



XHTML - Hvad 'De Skriftkloge' ikke fortæller dig - (Del 1)

Den første i en lille serie af artikler, hvor jeg prøver at gennemgå nogle af de emner indenfor brugen af XHTML, ikke så mange andre bruger tid på.

Du får bl.a. forklaringen på, hvorfor din XHTML parses som fejlfyldt HTML4.01 ;o)

Skrevet den **03. Feb 2009** af **olebole** | kategorien **Programmering / (D)HTML** | ★★★★★

Meningen med artiklerne

I den tid WWW har været en realitet, har forskellige buzz-fænomener hærget kodningen af sider til nettet. I disse år handler det hovedsageligt om XHTML, CSS og 'tabel-o-fobi'.

XHTML og DIV/CSS lovprises i utallige artikler og tutorials, mens brug af tabeller ukritisk forkastes, som værende langt værre end landsforrædderi og sodomisk omgang med døde egernbørn ... til sammen!

I dag er webkodning ikke, hvad det var for fem år siden (mere om det senere) - og de fleste tutorials/artikler glemmer at fortælle, hvor komplekst det er blevet. Tendensen med at citere andre tutorials, hvor forfatteren (heller) ikke har gjort sit forarbejde godt nok, begynder desværre at smitte af på nettets kodere, der i stigende grad vildføres med løsvrevne 'bibelcitater' fra W3C - men ikke får den nødvendige forståelse til at kunne bruge teknologierne.

Hvor ofte ser man f.eks. en XHTML-tutorial, der slutter med ordene:

"Gør, som jeg har skrevet - og du vil spilde din tid, da dine sider stadig parses som HTML4!" ...?

Desværre er det tilfældet for stort set al XHTML, jeg til dato har set på WWW ... selv XHTML-koden på W3C's egne sider, parses i alle browsere som HTML4.01 - og endda som fejlfyldt HTML4.01!

Da mange kodere har opnået en nærmest religiøs begejstring for XHTML, medfører det meget ofte ophidsede diskussioner i trådene i forskellige koder-fora - ikke mindst her på Eksperten - hvis man formaster sig til at gøre opmærksom på, at 'Kejseren ikke har tøj på' :)

Selvom det altså er at stikke hånden i en hvepserede, vil jeg i et antal mindre artikler prøve at råde bod på de værste misforståelser omkring XHTML.

Artiklerne skal ikke ses som en begynder tutorial i XHTML, men som et forsøg på 'religiøs debriefing' og uddybende forklaring for kodere, der allerede har stiftet bekendtskab med sproget :)

Artiklerne vil ikke på nogen måde være en kort og fyldestgørende indføring i XHTML. Den slags indgår ikke i min virkelighed. Jeg er af den faste overbevisning, at den eneste måde, man lærer teknologien ordentligt på, er ved at bruge mange, mange timer på W3C.

Artiklerne er netop et forsøg på at vise, hvor meget, der gemmer sig bag det, de almindelige tutorials fortæller - og hvor vigtigt, det er at gå i dybden med tingene. Meget ofte bliver helt essentielle ting nemlig aldrig fortalt og i værste fald kan læseren slet ikke bruge den viden, han tror at have fået, til noget somhelst.

Webkodning er blevet at sammenligne med et decideret fag, som det kræver en hel del at lære ... og den nært forstående omvæltning er nok større, end de fleste forestiller sig og vil kræve langt mere af koderen, end tilfældet er idag.

Først lidt historik

Da WWW blev skabt og introduceret i 1992, skete det fordi, der i forbindelse med oprettelsen af atomforsknings centret CERN var opstået et behov for global vidensdeling. Nettet var allerede virkelighed

og næsten alle universiteter verden over kommunikerede i forvejen via e-mail.

HTML var i den forbindelse tænkt som en enkel måde at layout'e sider med videnskabelige data. Overskrifter, tekstblokke og tabeller med data er, hvad HTML blev skabt til ... og det var lynhurtigt at lære for de forskere, det var tiltænkt.

De første HTML-versioner var derfor også temmelig loose. De ting, HTML skulle bruges til, var simple opstillinger, der ikke krævede særlig meget af browseren ... der var ingen grund til at optimere ved at lukke tags og koden var som regel simpel og overskuelig.

