



# CAPTCHA Creation and Integration in php

Sisoft Technologies Pvt Ltd

SRC E7, Shipra Riviera Bazar, Gyan Khand-3, Indirapuram,  
Ghaziabad

Website: [www.sisoft.in](http://www.sisoft.in) Email: info@sisoft.in

Phone: +91-9999-283-283



## Generate a Random String:-

All the code from this section will go in the **captcha.php** file. Let's begin by writing the function to create the random string.

```
<?php

$permitted_chars = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ';

function generate_string($input, $strength = 5) {
    $input_length = strlen($input);
    $random_string = "";
    for($i = 0; $i < $strength; $i++) {
        $random_character = $input[mt_rand(0, $input_length - 1)];
        $random_string .= $random_character;
    }

    return $random_string;
}

$string_length = 6;
$captcha_string = generate_string($permitted_chars, $string_length);

?>
```

## Render the CAPTCHA Background

Once we have our random string, it's time to write the code to create the background of the CAPTCHA image. The image will be 200 x 50 pixels in size and will use five different colors for the background.

```
<?php  
  
$image = imagecreatetruecolor(200, 50);  
  
imageantialias($image, true);  
  
$colors = [];  
  
$red = rand(125, 175);  
$green = rand(125, 175);  
$blue = rand(125, 175);  
  
for($i = 0; $i < 5; $i++) {  
    $colors[] = imagecolorallocate($image, $red - 20*$i, $green - 20*$i,  
    $blue - 20*$i);  
}  
  
imagefill($image, 0, 0, $colors[0]);  
  
for($i = 0; $i < 10; $i++) {  
    imagesetthickness($image, rand(2, 10));  
    $rect_color = $colors[rand(1, 4)];  
    imagerectangle($image, rand(-10, 190), rand(-10, 10), rand(-10, 190),  
    rand(40, 60), $rect_color);  
}  
  
?>
```



## Render the CAPTCHA String

For the final step, we just have to draw the CAPTCHA string on our background. The color, y-coordinate, and rotation of individual letters is determined randomly to make the CAPTCHA string harder to read.

```
<?php

$black = imagecolorallocate($image, 0, 0, 0);
$white = imagecolorallocate($image, 255, 255, 255);
$textcolors = [$black, $white];

$fonts = [dirname(__FILE__).'\\fonts\\Acme.ttf', dirname(__FILE__).'\\fonts\\Ubuntu.ttf',
dirname(__FILE__).'\\fonts\\Merriweather.ttf',
dirname(__FILE__).'\\fonts\\PlayfairDisplay.ttf'];

$string_length = 6;
$captcha_string = generate_string($permitted_chars, $string_length);

for($i = 0; $i < $string_length; $i++) {
    $letter_space = 170/$string_length;
    $initial = 15;

    imagettftext($image, 20, rand(-15, 15), $initial + $i*$letter_space, rand(20, 40),
$textcolors[rand(0, 1)], $fonts[array_rand($fonts)], $captcha_string[$i]);
}

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

?>
```

## Adding the CAPTCHA to Our Contact Form

We will be using sessions to store the CAPTCHA text and then validating the text entered by website visitors. Here is the complete code of our **captcha.php** file:

```
?php
session_start();

$permitted_chars = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ';

function generate_string($input, $strength = 10) {
    $input_length = strlen($input);
    $random_string = "";
    for($i = 0; $i < $strength; $i++) {
        $random_character = $input[mt_rand(0, $input_length - 1)];
        $random_string .= $random_character;
    }

    return $random_string;
}

$image = imagecreatetruecolor(200, 50);

imageantialias($image, true);

$colors = [];

$red = rand(125, 175);
$green = rand(125, 175);
$blue = rand(125, 175);

for($i = 0; $i < 5; $i++) {
    $colors[] = imagecolorallocate($image, $red - 20*$i, $green - 20*$i, $blue - 20*$i);
}
```

## Adding the CAPTCHA to Our Contact Form (Contd...)

```
imagefill($image, 0, 0, $colors[0]);

for($i = 0; $i < 10; $i++) {
    imagesetthickness($image, rand(2, 10));
    $line_color = $colors[rand(1, 4)];
    imagerectangle($image, rand(-10, 190), rand(-10, 10), rand(-10, 190), rand(40, 60), $line_color);
}

$black = imagecolorallocate($image, 0, 0, 0);
$white = imagecolorallocate($image, 255, 255, 255);
$textcolors = [$black, $white];

$fonts = [dirname(__FILE__).'fonts\Acme.ttf', dirname(__FILE__).'fonts\Ubuntu.ttf',
dirname(__FILE__).'fonts\Merriweather.ttf', dirname(__FILE__).'fonts\PlayfairDisplay.ttf'];

$string_length = 6;
$captcha_string = generate_string($permitted_chars, $string_length);

$_SESSION['captcha_text'] = $captcha_string;

for($i = 0; $i < $string_length; $i++) {
    $letter_space = 170/$string_length;
    $initial = 15;

    imagettftext($image, 24, rand(-15, 15), $initial + $i*$letter_space, rand(25, 45), $textcolors[rand(0, 1)],
$fonts[array_rand($fonts)], $captcha_string[$i]);
}

header('Content-type: image/png');
imagepng($image);
imagedestroy($image);
?>
```

## Adding the CAPTCHA to Our Contact Form (Contd...)

Now, you simply have to add the following HTML code above the **Send Message** button:

```
<div class="elem-group">
  <label for="captcha">Please Enter the Captcha Text</label>
  <i class="fas fa-redo refresh-captcha"></i>
  <br>
  <input type="text" id="captcha" name="captcha_challenge" pattern="[A-Z]{6}">
</div>
```

