

Diskreta DOM-skript

Lars Gunther
Högskolan Väst
461 86 Trollhättan
gunther@keryx.se

SAMMANFATTNING

Only the adoption of unobtrusive DOM scripting will do away with outdated, uninformed, and inaccessible development methods. Only the adoption of unobtrusive DOM scripting will allow JavaScript, and thus web development in general, to attain its full potential.

www.webstandards.org/action/dstf/manifesto/

Sedan ett par år tillbaka så har ett nytt synsätt på javascript slagit igenom på bred front. Sedan webbutvecklare världen över fått upp ögonen för separation av innehåll från design har tiden kommit att också separera innehåll från beteende.

Detta nya synsätt går under många namn: DOM-skriptⁱ är kanske det vanligaste. Bland de som befinner sig i rörelsens mitt talas det dock oftast om "unobtrusive JavaScript", ungefärligen översatt till "diskreta javascript" på svenska.

Med denna nya praxis används javascript endast för att förhöja användbarheten på en sida, och aldrig så att en sida får försämrade användbarhet.

Nyckelord

- Javascript
- DOM
- Standarder
- Webbutveckling
- Användbarhet
- Tillgänglighet.
- Diskret, icke-inkräktande
- DHTML
- Ajax

BAKGRUND

Webbstandarder blev en bred rörelse ungefär år 2003. Epokgörande var bl.a. boken *Designing With Web Standards* av Jeffrey Zeldmanⁱⁱ Under åren 2004 och 2005 kan webbstandardrörelsen ha sägas vara på väg att erövra "front end design" utvecklarna. Idag ges idag i princip inte ut några böcker längre som inte anammat standarder (undantaget litteratur som vänder sig direkt till skolan och som

skrivits av mindre kunniga specialiserade läroboksförfattare).

Till en början handlade rörelsen mest om CSS, för att sedan med ett nyvaknat intresse ta sig an HTML igen med betoning på språkets semantik.ⁱⁱⁱ Validering ses inte längre som det enda nödvändiga, utan man eftersträvar korrekt användning av element och attribut utifrån deras språkligt logiska och strukturella funktion.^{iv} Element som tidigare missbrukats i grafiskt syfte, såsom `blockquote`, tabeller och rubrikelement, återerövrades för att fylla strukturell eller semantisk funktionalitet. Presentationen flyttas i största möjliga mån till CSS-filerna. HTML-element med endast fysiskt syfte undviks, speciellt de som nedvärderats av W3C.

Till en början talades det lite om javascript på sidor byggda enligt standarderna. Tekniken sågs som motsatt allt vad webbstandardrörelsen gick ut på och med nyvaknad kunskap om – och webbläsarstöd för – CSS, så gjordes försök att låta CSS sköta sådant som traditionellt varit javascripts domäner, såsom bild "roll overs" eller rullgardinsmenyer.^v

De problem man såg med javascript härhörde från DHTML-eran och gällde bland annat:

1. Låg grad av *standardisering* mellan Netscape och Internet Explorer, med ständig kodgaffling som följd. Standardernas (från Java återanvända) motto är "run once, run everywhere" och står i bjärt kontrast till detta.
2. DHTML bidrog till den röra av dåligt skriven HTML man just börjat lyfta sig ur. Röran kallas ofta tagg-gröt på svenska, "tag soup" på engelska.^{vi}
3. DHTML förknippades endast med *försämrade* användbarhet, såsom pop-ups, reklam, onödiga animationer, etc.^{vii}
4. DHTML förknippades med försämrade eller fullständigt obefintlig *tillgänglighet*. WCAG 1.0 varnar för skript som ett stort problem.^{viii}

TES

Det finns idag en utbredd acceptans för javascript inom webbstandardrörelsen och ett sätt att använda det på ett sätt som endast ökar en sidas användbarhet och tillgänglighet. Detta sätt att använda javascript kallas "icke inkräktan-

de javaskript” eller ”diskreta javaskript” efter den engelska termen ”unobtrusive”.

Det pågår en slags kamp mellan utvecklare och ”evangelister” för standarder och mindre insatta eller mindre nogräknade utvecklare. Det finns en stor fruktan bland de första att den hausse som just nu finns runt begreppen ”webb 2.0” och ”AJAX” leder till att många misstag från DHTML-eran återupprepas.^{ix}

METOD OCH MATERIAL

Denna uppsats bygger till största delen på min egen medverkan i webbstandardrörelsen under de senaste fem åren (2001-2006) och det material jag använt är:

- Böcker. Se litteraturlista.
- Föredrag/podcasts. Se lista.
- Artiklar, främst från bloggar, med tillhörande kommentarer.
- Forum och mejl. Adresser redovisas på slutet.
- Speciell tonvikt har lagts vid ”de tre stora guruerna”^x och deras ord, samt utvecklare förknippade med *the Web Standards Project Dom Scripting Task Force*^{xi}. Artiklar och böcker som någon av dessa har skrivits föregås i referenserna av ”DSTF”.

