***PHP handling file uploads - Dec 13, 2008 at 22:00 PM by Rajmeet Ghai***

PHP handles file uploads through different method.

**POST method uploads**: This allows user to upload both text and binary files. PHP has a number of authentication and file manipulation functions, a control over upload is possible.

Files can be uploaded in PHP by using the tag type=”file”. An upload form must have encytype="multipart/form-data" , method also needs to be set to method="post". Also, hidden input MAX\_FILE\_SIZE before the file input. To restrict the size of files

E.g.

<form enctype="multipart/form-data" action="sampleuplaod.php" method="POST">
<input type="hidden" name="MAX\_FILE\_SIZE" value="1000" />

Finally, using move\_uploaded\_file () the file can be uploaded. Parameters will be source file and destination file

**PHP security tips**

* Avoid the use of global variables. Hence it must be ensured that register\_globals option is not enabled.
* Use of variables designed to be set by GET or POST requests.
* Store passwords in an encrypted format
* Avoid storing credit card and other secured information. Trust a third party gateway.
* Make use of server side validations and avoid trusting the user input.
Example: if the expected value is integer, use the intval function.
$post\_id = intval($\_GET['post\_id']);
mysql\_query("SELECT \* FROM post WHERE id = $post\_id");
* Avoid using user input directly in the query. Mysql\_real\_escape\_string()
* Always use the updated version of php.

**How can we encrypt the username and password using PHP?**

User names and passwords in PHP can be encrypted using md5 function.
MD5 function calculates the md5 hash of a string. It is basically used for encryption. It is also used for digital signature applications, where a large file must be "compressed" in a secure manner.

Example:

Md5($str);

Crypt() function can also be used to encrypt a string,. It used MD5, DES or blow fish algorithms for encryption.

Syntax:

Crypt(str, salt)

Salt is an optional parameter used to increase the number of characters encoded, to make the encoding more secure

**Explain the changing file permission and ownership using PHP's chmod() function.**

Chmod() is used for changing permissions on a file.

Syntax:

Chmod(file, mode)

Mode here specifies the permissions as follows:

* The first number is always zero
* The second number specifies permissions for the owner
* The third number specifies permissions for the owner's user group
* The fourth number specifies permissions for everybody else

Possible values (to set multiple permissions, add up the following numbers)

* 1 = execute permissions
* 2 = write permissions
* 4 = read permissions

Example:
// everything for owner, read for owner's group
chmod("test.txt",0740);

**PHP sessions introduction**

When a user logs in an application, his details are usually stored in a session variable. This information is available to all pages in one application. Sessions in PHP work using a unique id for each visitor.

**Starting a php session**: This tag must also appear before the HTML tag.

Session\_start();

**Storing a session variable**: Here in the sample variable value is set to 1

Session\_start();

$\_session[‘sample]=1;

**Destroying a session**:

Session\_destory();

**What is session\_start() ?**

When a user logs in an application, his details are usually stored in a session variable. This information is available to all pages in one application. Sessions in PHP work using a unique id for each visitor.

Starting a php session: This tag must also appear before the HTML tag.

Session\_start();

Storing a session variable: Here in the sample variable value is set to 1

Session\_start();

$\_session[‘sample]=1;

**What is session hijacking?**

Session hijacking is the misuse of a valid computer session. It is used to attain unauthorized and illegal access to a system. This access is attained using the “brute force” attack where in he tries multiple id’s to login in a system while the session is in progress. The most common method of session hijacking is IP spoofing where an attacker uses source-routed IP packets to insert commands into an active communication between two systems on a network and pretending itself as one of the authenticated users.

**What is meant by Session Clustering?**

Session clustering is used to provide scalability for keeping the session data in synch across a “cluster” of PHP servers. The sessions reside on the machine in which they are created. These sessions are then delivered from one machine to another. This delivery is fully distributed. The Zend Session manager is used for transferring sessions from the system (session storage) to remote addresses.

**How many ways I can register the variables into session?**

Global variables in PHP can be registered using the session\_register() function. It accepts different number of arguments, any of which can be either a string holding the name of a variable or an array consisting of variable names or other arrays

Example:
Session\_register(“smple”);

$\_session can also be used for registering variables.

Example:
$\_SESSION['count'] = 0;

**How many ways can we get the value of current session id?**

Using session\_id() function, the current value of the session can be found.

Syntax:
String session\_id(string $id);

**Short note on Starting a session**

* **Adding session data.**Here in the sample variable value is set to 1
<?php
       Session\_start();
       $\_session[‘sample]=1;
?>
* **Reading session data**Once the data is set, it immediately becomes available to read in the $\_SESSION array.
<?php
       $\_SESSION[‘sample’]=1;
       Print $\_SESSION [‘sample’];
?>
* **Removing session data**The session data can be removed using the unset() function. Only specific elements of the $\_SESSION array should be unset.
<?php
$\_SESSION[‘sample’]=1;
Print $\_SESSION [‘sample’];
Unset ($\_SESSION[‘sample’);
?>
* **Ending a session**A session lasts until the browser window is not closed. In order to explicitly end the session Session\_destory(); is used for ending the session.

