Name : Mayur Purushvani

htmletities for security :

htmlentities is the better function to use as it encodes all possible characters.

Example :

```
<?php

if (isset($_GET['day']) && isset($_GET['date']) && isset($_GET['year'])) {
    $day = htmlentities($_GET['day']);
    $date = htmlentities($_GET['date']);
    $year = htmlentities($_GET['year']);

    if (!empty($day) && !empty($date) && !empty($year)) {
        echo 'It is' . $day . ' ' . $date . ' ' . $year;
    } else {
        echo 'Fill the form first!';
    }
}

?>

<form action="html_entities.php" method="GET">

    Day : <br><input type="text" name="day"><br>
    Date : <br><input type="text" name="date"><br>
    Year : <br><input type="text" name="year"><br><br>
    <input type="submit" value="Submit">
</form>
```

The <**iframe**> tag specifies an inline frame. An inline frame is **used** to embed another document within the current HTML document.

Session in PHP :

A session is started with the session_start() function.

Session variables are set with the PHP global variable: $_SESSION.

Example :

Set.php :

```
<?php

session_start();

$_SESSION['username'] = 'Mayur';

?>
```

View.php :

```
<?php

session_start();

if(isset($_SESSION['username'])) {
    echo 'Welcome, '.$_SESSION['username'];
}
else {
    echo 'Please logi first!';
}

?>
```

First run the view.php page, you can see that it will give the result as please login first. Then, run the set.php and again run view.php, it will show the welcome message.

Let’s assume that set.php is your login page and view.php is your dashboard page.

Unset the session :

It is like a logout functionality. When we’ve a multiple sessions in all over website, and we’ve to remove all the sessions at once, then we use session_destroy() metod. Or else we use unset($_SESSION[‘’]).

```
<?php
session_start();
unset($_SESSION['username']);

//session_destroy(); //Or you can use this. It will remove all the sessions in your website at once.
?>
```

Cookie Example :

A cookie is often used to identify a user. A cookie is a small file that the server embeds on the user's computer. Each time the same computer requests a page with a browser, it will send the cookie too. With PHP, you can both create and retrieve cookie values.

We have a 3 arguments I cookie, first is key, second is value ad third is timestamp.

In below example the cookie will set only for 10 seconds, and then it destroyed.

Example :

View.php :

```
<?php

echo $_COOKIE['username'];

?>
```

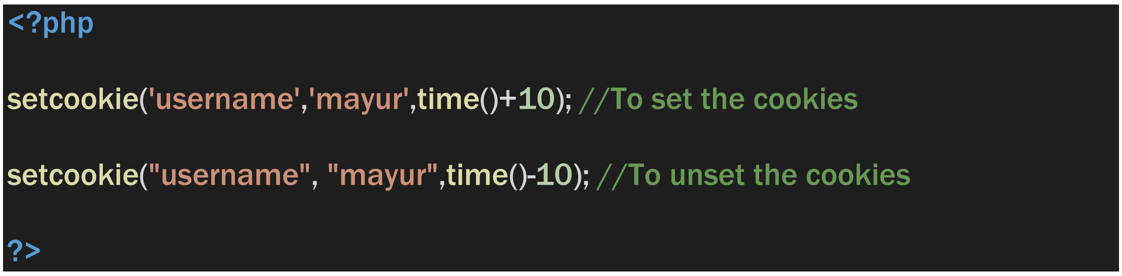
Set.php :

```
<?php

setcookie('username','mayur',time()+10);

?>
```

Deleting cookie from browser :



```php
<?php

setcookie('username','mayur',time()+10); //To set the cookies

setcookie("username", "mayur",time()-10); //To unset the cookies

?>
```

Difference between session and cookies :

The main difference between a session and a cookie is that session data is stored on the server, whereas cookies store data in the visitor’s browser.

Sessions are more secure than cookies as it is stored in server. Cookie can be turned off from browser.

Data stored in cookie can be stored for months or years, depending on the life span of the cookie. But the data in the session is lost when the web browser is closed.

File Handling :

We have 3 modes of files :

w – write something in the file.
r – Read something from the file.
a – Append something into the file.

