



Upload af billeder/filer for nybegyndere Del. 2

I denne Del vil jeg ikke gaa saa dybt i fil upload som i sidste. Jeg vil i stedet forklare om funktioner der kan bruges til at finde tilhørende oplysninger af filer.

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

For at begynde et sted vil jeg forklare lidt om de oplysninger man kan finde med variabelen `$_FILES` (eller, `$HTTP_FILES_VAR`).

\$_FILES er opbygget i array's, præcis ligesom `$_GET` og `$_POST`, der for, naar du sender en fil (med POST) bliver filen automatisk uploadet til et sted der kun er midlertidigt, for at komme i kontakt med disse filer bruger vi `$_FILES`. Den første array finder filnavnet. Dvs. hvis du har en form med et input af typen "file" (`<input type="file" name="fil" />`) er `name=""` det navn vi skal bruge i array'et for at finde frem til filen. Og det gør vi med `$_FILES['fil']`. Naar vi nu skal finde oplysninger om denne fil, fx placering/navn/type sætter vi en ny array paa som kan være ex name: `$_FILES['fil']['name']`.

`$_FILES['fil']['tmp_name']`

er nok den mest optimale at begynde med.

Med Array'en `tmp_name` kan vi finde den midlertidige fils placering, og det er den vi bruger i upload funktioner. Hvis man echo'er `$_FILES['fil']['tmp_name']` vil det noglelunde saadan her ud: `C:\2d324n24kkm`. Hvertfald noget i den stil er ikke helt sikker :)

`$_FILES['fil']['name']`

kan vi tage som den næste.

Det er simpelthen bare det oprindelige navn paa filen, som php.net siger: *The original name of the file on the client machine.*

`$_FILES['fil']['type']`

Denne funktion bruger vil til at finde MIME typen af filen, hvis vi tager et GIF billede som ex vil echo `$_FILES['detteerengiffil']['type']` vil den echo'e `image/gif` MIME typen er den samme man bruger som header (evt.).

`$_FILES['fil']['size']`

udsteder bare størrelsen i bytes. Bytes formaten er 1 Kilo bytes = 1024 bytes.

`$_FILES['fil']['error']`

vil jeg ikke komme ind paa, men der kan læses alt om den paa:

Jeg vil nu give et eksempel hvor jeg bruger bla. `$_FILES['fil']['size']` og `$_FILES['fil']['type']` (jeg vil ikke sige koden virker optimalt da dette kun er et eksempel!)

```
<?
if($_FILES['fil']['size'] > 10240 && $_FILES['fil']['type'] != "image/gif")
{
    echo "Noget er galt! Files type er: " . $_FILES['fil']['type'] . ", og filen fylder " . $_FILES['fil']['size'] . " bytes.
    Den skal overholde filtypen image/gif som er den eneste du må bruge, derudmover skal filen fylde
    mindre end 10240 bytes (10kb).";
}
else
{
    //Upload af filen ($_FILES['fil']['tmp_name']) slutter som regel med move_uploaded_file()
    $filnavn=$_FILES['fil']['name'];
    $filnavn=str_replace(" ", '_', $filnavn);
    //osv osv...

    echo"Filen blev uploadet succesfuld!";
}
?>
```

Her (burde) den finde ud af om filen fylder for mindre end det den maa, eller overholder filtype grænsen. Som et nyt eksempel satte jeg en `str_replace` der fjerner alle mellemrum i det navn filen havde paa klient computeren. Og derfor, hvis man ikke selv bestemmer filnavnet, sikre man sig at filen ikke faar et eller andet underligt navn med en masse mellemrum, hvilket ikke er smart paa webservere!

Ikke uploadede filer

Hvis man nu gerne vil finde oplysninger om en fil, som ikke er uploadet via en form i det øjeblik, men ligger paa serveren findes der self. PHP funktioner til at finde oplysninger om dem, hvilket ogsaa kan bruges til upload af filer! Disse funktioner er en del mere avancerede end de overstaende.

getimagesize()

Denne funktion bliver brugt i de fleste tilfælde til oplysninger om billedet, hvilket ogsaa er oplyst i funktionens navn (find billede størrelse (groft oversat)).