De ”tre stora guruerna” är

1. Brendan Eich^{xii}, upphovsmannen till javaskript.
2. David Flanagan^{xiii}, författare till ”noshörningsboken” (JavaScript: The Definitive Guide).
3. Douglas Crockford^{xiv}, skribent om stilguider, konventioner, best-practice och upphovsman till bl.a. *JSON* och *jslint*.^{xv}

Det jag dokumenterat i denna uppsats är *framväxten* och delvis *användarnyttan* av diskreta javaskript. Somliga referenser går därför till artiklar som har några år på nacken och som kanske inte längre beskriver den allra bästa tänkbara javaskriptkoden. Ett speciellt resonemang förs om AJAX

BEGREPP

Unobtrusive: tillbakadragen, inte påträngande (påflugan), försynt, diskret (*Stora Engelsk-svenska ordboken*)

Begreppet *unobtrusive scripting* myntades av Stuart Langridge i en artikel redan i november 2002.^{xvi} Termen är närbesläktad med begreppet *progressive enhancement*, vilket användes första gången om webbutveckling av en annan frontfigur inom webbstandardrörelsen, Steve Champeon, för första gången i mars 2003.^{xvii} Ett tredje begrepp som

ofta används är *graceful degradation*^{xviii}, ett annat ord för feltolerans.

Det finns idag inga begrepp på svenska som utgör vedertagna motsvarigheter till dessa tre begrepp. De svenska översättningar jag här lanseras måste ses som preliminära. (Jag har mejlat svenska datatermgruppen^{xix} om dem, men utan att ha fått några förslag.)

Feltolerans

När Netscape 2.0 lanserades så infördes tre sätt att hantera webbläsare som saknade nödvändigt stöd för javaskript:^{xx}

1. Man började lägga skripten inom HTML-kommentarer för att de skulle ignoreras av webbläsare som inte förstod skript-taggen (Mosaic och Netscape <= 1.2). Ironiskt nog används tekniken helt i onödan än idag!
2. Man införde attributet *language* på starttaggen, för att kunna specificera vilken version av javaskript man använde. Tanken var att webbläsare skulle ignorera nyare versioner som de inte förstod. Detta har aldrig fungerat och attributet är borttaget ur HTML-standarderna. Detta till trots används det ofta än idag också.
3. Man införde elementet *noscript* vars syfte var att förmedla samma innehåll som det skriptade, fast på ett kanske mindre tilltalande sätt. Men dess vanligaste användningsområde kom att bli att ge meddelanden som ”du måste ha javaskript för att kunna se innehållet”, vilket är ett missbruk av elementet på samma sätt som man missbrukade *noframes* till att ge oanvändbar information till den som hade en webbläsare utan stöd för tekniken.

Inom webbstandardrörelsen föreslås helt andra tillvägagångssätt, som räddar allt det som var gott i tanken bakom teknikerna beskrivna ovan, men som inte *kan* missbrukas. Dessa åtgärder har ju med all tydlighet visat sig otillräckliga, men själva tanken ses fortfarande som tilltalande.

Tilltagande förbättring

Principen med tilltagande förbättring är att inget *innehåll* förmedlas enbart med javaskript, utan att javaskripten utgör endast förbättringar av en sidas tillgänglighet och användbarhet. Basen för innehållet som sådant är endast HTML-koden och den som tar del av en hemsida genom en användaragent utan stöd för javaskript, eller vars stöd är otillräckligt, han eller hon missar ingen information. Det som tappas är förlagt till ett eget gränssnittslager: *beteendet*.^{xxi}

DOM-skriptning

Ur strikt standardsynpunkt markerar denna term att man övergår från proprietär kod till att använda det API som

W3C definierat för interaktion med dokumentet. DOM-skriptning är det begrepp man enats om att använda inom en stor grupp standardmedvetna utvecklare.^{xxii}

I praktiken är det omöjligt att använda endast av W3C och ECMA definierad syntax, eftersom (1) stödet i framför allt Internet Explorer är för dåligt och (2) somliga saker man som utvecklare vill göra ligger ännu helt utanför standarderna. Arbete pågår med att formellt standardisera de viktigaste de-facto standarderna, såsom *innerHTML*,^{xxiii} *XmlHttpRequest* och *Window*-objektet.^{xxiv}

Begreppet DOM-skriptning rymmer dock mer än bara en övergång till DOM-metoder. Det är en filosofi som ser sig själv såsom varandes motsats till DHTML-erans dåliga JavaScript.

Diskreta javascript

Av alla termer för att beskriva det nya tänkesättet är det kanske denna som blivit mest utbredd. David Flanagan anger tre mål för diskreta javascript:^{xxv}

1. All javascriptkod, inklusive händelsehanterarna, skall skiljas från HTML-koden och endast återfinnas i separata filer.^{xxvi}
2. Skripten skall vara feltoleranta. Detta uppnås genom *funktionsstestning* ("capability testing") också kallat *objekt detektering* ("object detection" ibland "feature testing"). Detta står i motsats till browsersniffning.^{xxvii} Att koda mot en specifik webbläsare, annat än för att hantera mycket specifika buggar, anses vara ett stort misstag.
3. Skripten får inte förstöra en hemsidas tillgänglighet. Då har det *inkräktat* ("obtrude") på användare med funktionsnedsättningar.

