



## Log ind med PHP

**Med denne guide lærer du hvordan du kan logge ind på din hjemmeside med PHP. Guiden viser dig hvordan koderne skal opstilles, og hvad du kan gøre ellers.**

Skrevet den **09. May 2011** af **dab93** | kategorien **Programmering / Andre** | ★★☆☆☆☆

Det indledene er at du skal benytte dig af HTML.

I din index.php fil lægger du ud med at skrive

```
session_start();
```

Derefter skal du igang med "post" kodningen.

```
if(isset($_POST['logon'])) {  
    //her skal den tjekke om brugernavn og password hænger sammen med det den skal hente  
}
```

```
if(isset($_POST['logon'])) {  
    $username = $_POST['username'];  
    $password = $_POST['password'];  
    $check = mysql_query("SELECT * FROM databasenavn WHERE username = '". $username ." '&&  
password = '". $password ." '");  
  
    if(mysql_num_rows($check)==1) {  
        $true = true;  
    }  
}
```

Nu skal du få din \$true til at hænge sammen med din \$\_SESSION

```
if($true == true) {  
    $row = mysql_fetch_assoc($check);  
  
    $_SESSION['id'] = $row['id'];  
    $_SESSION['username'] = $row['username'];  
    $_SESSION['password'] = $row['password'];  
    $_SESSION['admin'] = 1;  
} else {  
    header("Location: ?s=logon");  
}
```

I midten af din index skal du lave det man kalder for en "switch", som vil sige at du kan skifte imellem forskellige sider frem for hvad din url adresse omhandler.

```
switch ($_GET['s']) {
    case "logon": include("logon.php");
    break;
}
```

Hvis du skal have flere sider i din switch skal du gøre som følgende

```
switch ($_GET['s']) {
    case "frontpage": include("frontpage.php");
    break;

    case "logon": include("logon.php");
    break;
}
```

Opret nu en fil kaldet "logon.php", og indsæt følgende kode

```
<form method="post" action="?s=logon">
<input type="text" name="username" /><br />
<input type="password" name="password" /><br />
<input type="submit" name="logon" value="Log på" />
</form>
```

Hvis du har flere spørgsmål eller rettelser, så skriv det i kommentar-feltet.

Leder du efter et sikkert script? Så kig her:

<http://phpsec.org/articles/2005/password-hashing.html>

### **Kommentar af rebox d. 02. May 2011 | 1**

Udover sikkerhedsproblemerne med SQL injection synes jeg det er værd at bemærke kodestilen - den gør koden lettere ulæselig (her tænkes der specifikt på switchen). Navngivning af variabler er også et godt emne:

```
$true = true;
```

Det er ikke gennemskueligt hvad variabelen bruges til.

Jeg ser ikke nogen umiddelbar mening med at bruge en switch, når der ikke defaultes til en værdi - og implicit er det åbenbart også at ens struktur skal gøres op i en form for 'include'-visning ud fra en \$\_GET variabel.

Men absolut brugbart hvis man anvender strukturen (og forbedre sikkerheden) - man kunne med fordel lægge et eksempel på en databasestruktur ved.

### **Kommentar af dab93 d. 02. May 2011 | 2**

```
CREATE TABLE databasenavn
(
    id int AUTO_INCREMENT,
```

```
username varchar(255),
password varchar(255),
)
```

### **Kommentar af aryan d. 02. May 2011 | 3**

har du ikke en med java+html? eller step by step guide til hvordan forbinde man de to? på forhånd tak...

### **Kommentar af danco d. 02. May 2011 | 4**

Jeg synes godt nok heller ikke denne guide er meget bevendt. Jo ganske vidst er den brugbar som hovedstruktur men som rebox nævner er der nogle sikkerhedshuller som der ikke umiddelbart bliver gjort opmærksom på.

Hvis jeg var total PHP og programmerings rookie og læste din artikel ville jeg jo implementere og tro at altid var i skønneste orden.

Den SQL create statement du har postet senere giver heller ikke meget mening. For det første hedder det ikke "databasenavn" men tabel navn og dernæst er der vil næppe nogen grund til at anvende varchar(255) på begge felter. Password vil blive krypteret med MD5 eller SHA1 og kan tabellen skal derfor kun håndtere henholdsvis char(32) og char(42)

### **Kommentar af rix17172 d. 03. May 2011 | 5**

må også sige at sikkerheds hullerne er for store jeg kan nemt og smertefrit gå ind i databasen og da password bare ligger uden krypteret er det bare at hugge dem

forbedringer nemmer oversigt

sikere mod sql inflektion

krytering med password

forklare beder hvad de forskellige stum kode bruges til

### **Kommentar af rebox d. 03. May 2011 | 6**

#3

Du kan stille spørgsmål til dine implementeringsforsøg fra guiden i den kategori som dit spørgsmål omhandler.

#5

Jeg vil give dig ret i at der er ikke meget forklaring på hvad koden gør - det er mere en "'step-by-step' for at få et script der kan virke" ting end en faktisk guide.

Overordnet set, synes jeg desværre at guides oftere og oftere bliver skrevet af personer med for lidt erfaring med de pågældende teknologier. Det er naturligvis ikke en opfordring til at disse ikke skal skrive guides, men bare sætte sig mere ind i sit materiale.

