Date:2021-01-29

* Using htmlentities for security:
  + Some time user may input data as a html format then it is very harmful for a form because that code generate an another output.
  + Then htmlentities() are useful it does not run a html tag and just saw a text only.
  + Syntax:
    - htmlentities(string);
* Session in php:
  + Session are store in server and hidden from the user.
  + For setting a session:
    - Ex:
      * <?php
      * session\_start();
      * $\_SESSION['username'] = 'jay';
      * ?>
  + For get a session:
    - Ex:
      * <?php
      * session\_start();
      * echo "Welcome ".$\_SESSION['username'];
      * ?>
* Unsetting a session:
  + unset($\_SESSION[session\_name]: for particular session unset.
  + session\_unset() – for remove an all session.
    - Ex:
      * <?php
      * session\_start();
      * //here we create a 2 session uname and age
      * $\_SESSION['uname'] = 'jay';
      * $\_SESSION['age'] = '21';
      * //lets print a session
      * echo $\_SESSION['uname']."<br>";
      * echo $\_SESSION['age'];
      * //for remove a session name uname
      * unset($\_SESSION['uname']);
      * //lets print a session
      * echo "<br>";
      * print\_r($\_SESSION);
      * //for unset all session
      * session\_unset();
      * echo "<br>";
      * print\_r($\_SESSION);
      * ?>
* Creating a cookie:
  + Cookie is stored in a client side.
  + It means it store in a browser.
  + User can disable a cookie.
  + For create a cookie…
    - setcookie(‘name’,’value’,’expiretime in milisecond’);
    - Ex:
      * <?php
      * //set cookie name favcolor
      * setcookie('color','yellow',time() + 86400);//for a 1 day
      * ?>
  + for retrieve a value of cookie.
    - $\_COOKIE[‘cookie\_name’];
    - Ex:
      * <?php
      * //get cookie value
      * if (isset($\_COOKIE['color'])) {
      * $color = $\_COOKIE['color'];
      * echo "<body bgcolor = $color>";
      * echo "<h1>you got website in your faviourite color...</h1>";
      * } else {
      * echo "goto set\_cookie.php and then come here...";
      * }
      * ?>
* Deleting cookie:
  + There are not an any function for delete a cookie.
  + Ex:
    - * <?php
      * //set cookie name favcolor
      * setcookie('color','yellow',time() + 86400);//for a 1 day
      * //print cookie..
      * echo "<br>";
      * print\_r($\_COOKIE);
      * //now we delete a cookie
      * setcookie('color','yellow',time() - 86400);
      * //print cookie
      * echo "<br>";
      * print\_r($\_COOKIE);
      * ?>
* File handling – writing a file:
  + For write a file we use..
    - fopen(‘filename’,’mode {for write w});
    - now our file is ready to write
    - fwrite($handler variable,$data);
    - and last fclose();
* file handling – read a file:
  + Ex:
    - * <?php
      * $file = file('file\_read\_ex.txt') or die('unable to open filr...');
      * foreach ($file as $data) {
      * echo "$data ,";
      * }
      * sfclose($file);
      * ?>
* File handling – append a file:
  + - Ex:
      * <?php
      * $file = fopen('file\_write\_ex.txt', 'w');
      * $text = "Hello i am jay \n i am a student.";
      * fwrite($file, $text);
      * fclose($file);
      * ?>
* Explode function with file handling:
  + explode is used to convert string in to aarray.
  + Ex:
    - * <?php
      * $file = fopen('explode.txt', 'r');
      * $data = fread($file, filesize('explode.txt'));
      * $data1 = explode(',', $data);
      * fclose($file);
      * foreach ($data1 as $name) {
      * echo "$name<br>";
      * }
      * ?>
* Implode with file handling:
  + Used for convert array to string.
  + Ex:
    - * <?php
      * $data = array('jay', 'hemal' ,'mahir');
      * $str = implode(',', $data);
      * $file = fopen('implode.txt', 'w');
      * fwrite($file, $str);
      * fclose($file);
      * ?>
* File handling listing file:
  + opendir(‘folder name’) – for open a directory
  + readdir(handler) – return a files.
  + Ex:
    - * <?php
      * $dir = 'files';
      * if($file = opendir("$dir/")) {
      * echo "looking inside's $dir :<br>";
      * while ($files = readdir($file)) {
      * echo "<a href='$dir/$files'>$files</a> <br>";
      * }
      * }
      * ?>
* Checking is file exis:
  + File\_exists(‘file name’): return true if file is exists in directory.
  + Ex:
    - * <?php
      * $filename = 'explode.txt';
      * if(file\_exists($filename)) {
      * echo "File already exists....";
      * } else {
      * echo "file not exists";
      * }
      * //for inside another directory
      * $filename1 = 'files/first.txt';
      * if(file\_exists($filename1)) {
      * echo "File already exists....";
      * } else {
      * echo "file not exists";
      * }
      * ?>
* Delete file:
  + unlink(file name) – return true if file delete.