En fjärde punkt – som Flanagan behandlar i sin bok – skulle kunna vara att skript inte skall inkräkta på varandra. Genom att alla funktionalitet kapslas in i objekt, så undviks exempelvis konflikter i namngivningen av globala variabler.^{xxviii}

Nästkommende version av JavaScript kommer att stödja *privata klassvariabler*, *bibliotek*, *paket* och *"block scope variables"*^{xxix} men till dess får liknande funktionalitet återskapas inom ramen för befintlig teknik. Douglas Crockford upptäckte en metod att med hjälp av *closures* skapa privata medlemsvariabler.^{xxx} Dessa mer avancerade tekniker är dock sällan det som nämns som exempel på diskreta skript.

Alla grundtankar inom den diskreta DOM-skriptningens skola för javascript finns sammanfattade i "The JavaScript Manifesto"^{xxxi} Där slås det också fast:

Unobtrusive scripting is not a coding style; it's a way of thinking.

AJAX OCH "RICH USER INTERFACES"

Paul-Peter Koch, allmänt känd som *ppk*, beskriver användningen av javascript som en cirkelgång mellan "tunna" och "tjocka" faser.^{xxxii}

1. Den första "tunna" fasen var under perioden mars 1996 (Netscape 2.0 släpps) och varade så länge som skript främst inriktade sig på mindre effekter i generation 2 och 3 webbläsare.
2. Den första "tjocka" fasen är detsamma som DHTML-eran. Ivrigt pådrivna av både Netscape och Microsoft kodas det friskt mängder av proprietär eller browsersniffande skräpkod för deras generation 4 webbläsare. Man kan fortfarande hitta rester av "skript" från den här tiden, lätt igenkända på deras groteska utförande och totala begreppsförvirring.
3. Den andra "tunna" fasen innebar i princip att det mesta av javascript togs bort från sidorna. Till största delen hade tekniken använts för att störa användare (så till den milda grad att idag har alla webbläsare inbyggd pop-up blockering och förbjuder skript att dölja deras "krom" eller sätta värde på statusraden) och förstöra sidors tillgänglighet och användbarhet. I och med att Netscape 4 försvunnit från marknaden och det blivit ett genombrott för CSS-baserad layout utifrån standarder, så skapades förutsättningarna för ett ansvarsfullt sätt att hantera tekniken.
4. Den andra "tjocka" fasen är vi mitt uppe i, samtidigt som den tunna. Den utgörs av den rådande vågen av "Ajax". PPK fördömer onödigt komplicerade *frameworks* såväl som nedsatt användbarhet och tillgänglighet.
5. PPK förutspår att denna tjocka fas kommer att dö ut, eftersom den skapat onödigt tungrodda sajter, som gör dess utvecklare långsamma att reagera, eftersom den skapar sajter med *lägre* användbarhet, som i sin tur är sämre affärer, och eftersom den ger stora problem med tillgängligheten.

Vad är AJAX

Begreppet myntades Jesse James Garret i februari 2005^{xxxiii}. Innan dess kallades tekniken ofta för "remote scripting".^{xxxiv} Kärnan i tekniken är att en *del* av en sida uppdateras med ny information efter kommunikation med servern, utan att sidan som helhet behövs uppdateras. "The use of scripting to cause portions of a page to refresh without reloading the entire page." (Joe Clark, 2006-04-28)

Tidigare användes oftast en *iframe* för att åstadkomma detta, men på senare tid har funktionen *XmlHttpRequest* (XHR)^{xxxv} blivit den man främst tänker på. (Det mest omta-

lade exemplet på Ajax, Google Maps, använder sig dock ironiskt nog av iframe-tekniken!)

Själva akronymen AJAX har ofta ifrågasatts eller hånats. Den verkar dock vara här för att stanna. Som akronym har den följande problem:

- A står för asynkron, men man kan använda XMLHttpRequest i synkront läge också. Det är dock inte alls att rekommendera då webbläsaren läses upp. Dessutom kan ordet asynkron sägas syfta på *interaktionen* mellan webbplats och användare, snarare än på XMLHttpRequest som objekt.
- J står för javaskript. Inga problem. Ajax inkluderar således inte alternativa sätt att skapa liknande effekter, såsom Flash eller Java.
- Nästa A står kort och gott för "and", dvs. "och". Inga problem.
- X står för "XML". Det finns dock (minst) tre sätt för att överföra data mellan server och klient:
 1. Ren text.
 2. XML
 3. JSON

Av dessa är ren text enklast och fördras av många. Den har också blivit ett *mikroformat*: AHAH^{xxxvi}. XML som dataformat lider av att nuvarande DOM-metoder i javaskriptmotorerna resulterar i långsammare exekvering än exempelvis JSON.^{xxxvii} Utvecklare som kommer från back-end miljö tycks fördras XML, som är invant för dem och för vilken deras primära språk (PHP, ASP, JSP, etc) ofta har utmärkt stöd.^{xxxviii} Utvecklare från "front-end" miljö verkar föredra JSON.

Problemen med Ajax

Inom webbstandardrörelsen diskuteras det flitigt hur man skall komma till rätta med de tillgänglighets och användbarhetsproblem som skapas med Ajaxmetodiken.

