



Grundlæggende PHP

I denne artikel, vil der blive gennemgået de grundlæggende PHP-funktioner, såsom udskrift til skærmen, tid og dato og if-sætningen.

Skrevet den **04. Feb 2009** af **anders_h** I kategorien **Programmering / PHP** | ★★★★★

I denne artikel vil jeg skrive lidt om grundfunktionerne i PHP. Artiklen forudsætter, at du i forvejen ved hvad PHP er.

Udskrift til skærmen

Når man skal skrive tekst eller andet grafik til skærmen, bruges kommandoen echo. I den kommando kan man skrive almindelig HTML, der er sådan set ingen hokus pokus i det.

Vi starter ud med et eksempel:

```
<?php
    echo "Denne tekst bliver udskrevet til skærmen";
?>
```

Teksten mellem apostrofferne, " og ", bliver skrevet ud på klientens skærm når siden åbnes. Det kan dog anbefales, at hvis teksten (eller HTML-koden) er lang, simpelthen at lukke PHP-koden med ?>, skrive teksten eller HTML-koden og derefter åbne PHP igen med <?php

Hvis der er apostroffer i den tekst du vil have udskrevet, kan man med fordel gøre således:

```
<?php
    echo \"<span id='test'>blabla[/span]\";
?>
```

Med et backslash (\) foran apostroffen i starten og slutningen af udskriften kan man bruge apostroffer inde i selve teksten. Så ved PHP-serveren nemlig at echo-sætningen ikke slutter ved et apostrof, men først ved et apostrof hvor der står et backslash foran.

Variabler

Med en variabel har man mulighed for at genbruge den samme værdi senere i PHP-dokumentet. I denne sektion vil der blive forklaret hvordan man sætter en variabel og derefter udskriver værdien af variabelen. Selvom eksemplerne i den her sektion er små og ikke har nogle specielle fordele frem for almindelig HTML, vil du senere finde ud af hvor nødvendige variabler egentlig er i PHP-programmering.

Her er så et eksempel på en variabel:

```
<?php
    $til = "Kasper";
    $fra = "Nikolaj";
```

```
echo "Hej $til<br>Lige en hilsen :-)<br>Hilsen $fra";  
?>
```

Som du kan se, skal der være et \$ foran selve variabelens navn. Det er en god idé at bruge et navn til en variabel som er let at huske, og som er et nøgleord til variabelens indhold.

I ovenstående eksempel tildeler vi variabelen til værdien Kasper og fra værdien Nikolaj. Det bliver så udskrevet som et brev på skærmen, som vil se sådan ud:

```
Hej Kasper  
Lige en hilsen :-)  
Hilsen Nikolaj
```

Bemærk:
 er ganske almindelig HTML, som laver et linjeskift.

Variabler behøver ikke nødvendigvis at indeholde tekst. Man kan selvfølgelig også definere tal.

Vi starter ud med et eksempel:

```
<?php  
    $tal1 = 5;  
    $tal2 = 10;  
  
    $resultat = $tal1 + $tal2;  
  
    echo "$tal1 + $tal2 = $resultat";  
?>
```

I ovenstående eksempel tildeler vi variabelen tal1 værdien 5, og variabelen tal2 værdien 10. I en anden variabel som vi kalder resultat lægger vi de to variabler sammen. Efter det, udskriver vi det hele med echo. Som du nok har bemærket, er der ingen apostroffer omkring talvariabler. Det kunne der godt være, men så ville PHP bare opfatte dem som almindelige tekstvariabler, og man vil derfor ikke kunne regne med værdierne, som vi gør i ovenstående eksempel.

Eksemplet vil udskrive følgende på skærmen:

```
5 + 10 = 15
```

Man kan selvfølgelig benytte alle de 4 regnearter i PHP, og derudover kan man benytte en masse andre matematiske funktioner. Du kan finde en komplet liste over disse i PHP's egen manual;

<http://dk2.php.net/manual/en/ref.math.php>

Vi skal da lige have et eksempel mere :-)

```
<?php  
    $tal1 = 10;  
    $tal2 = 200;
```

```
$resultat = $tal1 * $tal2 / 100;

echo "$tal1 procent af $tal2 = $resultat";

?>
```

I dette eksempel vil vi finde ud af hvad 10% er af 200. Vi ganger 10 med 200 (altså tal1 * tal2) og dividerer med 100.

Det vil så udskrive på skærmen:

```
10 procent af 200 = 20
```

Man kan ændre en variabels værdi senere i dokumentet. Det er der vist et eksempel på her:

```
<?php
    $tal = 5;

    echo $tal;
    echo "<br>";

    $tal = 10;

    echo $tal;

?>
```

Først bliver tallet 5 udskrevet, fordi variabelen først indeholder 5. Bagefter (efter linjeskiftet
) bliver variabelens værdi ændret til 10. Når man så udskriver variabelen igen, vil der selvfølgelig blive udskrevet den opdaterede version, nemlig 10.

