



Webdesign - At bruge "includes"

Introduktion til "includes". Includes gør det muligt for webdesigneren at ændre hundredvis af sider, ved kun at ændre på én enkelt fil! En vis viden om HTML forudsættes. SSI, PHP, ASP, FrontPage og JavaScript gennemgås kort.

Skrevet den **11. Feb 2009** af **old-faithful** | kategorien **Internet / Generelt** | ★★★★★

1. Hvad er en "include"?

Direkte oversat betyder "include" *omfatte, indbefatte* eller *inkludere*. Dette er faktisk en meget rammende beskrivelse.

Includes virker kort sagt ved at man indsætter et kodelykke i sit webdokument. Dette kodelykke gør så at et andet webdokument automatisk vises, der hvor man har indsat koden. Brugeren ser dog aldrig dette. For ham, vil det se ud som én sammenhængende side.

2. Hvorfor bruge includes?

Det smarte ved includes er, at man kan let kan ændre på et i princippet uendeligt antal sider, ved blot at ændre på én fil.

Det kan f.eks. være at du har en Copyright-besked nederst på alle dine hjemmesides undersider, hvor der står et årstal. Medmindre du vil bruge et script til at ændre årstallet automatisk, bliver det hurtig en enorm opgave at ændre årstallet i masser af sider.

Eller måske har du en hjemmeside, med en navigationsmenu, som skal ændres eller opdateres. Bruger du includes, kan du blot ændre menuen i én fil, og så vil den opdaterede menu automatisk dukke op på alle sider, der bruger denne menu. Men bruger du ikke includes, kan du blive nødt til at sidde og bruge timevis på at ændre hver enkelt side automatisk (se dog afsnittet om alternativer til includes).

Eller måske bruger du et logo som du er blevet træt af, på alle dine sider. Har du indsat logo-koden med includes, skal du blot ændre denne kode i én fil, og vupti, så har du et nyt logo på alle dine sider.

Du kan selvfølgelig bruge flere includes i samme skabelon. Således kan du eksempelvis bruge includes til samme sides logo, menuer, ophavsretsbeskeder m.m.

Includes er naturligvis mest relevante, for dem der ikke bruger rammer (frames) til menuer.

De fleste større hjemmesider bruger en eller anden form for includes. Men man kan naturligvis ikke umiddelbart se det. Includes er lavet så de kan implementeres sømløst ind i sidens design.

Din hjemmeside er måske lille og undseelig i dag, men du vil ærge dig den dag dine hjemmeside har vokset sig stor og stærk, og du ingen lette muligheder har for at ændre på sidens generelle opbygning og udseende. Og kobler du brugen af includes sammen med CSS (cascading style sheets) kan du hurtigt få stablet en solid, dynamisk hjemmeside på benene. Derfor: Læs artiklen og lær at bruge includes!

3. Generelt - Hvordan laver man includes?

Som nævnt skal du først have en side (herefter »*hovedsiden*«), som du vil inkludere indhold i (herefter »*indholdssiden*«). Denne side indsætter du så et kodestykke i. Dette kodestykke gør, at det angivne indhold hentes ind på samme sted som kodestykket står.

Det er vigtigt at du ikke inkluderer en hel side med <HTML>, <HEAD> og <BODY> tags. Indholdssiden skal kun indeholde den HTML-kode (eller lignende) som kræves på det pågældende sted i dokumentet. De nævnte tags findes jo allerede i hovedsiden!

Det er ofte fordelagtigt at indsætte nævnte kodestykke i et afgrænset område, f.eks. en tabel eller en tabelcelle. Ellers kan indholdssidens indhold brede sig ud over andre dele af siden.

Du kan sagtens indsætte flere "includes" i samme webdokument.

Det anbefales at lægge alle sine include-filer i én mappe, så de er lette at holde styr på. Det kunne f.eks. være en mappe kaldet "includes" eller "skabelon".

Der findes flere forskellige metoder at inkludere indhold på, bl.a. SSI (server side includes), PHP, ASP og JavaScript. Disse vil jeg gennemgå i det følgende:

3a. Metode I: SSI-includes

Server Side Includes er måske en af de letteste måder at inkludere indhold.

Det kodestykke du skal indsætte, når du bruger SSI, ser f.eks. sådan her ud:

```
<!--#include virtual="/mappe/file.htm" -->
```

"Mappe/file.htm" er navnet og placeringen på indholdssiden. Dette kan du naturligvis erstatte med placeringen og navnet på dine egne sider.