Där finns problem utifrån ett flertal aspekter:

1. Javaskript i sig kan vara problematiskt. Om en sidas information inte delges på ett linjärt sätt, utan ett område man redan passerat (tekniskt eller mentalt) uppdateras, hur kan man då bli varse det. "Yellow fade"^{xxxix} hjälper något, men metoden hjälper inte den som tar del av sidan via talsyntes.
2. Utförandet av javaskripten kan vara problematiska. De kan antingen göras med insikt om teknikens problematiska sidor eller så kan de förvärras. Christian Gross föreslår exempelvis ett utvecklingsmönster ("design pattern") för AJAX som han kallar "Content Chunking Pattern", vilket återin-

roducerar precis alla användbarhets- och tillgänglighetsproblem som någonsin funnits med ramar, fast i värre form och ofta så att skadan omöjligt, eller bara med yttersta ansträngning, kan lindras. Detta trots att hans bok har orden "Best Practices" i titeln!^{xl}

Exempel på problem med AJAX:

- a. "Bookmarkability" – om en sida inte motsvarar en URL, hur skall man då kunna hitta den via en sökmotor, spara den som favorit eller kunna mejla länken till en vän?
- b. Tillbakaknappen! Jakob Nielsen (m.fl.) har en mängd saker att säga om den!^{xli} Dåliga javaskript förstör dess funktionalitet.
- c. "Custom Widgets", liksom ej förväntat beteckande på länkar och andra delar av sidan. En sida bör alltid uppträda som förväntat. Detta bryter oerfarna javaskriptsutvecklare ofta mot.
- d. "Code bloat". Många använder exempelvis i onödan stora "toolkits" bara för att göra enkla effekter. Det som kunde ha gjorts med 50 egenskrivna rader kod blir i stället 20 egenskrivna plus ett par tusen inlånade, med följder som onödigt högt uttag av bandbredd, längre nedladdningstider och lägre fart i applikationen som sådan.

3. CMS och server-side relaterade problem. Många utvecklare är beroende av – eller använder aningslöst – de javaskript som automatiskt genereras av deras utvecklingsplattform eller CMS. *doPostBack* i asp.NET är måhända mycket trevligt för utvecklaren, men ur tillgänglighets och användbarhetsperspektiv är tekniken usel.

Synsättet bland standardmedvetna utvecklare är: Dessa, och många andra problem, är den kanske stora utmaningen just nu. AJAX är "hett", det är "inne", det är "cool" – och investerare kastar pengar på allt som kallas "web 2.0". I denna trends kölvatten ser vi många misstag begås igen och igen.^{xlii}

It must degrade well. It must still be accessible. It must be usable. If not, it is a cool useless piece of rubbish for some or many people.^{xliii}

Hijax eller Client/SOA

Ett sätt att använda AJAX, utan att tappa någon användbarhet eller tillgänglighet, går under namnet Hijax.^{xliv} Begreppet myntades av Jeremy Keith. Och håller på att vinna viss spridning.^{xlv}

I princip är det en tillämpning av AJAX utifrån principen att man tänker sig en sida med AJAX från början, men först utvecklar man en som fungerar AJAX förutan. Sedan lägger man på *användbarhetslagret* med DOM-skript enligt tidigare angivna principer. Metoden kräver att man kan skicka två slags svar från servern: Hela sidor till den som inte kan ta del av javascript och delar av sidor för de flesta som kan göra det.

Motsatt synsätt representeras av den sortens applikation som inte längre kan ses som en "sida", den sortens rika användargränssnitt som mer påminner om regelrätta program.^{xlvi} Webbaseade ordbehandlare och kalkylprogram påvisar enkelt likheten med traditionella applikationer, men det finns fler. Ett begrepp som vunnit visst genomslag för denna sista slags applikation är "Client/SOA".^{xlvii} Hur de skall kunna göras tillgängliga är en fråga som återstår att lösa.

REFLEXION

Javascript har fått ett dåligt rykte. Inom svensk akademisk utbildning finns det inga kurser på högre nivå än A (mig veterligen) som över huvud taget behandlar front-end design inom webbutvecklingen. Att man skulle kunna bygga avancerade javascript med för användaren nyttig funktionalitet, förbättrad tillgänglighet och utnyttjandes objektorienterade mönster, den tanken tycks inte föresväva utbildningsanordnare i Sverige.

Problemet är dock inte endast svenskt. I The JavaScript Manifesto kan man läsa följande:

The development of such scripts can be shoved off on junior programmers, which allows the seniors to continue with their serious work, which of course does not include immature and simplistic technologies like, for instance, JavaScript. Besides, this process serves to consolidate the natural order of society in serious programming, *where*

front enders know their place at the bottom of the social pyramid.^{xlviii} (Betoning tillagd)

Inom svensk webbutveckling sker fortfarande det mesta som rör javascript av utvecklare som inte verkar tro sig behöva vidareutbildning. Jag har under åren 2005 och 2006 granskat de flesta webbsystem som används inom Trollhättans stad, liksom av samtliga politiska partier, lokalt och på riksplånet. Inte ens de som tagits i bruk under tiden granskningen pågick kan sägas ha kod av god kvalitet. Andra webbläsare än Internet Explorer version 6 (och ibland Firefox) utesluts helt i onödan, endast på grund av inkompetensen hos webbapplikationernas utvecklare. I princip inga uppfyller ens grundläggande krav på tillgänglighet. Sidorna är nedludade med in-line javascript, som ofta ger intryck av ren "klipp-och-klistra" metodik.

