



Simpel e-mailfunktion med validering

Her kan du læse, hvordan du laver en e-mailfunktion med validering. Den tjekker om man har udfyldt alle felter, og om det udfyldte navn og den udfyldte e-mailadresse er korrekt.

Jeg har forklaret de enkelte dele grundigt.

Skrevet den **03. Feb 2009** af **mstorgaard** I kategorien **Programmering / PHP** | ★★★★★

Indledning

Jeg synes efterhånden, at det er lang tid siden, at der er blevet skrevet en e-mailfunktion, så det måtte blive min tur til at bidrage med en artikel. Jeg har forsøgt at skrive artiklen i så korrekt PHP som muligt, men I er velkomne til at kommentere på det, hvis I finder det utilstrækkeligt.

Indhold

Funktionen indeholder blandt andet et tjek af om felterne er udfyldt og yderligere tjek om navn og e-mail er korrekt udfyldt. Jeg har med vilje valgt ikke at foretage yderligere validering på emne og besked, da de let kan indeholde unormale tegn, som jeg nødt vil komme til at validere væk.

Når der så fremkommer en fejlmeddelelse, for at fx e-mailadresse ikke er udfyldt, så gør funktionen at den skriver informationerne ned i felterne igen, så man ikke skal bruge tid på at skrive det hele ind igen.

Yderligere sletter funktionen de variabler, når e-mailen bliver sendt, således man ikke lægger op til at folk skal udnytte formen til at udsende spam.

Funktionen

Hele funktionen kan ses herunder. Jeg vil efterfølgende forklare de enkelte dele.

```
<?php
if(isset($_POST['navn'])) { $navn = $_POST['navn']; }
if(isset($_POST['email'])) { $email = $_POST['email']; }
if(isset($_POST['emne'])) { $emne = $_POST['emne']; }
if(isset($_POST['besked'])) { $besked = $_POST['besked']; }
if(isset($_POST['submit'])) {
if(empty($_POST['navn']) || empty($_POST['email']) || empty($_POST['emne']) ||
empty($_POST['besked'])) {
if(empty($_POST['navn'])) { echo "Navnet er ikke udfyldt<br />"; }
if(empty($_POST['email'])) { echo "E-mailadressen er ikke udfyldt<br />"; }
if(empty($_POST['emne'])) { echo "Emnet er ikke udfyldt<br />"; }
if(empty($_POST['besked'])) { echo "Beskeden er ikke udfyldt<br />"; }
} else {
if(!preg_match("[a-zA-zæøåÆØÅ]", $_POST['navn']) || !preg_match("^([a-zA-ZæøåÆØÅ0-9._-]+@[a-zA-ZæøåÆØÅ0-9-]+\.[a-zA-ZæøåÆØÅ]{2,5})$", $_POST['email'])) {
if(!preg_match("[a-zA-zæøåÆØÅ]", $_POST['navn'])) { echo "Navnet er ikke korrekt<br />"; }
if(!preg_match("^([a-zA-ZæøåÆØÅ0-9._-]+@[a-zA-ZæøåÆØÅ0-9-]+\.[a-zA-ZæøåÆØÅ]{2,5})$",
$_POST['email'])) { echo "E-mailadressen er ikke korrekt<br />"; }
} else {
mail("michael@storgaard.name", $_POST['emne'], $_POST['besked'], "From: " . $_POST['navn'] . " <" .
$_POST['email'] . ">");
echo "Beskeden er sendt<br />";
$navn = "";
$email = "";
$emne = "";
}
```

```

$besked = "";
}
}
}
?>
<form method="post" action="<?php echo $_SERVER['PHP_SELF']; ?>">
  Navn: <input type="text" name="navn" value="<?php echo $navn; ?>" /><br />
  E-mail: <input type="text" name="email" value="<?php echo $email; ?>" /><br />
  Emne: <input type="text" name="emne" value="<?php echo $emne; ?>" /><br />
  Besked: <textarea name="besked"><?php echo $besked; ?></textarea><br />
  <input type="submit" name="submit" value="Send" />
</form>

```

Denne del gør at variabler bliver tildelt felternes indhold. Dette bruges til at udfylde feltene igen, hvis der fremkommer en fejlmeddelelse:

```

if(isset($_POST['navn'])) { $navn = $_POST['navn']; }
if(isset($_POST['email'])) { $email = $_POST['email']; }
if(isset($_POST['emne'])) { $emne = $_POST['emne']; }
if(isset($_POST['besked'])) { $besked = $_POST['besked']; }

```

Denne del sætter processen i gang, hvis der er blevet trykket på Send-knappen:

```

if(isset($_POST['submit'])) {

```

Denne del tjekker om alle feltene er udfyldt. Først tjekker den om nogen af feltene er tomme, og hvis en af dem er tomme, går den videre og tjekker hvilke eller hvilket felt, som ikke er udfyldt. Til sidst sender den os videre, hvis feltene er udfyldte:

```

if(empty($_POST['navn']) || empty($_POST['email']) || empty($_POST['emne']) ||
empty($_POST['besked'])) {
if(empty($_POST['navn'])) { echo "Navnet er ikke udfyldt<br />"; }
if(empty($_POST['email'])) { echo "E-mailadressen er ikke udfyldt<br />"; }
if(empty($_POST['emne'])) { echo "Emnet er ikke udfyldt<br />"; }
if(empty($_POST['besked'])) { echo "Beskeden er ikke udfyldt<br />"; }
} else {

