



CAPTCHA: et script til at kende forskel på computere og mennesker!

En CAPTCHA er en metode til at kende forskel på mennesker og computere via teksgenkendelse på et billede. Dette bruges ofte for at undgå spam af ens side via forms og andre input. Kan selvfølgelig bruges i alle tænkelige og utænkelige scenarier hvor man v

Skrevet den **03. Feb 2009** af **hkb-x** i kategorien **Programmering / PHP** | ★★★★★

CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) er en metode til at kende forskel på mennesker og computere via teksgenkendelse på et billede. Det bruges oftest i sign-up form på hjemmesider for at undgå spam (ikke spam som i spam mails men f.eks. kan man undgå at en computer begynder at fylde hele ens gæstebog med reklamer osv).

Hele princippet med dette CAPTCHA system er at der dannes en tekststreng på 4 tegn angivet i \$tegn og den gemmes så i sessionen \$_SESSION['tekst'] til senere validering.

Helt basalt dannes der et billede på 200x60 pixels der så lægges følgende lag oven på:

1. Først laves der en baggrund af tilfældige lyse pixels.
2. På den kaster vi så vores tekst som vi tilfældigt udvælger fra \$tegn og gemmer i \$_SESSION['tekst'].
3. Der efter kaster vi en blanding af linier og cirkler overn på teksten.
4. Til sidst kaster vi så et begrænset antal pixels overn på både tekst og linier.

Nå teksten genereres bliver den både placeret tilfældigt i forhold til hinanden (dog baseret på den forriges position) samt de får en tilfældig vinkel, skriftstørrelse og skrifttype.

De forskellige skrifttyper angives i arrayet \$skrifttyper[] og kan ændres efter for godt befindende.

Meningen er så at brugeren aflæser billedet og indtaster de 4 tegn i et input-felt der sennes sammen med det andet data. Serveren kan der efter validere brugerens input ved at sammenligne det med SESSION['tekst'].

Jeg vil gennemgå de enkelte dele bid for bid men først tager vi hele scriptet her:

(Det kan også findes i en pænere .phps version her: <a href="<http://www.hkb.it/scripts/PHP/CAPTCHA.phps>">http://www.hkb.it/scripts/PHP/CAPTCHA.phps)

```
<?
session_start();

header("Content-type: image/png");
header("Cache-Control: no-cache, must-revalidate");
header("Pragma: no-cache");
```

```

$skrifttyper[] = "Arial.ttf";
$skrifttyper[] = "Tahoma.ttf";
$skrifttyper[] = "Impact.ttf";
$skrifttyper[] = "Verdana.ttf";
$skrifttyper[] = "Georgia.ttf";

// Basis billedet

$billede = imagecreatetruecolor(200, 60);

// En god bargrundsstøj men kun med lyse farver
for($i=0;$i<(200*60);$i++) {

    $color = imagecolorallocate($billede, mt_rand(155, 255), mt_rand(155, 255), mt_rand(155, 255));

    imagesetpixel($billede, ($x - 1), $y, $color);

    $x += 1;

    if($x > 200) {

        $y += 1;
        $x = 1;

    }
}

// Så selve teksten

$tegn = "123456789ABCDEFGHIJKLMNPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz";

for($i=0;$i<4;$i++) {

    $input .= substr($tegn, mt_rand(0, (strlen($tegn) - 1)), 1);

}

$_SESSION['tekst'] = $input;

$x = mt_rand(10, 60);
$y = mt_rand(35, 50);

for($i=0;$i<strlen($input);$i++) {

    $color = imagecolorallocate($billede, mt_rand(0, 155), mt_rand(0, 155), mt_rand(0, 155));

```

```

$font = "C:/Windows/Fonts/".$skrifttyper[mt_rand(0, (count($skrifttyper) - 1))];

imagettftext($billede, mt_rand(25, 30), mt_rand(-10, 10), $x, $y, $color, $font, substr($input, $i, 1));

$x += mt_rand(25, 40);
$y += mt_rand(-5, 5);
}

// Derefter nogle streger og cirkler
for($i=0;$i<mt_rand(4, 6);$i++) {

    $color = imagecolorallocate($billede, mt_rand(0, 200), mt_rand(0, 200), mt_rand(0, 200));

    if(mt_rand(0, 2) == 1) {

        imageellipse($billede, mt_rand(50, 150), mt_rand(10, 50), mt_rand(10, 60), mt_rand(10, 60),
        $color);

    } else {

        imageline($billede, mt_rand(0, 50), mt_rand(0, 60), mt_rand(100, 200), mt_rand(0, 60), $color);

    }

}

// Og igen nogle tilfældige pixels overnpå for at gøre forvirringen total
for($i=0;$i<mt_rand(250, 500);$i++) {

    $color = imagecolorallocate($billede, mt_rand(0, 255), mt_rand(0, 255), mt_rand(0, 255));

    imagesetpixel($billede, mt_rand(0, 200), mt_rand(0, 60), $color);

}

imagepng($billede);

?>

```

Det var lidt af en mundfuld men lad os tage det lidt af gangen.

