Лекция 10 (часть 2)
Создание и работа с изображениями в PHP. Разработка Captcha

Разработка Captcha

Реализовывать captcha будем с помощью трех скриптов – один будет генерировать captcha (файл captcha.php), а второй загружать форму для ввода пользователем символов captcha (файл index.php), третий – выполнять проверку введенной пользователем символов (файл result.php).

Формирование капчи будет происходить в четыре этапа:

1. Формируем подложку (т. е. фоновый рисунок). Сделаем так, чтобы он рандомно менялся.
2. Рисуем случайным образом линии, которые затрудняют распознавание для машин букв капчи.
3. Генерируем текст капчи.
4. Рисуем случайным образом линии, только поверх текста капчи.

Разработка Captcha

Этап 1.

Задаем фоны для капчи в виде набора картинок (png). Их можно нарисовать самим или скачать подходящие из интернета, и размещаем их внутри подпапки background. Будем использовать размер - 200x60. Фонов может быть сколько угодно. В рассматриваемом примере их три и хранятся они в виде массива.

```
$img_arr = array("back1.png","back2.png","back3.png");
```

Далее генерируем (выбираем) фон для капчи случайным образом и переносим его на изображение капчи.

```
$img_fn = $img_arr[rand(0, count($img_arr)-1)];
$im = imagecreatefrompng($img_fn);
```

Разработка Captcha

Этап 2.

Создаем случайные линии под будущим текстом капчи. Количество линий будет определятся случайным образом от 3 до 7.

```
$linenum = rand(3, 7);
```

Цвет также определяется случайным образом для каждой линии. Это удобно сделать через цикл. В этом же цикле разумным является и реализация отображения непосредственно линий.

```
for ($i=0; $i<$linenum; $i++)
    {
        $color = imagecolorallocate($im,
             rand(0, 255), rand(0, 200), rand(0, 255)
         );
        imageline($im,
             rand(0, 10), rand(1, 60),
             rand(160, 200), rand(1, 60), $color);
    }
```

Разработка Captcha

Этап 3.

Инициализируем переменную, в которой будет содержаться текстовое значение капчи:

```
$captcha = '';
```

Далее организовывается цикл с количеством итераций, равным длине капчи (caplen):

```
for ($i = 0; $i < $caplen; $i++)
```

На каждом шаге цикла генерируется очередной символ капчи и рисуется на изображении.
Берем случайный символ из нашего алфавита и добавляем его в капчу:

```
$captcha .= $letters[ rand(0, strlen($letters)-1) ];
```

Разработка Captcha

Вычисляем положение сгенерированного символа на изображении по оси x:

```
$x = ($width - 20) / $caplen * $i + 10;
```

Это положение зависит от ширины изображения, длины капчи и порядкового номера символа. Далее мы добавляем немного "случайности" в это положение:

```
$x = rand($x, $x+4);
```

Вычисляем положение сгенерированного символа на изображении по оси y:

```
$y = $height - ( ($height - $fontsize) / 2 );
```

Положение зависит от размера шрифта и высоты изображения.

Разработка Captcha

Генерируем случайный цвет для символа. Этот цвет не должен быть слишком светлым, поэтому каждый из компонентов цвета (R, G и B) генерируем в диапазоне 0-100:

```
$curcolor = imagecolorallocate( $im, rand(0, 100), rand(0, 100), rand(0, 100) );
```

Для текущего символа случайным образом генерируем его угол наклона в диапазоне -25..25 градусов, чтобы буквы на капче "плясали":

```
$angle = rand(-25, 25);
```

И, наконец, рисуем символ со всеми полученными выше характеристиками на изображении:

```
imagettftext($im, $fontsize, $angle, $x, $y, $curcolor, $font, $captcha[$i]);
```

На этой строке оканчивается тело цикла. Когда цикл отработает, в переменной captcha будет содержаться текстовое значение капчи, а изображение im будет представлять из себя отрисованный текст капчи поверх фона и линий.

Разработка Captcha

Этап 4.

Повторяет код этапа 2 — снова рисуем линии, только теперь поверх капчи.