Da muligheden for brug af billeder blev introduceret, fandt grafisk kreative kodere hurtigt ud af at placere grafik-elementer i sindrige tabelhelveder spændt ud af et hav af transparente, 1*1 pixel gif'er - der blev strukket med height- og width-attributter.

Alle mulige sites skød op med alle tænkelige former for indhold - og grafikken fik en mere og mere fremtrædende plads på websider. WWW begyndte så småt at ligne det, vi kender idag.

Lynhurtigt blev simple og overskuelige HTML sider til gigantiske, komplicerede og skrøbelige 'sætterkasser', som krævede oceaner af kraft at parse - og var totalt uoverskuelige for både koderen og browseren.

I takt med, at brugen af WWW gik mere og mere mod underholdning, blev det nødvendigt at finde på en anden måde at strukturere koden på.

Samtidig blev den store browserkrig mellem Netscape og Microsoft udkæmpet på alle våben ... herunder inkompatibilitet. MS fandt nemlig ud af, der var store markedsandele i at fremstille en browser, der var så tilgængelig overfor dårlig kode som muligt - samtidig med, de indbyggede en masse special-stuff, som kun IE kunne bruge. Den strategi var en væsentlig bestanddel af MS' sejr i browserkrigen.

Bagdelen har været, at det tager uforholdsmæssig megen tid og koster en sydfransk bondegård at udvikle den slags browsere - og at disse bliver ekstremt store og klodsede.

Allerede idag - og endnu mere i den nærmeste fremtid - bliver browserklienter brugt i alle mulige andre sammenhænge end PC'ere. De skal kunne indgå i armbåndsure, sodavands automater, mobiltelefoner, vindjakkefoer, windsurfer master, osv, osv.

Som forholdene er idag, bliver mange af disse webklienter skræddersyet til formålet - men det ville være en kæmpe fordel at få strømlinet alm. webbrowsere og de sprog og teknologier, de benytter sig af, så disse kunne genbruges til alle mulige andre formål uden for mange konflikter. Så ville vi frit kunne lade vores PC'er, armbånds-sur, sodavands automat, mobiltelefon, vindjakke og windsurfer mast tale sammen i et sprog, de alle forstod - og alle kunne de sende, modtage og vise data-repræsentationer i forhold til deres anvendelser.

Derfor foreslog W3C (som bl.a. består af de væsentlige browser leverandører) at opdele WWW kode i følgende bestanddele:

1. data
2. markup (elementer til adskillelse af data med henblik på visning)
3. visuel strukturering (styling af markup)
4. funktion (scripting på data og elementer/styling)

Den slags kan ikke gøres i ét hug, når først teknikken er i brug, så det blev det gjort i etapper og udviklingen er i stadig fremdrift:

Netscape introducerede allerede i NS2.0 JavaScript (eller LiveScript, som det oprindeligt hed), der blev standarden for funktionalitet.

I IE3.0 implementeredes JScript, som er MS' version af ECMA-standard, på hvilken nu både JavaScript og JScript bygger.

Microsoft introducerede i IE3.0 CSS, som dannede standarden for den visuelle strukturering af elementerne.

NS implementerede desværre først CSS sammen med Gecko-motoren i version 6.0. Opbygningen af CSS kan på mange måder analogiseres med andre programmerings sprogs objekt-orientering. Nedarvning gennem klasser er 'hjertet' i CSS.

Derudover introducerede MS i IE5.0 standardiserede ECMA-DOM Bindings i overensstemmelse med W3C. Dem indførte NS i NS6.0. ECMA-DOM Bindings er bindinger mellem script og Dokument Objekt Modellen (f.eks. 'document.getElementById').

XML blev valgt som data-sprog - og selve markup-sproget (HTML) blev forsigtig tilnærmet XML i HTML4 - og ikke mindst i HTML4.01-Strict.