Det basale

Først kører vi `session_start()` så vi er klar når vi skal gemme vores tekststreng i en session. Der efter angiver vi et par headers der fortæller at det er et png billede og at det ikke skal gemmes i browserens cache (er så op til den enkelte browser om den vil godtage det). Vi definerer også lide de skrifttyper vi vil bruge i arrayet `$skrifttyper`.

Her efter danner vi et 'true color' billede på 200x60 px til at lave vores CAPTCHA på.

```
session_start();

header("Content-type: image/png");
header("Cache-Control: no-cache, must-revalidate");
header("Pragma: no-cache");

$skrifttyper[] = "Arial.ttf";
$skrifttyper[] = "Tahoma.ttf";
$skrifttyper[] = "Impact.ttf";
$skrifttyper[] = "Verdana.ttf";
$skrifttyper[] = "Georgia.ttf";

// Basis billedet

$billede = imagecreatetruecolor(200, 60);
```

Baggrundsstøj

En af de bedste måder at skjule teksten på er at lave masser af baggrundsstøj da det er meget nemmere for en computer at læse tekst på en ensfarvet baggrund. Dette gøres ved at vi kører samtlige pixels på billedet igenem og giver dem en tilfældig farve med `imagegetpixel()`. Jeg har dog valgt kun at bruge farver fra den lysere del af skalaen så de ikke blander sig for meget med teksten.

```
for($i=0;$i<(200*60);$i++) {

    $color = imagecolorallocate($billede, mt_rand(155, 255), mt_rand(155, 255), mt_rand(155, 255));

    imagegetpixel($billede, ($x - 1), $y, $color);

    $x += 1;

    if($x > 200) {

        $y += 1;
        $x = 1;

    }

}
```

Selve teksten

Selve teksten genereres ved at vælge 4 tilfældige tegn ud fra den teksstreng der angiver hvilke tegn vi vil bruge. Bag efter vælger vi så et startpunkt for teksten med kordinaterne \$x, \$y.

```
$tegn = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz";  
  
for($i=0;$i<4;$i++) {  
    $input .= substr($tegn, mt_rand(0, (strlen($tegn) - 1)), 1);  
}  
  
$x = mt_rand(10, 60);  
$y = mt_rand(35, 50);
```

Her efter kaster vi den ind på billedet med en tilfældig frave (dog kun mørke farver der ikke blander sig med baggrundsstøjen), en tilfældig skrifttype, en tilfældig vinkel, en tilfældig skriftstørrelse og på position \$x, \$y.

Bag efter angiver vi en ny position for \$x, \$y (der vil være placeringen af det næste tegn). \$x går fra 25 - 50 da bogstaverne jo ikke bare kan spredes tilfældigt men skal stå i rækkefølge hvor imod \$y går fra -5 - 5 da bogstaverne både kan placeres lidt nede og oppe i forhold til hinanden.

Til sidst gemes det enkelte bogstav i \$_SESSION['tekst'] til senere validering.

```
for($i=0;$i<strlen($input);$i++) {  
    $color = imagecolorallocate($billede, mt_rand(0, 155), mt_rand(0, 155), mt_rand(0, 155));  
    $font = "C:/Windows/Fonts/".$skrifttyper[mt_rand(0, (count($skrifttyper) - 1))];  
  
    imagettftext($billede, mt_rand(25, 30), mt_rand(-10, 10), $x, $y, $color, $font, substr($input, $i, 1));  
  
    $x += mt_rand(25, 40);  
    $y += mt_rand(-5, 5);  
  
    $_SESSION['tekst'] .= substr($input, $i, 1);  
}
```

Streger og Cirkler

Her efter genereres der nogle streger og cirkler oven på vores tekst. Der vælges tilfældigt om der skal laves en streg eller en cirkel, der er dog dobbelt så stor sandsynlighed for at tegne en streg.

```
for($i=0;$i<mt_rand(4, 6);$i++) {  
    $color = imagecolorallocate($billede, mt_rand(0, 200), mt_rand(0, 200), mt_rand(0, 200));
```

```

if(mt_rand(0, 2) == 1) {
    imageellipse($billede, mt_rand(50, 150), mt_rand(10, 50), mt_rand(10, 60), mt_rand(10, 60),
$color);
} else {
    imageline($billede, mt_rand(0, 50), mt_rand(0, 60), mt_rand(100, 200), mt_rand(0, 60), $color);
}
}
}

