Semantiline HTML

19 years tagasi Autor: AM

Autor: Rene Saarsoo

Kui HTML algselt loodi, oli selle eesmärgiks kirjeldada dokumendi struktuuri: pealkirjad, lõigud, tabelid, loendid jne. Aja jooksul hakkasid veebi vastu huvi tundma ka mitteakadeemilised ringkonnad, kes huvitusid rohkem sellest, kuidas kasutada uut meediat nii, et oma ettevõtmistele rohkem tähelepanu tõmmata ja uusi kliente leida.

Avastati, et pealkirju saab esitada kaunilt kujundatuna, kui kasutada h1 elemendi asemel hoopis pilti, ning et leheküljel võib teksti jaotada kenasti tulpadesse, kui kasutada tabeleid, millel on servadepaksus nulliks seatud.

Selliste tehnikate kasutamine oli veebi algusaegadel vältimatu, kui soovisid oma lehekülge külastajatele visuaalselt atraktiivsemaks muuta. Nüüdseks on aga brauserite areng jõudnud niikaugele, et hingematvalt kauni lehekülje loomiseks pole enam tarvis kasutada vanu trikke. Piisab sellest, kui märkida lehel olev tekst üles lähtuvalt selle semantikast ning kirjeldada seejärel stiililehes kuidas lehekülg välja peab nägema.

Presentatiivne lähenemine

Sa soovid luua lehekülge, kus on nimekiri puuviljadest ja iga puuvilja juures väike lühikirjeldus. Puuviljade nimed peaksid olema rasvases kirjas ning kirjeldused, kui vähemtähtsad osad, tillukeses kaldkirjas. Lehekülje alguses peaks olema rasvases kirjas punane tekst „Puuviljade loetelu", mis on eraldatud ülejäänud dokumendist horisontaaljoonega. Seega sa kirjutad:

```
<font color="red"><b>Puuviljade loetelu</b></font><br>

<hr>

<b>Apelsin</b><br>

<font size="2"><i>Magus tsitruseline.</i></font><br>

<br>

<b>Sidrun</b><br>

<font size="2"><i>Hapukas tsitruseline.</i></font><br>
```

See on presentatiivne lähenemine. Me mõtlesime, milline meie lehekülg peaks välja nägema, ning panime seejärel oma nägemuse vastavate HTML-i elementide abil kirja. Kuid HTML, nagu alguses öeldud, on mõeldud siiski selleks, et kirjeldada dokumendi struktuuri, märkides üles, mis on ühe või teise fraasi tähendus ehk semantika, mitte kajastades seda, milline üks või teine tekstijupp välja näeb. Selleks, et kirjeldada veebilehe kujundust on olemas märksa sobivam ja võimalusterikkam keel - CSS.

Semantiline lähenemine

Proovime luua eelpoolkirjeldatud lehte uuesti, ent lähtudes nüüd semantilisest vaatepunktist.

Kui järele mõelda, siis tekst „Puuviljade loetelu" pole midagi muud kui lehekülje pealkiri. Miks mitte ta siis ka vastavalt üles märkida:

```
<h1>Puuviljade loetelu</h1>
```

Järgnevad puuviljade nimed on samuti pealkirjad, lihtsalt ühe taseme võrra madalamad:

```
<h2>Apelsin</h2>
```

Viljade kirjeldused on jällegi harilikud tekstilõigud, mida tuleks üles märkida elemendiga p (paragrahv). Kuna kirjeldused moodustavad koos teema nimega ühtse terviku, siis võiksime täiendavalt lisada neid kokku ühendava elemendi div koos sobiliku klassiga puuvili. Kogu dokument oleks seega järgmine:

```
<h1>Puuviljade loetelu</h1>

<div class="puuvili">

<h2>Apelsin</h2>

<p>Magus tsitruseline.</p>

</div>

<div class="puuvili">
```

```
<h2>Sidrun</h2>

<p>Hapukas tsitruseline.</p>

</div>
```