Dette kan også være eksempelvis en tekstfil eller en html-fil. Ovenstående kode er bedst at bruge, hvis indholdssiden ikke ligger i samme mappe som hovedsiden. Ligger indholdssiden derimod i samme mappe (eller en undermappe i denne mappe), kan du bruge flg. kode:

```
<!--#include file="file.htm" -->
```

Det er som regel nødvendigt at omdøbe alle hovedsider, så filnavnet ender på *.shtml* (eller *.shtm*). Dette skyldes at serveren ikke ellers lægger mærke til, at noget skal inkluderes. Du kan omgå denne begrænsning på nogle servere, ved at oprette en fil kaldet **.htaccess** og indsætte flg. i denne fil:

```
Options Includes FollowSymLinks
AddType text/x-server-parsed-html .html
AddType text/x-server-parsed-html .htm
AddType text/x-server-parsed-cgi .cgi
```

Bruger du ovenstående kode (og understøtter din udbyder det) behøver du således IKKE at omdøbe dine filer til *.shtml*-filer.

Ikke alle hjemmesideudbydere understøtter SSI. For at teste om din udbyder understøtter SSI, kan du bruge testen som beskrives på flg. side:

- <http://www.htmlgoodies.com/beyond/webmaster/article.php/3473341>

Du kan læse mere om SSI her:

<http://forums.seochat.com/search-engine-articles-59/ssi-why-and-how-to-use-server-side-includes-10825.html>

3b. Metode II: PHP-includes

Du kan også bruge PHP-includes. Dette kræver at din udbyder understøtter PHP.

Koden til at indsætte PHP-includes ser sådan her ud:

```
<?php include('/mappe/fil.php'); ?>
```

Som regel skal hovedsiderne gemmes med filnavne der ender på ".php", før det virker. Du kan inkludere php-filer, html-filer, tekstfiler osv. ved at bruge PHP-includes.

En væsentlig fordel ved at bruge PHP-includes er, at du kan inkludere filer fra helt andre hjemmesider (andre URL'er). Det kan du ikke med eksempelvis SSI.

Læs mere om PHP-includes: <http://www.tizag.com/phpT/include.php>

3c. Metode IV: ASP-includes

ASP-includes skrives på samme måde som SSI-includes, f.eks.:

```
<!--#include virtual="/mappe/file.inc" -->
```

eller...

```
<!--#include file="file.inc" -->
```

Du skal sørge for at dine hovedsiders filnavne ender på ".asp". Du kan inkludere filer af typen .asp, .html, .txt osv.

Læs mere om ASP-includes: http://www.w3schools.com/asp/asp_incfiles.asp

3d. Metode V: JavaScript includes

Du kan også bruge JavaScript til at inkludere filer. Denne mulighed er - efter min mening - ikke så smart at benytte, dels af hensyn til søgemaskineplacering, dels fordi koden er mere besværlig at implementere.

I hovedsiden skal du indsætte flg. tekststykke:

```
<SCRIPT language="JavaScript" src="http://www.yourdomain.com/content/yourfile.js"></SCRIPT>[/span]
```

Derudover skal din indholdsside (her *yourfile.js*) opbygges på en helt bestemt måde. Et eksempel herpå:

```
<!--  
document.writeln('linje 1');  
document.writeln('linje 2');  
document.writeln('linje 3');  
document.writeln('osv.');
```

Først i filen skal der altså stå `<!--` og til sidst i filen skal der stå `//-->`. Den HTML-kode som du vil indsætte, skal stå mellem **document.writeln('** og **');**

Derudover er der visse tegn som du ikke må bruge direkte i koden. Du kan dog bruge det gratis program JS-DUC til at omforme indholdet automatisk for dig: <http://digital.hollmen.dk/products/js-duc/index.htm>

En meget god artikel om JavaScript includes, kan du læse her: http://www.web-source.net/dynamically_update.htm

3e. Metode VI: FrontPage includes

Microsoft's FrontPage program har også en særlig måde at lave includes på. For at bruge denne slags include, skal din hjemmesideudbyder understøtte FrontPage extensions.

Koden som du skal indsætte i hovedsiden er:

```
<!--webbot bot="Include" U-Include="file.htm" TAG="BODY" -->
```

Denne kode kan du indsætte ved at bruge FrontPage menuen Insert -> Component -> Include page (engelsk udgave).

Bemærk at det i nogle tilfælde kan være problematisk at bruge JavaScript i indholdssiden, grundet det forhold, at JavaScript ofte kræver at et vist kodelykke er indsat i dokumentets <HEAD> sektion.