Далее необходимо где-то сохранить значение капчи, чтобы основной скрипт, который использует ее, мог проверить значение пользователя. Лучшее для этого место — сессионная переменная. Инициализируем сессию и сохраняем значение капчи:

```
session_start();
$_SESSION['captcha'] = $captcha;
```

Наконец, выводим сформированное изображение captcha:

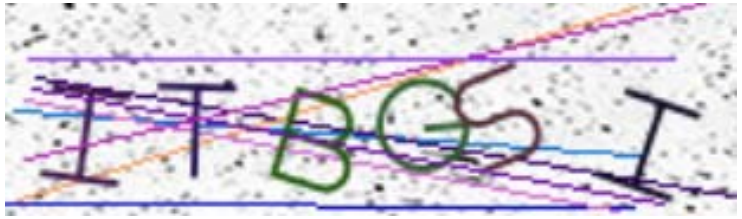
```
imagepng($im);
```

И освобождаем память, выведенную под изображение:

```
imagedestroy($im);
```

Разработка Captcha

Результат



Файл captcha.php

```php
<?php
  $letters = 'ABCDEFGKIJKLMNOPQRSTUVWXYZ';

  $caplen = 6;
  $width = 200;
  $height = 60;
  $font = __DIR__.'\gasalt.ttf';;
  $fontsize = 17;

  header('Content-type: image/png');

  $im = imagecreatetruecolor($width, $height);
  imagesavealpha($im, true);
  $bg = imagecolorallocatealpha($im, 0, 0, 0, 127);
  imagefill($im, 0, 0, $bg);
  // Задаем фоны для капчи. Можете нарисовать свой и загрузить его в папку /img.
Фонов может быть сколько угодно
  $img_arr = array("back1.png","back2.png","back3.png");
  // Генерируем (выбираем) фон для капчи случайным образом
  $img_fn = $img_arr[rand(0, sizeof($img_arr)-1)];
  $im = imagecreatefrompng($img_fn);
```

Файл captcha.php

```
// Случайные линии под текстом
    $linenum = rand(3, 7);
    for ($i=0; $i<$linenum; $i++)
    {
$color = imagecolorallocate($im, rand(0, 255), rand(0, 200), rand(0, 255));
imageline($im, rand(0, 10), rand(1, 60), rand(160, 200), rand(1, 60), $color);
    }
//формирование текста капчи
      $captcha = '';
      for ($i = 0; $i < $caplen; $i++)
      {
        $captcha .= $letters[ rand(0, strlen($letters)-1) ];
        $x = ($width - 20) / $caplen * $i + 10;
        $x = rand($x, $x+4);
        $y = $height - ( ($height - $fontsize) / 2 );
$curcolor = imagecolorallocate( $im, rand(0, 100), rand(0, 100), rand(0, 100) );
        $angle = rand(-25, 25);
imagettftext($im, $fontsize, $angle, $x, $y, $curcolor, $font, $captcha[$i]);
      }
```

Файл captcha.php

```
// Опять линии, уже сверху текста
    $linenum = rand(3, 7);
    for ($i=0; $i<$linenum; $i++)
    {
  $color = imagecolorallocate($im, rand(0, 255), rand(0, 200), rand(0, 255));
  imageline($im, rand(0, 10), rand(1, 60), rand(160, 200), rand(1, 60), $color);
    }

  session_start();
  $_SESSION['captcha'] = $captcha;

  imagepng($im);
  imagedestroy($im);
?>
```

Файл captcha_index.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html;charset=utf-8" />
  </head>
  <body>
  <form action="result.php" method="post">
      <img src='captcha.php' id='capcha-image'>
      <br>  
      <a href="javascript:void(0);" onclick="document.getElementById('capcha-
image').src='captcha.php?rid=' + Math.random();">Обновить капчу</a>
      <br>
      <br> 
      <input type="text" name="captcha" /><br />
      <br>   
      <input type="submit" value="Проверить" />
    </form>
  </body>
</html>
```

Файл result.php

```php
<?php
//Если передана капча
if (isset($_POST['captcha']))
{
    //Получаем введенную пользователем капчу
    $code = $_POST['captcha'];

    session_start();

    //Сравниваем
    if ( isset($_SESSION['captcha']) && strtoupper($_SESSION['captcha']) ==
strtoupper($code) )
        echo 'Правильно!';
    else
        echo 'Неправильно!';

    //Удаляем из сессии код капчи
    unset($_SESSION['captcha']);
    exit();
}
?>
```

Разработка Captcha