Write into the file Example :

This will create a names.txt file automatically and write the content.And after writing the content, it will close the file.

```
<?php

$handle = fopen('names.txt', 'w');

fwrite($handle, 'Mayur'."\n");
fwrite($handle, 'max');
fclose($handle);
echo "Written!";

?>
```

Read from file and append into file Example :

```
<?php

if (isset($_POST['name'])) {
    $name = $_POST['name'];
    if (!empty($name)) {

        $handle = fopen('names.txt', 'a');
        fwrite($handle, $name . "\n");
        fclose($handle);

        echo "Curret name in file is : ";

        $count = 1;
        $readin = file('names.txt');
        $readin_count = count($readin);

        foreach ($readin as $fname) {
            echo trim($fname);
            if ($count < $readin_count) {
                echo ', ';
            }
            $count++;
        }
```

```
  }
}
?>

<form action="file.php" method="post">
  Name : <input type="text" name="name"><br><br>
  <input type="submit" name="Submit">
</form>
```

Implode in PHP :

The implode() function returns a string from the elements of an array.

The implode() function accept its parameters in either order. However, for consistency with explode(), you should use the documented order of arguments.

The separator parameter of implode() is optional. However, it is recommended to always use two parameters for backwards compatibility.

Example :

```
<?php

$names_array = array('mayur','max','mokey');
$string =  implode(', ', $names_array);

echo $string;

?>
```

File Directory Example :

opendir() and readdir() methods we can use into a file directories example.

```
<?php

$directory = 'files';
if ($handle = opendir($directory)) {
   echo 'Looking into your \'' . $directory . '\' : <br>';

   while ($file = readdir($handle)) {
      if ($file != "." && $file != "..") {
         echo '<a href="' . $directory . '/' . $file. '">' . $file . '</a><br>';
      }
   }
} else {
   echo 'Please make the directory first!';
}
?>
```

File_exists() Example :

```
<?php

$filename = "file.txt";
if (file_exists($filename)) {
   echo 'file already exists!';
} else {

   $handle = fopen($filename, 'w');
   fwrite($handle, "nothing!");
   fclose($handle);
}
?>
```

File Reame and Delete example :

```
<?php

$filename = 'filetodelete.txt';

if (@unlink($filename)) {
   echo 'File <strong>' . $filename . '</strong> has been deleted!';
} else {
   echo 'File not deleted!';
}

$filename2 = 'filetorename.txt';
$rand = rand(100, 999);
if (@rename($filename2, $rand .'.txt')) {
   echo 'File <strong>' . $filename2 . '</strong> has been renamed with <strong>' . $rand . ' .txt </strong>';
} else {
   echo 'File not renamed!';
}
?>
```

Uploading the file Example :

We use move_upload_file() method to upload the file.

```
<form method="post" action="file.php" name="file" enctype="multipart/form-data">
   <input type="file" name="file"><br><br>
   <input type="submit" value="Submit">
</form>

<?php

$name = @$_FILES["file"]["name"];
$tmp_name = @$_FILES['file']['tmp_name'];

if (isset($name)) {
   if (!empty($name)) {
      $location = 'uploads/';
      if (move_uploaded_file($tmp_name, $location . $name)) {
         echo 'Uploaded!';
      } else {
         echo 'There was an error';
      }
   } else {
      echo 'Please choose a file';
   }
} ?>
```

File Size restriction :

We use the sustr function to get the file extension.

In filename, we can get the after the dot “.” Extension will returned.

```
<form method="post" action="file.php" name="file" enctype="multipart/form-data">
    <input type="file" name="file"><br><br>
    <input type="submit" value="Submit">
</form>

<?php

$name = @$_FILES["file"]["name"];
$tmp_name = @$_FILES['file']['tmp_name'];
$extension = strtolower(substr($name, strpos($name, '.') + 1));
$type = @$_FILES['file']['type'];
$size = @$_FILES['file']['size'];
$max_size = 100000;

if (isset($name)) {
    if (!empty($name)) {
        if (($extension == 'jpg' || $extension == 'jpeg') && ($type == 'image/jpg' || $type == 'image/jpeg') && $size <= $max_size) {
            $location = 'uploads/';
            if (move_uploaded_file($tmp_name, $location . $name)) {
                echo 'Uploaded!';
            } else {
                echo 'There was an error';
            }
        } else {
            echo 'File must be jpg/jpeg ad must be 2mb or less';
        }
    } else {
        echo 'Please choose a file';
    }
}
?>
```

Creating a hit counter into a file :

Make a 3 files : count.txt, index.php and count.php.

count.php :

```php
<?php

function hit_count()
{
   $filename = 'count.txt';

   $handle = fopen($filename, 'r');
   $current = fread($handle, filesize($filename));
   fclose($handle);

   $current_inc = $current + 1;
   $handle = fopen($filename, 'w');
   fwrite($handle, $current_inc);
   fclose($handle);
}
?>
```

Index.php :