## Adding the Javascript to Contact Form:-

```
var refreshButton = document.querySelector(".refresh-captcha");
refreshButton.onclick = function() {
  document.querySelector(".captcha-image").src = 'captcha.php?' + Date.now();
}
```

The final step in our integration of the CAPTCHA we created with the contact form involves checking the CAPTCHA value input by users when filling out the form and matching it with the value stored in the session. Update the **contact.php** file:

```
<?php
session_start();

if($_POST) {
    $visitor_name = "";
    $visitor_email = "";
    $email_title = "";
    $concerned_department = "";
    $visitor_message = "";

    if(isset($_POST['captcha_challenge']) && $_POST['captcha_challenge'] == $_SESSION['captcha_text']) {

        if(isset($_POST['visitor_name'])) {
            $visitor_name = filter_var($_POST['visitor_name'], FILTER_SANITIZE_STRING);
        }

        if(isset($_POST['visitor_email'])) {
            $visitor_email = str_replace(array("\r", "\n", "%0a", "%0d"), "", $_POST['visitor_email']);
            $visitor_email = filter_var($visitor_email, FILTER_VALIDATE_EMAIL);
        }

        if(isset($_POST['email_title'])) {
            $email_title = filter_var($_POST['email_title'], FILTER_SANITIZE_STRING);
        }

        if(isset($_POST['concerned_department'])) {
            $concerned_department = filter_var($_POST['concerned_department'], FILTER_SANITIZE_STRING);
        }
    }
}
```

## The final step in our integration of the CAPTCHA we created with the contact form (Contd...)



```
if(isset($_POST['visitor_message'])) {
    $visitor_message = htmlspecialchars($_POST['visitor_message']);
}

if($concerned_department == "billing") {
    $recipient = "billing@domain.com";
}
else if($concerned_department == "marketing") {
    $recipient = "marketing@domain.com";
}
else if($concerned_department == "technical support") {
    $recipient = "tech.support@domain.com";
}
else {
    $recipient = "contact@domain.com";
}

$headers = 'MIME-Version: 1.0' . "\r\n"
.'Content-type: text/html; charset=utf-8' . "\r\n"
.'From: ' . $visitor_email . "\r\n";

if(mail($recipient, $email_title, $visitor_message, $headers)) {
    echo '<p>Thank you for contacting us. You will get a reply within 24 hours.</p>';
} else {
    echo '<p>We are sorry but the email did not go through.</p>';
}
else {
    echo '<p>You entered an incorrect Captcha.</p>';
}

} else {
    echo '<p>Something went wrong</p>';
}

?>
```

## Pagination in PHP

Its always possible that your SQL SELECT statement query may result into thousand of records. But its is not good idea to display all the results on one page. So we can divide this result into many pages as per requirement.

Paging means showing your query result in multiple pages instead of just put them all in one long page.

MySQL helps to generate paging by using **LIMIT** clause which will take two arguments. First argument as OFFSET and second argument how many records should be returned from the database.

Below is a simple example to fetch records using **LIMIT** clause to generate paging.

### Example

Try out following example to display 10 records per page.



```
<html>
<head>
<title>Paging Using PHP</title>
</head>
<body>
<?php $dbhost = 'localhost:3036'; $dbuser = 'root'; $dbpass = 'rootpassword'; $rec_limit = 10;
$conn = mysql_connect($dbhost, $dbuser, $dbpass);
if(! $conn ) { die('Could not connect: ' . mysql_error()); }
mysql_select_db('test_db');
/* Get total number of records */
$sql = "SELECT count(emp_id) FROM employee ";
$retval = mysql_query( $sql, $conn );
if(! $retval )
{
die('Could not get data: ' . mysql_error());
}
$row = mysql_fetch_array($retval, MYSQL_NUM );
$rec_count = $row[0];
if( isset($_GET{'page'}) ) {
$page = $_GET{'page'} + 1;
$offset = $rec_limit * $page ;
}else {
$page = 0;
$offset = 0;
}
}
```



```
$left_rec = $rec_count - ($page * $rec_limit);  
$sql = "SELECT emp_id, emp_name, emp_salary "  
"FROM employee "  
"LIMIT $offset, $rec_limit";  
$retval = mysql_query( $sql, $conn );  
if(! $retval )  
{  
die('Could not get data: ' . mysql_error());  
}  
while($row = mysql_fetch_array($retval, MYSQL_ASSOC)) {  
echo "EMP ID :{$row['emp_id']} <br> "  
"EMP NAME : {$row['emp_name']} <br> "  
EMP SALARY : {$row['emp_salary']} <br> "  
"-----<br>";  
}  
if( $page > 0 ) {  
$last = $page - 2;  
echo "<a href = \"$_PHP_SELF?page = $last\">Last 10 Records</a> |";  
echo "<a href = \"$_PHP_SELF?page = $page\">Next 10 Records</a>";  
}else if( $page == 0 ) {  
echo "<a href = \"$_PHP_SELF?page = $page\">Next 10 Records</a>";  
}else if( $left_rec < $rec_limit ) {  
$last = $page - 2; echo "<a href = \"$_PHP_SELF?page = $last\">Last 10 Records</a>";  
}  
mysql_close($conn); ?>  
</body> </html>
```



# Thank You

Sisoft Technologies Pvt Ltd  
SRC E7, Shipra Riviera Bazar, Gyan Khand-3, Indirapuram, Ghaziabad  
Website: [www.sisoft.in](http://www.sisoft.in) Email: [info@sisoft.in](mailto:info@sisoft.in)  
Phone: +91-9999-283-283