Denne funktion udskriver et array og det betyder du ikke kan udskrive meget andet end et kort "Array." hvis du prøver at `echo`'e den (`echo getimagesize("billede.gif")`). Derfor skal vi bruge en kode som kan fortælle os om hvordan pokker vi faar fat i de array navne.

En simpel udskrivning kan gøres saadan her:

```
<?
$billede=getimagesize("test_billede.gif");
```

```
foreach($billede as $info)
{
    echo $info . "\n";
}
?>
```

Men den kode vil kun fortælle os værdierne i array'en derfor maa vi gøre noget mere ved det loop (foreach()).

Og jeg har valgt at gøre saadan her:

```
<?
$billede=getimagesize("test_billede.gif");

foreach($billede as $info => $i)
{
    echo $info . " -> " . $i . "\n";
}
?>
```

Nu ser vi lige pludselig noget andet!

\$info er blevet til en oplysning om hvad array navnet hedder til array informationen. \$i blev til værdien af array'et. Og derfor vil det se saadan her ud:

```
0 -> 300
1 -> 300
2 -> 1
3 -> width="300" height="300"
bits -> 6
channels -> 3
mime -> image/gif
```

Den kode oplyste os om alt hvad vi skulle bruge.

```
$billede=getimagesize("test_billede.gif");
```

\$billede[0] og **\$billede[1]** giver os dimensionerne paa billedet, 0 er bredde og 1 er højde.

\$billede[2] har jeg faktisk ikke rigtig nogle anelse om hvad gør.

\$billede[3] fortæller os i fuld HTML sætning om dimensionerne, og af den grund (HTML) kan variabel placeres direkte i en HTML kode

fx

```
 />
```

som giver os:

```

```

\$billede[bits] siger sig selv størrelsen i bits.

\$billede[channels] ...tror jeg er farve "kanaler".

\$billede[mime] er filtypen i MIME form igen image/gif eller image/jpeg osv..

MEN HOV! Nu mangler vi jo den præcise størrelse i bytes!
og den finder vi med en simpel *filesize()*

fx

```
<?
echo "Filens fylder " . filesize("test_billede.gif") . " bytes";
?>
```

Det var saa de mest alm. oplysninger. Men der findes self. MANGE, som alle kan findes paa <http://php.net>

Det var alt jeg ville fortælle i denne Del. Det var self. ikke meget om fil/billede upload, men disse funktioner kan bruges til at sætte limits over for brugere om hvor meget de fx kan tillade sig at deres billeder fylder. Jeg kunne meget godt tænke mig senere at komme med noget resize af billeder, og vil nok ogsaa gøre det :) Haaber at i (nybegyndere) lærte noget af denne lille artikel.

Hvis nogle synes der er ting jeg mangler at faa belyst lytter jeg gerne til kommentare :o)

Kommentar af karstenl d. 26. Oct 2005 | 1

Samme artikel/hjælp til i ASP. ville være super ;)

Kommentar af the_email d. 27. Oct 2005 | 2

Endnu en god artikel af nizo. Glæder mig til dem om resize af billeder

Kommentar af mikze d. 26. Oct 2005 | 3

Fin :-)

Kommentar af nyhuusdk d. 13. Aug 2006 | 4

Meget god artikkel, jeg har aldrig arbejdet med upload før, men denne og din første artikkel har hjulpet meget. jeg glæder mig til nr3 og håber på du laver en mere beregnet til billeder hvor den automatisk laver

tumbs og stadig gemmer det originale billede :)

Kommentar af mik2000 d. 04. Oct 2006 | 5

Hej
\$billede[2] har jeg faktisk ikke rigtig nogle anelse om hvad gør.
Denne giver typen af fil. F.eks. 1 for jpg, 2 for gif osv
Se mere på php.net

Kommentar af visualdeveloper d. 25. Oct 2005 | 6

TOP ;)

Kommentar af lazniak d. 27. Oct 2005 | 7

Lige hvad jeg havde brug for! Tak!

Kommentar af electricnet d. 27. Oct 2005 | 8

Virkelig nyttig! Det her er en virkelig god serie, glæder mig til næste del!

Kommentar af poptex d. 21. May 2006 | 9

God artikel, der var noget jeg ikke viste om upload af filer.