```

Denne del tjekker om navn og e-mailadresse er udfyldt korrekt. Først tjekker den, via regexp, om nogen af dem indeholder fejl, og hvis de gør, går den videre og tjekker hvilke eller hvilket felt, som indeholder fejl. Til sidst sender den os videre, hvis navn og e-mailadresse er korrekt udfyldte:

```

if(!preg_match("[a-zA-zæøåÆØÅ]", $_POST['navn']) || !preg_match("^[a-zA-ZæøåÆØÅ0-9._-]+@[a-zA-ZæøåÆØÅ0-9.-]+\.[a-zA-ZæøåÆØÅ]{2,5}$", $_POST['email'])) {
if(!preg_match("[a-zA-zæøåÆØÅ]", $_POST['navn'])) { echo "Navnet er ikke korrekt<br />"; }
if(!preg_match("^[a-zA-ZæøåÆØÅ0-9._-]+@[a-zA-ZæøåÆØÅ0-9.-]+\.[a-zA-ZæøåÆØÅ]{2,5}$",
$_POST['email'])) { echo "E-mailadressen er ikke korrekt<br />"; }
} else {

```

Denne del afsender e-mailen, fortæller at den har gjort det og resætter variablerne med felternes indhold. Første del af mail() er din e-mail, som beskeden sendes til. Herefter indskrives de forskellige felters

indhold. Herefter skriver den at beskeden er sendt, hvorefter den tømmer variablerne, så den senest afsendte besked ikke kommer til at stå i felterne:

```
mail("navn@domæne.dk", $_POST['emne'], $_POST['besked'], "From: " . $_POST['navn'] . " <" .  
$_POST['email'] . ">");  
echo "Beskeden er sendt<br />";  
$navn = "";  
$email = "";  
$emne = "";  
$besked = "";
```

Denne del bruges til at afslutte de forskellige dele. Den er der ikke så meget at sige om, men syntes lige den skulle med (:

```
}  
}  
}
```

Og får lige at få de sidste tabte sjæle med, så kan jeg fortælle at jeg bruger dette til at indlemme mine PHP-koder:

```
<?php  
?>
```

Denne del er selve formen. Først definerer jeg hvilken måde jeg bruger til at sende med, og derefter bruger jeg \$_SERVER['PHP_SELF'] til at udføre vores submit i denne fil. De næste er inputfelterne, og så indsætter jeg en værdi, hvis det skal bruges. Dem har jeg defineret i starten af funktionen, og hvis e-mailen er blevet sendt er de variabler så blevet tømt igen. Den sidste del er knappen, som jeg bruger til at sende. Det vigtige ved den er at den hedder submit, da det er den, som sætter funktionen i gang, når der er blevet trykket sendt.

```
<form method="post" action="<?php echo $_SERVER['PHP_SELF']; ?>">  
  Navn: <input type="text" name="navn" value="<?php echo $navn; ?>" /><br />  
  E-mail: <input type="text" name="email" value="<?php echo $email; ?>" /><br />  
  Emne: <input type="text" name="emne" value="<?php echo $emne; ?>" /><br />  
  Besked: <textarea name="besked" cols="10" rows="3"><?php echo $besked; ?></textarea><br />  
  <input type="submit" name="submit" value="Send" />  
</form>
```

Afslutning

Tak, fordi du læste med, og jeg håber, du kan bruge min artikel til noget.

Giv endelig konstruktive kommentarer.

Opdateringer

27-02-2007: ÆØÅ-validering tilføjet

27-02-2007: ereg() erstattet med preg_match()

Kommentar af a1 d. 01. Mar 2007 | 1

ja, men e-mail systemet (smtp/mail) har ALDRIG ville/kunne understøtte mærkelige navne (tegn).

Du kan heller ikke sende en mail til f.eks. minsøn@dinmæil.dk fra outlook...

Kommentar af noob2003 d. 26. Feb 2007 | 2

surt at hedde Søren Sørensen, så kan man ikke benytte den :(

ÆØÅ æøå

Kommentar af sneska d. 28. Sep 2011 | 3

Hvordan skal regex tjekket/validering så se ud for at give mening?

Er dette nok sikkerhed for email tjek ?

```
function isValidEmail($email)
```

```
{  
return preg_match('/^([a-zA-Z0-9])+([a-zA-Z0-9\._-])*@([a-zA-Z0-9_-])+([a-zA-Z0-9\._-]+){2,4}+$/',  
$email);  
}
```

Og er det nødvendigt at sikre at man ikke bruger special tegn som ÆØÅ?

Og en sidste ting - Subject feltet SKAL være i formen, men kan man ikke bruge den til andet en emne hvis man ikke har behov for et sådan felt? eller lade være med at bruge den og sætte css'en til display none ?

Håber der er nogen der kan hjælpe :) tak :)