

# Captcha Tutorial

## == Definition ==

CAPTCHA bedeutet wörtlich "Vollautomatischer öffentlicher Turing-Test zur Unterscheidung von Computern und Menschen".

Warum gibt es solche Bots überhaupt. Webseiten bei Google werden besser gefunden um so mehr Webseiten auf diese Webseite verlinken. Viele Webseiten verlinken auf die Wikipedia, daher wird diese Seite sehr gut gefunden.

S.g. Bots versuchen auf möglichst vielen Webseiten Links zu der eigenen Seite zu platzieren, damit diese Seite besser bei Google gefunden wird.

Das stört natürlich bei steigender Anzahl an Bots sehr.

## == Aufbau ==

Das typische Captcha benötigt 3 Websiteaufrufe.

Zuallerst wird die geschützte Seite aufgerufen (1)

Diese geschützte Seite bindet ein Bild ein, das in Wirklichkeit in PHP Script ist, dass das Bild dynamisch generiert (2)

Sobald der Benutzer auf "Absenden" klickt gelangt man zur Zielseite, wo man das Captcha auf Gültigkeit prüfen muss (3)

Konkret wird das Captcha beim Websiteaufruf Nummer 2 in der Session gespeichert und wird beim Aufruf Nummer 3 wieder abgefragt.

## == Seite 1 ==

Fangen wir auch beim Tutorial mit Seite 1 an. Das Formular muss kein PHP beherrschen.

## Quellcode

```
1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
2. <html xmlns="http://www.w3.org/1999/xhtml" dir="ltr" xml:lang="de"><head>
3. <title>Captcha mit PHP</title>
4. <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
5. </head>
6. <body>
7. <form method="post">
8. <fieldset>
9. <legend>Sicheres Formular</legend>
10. <table>
11. <tr>
12. <td>Benutzername:</td>
13. <td><input type="text" name="username" /></td>
14. </tr>
15. <tr>
16. <td>Sicherheitsfrage:</td>
17. <td><br/>
18. <input type="text" name="captcha" /></td>
19. </tr>
20. <tr>
21. <td>&nbsp;</td>
22. <td><input type="submit" /></td>
23. </tr>
24. </table>
25. </fieldset>
26. </form>
27. </body>
28. </html>
```

Alles anzeigen

## == Seite 2 ==

Die captchaimage.php ist ein PHP Script, die das Bild zurückliefert, nachdem das Bild in die Session gespeichert wurde. Wir benutzen dazu die Klasse "Captcha", die unter der LGPL lizenziert ist, also ohne Gebühren oder Namensnennung in kommerziellen und nicht kommerziellen Projekten eingebunden werden darf.

## Quellcode

```
1. <?php
2. session_start();
3. require_once(dirname(__FILE__).'/captcha/Captcha.php');
5. header("Content-type: image/png");
6. $captchaString = Captcha::create();
7. $image = new CaptchalImage($captchaString);
```

== Schritt 3 ==

Auf der Zielseite müssen wir uns das Captcha prüfen. Das geht wie folgt. In der Zeile 18 müsste man noch die geschützte Funktion einbauen. Beispielweise könnte man an dieser Stelle den Benutzer anlegen.

## Quellcode

```
1. function validateCaptcha() {
2. try {
3. require_once(dirname(__FILE__).'/captcha/Captcha.php');
4. // was there a response?
5. if (!isset($_POST["captcha"])) {
6. throw new Exception("no captcha supplied");
7. }
8. Captcha::validate($_POST["captcha"]);
10. } catch(Exception $e) {
11. return false;
12. }
13. return true;
14. }
16. if($_POST) {
17. if(validateCaptcha()) {
18. echo 'Benutzer "'.htmlentities($_POST['username'])."' erfolgreich angelegt.';
20. exit;
21. } else {
22. echo 'Captcha ungültig.';
23. }
24. }
```

Alles anzeigen

== Weitere Features ==

Die genutzte Captcha Bibliothek nutzt s.g. mnemonische Passwörter und vereinfacht damit unbewusst das Verständnis beim menschlichen Nutzer.

Die Mnemonik ist die Gedächtniskunst. Sie soll mit Regeln, Kniffen und Tricks das Erinnerungsvermögen unterstützen und leistungsfähiger zu machen.

Im realen Leben ist die Eselsbrücke eine bekannte mnemonische Technik.

Natürlich können spezielle Algorithmen Passwörter auf diese Art um einen Bruchteil schneller knacken. Allerdings ist der Aufwand der Anpassung immens groß.

== Demo + Download ==

Zusammengesetzt findet ihr das ganze inkl der benötigten Schriftarten unter [demo.easy-coding.de/php/captcha-tutorial](http://demo.easy-coding.de/php/captcha-tutorial)

Hier gehts zum [Download](#)