Fordelen ved XML's lukkede tags er, at disse gør det langt lettere/hurtigere for browserens parser at finde de enkelte elementers begyndelse og slutning. Enhver, der har arbejdet med mønster genkendelse i RegExp, vil vide, hvor meget enklere det er at finde noget, hvis man ved præcist, hvad der afgrænser det. Jo mere stramt, man definerer standarden, jo mindre kan parseren laves og jo mindre kraft kræver klienten.

Desuden kan man lægge elementer under forskellige namespaces og derved definere, til hvad og hvordan de skal bruges. Det åbner for ekstrem stor fleksibilitet og der kan konstrueres specialprog, der kan bruges sammen - og i samme dokument - uden at konflikte.

Ønsket var som sagt at komme væk fra gigantiske browsere, der retter op på alverdens HTML-fejl - henmod strømlinede klienter, der kan bruges i mindre og mindre enheder. For ikke udviklingen skulle gå helt i stå, måtte ansvarerne på nettet nu fordeles mere rimeligt. Ansvar for at klienterne overholdt gældende standarder skulle påhvile leverandørerne - og ansvar for at koderne overholdt standarderne skulle påhvile de, der skriver koderne.

Hvis man gjorde markup sproget kompatibelt med - eller ligefrem en afart af - XML, kunne de enkelte browser (og tredieparts) leverandører ovenikøbet levere speciel funktionallitet i tags under et andet namespace. Dermed kunne man let og enkelt filtrere markup'en til forskellige klienter i samme dokument.

Målet var altså et stramt markup-sprog, der levede op til XML ... 'Extended HyperText Markup Language' eller XHTML blev det døbt. I sin endelige form er det ren XML og kræver således perfekt kode for at blive parsed/vist.

Nettet drives ikke længere af universitets forskere og potheads i californiske garager ... det drives på hyperliberal vis af markedskræfter og erhvervsliv. For dem har det aldrig været mere presserende at få implementeret standarderne fuldt og helt, så formodentlig går der ikke mere end et par browsergenerationer, før standarden på nettet hedder XHTML2.0 - uden bagudkompatibilitet for HTML. Sikkerhedsnettene i form af bl.a. HTML-kompatibilitet er udelukkende noget, der hører til i overgangs versionerne 1.0 og 1.1.

Ligesom der rundt omkring stadig lever Amiga-geeks i støvede skabe, vil der sikkert i en tid fremover være et parallelt HTML-net ... men der, hvor det virkelig sner, kommer den helt afgjort til at stå på XHTML. Fordele og muligheder ved XHTML/XML er så store, at der ikke er nogen - bare lettere avanceret - koder, der vil gide beskæftige sig med HTML som andet end et nostalgisk, historisk fænomen. Det må således forventes, at der indenfor de næste par år kommer til at ske den måske største omvæltning på WWW, vi har været vidne til siden 1992.

Umiddelbart er der således al mulig god grund til at skrive XHTML ... men stop en halv og læs lidt om overgangs problemerne <a href="<http://www.eksperten.dk/artikler/538>">[HER] ;o)

super artikel, 13 herfra!

Kommentar af jensgram d. 07. Dec 2004 | 2

Ind med det!

Kommentar af tentakkel d. 16. Apr 2006 | 3

En RIGTIG artikel :)

Kommentar af roenving d. 10. Dec 2004 | 4

-- og også jeg må tilslutte mig rosen: Godt at have det samlet sammen !-)

Kommentar af skovenborg d. 08. Dec 2004 | 5

Flot skrevet, og meget spændende læsning. Specielt historien bag XHTML er ret så fint skrevet og interessant at vide. Også en god idé at splitte artiklerne op så man får bedre overblik.

Kommentar af basementjack d. 08. Dec 2004 | 6

Din er artikel er længeventet.. ;)

Kommentar af popsy d. 07. Dec 2004 | 7

Hmm... Touche! Godt og velskrevet, rigtigt interessant.

Kommentar af phatlasse (nedlagt brugerprofil) d. 07. Dec 2004 | 8

Super lækker læsning.... :-)

Kommentar af 3xm d. 07. Dec 2004 | 9

herligt at se en lang, dum diskussion i en spørgsmålstråd blive kogt ned til de hårde facts.