Tid og dato

PHP har mange indbyggede funktioner, her iblandt en dato funktion, som kan udskrive tid og dato. Jeg vil ikke gå i dybden med så mange af PHP's indbyggede funktioner, da der er så mange, og fordi denne tutorial kun vil forklare de mest basale ting i PHP. Men da netop tid og dato er meget brugt, vil jeg dog forklare lidt om dét.

Funktionen vi skal kigge på, hedder date.

...og vi starter ud med et eksempel:

```
<?php
    $dato = date("d/m-Y");

    echo $dato;

?>
```

Der er ikke apostroffer omkring indholdet af echo, da det ikke er nødvendigt når der bare skal udskrives én enkelt variabel.

Umiddelbart kan det godt se lidt avanceret ud, men der er faktisk et system i date funktionen  d'et udskriver dagen i måneden, skråstregen betyder bare at vi vil have / til at adskille dagen i måneden

fra månedsnummeret. m betyder at månedsnummeret skal udskrives. Bindestregen betyder at vi vil have adskilt månedsnummeret fra året med -. Y udskrives året. Det vi skriver, er selve syntaksen til, hvordan vi vil have udskrevet datoen.

Der er forskel på store og små bogstaver, d er derfor ikke det samme som D.

Alt efter hvilken dato det er, vil ovenstående eksempel udskrive følgende på skærmen:

```
05/08-2005
```

Hvis vi også vil have klokkeslettet med, bruger vi bogstaverne H og i. H udskrives timerne, mens i udskrives minutterne. Date kunne derfor se således ud:

```
<?php
    $dato = date("d/m-Y H:i");

    echo $dato;
?>
```

Det er selvfølgelig fuldstændig valgfrit hvordan man vil have stillet det op, men hvis man gør ligesom jeg har gjort ovenover, vil det cirka udskrive det her på skærmen:

```
05/08-2005 10:18
```

Der er mange andre forskellige bogstaver man kan bruge, bl.a. til at udskrive navnet på dagen (dog på engelsk). En komplet liste over de forskellige "bogstaver" er at finde her:

<http://dk2.php.net/manual/en/function.date.php>

If-sætningen

If betyder jo oversat fra engelsk hvis. Med den sætning, kan man tjekke om et eller flere udsagn er sande eller falske.

Forestil dig f.eks. et login-system, hvor en bruger skal have mulighed for at logge ind. Der kan man lave en if-sætning, som siger; [hvis brugeren er logget ind, udfør én funktion, hvis ikke - udfør en anden funktion].

Men et eksempel er jo altid godt:

```
<?php
    $film = "Olsen Banden";

    if ($film == "Olsen Banden")
    {
        echo "Filmen er Olsen Banden";
    }
?>
```

I dette eksempel tildeler vi variabelen film værdien Olsen Banden. If-sætningen spørger således: "Hvis film er det samme som Olsen Banden gør følgende:". Imellem klammerne { og } laver man en handling hvis udsagnet af sætningen er korrekt. Hvis udsagnet ikke er korrekt bliver der ikke udskrevet noget på

skærmen, men da det i dette tilfælde er helt rigtigt, vil det selvfølgelig udskrive

```
Filmen er Olsen Banden
```

på skærmen. Hvis du undrer dig over hvorfor der er dobbeltlighedstegn (==) i if-sætningen, er det fordi der bliver brugt dobbeltlighedstegn hvis det er en tekst og et enkelt lighedstegn hvis det er et tal. Et eksempel med tal kunne f.eks. se sådan her ud:

```
<?php
    $tal = 20;

    if ($tal == 20)
    {
        echo "Tallet er 20";
    }
?>
```

Men hvad nu hvis man kun vil have udskrevet noget hvis udsagnet er forkert? Så må vi bruge tegnet !, altså et udråbstegn, som i PHP-sprog betyder ikke. Vi kan prøve at lave eksemplet ovenfor om til, at den kun skal udskrive noget hvis udsagnet er falsk:

```
<?php
    $tal = 20;

    if ($tal != 20)
    {
        echo "Tallet er ikke 20";
    }
?>
```

Dette eksempel vil ikke udskrive noget som helst, da tal jo rent faktisk indeholder 20, og dermed er udsagnet falsk.

Bemærk: Hvis det er en tekstvariabel, skal man ikke skrive != men bare != som der også blev gjort i dette eksempel. Hvorfor - ved jeg ikke, men sådan er det bare ...