For at sikre at FrontPage includes virker bedst muligt, bør du som regel bruge programmets indbyggede "Publish" funktion (og ikke f.eks. FTP-upload).

4. Includes og søgemaskiner/SEO

Den almindelige konsensus er, at SSI, PHP og ASP includes ikke påvirke placeringen i søgemaskiner, eller søgemaskinernes indeksering. Dette skyldes at søgemaskinerne får det "færdige resultat" serveret af serveren. Disse include-typer er med andre ord serverbaserede og serveren sørger for at sammensætte indholdet.

Modsat er JavaScript-includes *klientbaserede*. Her er det brugerens computer (eller søgemaskinen) der selv stykker dokumenterne sammen. Dette er en klar ulempe, hvis pågældende søgemaskine ikke understøtter JavaScript-includes, for så læses indholdssiden slet ikke! Mens der formentlig er flere søgemaskiner der understøtter JavaScript includes i dag, er det sikrest ikke at benytte JavaScript-includes, hvis man vil være sikker på at hele ens side indekseres af søgemaskinerne.

5. Bemærkninger om hyperlinks og filplaceringer

Det er smart at vide lidt om hvordan websider "finder" hinanden - dvs. hvordan du bruger links (sidehenvisninger). Viden herom er også relevant, når man bruger includes, særligt når websider ligger i forskellige undermapper men bruger samme skabelon. Det er nemlig vigtigt, at overveje hvorvidt relative eller absolutte henvisninger i include-koden skal bruges. Afhængig af din mappestruktur, kan valg af den forkerte type henvisninger/links betyde, at henvisningerne "viser forkert" (og at filer således ikke kan findes).

Kort sagt findes der *relative* og *absolutte* links. Relative links bruger pågældende dokumentets placering, som udgangspunkt for at finde filer/sider, der henvises til. Ved absolutte links er det dog som udgangspunkt ligegyldigt hvor det henvisende dokument ligger. Som en metafor kunne man bruge en vejvisningshistorie:

Lad os sige at du leder efter moster Ernas hus, og du spørger en fremmede hvor det ligger. Denne fremmede siger måske "det er andet hus på højre hånd" (en relativ angivelse). Men havde den fremmede stået et andet sted og sagt det samme, ville du ikke kunne finde frem til moster Ernas hus. Havde den fremmede derimod givet dig en (absolut) adresse: "Lillebyvej 4, 1234 Lilleby", ville det være fuldstændig ligemeget hvor du stod.

Typisk henviser et relativt link til en fil i en undermappe, f.eks. "billeder/mitbillede.jpg". Men flytter du så dokumentet med denne henvisning, til eksempelvis en anden mappe, leder dokumentet stadig efter "billeder/mitbillede.jpg" (men kan ikke finde det!). Et absolut link benytter derimod hele filens "adresse", f.eks. "<http://www.minside.dk/billeder/mitbillede.jpg>". Da hele adressen fremgår, er det derfor irrelevant hvor den henvisende fil ligger i forhold til den fil der henvises til. Du kan læse mere om principperne her: <http://www.webreference.com/html/tutorial2/3.html>

Der findes en tredje slags henvisning - "root relative" links - som er en mellemting mellem absolutte og relative henvisninger. Sådanne "rodrelative" henvisninger starter altid med en skråstreg "/". Denne skråstreg angiver, at der skal ledes i forhold til rodmappen, og ikke i forhold til den (under)mappe som den henvisende fil ligger i. Lad os sige at du har en side der hedder:

"<http://www.minside.dk/minmappe/side.shtml>". Indeholder denne side en henvisning til "/billeder/mitbillede.jpg", vil siden kigge i "<http://www.minside.dk/billeder/mitbillede.jpg>". Udelader man derimod denne først skråstreg ("billeder/mitbillede.jpg"), leder siden efter "<http://www.minside.dk/minmappe/billeder/mitbillede.jpg>". En lille forskel, med stor virkning.

Hvad angår includes, er det formentlig sikrest at bruge enten absolutte eller "root relative" links.

6. Alternativer til *includes*

Hvis du ikke vil (eller kan) bruge includes, kan du overveje andre muligheder.

Content management systems

For det først kan du benytte et CMS (content management system). CMS'er har den fordel at sidens opbygning i høj grad styres automatisk af systemet. CMS'er er særligt smart hvis man har store mængder data som skal kunne styres på en effektiv måde, evt. af flere personer. Det er derfor ikke overraskende, at mange avis- og nyhedssider, portaler og andre store sites, benytter sig af CMS'er. Omvendt er det ikke sikkert at du kan finde et CMS som passer til lige netop de krav du stiller til brugervenlighed, funktionalitet, design, navigation m.m.