### **Kommentar af dab93 d. 03. May 2011 | 7**

#6

Hentyder du til at jeg ikke har erfaring? Tror du at koden bare er blevet taget fra nettet? Jeg har faktisk lært at håndtere dette script gennem en kammerat som er uddannet Web Integrator, så jeg vil ikke ligefrem sige at det er amatør kode. Derimod er mit gamle script lidt mere amatør stil, men ikke det her.

Jeg går meget op i php, og jeg har selvfølgelig ikke lært det fulde, men jeg skal snart på uddannelse som Web Integrator.

Kode for md5(kryptering):

```
password = "". md5($password) ."
```

Men hvis der bare kommer kritik uden en smugle form for input, altså en slags forklaring på hvordan man laver bedre sikkerhed, så syntes jeg personligt at det er tarveligt af jer. Jeg laver en guide for at få folk med mindre forstand til at kunne benytte sig af det, men de skal selvfølgelig have en lille forståelse på hvordan det hænger sammen. Men ligemeget hvad jeg laver kommer der åbenbart ikke andet end kritik, som om jeg er amatør.

#### **Kommentar af rix17172 d. 04. May 2011 | 8**

Når man laver eb sådan en guid som denne her ville folk som endnu ikke har forstand på php bare tage dwn og tro at den er iorden vi kommer med input på hvordan den kan blive beder tag det ikke som kritik men tag dett til dig og lære det :-)

#### **Kommentar af repox d. 04. May 2011 | 9**

Arh, så skidt, da...

#7

Nej, jeg hentyder til at du mangler erfaring - ikke at du ikke har nogen. Det er i øvrigt heller ingen garanti for et perfekt resultat blot fordi du får hjælp fra en webintegrator. Det må også være tydeligt for dig selv at koden har mangler, når jeg ikke er den eneste der nævner problematikker.

Du har i øvrigt også fået masser af input til dine sikkerhedsproblemer - hertil kan SQL injection og kryptering nævnes.

#### **Kommentar af adionditsak d. 08. May 2011 | 10**

Jeg går på uddannelse som webintegrator og er færdig til oktober, skal herefter på datamatiker.

Jeg kan også se fejlene i din kode, men for det første er det fedt at se sådan en guide blive lavet, om den så er god eller dårlig er en anden sag :-P

Ville være spændende og se din version af logind guiden Repox, kan du ikke sætte en op?

Der er nogle mærkelige ting i koden der er lidt unødvendige, og rimelig alternativ opstilling af logind script i det hele taget.

Hvis du godt kan lide PHP er webintegrator uddannelsen helt sikkert noget for dig. Grundforløbet er stortset kun HTML og CSS, men i hovedforløbet kommer der masser af PHP :-D

#### **Kommentar af dab93 d. 08. May 2011 | 11**

#12

Jeg skal også læse til datamatiker :)

Men hvordan ville du sætte det op selv?

#13

Jeg kan godt lig den måde du udtaler jeg på til sidst :D

#### **Kommentar af dab93 d. 09. May 2011 | 12**

Jeg fandt et sikkert login-script, men ved ikke om det fungere, så prøver selv at gøre det sikkert med

mysql\_real\_escape\_string() og sha1() :-)

<http://phpsec.org/articles/2005/password-hashing.html>

### Kommentar af hollow d. 13. May 2011 | 13

Hovsa kom til at poste mit kode uden at sige noget om hvad det gør :)

Jeg plejer at bruge denne funktion til at kryptere de passwords som folk indtaster i mine systemer. mine `<code>` tags var et fejlslagent forsøg på at få det til at ligne kode :)

Det funktionen gør er at lave en kryptering af kodeordet i dette tilfælde bliver kodordet saltet og krypteret 321 gange, det skulle kunne rette en smule op på krypterings problemet.

Funktionen returnerer en 32 char lang streng.

### Kommentar af hollow d. 13. May 2011 | 14

Og nu kom jeg så til at skrive min kommentar uden at skrive koden fordi det åbenbart var et preview. hehe :) det kører bare. denne gang uden `<code>` tags

```
function encrypt_password($x){  
  
    $encrypt_this = $x;  
    $encryption_times = 321; // Encryption times  
    $encryption_salt = '9jgoaag1!42#og'; // spice it up  
  
    for($zo=0;$zo<$encryption_times;$zo=$zo+1)  
    {  
        $encrypt_this = hash('sha512',$encryption_salt.$encrypt_this);  
        $encrypted_password = md5($encrypt_this.$encryption_salt);  
    }  
    return $encrypted_password;  
}
```

### Kommentar af hollow d. 13. May 2011 | 15

Man kunne lave salten specifik for brugeren og det samme kunne gøres med antallet af krypteringer, det skal bare gemmes i databasen sammen med de øvrige data om brugeren

### Kommentar af magic-mouse d. 26. May 2011 | 16

Selv synes jeg at denne guide er fantastisk. Han dækker div. sikkerhedshuller, og går igennem en god læsbar struktur.

<http://www.youtube.com/watch?v=foptrvfCqdk>