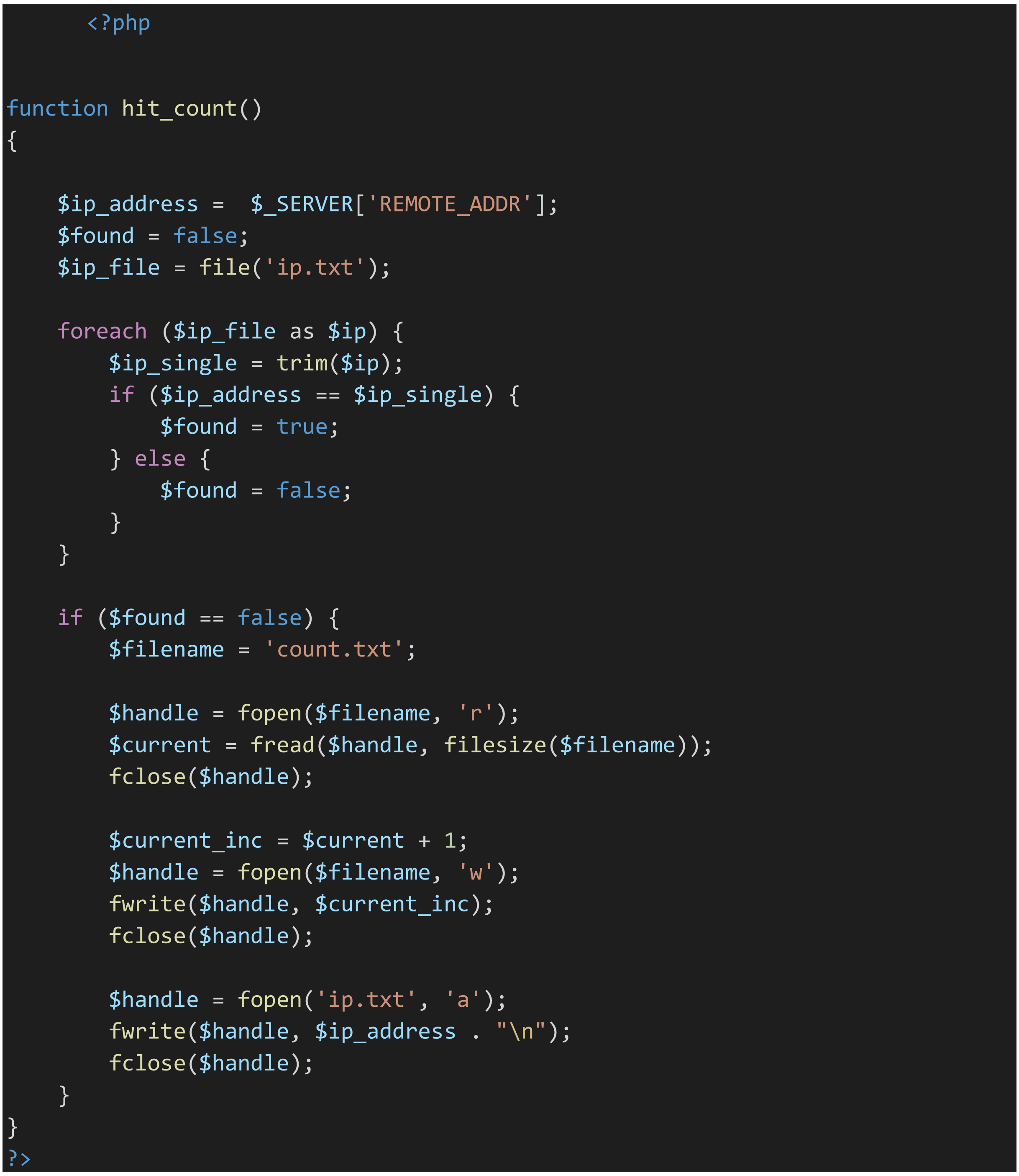
```php
<?php

include 'count.php';

hit_count();

?>
```

Creating a file based unique hit counter Example :



```
      <?php


function hit_count()
{

    $ip_address =  $_SERVER['REMOTE_ADDR'];
    $found = false;
    $ip_file = file('ip.txt');

    foreach ($ip_file as $ip) {
        $ip_single = trim($ip);
        if ($ip_address == $ip_single) {
            $found = true;
        } else {
            $found = false;
        }
    }

    if ($found == false) {
        $filename = 'count.txt';

        $handle = fopen($filename, 'r');
        $current = fread($handle, filesize($filename));
        fclose($handle);

        $current_inc = $current + 1;
        $handle = fopen($filename, 'w');
        fwrite($handle, $current_inc);
        fclose($handle);

        $handle = fopen('ip.txt', 'a');
        fwrite($handle, $ip_address . "\n");
        fclose($handle);
    }
}
?>
```