



Denne guide er oprindeligt
udgivet på Eksperten.dk

Forbedret udgave af php tutorials

Jeg kigge mig lidt omkring i php artiklerne her på eksperten, og lærte mig fra alle de fejl som andre har begået, dem vil jeg prøve at rette, og lave en "forbedret udgave".

Skrevet den **12. Feb 2009** af **udvikler** | kategorien **Programmering / PHP** | ★★★★★

Tutorialen består af 8 trin:

- 1) Installation af apache,php,mysql og phpmyadmin
- 2) Kort forklaring til forbedring af php installation
- 3) Igang med det basale
- 4) variabler
- 5) if sætninger
- 6) funktioner
- 7) samarbejde mellem php og mysql

Vi går igang med det samme:

1)

Wamp er et program som installere apache,php,mysql og phpmyadmin på ens computer uden at man selv skal røre en finger.

Bedre kunne det da ikke være ? Man kan downloade det herfra:

http://prdownloads.sourceforge.net/wampserver/wamp5_1.5.0c.exe?download

eller hvis du vil, kan du først læse lidt om programmet på deres hjemmeside:

<http://www.wampserver.com/en/index.php>

Jeg bruger selv programmet, og kan kun anbefale det varmt.

Når du har hentet det, så er der ikke så mange trin, så der vil jeg ikke gå så meget i dybden, du installere det ligesom

alle andre programmer. Vælger hvor det skal gemmes osv.

...Sådan, nu er det installeret, videre til trin 2!

2)

Nu kan du selv teste om det virker ved hjælp af at lave følgende dokument og gemme det i C:\wamp\www\

```
<?php  
phpinfo();  
?>
```

gem det som info.php

åben det derefter på adressen 127.0.0.1\info.php

nu burde der komme en hele masse frem, det er alle dine php settings, der står i bund og grund hvad din

version af php

tillader. Det vil jeg dog ikke gå i dybden med, det var bare for at se om php virker. Du tænker sikkert også hvad 127.0.0.1 står for.

Jo det er skam bare adressen til din apache server, så du kan teste alle dine dokumenter.

NU skal vi så lige have optimeret php, da det kun virker hvis du skriver <?php som er start tagget i php, mens vi istedet bare kunne skrive <?.

Gå ind i disse to filer:

C:\wamp\php\php\ (konfigurations filen)

og C:\wamp\apache\php\ (konfigurations filen)

Find linierne

short_open_tag = Off

og ændre dem til

short_open_tag = On

Prøv dernæst at ændre dit info.php dokument som vi lavede tidligere til

```
<?
phpinfo();
?>
```

Åben det igen og se om der stadig står det samme ?

Gør der det har du gjort det korrekt, ellers så læs dette trin forfra igen :)

3)

Nu er hele molevitten installeret, så kan vi gå igang!

Vi starter med det hele basale og begynder så småt at udskrive tekst på skærmen.

```
<?php
echo "Hello world!";
?>
```

gem filen og åben den dernæst

Du burde få udskrevet **Hello world!**

echo er en php funktion som udskriver noget på skærmen, dvs den der skaber outputtet.

du kan læse mere om den her: <http://dk2.php.net/manual/en/function.echo.php>

og alle sætninger i php skal afsluttes med en ; det er der vel ikke noget underligt i, så der for i ikke en lang forklaring :)

Nu vil jeg så lære dig hvordan du kan kommentere din kode, dvs lægge noget tekst ind som ingen kan se i browseren med kun i selve kilde koden. De skal være til gavn for programmøren så han senere hen kan komme i tanke om

hvorfor han lige netop
skrev den kode, eller gjorde det på netop den måde som han gjorde.

en kommentar kan skrives sådan her

```
<?php
//dette er en kommentar
echo "Hello world!";
?>
```

det var den korte udgave, og den holder kun på en linie, dvs det IKKE er okay at skrive den sådan her

```
<?php
//dette er en kommentar
det er dette ikke
echo "Hello world!";
?>
```

den lange kommentar form ser således ud:

```
<?php
/*
dette er en kommentar
det er dette også
```

```
og også dette :)
*/
echo "Hello world!";
?>
```

som i kan se strækker den sig over flere linier, den kan faktisk blive ved lige så langt du vil, eller rettere
sagt lige indtil
du sætter slut tegnet */

det var den basale afdeling. Videre til 4. trin

4)

En variabel kan gemmes på følgende måde

```
<?php
$minvariabel = "hejsa";
?>
```

nu kan du så bruge echo fra eksemplet i trin nr 3 til at udskrive variabelen

```
<?php
$minvariabel = "hejsa";

echo $minvariabel;
?>
```

Læg mærke til at jeg har undladt at skrive "" omkring variabelen, det er fordi at det ikke er nødvendigt for
variabler.

