PHP - Web Concepts

This session demonstrates how PHP can provide dynamic content according to browser type, randomly generated numbers or User Input. It also demonstrated how the client browser can be redirected.

Identifying Browser & Platform

PHP creates some useful **environment variables** that can be seen in the phpinfo.php page that was used to setup the PHP environment.

One of the environment variables set by PHP is **HTTP_USER_AGENT** which identifies the user's browser and operating system.

PHP provides a function getenv() to access the value of all the environment variables. The information contained in the HTTP_USER_AGENT environment variable can be used to create dynamic content appropriate to the browser.

Following example demonstrates how you can identify a client browser and operating system.

NOTE − The function preg_match()is discussed in PHP Regular expression session.

```
<html>
   <body>
   
      <?php
         function getBrowser() {
            $u_agent = $_SERVER['HTTP_USER_AGENT'];
            $bname = 'Unknown';
            $platform = 'Unknown';
            $version = "";
            
            //First get the platform?
            if (preg_match('/linux/i', $u_agent)) {
               $platform = 'linux';
            }elseif (preg_match('/macintosh|mac os x/i', $u_agent)) {
               $platform = 'mac';
            }elseif (preg_match('/windows|win32/i', $u_agent)) {
               $platform = 'windows';
            }
            
            // Next get the name of the useragent yes seperately and for good reason
            if(preg_match('/MSIE/i',$u_agent) && !preg_match('/Opera/i',$u_agent)) {
               $bname = 'Internet Explorer';
               $ub = "MSIE";
            } elseif(preg_match('/Firefox/i',$u_agent)) {
               $bname = 'Mozilla Firefox';
               $ub = "Firefox";
            } elseif(preg_match('/Chrome/i',$u_agent)) {
```

```
      $bname = 'Google Chrome';
      $ub = "Chrome";
   }elseif(preg_match('/Safari/i',$u_agent)) {
      $bname = 'Apple Safari';
      $ub = "Safari";
   }elseif(preg_match('/Opera/i',$u_agent)) {
      $bname = 'Opera';
      $ub = "Opera";
   }elseif(preg_match('/Netscape/i',$u_agent)) {
      $bname = 'Netscape';
      $ub = "Netscape";
   }

   // finally get the correct version number
   $known = array('Version', $ub, 'other');
   $pattern = '#(?<browser>' . join('|', $known) . ')[/ ]+(?<version>[0-9.|a-zA-Z.]*)#'

   if (!preg_match_all($pattern, $u_agent, $matches)) {
      // we have no matching number just continue
   }

   // see how many we have
   $i = count($matches['browser']);

   if ($i != 1) {
      //we will have two since we are not using 'other' argument yet

      //see if version is before or after the name
      if (strripos($u_agent,"Version") < strripos($u_agent,$ub)){
         $version= $matches['version'][0];
      }else {
         $version= $matches['version'][1];
      }
   }else {
      $version= $matches['version'][0];
   }

   // check if we have a number
   if ($version == null || $version == "") {$version = "?";}
   return array(
      'userAgent' => $u_agent,
      'name'      => $bname,
      'version'   => $version,
      'platform'  => $platform,
      'pattern'   => $pattern
   );
}

// now try it
$ua = getBrowser();
$yourbrowser = "Your browser: " . $ua['name'] . " " . $ua['version'] .
```

```
            " on " .$ua['platform'] . " reports: <br >" . $ua['userAgent'];

         print_r($yourbrowser);
      ?>

   </body>
</html>
```

This is producing following result on my machine. This result may be different for your computer depending on what you are using.

It will produce the following result −

```
Your browser: Google Chrome 54.0.2840.99 on windows reports:
Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
   Chrome/54.0.2840.99 Safari/537.36
```

Display Images Randomly

The PHP **rand()** function is used to generate a random number.i This function can generate numbers with-in a given range. The random number generator should be seeded to prevent a regular pattern of numbers being generated. This is achieved using the **srand()** function that specifies the seed number as its argument.

Following example demonstrates how you can display different image each time out of four images −

Live Demo

```
<html>
   <body>

      <?php
         srand( microtime() * 1000000 );
         $num = rand( 1, 4 );

         switch( $num ) {
            case 1: $image_file = "/php/images/logo.png";
               break;

            case 2: $image_file = "/php/images/php.jpg";
               break;

            case 3: $image_file = "/php/images/logo.png";
               break;

            case 4: $image_file = "/php/images/php.jpg";
               break;
         }
         echo "Random Image : <img src=$image_file />";
      ?>
   </body>
</html>
```

It will produce the following result −

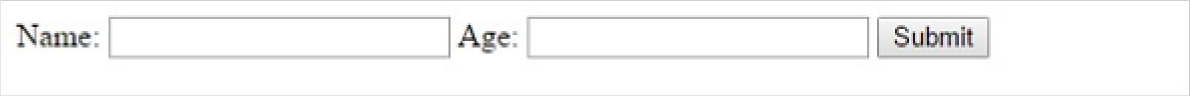
Using HTML Forms

The most important thing to notice when dealing with HTML forms and PHP is that any form element in an HTML page will automatically be available to your PHP scripts.

Try out following example by putting the source code in test.php script.

```
<?php
   if( $_POST["name"] || $_POST["age"] ) {
      if (preg_match("/[^A-Za-z'-]/",$_POST['name'] )) {
         die ("invalid name and name should be alpha");
      }
      
      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>
```

It will produce the following result −



- The PHP default variable **$_PHP_SELF** is used for the PHP script name and when you click "submit" button then same PHP script will be called and will produce following result −
- The method = "POST" is used to post user data to the server script. There are two methods of posting data to the server script which are discussed in PHP GET & POST chapter.

Browser Redirection

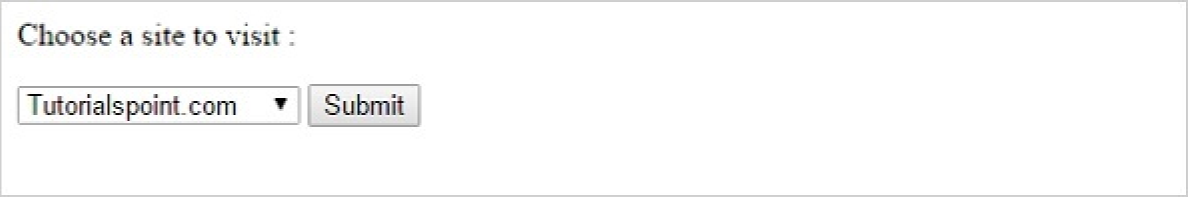
The PHP **header()** function supplies raw HTTP headers to the browser and can be used to redirect it to another location. The redirection script should be at the very top of the page to prevent any other part of the page from loading.

The target is specified by the **Location:** header as the argument to the **header()** function. After calling this function the **exit()** function can be used to halt parsing of rest of the code.

Following example demonstrates how you can redirect a browser request to another web page. Try out this example by putting the source code in test.php script.

```
<?php
   if( $_POST["location"] ) {
      $location = $_POST["location"];
      header( "Location:$location" );
      
      exit();
   }
?>
<html>
   <body>
   
      <p>Choose a site to visit :</p>
      
      <form action = "<?php $_SERVER['PHP_SELF'] ?>" method ="POST">
         <select name = "location">.
         
            <option value = "http://www.tutorialspoint.com">
               Tutorialspoint.com
            </option>
         
            <option value = "http://www.google.com">
               Google Search Page
            </option>
         
         </select>
         <input type = "submit" />
      </form>
      
   </body>
</html>
```

It will produce the following result −



Choose a site to visit :