the **session destroy()** function.

Cont...

Example: Start php session

```
<?php
    session_start();
    if(

isset( $_SESSION['counter'] ) )
    {
$_SESSION['counter'] +=
    1;
    }
    else {
$_SESSION['counter'] =
    1;  }
```

```
this page ".
$_SESSION['counter'];
$msg .= "in this
session.";
```

```
?>
```

```
<html>
<head>
<title>Setting up a
    PHP session</title>
</head>
<body>
<?php echo
    ( $msg ); ?>
</body>
</html>
```

Cont...

- **isset()** function to check if session variable is already set or not.

```
<?php
    unset($_SESSION['counter']);
?>
```

- You can completely destroy the session by calling the **session_destroy()** function:

```
<?php
    session_destroy();
?>
```

- **Note:**

Sessions and Cookies are useful mechanism to control the state of Login and Logout over website

PHP File Uploading

- A PHP script can be used with a HTML form to allow users to upload files to the server.
- Initially files are uploaded into a temporary directory and then relocated to a target destination by a PHP script.
- Information in the **phpinfo.php** page describes
 - Temporary directory used for file uploads as ***upload_tmp_dir***
 - Maximum permitted size of files uploaded as ***upload_max_filesize***.
 - These parameters are set into PHP configuration file **php.ini**
- An uploaded file could be a text file or image file or any document.

Cont...

The process of uploading a file follows these steps

- The user opens the page containing a HTML form featuring a text files, a browse button and a submit button.
- The user clicks the browse button and selects a file to upload from the local PC.
- The full path to the selected file appears in the text filed then the user clicks the submit button.
- The selected file is sent to the temporary directory on the server.
- The PHP script that was specified as the form handler in the form's action attribute checks that

Cont...

Sample upload form using HTML

```
<html>
<head><title>File Uploading Form</title>
</head>
<body>
<h3>File Upload:</h3>
Select a file to upload: <br />
<form          action="/php/file_uploader.php"
  method="post"
                                enctype="multipart/form-
  data">
<input  type="file"  name="file"  size="50"
  /><br />
<input type="submit" value="Upload File" />
</form>
```

Cont...

Creating an upload script:

- A Global variable '***\$_FILES***' which is an associate two dimension array and keeps all the information related to uploaded file.
- So if the value assigned to the input's name attribute in uploading form was file, then PHP would create following five variables:
 - ***\$_FILES['file']['tmp_name']***- the uploaded file in the temporary directory on the web server.
 - ***\$_FILES['file']['name']*** - the actual name of the uploaded file.
 - ***\$_FILES['file']['size']*** - the size in bytes of the uploaded file.
 - ***\$_FILES['file']['type']*** - the MIME type of the

Cont...

Example: code for upload script

```
<?php
if( $_FILES['file']['name'] != "" )
{
    copy(                                     $_FILES['file']['name'],
"/var/www/html" ) or
    die( "Could not copy file!");
}
else{
    die("No file specified!");
}

?>
```

Cont...

```
<html>
<head>
<title>Uploading Complete</title>
</head>
<body>
<h2>Uploaded File Info:</h2>
<ul>
<li>Sent file:  <?php echo  $_FILES['file']
                 ['name']; ?>
<li>File size:  <?php echo  $_FILES['file']
                 ['size']; ?> bytes
<li>File type:  <?php echo  $_FILES['file']
                 ['type']; ?>
</ul>
</body>
```

PHP - Date and Time Functions

- Allow you to get the date and time from the server where your PHP scripts are running.
- You can use these functions to format the date and time in many different ways.

Function PHP

Description

- `checkdate()`
3 Validates a Gregorian date
- `date_date_set()`
5 Sets the date
- `date_modify()`
5 Alters the timestamp
- `date_sunrise()`
5 Returns time of sunrise for a given day/ location
- `date_sunset()`
Returns the time of sunset

Cont...

Function PHP

Description

- `gmdate()` Formats a GMT/UTC date/time
3
- `idate()` Formats a local time/date as
integer 5
- `microtime()` Returns microseconds for current time
3
- `time()` Returns the current time as a
Unix timestamp
3
- `timezone_name_get()` Returns the name of
the timezone
5
- `timezone_offset_get()` Returns the timezone
offset from GMT

PHP -Regular Expressions

- Regular expressions are nothing more than a sequence or pattern of characters itself.
- Provide the foundation for pattern-matching functionality.
- Using regular expression you can search a particular string inside a another string, you can replace one string by another string and you can split a string into many chunks.
- PHP offers functions specific to two sets of regular expression functions, each corresponding to a certain type of regular expression.
 - POSIX Regular Expressions
 - PERL Style Regular Expressions

Cont...

POSIX Regular Expressions

1) Brackets ([]):

- Have a special meaning when used in the context of regular expressions.
- They are used to find a range of characters.

Example:

Expression	Description
-------------------	--------------------

- | | |
|---------|--|
| • [0-9] | It matches any decimal digit from 0 through 9. |
| • [a-z] | It matches any character from lowercase a through lowercase z. |
| • [A-Z] | It matches any character from uppercase A through uppercase Z. |
| • [a-Z] | It matches any character from lowercase a through uppercase Z. |

Cont...

2) *Quantifiers*:

- The frequency or position of bracketed character sequences and single characters can be denoted by a special character.
- Each special character having a specific connotation.

Example:

Expression

Description

- p^+ It matches any string containing at least one p .
- p^* It matches any string containing zero or more p 's.
- $p^?$ It matches any string containing zero or one p 's.
- $p\{N\}$ It matches any string containing a sequence of N p 's

Cont...