Variabler kan også indeholde tal, dog kan variabel navnet ikke starte med det tal eller indeholde mellemrum, brug istedet underscore (_).

Du kan også lave regnestykker med variabler:

```
<?php
$min_variabel = 5;
$din_variabel = 3;

echo $min_variabel + $din_variabel;

?>
```

nu vil den udskrive **8**

der er også de indbygge variabler

denne viser lidt information om din browser og dit styresystem:

```
<?php
echo $_SERVER["HTTP_USER_AGENT"];

?>
```

denne viser noget om ens server som vi installere før:

```
<?php
echo $_SERVER["SERVER_SOFTWARE"];

?>
```

og til slut har vi denne som viser stien til den side du har indsat koden på:

```
<?php
echo $_SERVER["PHP_SELF"];

?>
```

5)

if sætninger er som du kan høre "hvis sætninger". De kan tjekke noget for dig.

opbygningen for if sætninger er som følger

```
<?php
if (1 == 2) {
echo "3";
}

?>
```

koden ovenfor kan omformuleres til:

hvis 1 er det samme som 2 så skal php udskriver 3. denn her == tjekker om 1 er det samme som 2 og om det er samme type. dvs tal.

og det der står imellem { og } er det som skal ske hvis 1 er det samme som 2

man kan også tilføje en else

som du kan høre er det en "ellers" sætning. Det er så (hvis du ikke har gættet det) det som skal ske hvis ikke 1 er det samme som 2

det kunne se således ud

```
<?php
if (1 == 2) {
echo "3";
}else{
echo "4";
}
?>
```

denne her siger:

hvis 1 er det samme som 2 så skal php udskriver 3 hvis ikke 1 er det samme som 2 skal den udskrive 4. og nu vil du få udskrevet 4 fordi 1 er jo ikke det samme som 2.

du kan også tjekke om variabler indeholder det du tror:

```
<?php
$min_variabel = 2;

if ($min_variabel == 2) {
echo "den indeholder det jeg tror";
}else{
echo "den indeholder ikke det jeg tror";
}
?>
```

det sidste jeg vil vise er elseif det er "ellerhvis" sætninger.

```
<?php
$min_variabel = 2;

if ($min_variabel == 2) {
echo "den indeholder et 2 tal";
}
elseif ($min_variabel == 3) {
echo "den indeholder et 3 tal";
}else{
echo "den indeholder noget andet";
}
?>
```

denne siger:

hvis min variabel indeholder tallet 2 så skal php udskrive "den indeholder et 2 tal"
hvis min variabel indeholder tallet 3 så skal php udskrive "den indeholder et 3 tal"
ellers så skal php udskrive "den indeholder noget andet"

det var det jeg havde for if sætninger

6)

funktioner kan bruges til at gemme noget i og senere hen hente det igen.

vi starter med et eksempel:

```
<?php
function hejsa() {
    echo "Hejsa, hvordan har du det?";
}

hejsa();
echo " nu har vi udskrevet funktionen";

echo "og vi udskriver den lige igen: ";
hejsa();
?>
```

der er ikke så meget at forklare der, det er så simpelt som det kunne være. :)

evt. kan du læse videre her: <http://dk2.php.net/manual/en/language.functions.php>

7)

Nu er vi så kommet til sidste kapitel hvor jeg vil forklare her hvordan php kan samarbejde via mysql.

mysql er et sprog som gør php endnu mere sjovt. Det kan bruges til at lave forums, gæstebøger, fotoalbums, billede uploads osv. med.

der er næsten ikke grænser for hvad man kan med php og mysql. En stor del (for ikke at sige det hele) af eksperten er fx. opbygget i php og mysql.

jeg vil gå direkte igang med det.

Hvis du fx vil have indsat noget tekst ind i databasen som brugeren selv har skrevet kan du gøre sådan her:

Start med at kontakte databasen, det gør du på følgende måde:

```
$host="localhost";
$user="root";
$pass="";
$db="test";

mysql_connect($host,$user,$pass);
mysql_select_db($db);
```

det betyder bare at du kontakter databasen "test" på localhost med brugernavnet root som er standard for apache servere. Du skal ikke skrive noget kodeord da der ikke er noget som standard. Husk at hvis du under installationen angav et kodeord samt brugernavn, så skal du lige ændre det.

Nu er databasen "tændt" nu kan vi gå igang med selve det det handler om - at indsætte noget i databasen.