Search & Replace

Der findes også programmer som automatisk kan erstatte tekststykker i mange filer (f.eks. websider) på én gang. Hvis du allerede har en hjemmeside, som ikke bruger includes, er dette måske en mulighed, når du har brug for at opdatere mange sider hurtigt. Du kunne endda måske bruge en sådan funktionalitet til at indføre includes på dine sider. Jeg vil ikke gå mere i dybden med sådanne programmer her, men du kan

prøve at kigge på nogle af disse gratis muligheder:

- HTML Enforcer: <http://www.soft.tahionic.com/html-enforcer.html>
- gbReplace: <http://www.garybeene.com/free/gbreplace>
- Replace & Rename: <http://www.replsoft.com/randr.html> (download her: <http://www.garybeene.com/files/>)

Frames/rammer

At gå i dybden med fordelene og ulemperne ved "frames" og "iframes" ville være uden for artiklens område. Man kan blot sige, at i den forstand at rammer kan bruges til at vise samme indhold flere steder på samme site, kan de bruges på samme måde som includes.

7. Konklusion

Hvis ikke du vil bruge et CMS, bør du i det mindste altid bruge en form for includes, også hvis du ikke regner med at siden bliver meget større (man ved jo aldrig). Selv foretrækker jeg SSI eller PHP includes, men alle de nævnte typer kan bruges. De har hver sine krav, fordele og ulemper. Fælles for dem er dog, at du sparer dig selv for en masse arbejde senere, ved at implementere includes nu.

Når du har fået oprettet din hovedside med includes, er det en gode idé at have en sideskabelon liggende (som indeholder include-kodestykket). Denne skabelon bruger du så hver gang du opretter en ny side.

// Old-Faithful□

Appendiks: Svar til kommentarerne

*Jeg modtager gerne kommentarer, så jeg løbende kan forbedre og opdatere artiklen (og gerne med en angivelse af hvorfor *ahem* en god/dårlig karakter gives). Herefter følger tilbagemeldinger på konkrete kommentarer:*

Tdafoobar:

- Jeg benytter den syntaks der anvendes i adskillige tutorials. Har du nogen nogen nærmere forklaring på hvad du ikke kan lide?
- Jeg påstår ikke noget sted at CMS har noget med includes at gøre. Jeg nævner derimod CMS som et alternativ til at bruge includes. Der er en stor forskel.
- Men nu er det ikke MVC artiklen handler om, men derimod includes. Jeg er af den opfattelse at MVC som regel vil være mere krævende at implementere, og dermed mindre egnet til dels mindre sider, dels mindre tekniske brugere.

netro:

Hentyder du til ASP.NET master pages eller bare konceptet generelt?

countermads:

Noget specifikt du mener er dårligt formuleret?

Kommentar af konder d. 05. Nov 2006 | 1

Kommentar af triple-x d. 04. Nov 2006 | 2

udmærket.

Kommentar af mysitesolution d. 08. Dec 2006 | 3

Savner også reelt brug af includes. tdafoobar: fx include er en language construct bygget omkring en funktion, så include(fil) er ligeså rigtigt som include fil

Kommentar af netro d. 06. Dec 2006 | 4

Du mangler efter min mening nogle konkrete eksempler på brugen af includes. Der er for meget snak om alt og ingenting. Den bør være mere slagkraftigt. Målgruppen skulle måske lige overvejes igen og lix-tallet sættes lidt ned rundt omkring. Og så kunne det vel være relevant at nævne masterpages under alternativer.

Svar: Jeg tænkte her på asp.net (2.0) masterpages. PS: Det er da noget bøvet at kommunikere her.

Kommentar af saudoo d. 06. Dec 2006 | 5

Ikke dårligt... ikke dårligt...

Kommentar af countermands d. 05. Nov 2006 | 6

dårlig formuleret

Kommentar af jokeren25 d. 07. Dec 2006 | 7

God informativ artikel.

Kommentar af hoeks d. 01. Nov 2006 | 8

Kommentar af cronick d. 04. Nov 2006 | 9

Fin beskrivelse af de forskellige måder at inkluderer en fil..

Kommentar af tdafoobar d. 22. Nov 2006 | 10

a) dine php-includes har forkert syntax, include/require er et language-construct ,ikke en funktion. b) CMS har ikke en skid med includes at gøre. c) Det er meget mere funktionelt at bruge en MVC arkitektur, hvor View og Models splittes totalt ad, istedet for at bruge includes.