Utifrån sådana förutsättningar är det inte underligt att webbutvecklare som pläderat för standarder och tillgänglighet länge såg skripten som en fiende. Idag har dock pendeln svängt och de allra flesta inom webbstandardrörelsen ser dem som potentiella tillgångar.

Om däremot det sker ett utbrett anammande av AJAX i Sverige – det har hittills gått trögt då vårt land är idag inte tycks vara i frontlinjen av webbutveckling längre – så är risken stor att gamla synder kombineras med nya synder och vad vi då får komma att se är mer och värre grötkod än någonsin.

Ett stort hopp står dock till det faktum att AJAX kräver så pass avancerade skript att den som använder sig av dåliga metoder också som utvecklare blir alltför akterseglad för att kunna konkurrera. Det finns också en tydlig affärsnytta med att använda standarder och följa den bästa metodiken. Men i stället för att bara hoppas på att darwinistiska lagar slår ut de sämre webbyråerna, så önskar åtminstone jag att de goda sätten att göra webbplatser blir vad som lärs ut inom utbildningsväsendet och krävs i lag för offentlig upphandling.

KOMMENTARER TILL UPPSATSEN ("META-INFORMATION")

I noterna tar jag mig friheten att avvika från mallens tvåspaltupplägg, för att inte tvingas radbryta [URL:er](#).

Jag skriver ut ISBN på böcker jag refererar till, trots att det bryter mot Harvardsystemet. Idag är ISBN det som man använder för att identifiera en bok!

Utvidgning

Min tanke är att påvisa att det finns material nog för att skriva uppsats på C-nivå om front-end design. En sådan uppsats skulle kunna vara en utvidgning av denna, med fördjupad frågeställning och konkreta exempel, i form av programkod och komplettering av materialet genom undersökning av befintliga webbplatser och intervjuer.

Bieffekter

Denna uppsats skrivs primärt som inlämning i en kurs jag deltar i på Högskolan Väst. Men den kommer också att skickas till vänner inom webbstandardrörelsen, kollegor inom utbildningsväsendet, samt kontakter inom offentlig förvaltning. Eventuellt kommer den också att göras till underlag för lobbying i riksdagen, eftersom Sverige saknar en fungerande lagstiftning om tillgänglighet för webbplatser. På den punkten är vi löjligt akterseglade jämfört med USA, Australien, Storbritannien, med flera. Somliga formuleringar i uppsatsen och somligt faktaurval kommer sig av dessa åsyftade bieffekter.

Lars Gunther
2006-11-12

REFERENSER

Böcker, huvudsakligen konsulterade

ORD, Stora engelsk-svenska ordboken (1980), Nacka: Esselte Studium.

Flanagan, David (2006) *JavaScript, The Definitive Guide, 5th edition*, O'Reilly: Sebastopol, USA (ISBN 0-596-10199-6)

Koch, Paul-Peter, alias PPK (2006) *ppk on JavaScript*, Peachpit/New Riders: Berkeley, CA, USA (ISBN 0-321-42330-5)

Heilman, Christian (2006) *Beginning JavaScript with DOM Scripting and AJAX*, Apress: Berkeley, CA, USA (ISBN 1-59059-680-3)

Gross, Christian (2006) *AJAX Patterns and Best Practices*, Apress: Berkeley, CA, USA (ISBN 1-59059-616-1)

Edwards, James och Adams, Cameron, *The JavaScript Anthology : 101 Essential Tips, Tricks & Hacks*, SitePoint: Collingwood, Australien (0-9752402-6-9)

Keith, Jeremy (2005) *DOM Scripting (Web Design with JavaScript and the Document Object Model)*, Apress/Friends of Ed: Berkeley, CA, USA (ISBN: 1-59059-533-5)

Langridge, Stuart (2005) *DHTML Utopia: Modern Web Design Using JavaScript & DOM*, SitePoint: Collingwood, Australien (ISBN: 0-9579218-9-6)

En del av den övriga litteratur jag last om webbutveckling kan man se på <http://keryx.se/resurser/Webb.php>

Webbplatser, där jag läst många artiklar eller som innehåller referensmaterial eller som beskriver mitt eget deltagande i webbstandardrörelsen:

- *Netscape Devedge* (webbplatsen är återskapad på Mozillas webbplats http://devedge-temp.mozilla.org/index_en.html [2006-11-10 - informationen sägs vara på den adressen bara provisoriskt])
- Douglas Crockfords webbplats: <http://www.crockford.com>
- Mozilla Developers center: <http://developer.mozilla.org/>
Sidor att speciellt titta på inkluderar:
 - *About Javascript*: [http://developer.mozilla.org/en/docs/About JavaScript](http://developer.mozilla.org/en/docs/About_JavaScript)
 - *JavaScript Language Resources* (länkar bl.a. till ECMA-standarderna) [http://developer.mozilla.org/en/docs/JavaScript Language Resources](http://developer.mozilla.org/en/docs/JavaScript_Language_Resources)
 - *DOM* <http://developer.mozilla.org/en/docs/DOM>
- Andra tillverkare har liknande sidor som Mozilla (fast inte lika bra, i mitt tycke):
 - *Apple Developer Connection*: <http://developer.apple.com/>
 - *Opera Developer Community* <http://dev.opera.com/> (Webbplatsen togs i bruk under hösten 2006.)
 - *Microsoft Developer Network*: <http://msdn.microsoft.com/>