  + Ex:
    - * <?php
      * $file\_del = 'file\_for\_delete.txt';
      * if (@unlink($file\_del)) {
      * echo "File <strong>$file\_del</strong> deleted successfully.....";
      * } else {
      * echo "File <strong>$file\_del</strong> already deleted.....";
      * }
      * ?>
* Rename file:
  + rename(old filename, new filename);
  + Ex:
    - * <?php
      * $file = 'file\_rename.txt';
      * $rand = rand(1000,9999);
      * if(@rename($file, "$rand.txt")) {
      * echo "File <strong>$file</strong> renamed to <strong>$rand.txt</strong>successfully....";
      * } else {
      * echo "Error in rnameing file <strong>$file</strong>...";
      * }
      * ?>
* Uploading a file:
  + For upload a file move\_uploaded\_file(temp location,actual location);
  + Ex:
    - * <?php
      * if (isset($\_FILES['file']['name'])) {
      * if(!empty($\_FILES['file']['name'])) {
      * $name = $\_FILES['file']['name'];
      * // echo $size = $\_FILES['file']['size'];
      * // echo $type = $\_FILES['file']['type'];
      * $tmp\_loc = $\_FILES['file']['tmp\_name'];
      * // echo $error = $\_FILES['file']['error'];
      * // echo "<pre>";
      * // print\_r($\_FILES);
      * $location = 'uploads/';
      * if (move\_uploaded\_file($tmp\_loc, $location.$name)) {
      * echo "File Uploaded Successfully.....";
      * } else {
      * echo "Problem to upload file...";
      * }
      * } else {
      * echo "Please choose a file....";
      * }
      * }
      * ?>
      * <form action="#" method="post" enctype="multipart/form-data">
      * <input type="file" name="file"><br><br>
      * <input type="submit" value="Upload file">
      * </form>
* Restricting a file size:
  + Some time we have to deny user for not uploading a large files.
  + Then we can restrict user to upload a large file.
  + Ex:
    - * <?php
      * if (isset($\_FILES['file']['name'])) {
      * if(!empty($\_FILES['file']['name'])) {
      * $name = $\_FILES['file']['name'];
      * $size = $\_FILES['file']['size'];
      * // echo $type = $\_FILES['file']['type'];
      * $tmp\_loc = $\_FILES['file']['tmp\_name'];
      * $max = 1000;
      * // echo $error = $\_FILES['file']['error'];
      * // echo "<pre>";
      * // print\_r($\_FILES);
      * if ($size <= $max) {
      * $location = 'uploads/';
      * if (move\_uploaded\_file($tmp\_loc, $location.$name)) {
      * echo "File Uploaded Successfully.....";
      * } else {
      * echo "Problem to upload file...";
      * }
      * } else {
      * echo "File size must be 1kb or smaller....";
      * }
      * } else {
      * echo "Please choose a file....";
      * }
      * }
      * ?>
      * <form action="#" method="post" enctype="multipart/form-data">
      * <input type="file" name="file"><br><br>
      * <input type="submit" value="Upload file">
      * </form>
* Restrict user for extension:
  + Some time we have to restrict user to upload a specific type of files. Then we can use restrict extension.
  + Ex:
    - * <?php
      * if (isset($\_FILES['file']['name'])) {
      * if(!empty($\_FILES['file']['name'])) {
      * $name = $\_FILES['file']['name'];
      * $size = $\_FILES['file']['size'];
      * // echo $type = $\_FILES['file']['type'];
      * $tmp\_loc = $\_FILES['file']['tmp\_name'];
      * $ext = strtolower(substr($name ,strpos($name, '.')+1));
      * // echo $error = $\_FILES['file']['error'];
      * // echo "<pre>";
      * // print\_r($\_FILES);
      * if ($ext == 'jpg' || $ext == 'jpeg' || $ext == 'image/jpeg') {
      * $location = 'uploads/';
      * if (move\_uploaded\_file($tmp\_loc, $location.$name)) {
      * echo "File Uploaded Successfully.....";
      * } else {
      * echo "Problem to upload file...";
      * }
      * } else {
      * echo "File must be jpg , jpeg , image/jpeg....";
      * }
      * } else {
      * echo "Please choose a file jpg , jpeg , image/jpeg....";
      * }
      * }
      * ?>
      * <form action="#" method="post" enctype="multipart/form-data">
      * <input type="file" name="file"><br><br>
      * <input type="submit" value="Upload file">
      * </form>
* Creating a non-unique hit counter:
  + Counter is used for a check that how many time pages are refreshed.
    - Ex:
      * <?php
      * function hit\_count(){
      * $file = 'non\_unique\_hit\_counter.txt';
      * $handler = fopen($file, 'r');
      * $count = fread($handler, filesize($file));
      * fclose($handler);
      * $current\_count = $count + 1;
      * $handler = fopen($file, 'w');
      * fwrite($handler, $current\_count);
      * echo "<h1>This page is visited by $current\_count users.....</h1>";
      * }
      * hit\_count();
      * ?>
* Creating a unique hit counter:
  + <?php
  + function hit\_counter(){
  + $current = 0;
  + $found = '';
  + $current\_ip = $\_SERVER['REMOTE\_ADDR'];
  + //$current\_ip = '100.100.100.100';
  + $data = file("ip.txt");
  + foreach ($data as $data1) {
  + $ip\_single = trim($data1);
  + if ($ip\_single == $current\_ip) {
  + $found = true;
  + } else {
  + $found = false;
  + }
  + }
  + if ($found == false) {
  + $handler = fopen('ip.txt', 'a');
  + fwrite($handler, "$current\_ip\n");
  + fclose($handler);
  + $handler = fopen('uniqur\_hit\_counter.txt', 'r');
  + $count = fread($handler, filesize('uniqur\_hit\_counter.txt')+1);
  + fclose($handler);
  + @$tot\_user = $count + 1;
  + $handler = fopen('uniqur\_hit\_counter.txt', 'w');
  + fwrite($handler, $tot\_user);
  + fclose($handler);
  + }
  + $handler = fopen('uniqur\_hit\_counter.txt', 'r');
  + $data = fread($handler, filesize('uniqur\_hit\_counter.txt'));
  + fclose($handler);
  + echo "<h1>Total unique user visited this page $data times</h1>";
  + }
  + hit\_counter();
  + ?>