Vi skal nu have fat på else if. Denne funktion er integreret i if-funktionen, og gør det muligt at tjekke flere værdier i én if-sætning.

Vi kan jo lige tage et teoretisk eksempel på dansk:

```
Start PHP
```

```
Tal = 20
```

Hvis [Tal] er det samme som 5, gør
Udskriv "Tallet er 5" på skærmen.

Ellers hvis [Tal] er det samme som 10, gør
Udskriv "Tallet er 10" på skærmen.

Ellers hvis [Tal] er det samme som 20, gør
Udskriv "Tallet er 20" på skærmen.

Ellers
Udskriv "Tallet kunne ikke findes" på skærmen.

Slut Hvis

Luk PHP

Hvis eksemplet ovenfor ellers er til at finde ud af, bliver der først tjekket om variabelen Tal er det samme som 5. Hvis det er tilfældet, går PHP-serveren ikke videre i if-sætningen, men udfører det der står under 5. Altså i dette tilfælde, skal den udskrive "Tallet er 5" på skærmen. Hvis tal ikke er det samme som 5, går den videre til den næste, hvor der bliver tjekket om tallet er 10. Hvis ikke, går den videre til 20, som i dette eksempel jo er sandt, og så bliver der udskrevet "Tallet er 20" på skærmen. Men hvis nu variabelen tal, indeholdt værdien 25, ville ingen af "hvis'erne" jo være sande. Så kan man lave en "Ellers", som kun bliver udskrevet, hvis ingen af udsagnene i den pågældende if-sætning er sande. Hvis vi antager at ingen er sande, og vi ikke har en "Ellers", vil intet blive udskrevet.

Hvis man skulle lave mit teoretiske eksempel i praksis, ville det se sådan ud:

```
<?php
    $tal = 20;

    if ($tal = 5)
    {
        echo "Tallet er 5.";
    }
    else if ($tal = 10)
    {
        echo "Tallet er 10";
    }
    else if ($tal = 20)
    {
        echo "Tallet er 20";
    }
    else
    {
        echo "Tallet kunne ikke findes.";
    }

?>
```

Jeg håber på at jeg kan finde tid til at skrive en Del 2, hvis det skulle have interesse!

Mvh. Anders

Kommentar af reimar d. 06. Sep 2005 | 1

God artikel! Se også gratis god php-kursus på <http://www.learn-php-tutorial.com>

Kommentar af repsac d. 30. Aug 2005 | 2

:-D 'if (\$tal = 20)'.

Der mangler rigtig mange ting førend man kan begynde at bruge PHP til noget fornuftigt. Gå ned på dit lokale bibliotek og lån eksempelvis 'PHP Bible'. Det tager meget længere tid, men den kan du bruge til noget, hvis du vil bruge PHP til noget. Hvis du ikke vil bruge PHP til noget, men bare vide hvordan det dersens PHP-kode ser ud, så er denne artikel ok. Men så kan du jo bare søge på nettet, og så synes jeg webcafe.dk/artikler/php giver en langt bedre introduktion.

Kommentar af saudoo d. 03. Sep 2005 | 3

Lidt tynd. Med mindre der selvfølgelig kommer en nummer 2. :)

Kommentar af plazm d. 26. Aug 2005 | 4

du har lige en fejl i dine eksempler.

i if sætningerne skal det være == ellers bliver \$tal faktisk sat til noget andet i stedet for det bliver tjekket. Derudover er det en meget god artikel, men synes du burde have taget med at man kan bruge echo "Hej " . \$til . "
";

da det er noget pænere i mine øjne

Kommentar af wicez (nedlagt brugerprofil) d. 26. Aug 2005 | 5

God artikel - Gav mig lidt mere info om det basale php. Men som plazm også skriver, skal du nok holde variablerne udenfor dine strenge med ". \$var ." ;o)

Kommentar af venchil d. 26. Aug 2005 | 6

Sætter begynderen ind i grundlæggende PHP... Ret else-sætningen i din titel til if-else-sætningen? Som Plazm også siger, så bør du lige tjekke dine eksempler... :)

Kommentar af there-is-only-xul d. 01. Sep 2005 | 7

Meget rimeligt, og det bedste forsøg jeg har set indtil videre, dog vil jeg KRAFTIGT fraråde `echo "\blabla\";` , det anbefales heller ikke af PEAR standarden for php kode. brug `echo "\blabla\n";` istedet, det sidte `\n` er for at få linjeskift med i HTML "vis kilde" hvilket er god til debugging.

Kommentar af mikkel-php d. 11. Sep 2005 | 8

Jeg fik helt sikkert noget ud af den ;)