-
- *The WebstandardsProject* <http://webstandards.org/>
Lägg märke till:
 - *AJAX, Accessibility & Screen Readers*
<http://www.webstandards.org/2006/05/04/ajax-accessibility-and-screenreaders/>
 - *Advocacy in Education* (intervju med mig)
<http://www.webstandards.org/action/edutf/interviews/gunther-sv/>
 - *Letter to Skolverket*
http://www.webstandards.org/action/edutf/resources/letter_to_skolverket/
Denna skrivelse, som jag är huvudförfattare till finns också på adressen:
<http://keryx.se/wasp/>
 - *WASP welcomes Sweden* (Om mig som ombud för WASP)
<http://www.webstandards.org/2005/09/10/wasp-welcomes-sweden/>
 - *Mejlinglista för diskussion om webbstandarder i utbildning:*
<http://www.webstandards.org/action/edutf/discussion/>
 - *Wikipedia* (Några få av de artiklar jag har läst listas. Jag har också bidragit till några av dem.)
 - <http://en.wikipedia.org/wiki/JavaScript>
 - http://en.wikipedia.org/wiki/Document_Object_Model
 - http://en.wikipedia.org/wiki/Web_standards (bidragit själv)
 - http://en.wikipedia.org/wiki/Comparison_of_layout_engines_%28DOM%29 (bidragit själv)
 - http://en.wikipedia.org/wiki/DOM_scripting (mindre bidrag från mig)
 - *SitePoint* <http://www.sitepoint.com> En ledande plats om webbutveckling, som erbjuder
 - Nyhetsbrev (Jag prenumererar på flera av dem)
 - Stort forum (Jag deltar av och till)
 - Artiklar
 - Bloggar
 - Webcasts
 - *World Wide Web Consortium* <http://w3.org>
Jag har läst hundratals sidor på W3, inklusive standarderna för HTML 4.01, XHTML 1.0, DOM 1, DOM 2 och (deklar av) DOM 3, WCAG 1.0, I18n-material, etc.
 - *ThinkVitamin* <http://www.thinkvitamin.com/>
 - *456 Berea Street* <http://www.456bereastreet.com/>
Denna sajt drivs av Roger Johansson från Göteborg. Den är mycket läst, återpublicerad och respekterad. Roger är medundertecknare av min skrivelse till Skolverket och en av dem jag lärt mig mest av.
 - *The Web Standards Group* <http://webstandardsgroup.org/>
Ursprungligen en australisk grupp, men med en *synnerligen aktiv mejlinglista* där deltagarna kommer från hela världen. 2006-11-11 var det 4016 medlemmar från 99 länder! Här finns också artiklar och ett mycket uppskattat nyhetsbrev, samt en podcast som jag prenumererar på.
 - *Quirksmode* (webbplats, med blogg, artiklar, tester och kompatibilitetstabeller driven av PPK).
<http://www.quirksmode.org/>
 - *The New Amateur* http://www.quirksmode.org/blog/archives/2005/11/the_new_amateur.html
 - *W3C DOM Introduction* <http://www.quirksmode.org/dom/intro.html>

- *Vägledningen 24-timmarswebben* <http://purl.org/net/vl24h> [2006-11-12] Detta är en permalänk som ständigt går till senaste versionen, oavsett om webbplatsen byter namn eller om dokumentet bara flyttats internt.

Jag läser regelbundet en mängd nyhetssajter och bloggar om vad som händer inom området webbutveckling och om PHP. OPML tillgänglig på <http://keryx.se/opml.xml>

Podcasts om webbutveckling och standarder:

- *The Web Standards Group*: <http://webstandardsgroup.org/audio/>
- *./ with Imagination* <http://www.dustindiaz.com/> (också blogg och webbcasts) **Starkt fokus på javaskript.**
- *Boagworld* <http://www.boagworld.com/>
- *Practical Web design Magazine* <http://www.pwdmag.co.uk/> (Paul Boag gör deras podcast)
- *@media 2006* <http://www.vivabit.com/atmedia2006/blog/index.php/category/podcast/feed>
- *South By Southwest* <http://2006.sxsw.com/coverage/podcasts/> Detta är troligen världens största konferens om webbutveckling. The Web Standards Group arrangerade flera seminarier och hade sitt årsmöte där. Allt finns som podcast.
- *Web Essentials 2005* <http://odeo.com/channel/31434/view>
- *Web Standards Group London, Meetup* <http://muffinresearch.co.uk/wsg/>
- *WebAxe* <http://webaxe.blogspot.com> Podcast med fokus endast på tillgänglighet. Har bl.a. en intervju med Christian Heilmann.
- *ThinkVitamin* <http://feeds.feedburner.com/vitaminmasterfeed> Här finns också deras podcasts utskrivna att läsa. Föredömligt!
- *Webstock 2006* <http://www.webstock.org.nz/recordings.php> Konferens i Nya Zeeland, maj 2006. (Humor: Missa för allt i världen inte "The 8x5 session".)