Lav tekst feltet, og giv det navnet tekst ved hjælp af denne attribute name="tekst"
Du skal dog også huske at lave en <form> som du angiver til at være method="POST"

derefter laver du denne sætning

```
mysql_query("INSERT INTO test (tekst) VALUES ('$_POST[tekst]')") or die(mysql_error());
```

Her står der

Indsæt i databasen test feltet tekst som i mit dokument har værdien \$_POST[tekst] hvis dette ikke sker skal php bare udskrive en fejl

I databasen (evt. i phpmyadmin) så opretter du en database ved navnet test og laver et felt som hedder "tekst" (uden apostrofferne) og typen skal være text, nu trykker du bare gem.

Så er du klar til at teste det hele. Åben filen med tekst feltet og afprøv. Hvis ikke det virkede så har du gjort noget galt, du kan så prøve at starte forfra.

Det er et ret stort emne og derfor meget svært at forklare, jeg kan dog komme med et par gode links som kan få jer igang.

<http://www.webcafe.dk/artikler/php/phpmysqlintroduktion/>

og

http://netcoders.dk/articles.asp?mode=show_article&article_id=81&tech_id=2

Det var alt for denne gang, håber du kunne lide tutorialen. Tak!

Kommentar af ellebaek d. 09. Nov 2006 | 1

Vil ikke kalde denne artikel for en forbedret udgave af PHP tutorials.. Men derimod bare en række eksempler. Synes det er en meget overfladisk artikel, savner lidt mere info om hvad de forskellige elementer gør. Og hvorfor læres det kun at benytte <? når du resten af artiklen alligevel benytter <?php ?. Hvad gør de forskellige dele i WAMP? og endelig hvordan henter man de oplysninger ud igen, som lige er skrevet i databasen? Synes du springer meget meget nemt hen over noget af det der vel egentlig er kernen i PHP, nemlig samspillet med MySQL.

Alt i alt må jeg desværre sige meget mangelfuld, og bestemt ikke særlig oplysende... Der skal mange flere forklaringer ind i hvad du gør.

Kommentar af jih d. 20. Dec 2005 | 2

.... 1) brug "<?php .. ?>" istedet for "<? .. ?>", siden alle ikke kan se outputtet af koden, hvis du skriver "<? .. ?>". 2) et tal skal sammenlignes "if (\$variabel == 2) { .. }", mens en string skal sammenlignes "if (\$variabel == "hej med dig") { .. }". 3) Dette er ikke så meget af en tutorial, men et sæt eksempler af php.. hvad "forbedret" angår, vil jeg sige 2 thumbs down.. men der er så links med.. min personlige mening: Ikke værd 5 / 2.5 points..

Kommentar af plazm d. 28. Jan 2006 | 3

Tynd artikel, og forfatteren kan åbenbart ikke tage kritik.

Kommentar af mikze d. 20. Dec 2005 | 4

Okay. Godt delt op med overskriften og de 7 punkter, men dine koder syntes jeg personligt bliver forvirret når du opretter en tekst-variabel med \$minvariabel = hejsa; .. Så heller holde det overskueligt med "" omkring!

Kommentar af nizo d. 28. Dec 2005 | 5

Fra if() til mysql_query() ?
#Det her er ogsaa en kommentar
'og det er det her ogsaa

Men ellers en fin artikel, jeg synes dog ikke at du skal blande brugen af PHP med ind naar du lige har lært dem at installere Apache + PHP. De fleste nybegyndere finder nok en server hvor det hele er serveret, uden erfaringer med den slags kan det være ret svært for en nybegynder...

Kommentar af huset d. 20. Dec 2005 | 6

Ak - hvordan pokker skal man kunne få indsigt eller få startet udfra så tynd en artikel. Stop med at smide sådan noget hø ppå før du har mere styr på området.
Spild af 2,5 point!

Kommentar af there-is-only-xul d. 30. Dec 2005 | 7

WAMP er noget skrammel, og fuldt af fejl. Manual install af det hele er bedst , + men lærer en masse af det. + ting som \$_POST direkte i SQL er en 100% sikker måde at blive hacket på, du mangler en masse info omkring sikkerhed.

Kommentar af ksvendsen (nedlagt brugerprofil) d. 21. Dec 2005 | 8

Fint forklaret til en som, ikke kan finde ud af at sætte apache, php, mysql op..
Rettelsen: mysql_query("INSERT INTO test (tekst) VALUES ('\$_POST[tekst]')") or die(mysql_error()); Istedet for burde du nok lave en variable fx. \$tekst = \$_POST["tekst"]; og så bruge VALUES(" . \$tekst . ") - Men det er self. også kun min mening

Kommentar af danieru d. 23. Aug 2006 | 9

WAMP vil jeg desuden ikke anbefale. (en del fejl i)
"if (\$min_variabel == 2) {" - det ville være dejligt med et liste over hvad de betyder (Som "==" er lig med") kunne også være dejligt med at dine eksempler var fremhævet?