***PHP cookies - Dec 13, 2008 at 22:00 PM by Rajmeet Ghai***

A cookie is used for identification purposes. It is more commonly used to identify a user in a session. It is a small file the application inserts on the users computer. With PHP one can create and retrieve the cookie.

**Setting cookie in php:**

Cookies in PHP can be set using the setcookie() function. This must appear before the HTML tag.

**Syntax:**

Setcookie(name, value, expire, path, domain);

**Example**: here, the cookie name sample is assigned a value jim. The cookie expires after an hour.

Setcookie(“sample”, “jim”, time()+3600);

**Retrieving cookie value:**

The cookie that is set can be retrieved as shown below:

Echo $\_cookie[“user”];

Isset() function can be used to find if the cookie is set.

#### What is a Persistent Cookie?

Cookies are used to remember the users. Content of a Persistent cookie remains unchanged even when the browser is closed. ‘Remember me’ generally used for login is the best example for Persistent Cookie.

#### How to set cookies? How to reset/destroy a cookie?

**Setting cookie in php:**

Cookies in PHP can be set using the setcookie() function. This must appear before the HTML tag,

Syntax:

Setcookie(name, value, expire, path, domain);

Example: here, the cookie name sample is assigned a value jim. The cookie expires after an hour.

Setcookie(“sample”, “jim”, time()+3600);

**Reset/destroy cookie**

Cookies can be deleted either by the client or by the server. Clients can easily delete the cookies by locating the Cookies folder on their system and deleting them. The Server can delete the cookies in two ways:

* Reset a cookie by specifying expiry time
* Reset a cookie by specifying its name only

##### PHP cookies - October 01, 2009 at 17:00 pm by Vidya Sagar

#### How to set cookies in PHP?

The function setcookie() is used to define a cookie that is to be sent along with HTTP headers. The cookie must be sent prior to any output from the script as is the protocol restriction. After setting the cookies, they can be used when the next page is loaded by using $\_COOKIE or $HTTP\_COOKIE\_VARS arrays.

***PHP class - Dec 13, 2008 at 22:00 PM by Rajmeet Ghai***

PHP classes help to deal with data in an organized fashion. Every class definition in PHP begins with the keyword “class”. This is followed by the name of the class. Curly braces mark the beginning and end of the class.

Example: $ this is a pseudo variable which is used when a method is called from within an object context. It is a reference to the calling object.

<?php
Class sampleClass
{
          //member decklaration
          Public $var = ‘some value’;
          //method declaration
          Public function displayvar() {
               Echo $this -> var;
           }
}
?>

Instance of a class can be created by creating a new object and assigning it to a variable. Example: $instance = new sampleClass();

**PHP inheritance introduction**

Inheritance is a mechanism that extends and existing class. Inheriting a class would mean creating a new class with all functionality of the existing class in addition to some more. The created class is called as a subclass of the parent class. Parent is a keyword which we use to access members of a parent class. Inheritance is usually defined by using the keyword “Extend”. $this is used a reference to the calling object. Inheritance avoids redundancy of code.

**Example:**

Class employee extends manager : Here Employee is a subclass of manager.

Echo $this->name can be used to echo variable name of the parent class.

**What type of inheritance that PHP supports? Provide an example.**

PHP supports single level inheritance.

***HP objects & properties - Dec 13, 2008 at 22:00 PM by Rajmeet Ghai***

Every class in PHP must have new instances or object created. This object must be assigned to a variable. An object will always be assigned when creating a new object unless the object has a constructor defined that throws an exception on error

**Example:**

<?php
$instance = new sample();
?>

For accessing an objects property and In order to perform a method within an object; one needs to specify both the object name and the function name:

$result = $instance->setsample('value');

***PHP access control modifier - Dec 13, 2008 at 22:00 PM by Rajmeet Ghai***

Access control modifiers restrict the functions and classes to be accessed. Following are the list of access control modifiers:

**Public**: This property or method can be accessed from anywhere on the script.

**Private**: This property or method cannot be accessed from everywhere. It can only be used by the class or the object it is a part of.

**Protected**: This property or method can be used by the code in the class it is a part of.

**Abstract**: This property or method needs to be subclassed and cannot be used directly.

**Example:**

Class employee
{
        Public $name;
        Private $salary;
}
$emp = new employee;
$emp->name = “emp”;
Print $emp->name

**Constrcutor:**

Constructor methods for classes can be declared. Classes having a constructor method can call this method on each new object.

Syntax:

Void \_\_ construct(arguments);

Example:
Class base {
      Function \_\_constrcutor() {
               Print “In base class”;
      }
}

Parent constructors are not called implicitly if the child class defines a constructor. In order to run a parent constructor, a call to parent::\_\_construct() within the child constructor is required.

**Destructor:**

The destructor methods are called as soon as all references to a particular object are removed or when the object is explicitly destroyed.

Like constructors, parent destructors will not be called implicitly by the engine. In order to run a parent destructor, one would have to explicitly call parent::\_\_destruct() in the destructor body.

**Example:**

Function \_\_destruct()