Enskilda artiklar, utöver de som anges i noterna:

- Bobby Van der Sluis (2005-04-19) *Ten good practices for writing JavaScript in 2005* <http://www.bobbyvandersluis.com/articles/goodpractices.php> [2006-11-09]
- Scott Andrew LePera (2002-01-27) *Scripting For The 6.0 Browsers* (om DOM) http://www.scottandrew.com/weblog/articles/dom_1 [2006-11-09]
- Douglas Crockford (2001) *JavaScript The World's Most Misunderstood Programming Language* <http://www.crockford.com/javascript/javascript.html> [2006-11-09]
- Cameron Adams (2005-10-05) *JavaScript and the DOM* <http://www.themaninblue.com/writing/perspective/2005/10/05/presentation/index.htm> [2006-11-09]
- Jakob Nielsen (december 2005) *Why AJAX Sucks (most of the Time)* <http://www.usabilityviews.com/ajaxsucks.html> [2006-11-09]
- Joe Clark (2006-04-28) *Build Half a Product: Is Ajax accessible? At all?* <http://joelclark.org/access/research/ice//iceweb2006-notes.html> [2006-11-09]
- James Edwards (2006-05-25) *AJAX and Screenreaders: When Can it Work?* <http://www.sitepoint.com/article/ajax-screenreaders-work>
- Gez Lemon och Steve Faulkner (2006-05-25) *Making Ajax Work with Screen Readers* <http://juicystudio.com/article/making-ajax-work-with-screen-readers.php>
- Simon Willison (2003-12-10) *Enhancing Structural Markup with JavaScript* <http://www.sitepoint.com/article/structural-markup-javascript> [2006-11-09]

NOTER:

- i http://en.wikipedia.org/wiki/DOM_scripting [2006-11-02]
- ii Zeldman, Jeffrey (2003) *Designing With Web Standards*. Indianapolis, USA: Peachpit/New Riders Press. Boken kom 2003-05-14 (ISBN 0-7357-1201-8), en andra utgåva från 2006 är numera tillgänglig. <http://www.zeldman.com/dwss/>
- iii Ex: <http://www.ermanz.govt.nz/news-events/archives/presentations/semantichtml.html> [2006-11-02] <http://tantek.com/presentations/2004etech/realworldsemanticpres.html> [2006-11-02]
- iv DSTF: <http://simon.incutio.com/archive/2003/08/28/structuralVsSemantic> [2006-11-02] Sidan skrevs 28 augusti 2003 och är alltså från ”genombrottstiden”. Detta är en mycket tidig röst som hänvisar till flera pionjärer som talar om ”semantisk kodning” och själv går ytterligare ett steg längre och skiljer ut semantik från struktur.
- v <http://meyerweb.com/eric/css/edge/> [2006-11-02] De rena CSS-popup menyerna lanserades 2004. Ett annat känt ställe är <http://www.cssplay.co.uk/>
- vi http://en.wikipedia.org/wiki/Tag_Soup [2006-11-08]
- vii DSTF: Hur PPK i november 2003 försvarade att javaskript har en relevant plats i modern webbutveckling och kan användas för att öka en sidas användbarhet ger en bild av denna diskussion, såsom den fördes. http://www.quirksmode.org/oddsandends/future_javascript.html [2006-11-02]
- viii <http://www.w3.org/TR/WCAG10-HTML-TECHS/> och <http://www.w3.org/TR/WCAG10/#gl-structure-presentation>
- ix <http://2006.sxsw.com/interactive/programming/panels/?action=show&id=IAP060016>
- x Begreppet är mitt eget och inte allmänt känt. Andra bedömare kanske vill se en något annorlunda lista.
- xi DSTF; <http://www.webstandards.org/action/dstf/>
- xii Personlig blogg på adressen <http://weblogs.mozillazine.org/roadmap/> [2006-11-03]
- xiii Personlig blogg på adressen <http://www.davidflanagan.com/> [2006-11-03]
- xiv Personlig hemsida på adressen: <http://www.crockford.com/> [2006-11-03]
- xv <http://json.org> och <http://www.jshint.com/> (detta är ett verktyg som kontrollerar din kods kvalitet.)
- xvi DSTF: <http://www.kryogenix.org/code/browser/aqlists/> [2006-11-02]. Artikeln skrevs som sagt i november 2002.
- xvii http://www.hesketh.com/publications/progressive_enhancement_and_the_future_of_web_design.html På webbplatsen <http://www.hesketh.com> finns ett antal artiklar som skrevs under perioden 2001-2004 som hade ett revolutionerande innehåll på sin tid.
- xviii ”Graceful Degradation” ifrån ”Federal Standard 1037C/Glossary of Telecommunication terms”. http://www.its.bldrdoc.gov/fs-1037/dir-017/_2479.htm [2006-11-07]
- xix <http://www.nada.kth.se/dataterm/> [2006-11-01]
- xx Stöd för dessa punkter kan hämtas ur i princip vilken HTML- eller javaskriptbok bok från 90-talet som helst. Specifikt om deras nyare historik kan man läsa i *PPK (2006)* sid. 12-13. Douglas Crockford berör också ämnet ingående i ett föredrag från hösten 2006 som finns tillgängligt som screencast på adressen <http://www.webstandards.org/2006/10/18/video-presentation-douglas-crockford-on-the-theory-of-the-dom/> [2006-11-03]
- xxi http://en.wikipedia.org/wiki/Progressive_Enhancement [2006-11-02]
- xxii DSTF: http://www.quirksmode.org/blog/archives/2005/06/you_shouldve_be_1.html [2006-11-03]
- xxiii <http://whatwg.org/specs/web-apps/current-work/#innerhtml0>
- xxiv <http://www.w3.org/2006/webapi/> [2006-11-03] och <http://whatwg.org/specs/web-apps/current-work/#apis> [2006-11-03].