Kommentar af mm12010 d. 12. Dec 2004 | 10

Kommentar af xyborx d. 09. Dec 2004 | 11

Jeg er sådan set enig med de andre :) Godt gået

Kommentar af radiohead d. 11. Jun 2006 | 12

Thanks

Kommentar af webcreator d. 08. Jul 2005 | 13

Mange tak for en god artikel, der giver et glimrende indblik i hvad XHTML er og hvad vi kan forvente os af det. Jeg kan varmt anbefale at man læser hans næste artikel, som der linkes til sidst i dette dokument.

Kommentar af googolplex d. 09. Dec 2004 | 14

Kommentar af stich d. 18. May 2006 | 15

Fejlfyldt, løgnagtig og generelt ubrugelig - præcis ligesom mange af oleboles indlæg herinde.

Kommentar af mikkellbm d. 08. Dec 2004 | 16

Thumbs up!

Kommentar af htmlkongen d. 08. Dec 2004 | 17

Endnu en! :) /Htmlkongen

Kommentar af madsass d. 03. Jan 2005 | 18

Godt læsestof. Men lærer ikke rigtigt noget af det, andet end at du fortæller os utallige gange, at de fleste CSS/DIV sider parses som HTML 4.01.

Kommentar af freesoft d. 15. May 2005 | 19

God, meget god artikel.

Kommentar af bjoarn d. 25. Sep 2005 | 20

En god artikel, men jeg har lige en protest:

xhtml bliver ikke parset som html4, faktisk parser browserne ikke efter standarderne, men efter deres egne interne regler. Og de fleste browsere har 2-3 forskellige modes, som oftest aktiveres ved at bruge doctype. Her bliver html4 strict skåret over en karm med alle xhtml-doctypes. Hvis den skulle parse efter html4 skulle den jo ignorere tags der var en del af xhtml, hvilket den ikke gør. Ligesom at doctype html3.2 sagtens kan bruges sammen med tags fra html4.

Der hvor der tilgængæld er en fejl er i mime-typen som sendes i http'en. Den skal efter standarden være application/xhtml+xml i xhtml, hvilket de færreste servere er sat op til og som ihvertfald IE slet ikke kan håndtere.

Desuden en lille tilføjelse til "disse år handler det hovedsageligt om [...] 'tabel-o-fobi'"

Div'itis er også blevet et populært diskussion-emne. Refereret til sider som er bygget op af tusind divs med ids og classes

Kommentar af wicez (nedlagt brugerprofil) d. 21. Mar 2005 | 21

Genial artikel, hvis bare alle kunne skrive artikler af denne kaliber

Kommentar af el_barto (nedlagt brugerprofil) d. 06. Jan 2005 | 22

Som altid kloge ord fra ole (positivt ment!)

Kommentar af wanze d. 01. Feb 2005 | 23

Fantastisk! ;)

Kommentar af morteeart d. 07. Dec 2004 | 24

superb, endelig en god dansk forklaring på dette emne, som der tit (også i min egne spørgsmål) har diskuteret meget med forfatteren.

Kommentar af simonhans73 d. 30. May 2006 | 25

fint skrevet og spændende læsning

Kommentar af lazlowdk d. 11. Jan 2005 | 26

Kommentar af klubba d. 14. Feb 2005 | 27

Bullseye!

Kommentar af sommer89 d. 07. Jul 2006 | 28

Rigtig lækker artikel :-)

Kommentar af slst d. 09. Apr 2006 | 29

Kommentar af bootup d. 21. Dec 2006 | 30

Kommentar af bredbaandmobilt d. 10. Jan 2013 | 31

Hvor er kilderne ?

Man kan da ikke tage den slags seriøst uden kildehenvisninger.

Kommentar af olebole d. 13. Jan 2013 | 32

Hvis der er kildehenvisninger, du mener, jeg har glemt, er du mere end velkommen til at gøre mig opmærksom på, hvilke. Hvis jeg finder dem relevante, kommer de naturligvis øjeblikkeligt med - selvom der, som du sikkert er bevidst om, er tale om en dybt forældet guide, da ingen jo længere bruger XHTML.

Hvilke kildehenvisninger er det specifikt, du tænker på? Hvad mangler du uddybning af?