```

Tilfældige Pixels

Til sidst kaster vi en masse blandede pixels ud over det hele for at skjule bogstaverne yderligere. Dette gøres på samme måde som i starten bare 100% tilfældigt og med brug af hele fravespekteret.

```

for($i=0;$i<mt_rand(250, 500);$i++) {
    $color = imagecolorallocate($billede, mt_rand(0, 255), mt_rand(0, 255), mt_rand(0, 255));
    imagesetpixel($billede, mt_rand(0, 200), mt_rand(0, 60), $color);
}

```

Som det sidste kaster vi så det genererede billede afsted til brugeren.

```

imagepng($billede);

```

I praksis brug

Et eksempel på brug af billedet i praksis kan ses her: <http://www.hkb.it/Blog/2>

Billedet der sendes burde nu kun kunne læses af mennesker. Jeg har testet det op imod "What the font" (<http://www.myfonts.com/WhatTheFont/>) og den har aldrig været i stand til bare at komme tæt på bare et af bogstaverne.

En god ide når man bruger det i praksis er at give brugeren mulighed for at kunne reloadere billedet hvis det er for utydeligt. Det kan gøres med denne lille smule JavaScript.

```

<br />
Er billedet for utydeligt? så < a href="#"
onclick="document.getElementById('CAPTCHA').src='http://www.hkb.it/blog/CAPTCHA.hkb?' +
Math.random();">klik her</ a>.

```

Det er også grunden til at jeg genererer \$_SESSION['tekst'] i selve billedet.

Yderligere kommentarer

Da jeg bruger så mange tilfældige værdier har jeg bevist valgt at bruge mt_rand() istedet for bare at bruge rand(). Det giver faktisk en målbar forskel på genereringen af billedet.

Hvis din hjemmeside hostes på en server du ikke selv kan administrere har du højest sandsynligt ikke adgang til C:/Windows/Fonts/ så du må eventuelt prøve en anden sti. Jeg har endnu ikke testet om imagettftext() vil fungere hvis skrifttypen ikke er installeret på serveren men bare ligger i en tilfældig mappe.

Svar

obhat: Umiddelbart vil jeg ikke tro det da selve billefunktionerne bruger GDlib som er "indbygget" i PHP men om ASP har en lignende tilføjelse ved jeg ikke.

windcape: Jahr i know... er derfor jeg også har lavet den i .phps format: <a href="<http://www.hkb.it/scripts/PHP//CAPTCHA.phps>">http://www.hkb.it/scripts/PHP//CAPTCHA.phps

Kommentar af olebole d. 04. Jun 2007 | 1

Fin artikel - omend den ikke omhandler PHP's mest anvendte miljø: Linux (og andre *nix-kloner). At der mangler indrykninger, er ganske forståeligt med det elendige og outdatede interface, Eksperten stiller til rådighed for artikelskrivning. Det skal forfatteren ikke skældes ud for =)

Kommentar af langthjem d. 07. Jun 2007 | 2

God artikel. Har tit undret mig over hvordan det skulle gribes an. Det forklarer du jo godt. Jeg tror faktisk at den vil være let at overføre til ASP. Hvis man bruger ASPImage eller ASPJpeg til genereringen. Både ASPJpeg og ASPImage kan læse fonte fra en hvilken som helst mappe.

Kommentar af jensgram d. 19. Jun 2007 | 3

Nydeligt. Læste den ikke så intenst, men det ser fornuftigt og gennemarbejdet ud.

Kommentar af roenving d. 02. Aug 2008 | 4

Det'r zq i orden (og GDlib kan da også bruges i asp !-)

Kommentar af human d. 27. Jun 2007 | 5

Kommentar af w13 d. 18. Sep 2008 | 6

Meget nyttigt!

Kommentar af dandyman d. 10. Oct 2007 | 7

Har selv lavet captcha's før men denne ser bare endnu bedre ud

Kommentar af michaelthomassen d. 24. Jun 2007 | 8

Fonte kan "trækkes ud" af din egen windows installation og lægges op på serveren, hvis du ikke har adgang til fonte direkte derfra. Jeg havde ikke adgang til fonts på min webside (one.com), men ved at uploade dem og rette stien til, virkede det fint.

Kommentar af supreme-creme d. 25. Jan 2009 | 9

Hvad retter i adressen til fontsne til?

Jeg har smidt dem i en mappe kaldet fonts, men får stadig fejlen med at den ikke kan loade fontsne. Hvad gør i?

Kommentar af obhat d. 30. May 2007 | 10

Kunne det laves i ASP?

Kommentar af challenge d. 02. Dec 2007 | 11

Kommentar af windcape d. 30. May 2007 | 12

rimelig godt, dog er php koden lidt rodet, mangler indents og burde scopes af praktiske årsager. ||| obhat: Ja, hvis du installere et module der kan generere billeder. Konceptet bagved (at generere random bogstaver, og gemme i en session) er det samme for alle sprog.

Kommentar af grafik-anders d. 14. Jul 2011 | 13

Sorry, - den er væk nu ;)