-
- xxv Flanagan, 2006, sid 241-242.
- xxvi DSTF: Artikeln DHTML Utopia (<http://www.sitepoint.com/article/dhtml-utopia-modern-web-design> [2006-11-03] från sommaren 2005 och boken med samma namn av Stuart Langridge är en annan plats där detta förklaras i detalj.
- xxvii Flanagan sid. 258-260. DSTF: http://www.quirksmode.org/blog/archives/2006/08/the_dangers_of.html Tanken och tekniken beskrivs ingående i princip i alla moderna böcker om javascript, förutom de som specialiserat sig på AJAX och därför inte med nödvändighet anpassats efter den *diskreta* tekniken.
- xxviii Flanagan, kapitel 10.
- xxix Brendan Eich (2006-05-19) *JavaScript 2 and the Future of the Web* (Presentation från XTech 2006) <http://developer.mozilla.org/presentations/xtech2006/javascript/>
http://weblogs.mozillazine.org/roadmap/archives/2006/05/javascript_2_ecmascript_editio.html
- xxx Crockford (2001) *Private Members in JavaScript* <http://www.crockford.com/javascript/private.html> [2006-11-10] Se också Flanagan, sid. 163-164.
- xxxi DSTF: <http://www.webstandards.org/action/dstf/manifesto/>
- xxxii DSTF: PPK (2006) kapitel 1C.
- xxxiii Begreppet myntades i denna artikel av Jesse James Garret 2005-02-18: *Ajax: A New Approach to Web Applications* <http://www.adaptivepath.com/publications/essays/archives/000385.php> [2006-11-09]
- xxxiv Förhistorien och explosionen beskrivs i artikeln *Ajax, Atlas, W3C Oh My!* <http://www.cherny.com/webdev/14/ajax-atlas-w3c-oh-my> [2006-11-09]
Ett längre föredrag finns att se på adressen <http://terrychay.com/blog/article/liquid-remote-scripting.shtml> [2006-11-11].
Föredragshållare är Terry Chay, en pionjär inom området ”remote scripting” med PHP. Från föredragets abstract: “Remote Scripting has been with us for five years now, but Google Gmail and Google Maps have brought it to the forefront.” Uttalandet gjordes i augusti 2005.
- xxxv <http://en.wikipedia.org/wiki/XMLHttpRequest> [2006-11-11]
<http://developer.apple.com/internet/webcontent/xmlhttpreq.html> [2006-11-11]
- xxxvi *Rest/AHAH* på Microformats WIKI <http://microformats.org/wiki/rest/ahah> [2006-11-12]
- xxxvii Sean Kelly *Speeding Up AJAX with JSON* (April? 2006) <http://www.developer.com/lang/jscript/article.php/3596836> [2006-11-12]
- xxxviii Christian Gross, utvecklare från ASP-miljö, går så långt att han menar att allt annat än XML för dataöverföringen är ”bad practice”, utom i undantagsfall (Gross, sid. 12).
- xxxix <http://www.37signals.com/svn/archives/000558.php> (Som en humoristisk kommentar kan man notera att världens mest kända användare av Ruby, tillika utvecklarna av Rails använder PHP för sin blogg!)
- xl Gross (2006) kapitel 3.
- xli Jakob Nielsen/Loa Horanger (2006) *Prioritizing Web Usability*, Berkeley CA USA: New Riders Press (ISBN 0-321-35031-6), sid. 63-67. Ämnet berörs också i andra böcker och på en mängd hemsidor om användbarhet.
- xlii Kollaborativt dokument: *Ajax Mistakes* <http://swik.net/Ajax/Ajax+Mistakes> [2006-11-09]
- xliii <http://vanderwal.net/random/entrysel.php?blog=1770> [2006-11-09]
- xliv DSTF: <http://domscripting.com/blog/display/41> [2006-11-09]
- xlv Ett exempel på detta: <http://www.cherny.com/webdev/37/ajax-and-accessibility-youre-doing-it-wrong-hijax> [2006-11-09]
- xlvi <http://www.sitepoint.com/blogs/2006/02/15/two-kinds-of-ajax-html-vs-client-soa/>
- xlvii DSTF: <http://domscripting.com/presentations/xtech2006/>
- xlviii DSTF: <http://www.webstandards.org/action/dstf/manifesto/>