Examples:

- | Expression | Description |
|---------------------------------------|--|
| • <code>[^a-zA-Z]</code> | It matches any string not containing any of characters ranging from a through z and A through Z. |
| • <code>p.p</code> | It matches any string containing p, followed by any character, in turn followed by another p. |
| • <code>^.{2}\$</code> | It matches any string containing exactly two characters. |
| • <code>(.*</code> | It matches any string enclosed within <code></code> and <code></code> . |
| • <code>p(hp)*</code> | It matches any string containing a p followed by zero or more instances of the sequence hp. |

Cont...

3) Predefined Character Ranges

- Also known as character classes.
- It specifies an entire range of characters, for example, the alphabet or an integer set:

Expression	Description
• <code>[[:alpha:]]</code>	It matches any string containing alphabetic characters aA through zZ.
• <code>[[:digit:]]</code>	It matches any string containing numerical digits 0 through 9.
• <code>[[:alnum:]]</code>	It matches any string containing alphanumeric characters aA through zZ and 0 through 9.
• <code>[[:space:]]</code>	It matches any string containing a space.

Cont...

PHP's Regexp POSIX Functions

Function	Description
ereg()	searches a string specified by string for a string specified by pattern, returning true if pattern is found, else false
ereg_replace()	searches for string specified by pattern and replaces pattern with replacement if found.
eregi()	searches throughout a string specified by pattern for a string specified by string. The search is not case sensitive.
eregi_replace()	operates exactly like ereg_replace()

Cont...

Function

Description

- `split()` will divide a string into various elements, the boundaries of each element based on the occurrence of pattern in string.
- `spliti()` The `spliti()` function operates exactly in the same manner as its sibling `split()`, except that it is not case sensitive.
- `sql_regcase()` The `sql_regcase()` function can be thought of as a utility function, converting each character in the input parameter string into a bracketed expression containing two

Cont...

PERL Style Regular Expressions:- similar to their POSIX counterparts and used almost interchangeably

1) Metacharacters:- It is simply an alphabetical character preceded by a backslash that acts to give the combination a special meaning.

Character	Description
• .	a single character
• \s	a whitespace character (space, tab, newline)
• \S	non-whitespace character
• \d	a digit (0-9) and \D
non-digit	a
• \w	word character (a-z, A-Z, 0-9, <u>70</u>)
• \W	non-word character

Cont...

2) *Modifiers*

- Several modifiers are available that can make your work with regexps much easier, like case sensitivity, searching in multiple lines etc.

Modifier	Description
----------	-------------

- | | |
|-----|--|
| • i | Makes the match case insensitive |
| • m | Specifies that if the string has newline or carriage return characters, the ^ and \$ operators will now match against a newline boundary, instead of a string boundary |
| • o | Evaluates the expression only once |
| • s | Allows use of . to match a newline character |
| • x | Allows U 2 use whitespace in the expression for clarity |

Cont...

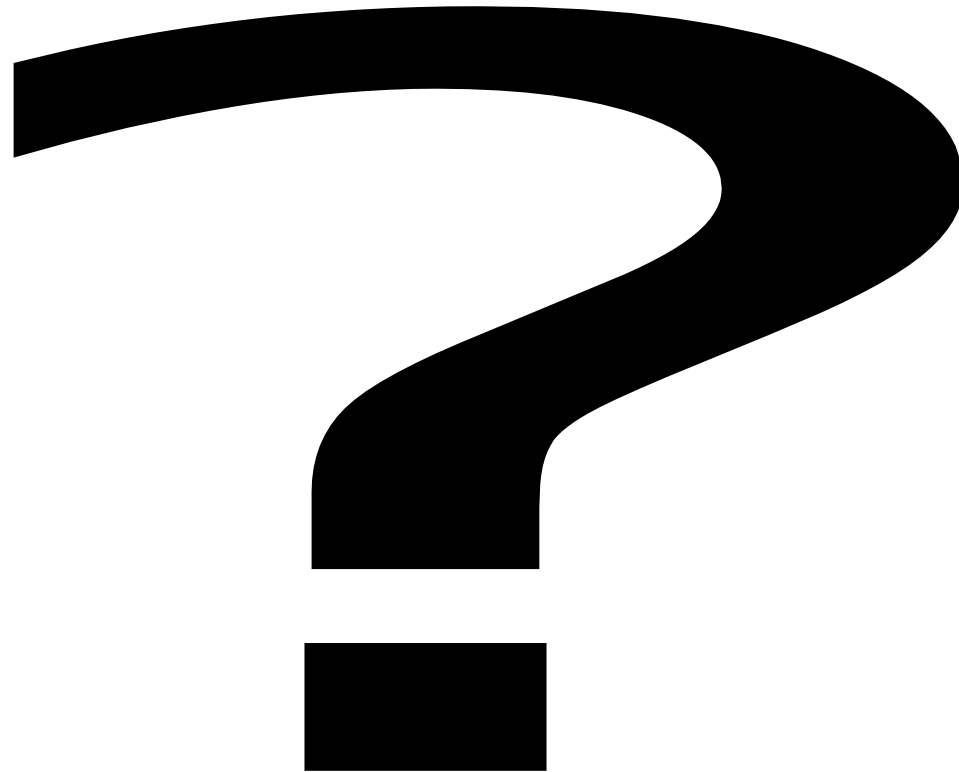
PHP's Regexp PERL Compatible Functions

Function

Description

- `preg_match()` searches string for pattern, returning true if pattern exists, else false.
- `preg_match_all()` matches all occurrences of pattern in string.
- `preg_replace()` operates like `ereg_replace()`, except that Regexp can be used in pattern and replacement input parameters.
- `preg_split()` operates like `split()`, except Regexp are accepted as input for pattern.
- `preg_grep()` searches all elements of input array, returning

QUESTIONS



**Thank
U!**