See näeb välja juba märksa semantilisem. Ent kui meie kirjeldused iga puuvilja kohta saavad olema vaid mõne lause pikkused ja puuvilju endid kaunis palju, siis tundub iga puuvilja jaoks eraldi pealkirja tekitamine ülepingutatuna. Tundub, et meie puuviljade nimekiri on rohkem nagu definitsioonide loend - miks mitte siis kasutada ka vastavat HTML-i elementi:

```
<h1>Puuviljade loetelu</h1>

<dl id=”puuviljad”>

<dt>Apelsin</dt>

<dd>Magus tsitruseline.</dd>

<dt>Sidrun</dt>

<dd>Hapukas tsitruseline.</dd>

</dl>
```

Nüüd on lehekülg veelgi kompaktsem, sest saime ära jätta hulga div elemente koos klassidega ning ajada läbi vaid kogu loendile määratud atribuudiga id. Ühtlasi on tulemus ka semantiliselt korrektsem.

Kui definitsioonide loendi element dl pole teile tuttav, siis teadmiseks, et selles sisalduvad elemendid dt ja dd märgivad vastavalt defineeritavat terminit (definition term) ning tollele antavat kirjeldust (definition description).

Milleks see kõik?

Kuid nüüd ei näe see ju välja hoopiski nii nagu oli plaanitud. Pealkiri on hirmus suur, viljade nimed pole rasvased, ning kirjeldused on esitatud inetu taandega, rääkimata pealkirjaalusest joonest, mis on hoopistükkis puudu. Tundub, et nüüd näevad kõik selle lehekülje külastajad igavat suurt tumedat pealkirja ning täiesti harilikku musta teksti.

Suure hulga kasutajate jaoks see kindlasti nii ongi, kuid lisaks tüüpilistele kasutajatele, kes kasutavad levinumaid brausereid nende kõige levinumates seadistustes, on ka hulk neid, kes kasutavad hoopis teistsuguseid veebilehitsejaid ning hoopis teistsuguse konfiguratsiooniga.

Näiteks nägemispuudega isik, kes kasutab veebi lehitsemiseks kõnesüntesaatorit, kuuleks esimese presentatiivse koodiga lehekülje puhul midagi taolist: „Puuviljade loetelu {paus} apelsin magus tsitruseline {paus} sidrun hapukas tsitruseline.” Viimase, semantilise HTML-iga lehekülje puhul, kostuks aga kõnesüntesaatorist midagi järgmist: „Pealkiri {lühike paus} puuviljade loetelu {paus} apelsin {lühike paus} magus tsitruseline {paus} sidrun {lühike paus} hapukas tsitruseline.”

Erinevus teksti esitamisel tuleb sellest, et esimeses näites kasutatud elemendil br pole muud tähendust kui visuaalne reavahetus ning kõnesüntesaator ei tea, kas reavahetust kasutati lõigu lõpetamiseks või üksnes visuaalsel eesmärgil (nagu näiteks luuletustes tehakse).

Riigiasutuste lehekülgede puhul on kindlasti oluline, et ka puuetega kodanikud saaksid neid lehti külastada, aga mis on kõigest sellest kasu firmale, keda huvitavad vaid suurest rikkusest pakatavad kliendid?

Kõige rikkamad olendid Internetis on mitmesugused otsingurobotid, kelle võimuses on teie lehele kutsuda oma miljoneid sõpru. Otsingurobotid on aga täiesti pimedad ning peavad oma arvamuse leheküljest kujundama üksnes HTML-i põhjal.

Esimese mittesemantilise variandi puhul paneb otsingurobot tähele, et lehel leidub sõna „puuviljad” ning mõtleb: „Paistab, et sellel leheküljel räägitakse muuhulgas ka puuviljadest.” Kui otsingurobot külastab aga semantiliselt märgendatud lehekülge siis märkab ta, et leheküljel on suur pealkiri „Puuviljade loetelu”, ning mõtleb hoopis nii: „Ohhoo! Siin on üks lehekülg, mis on tervenisti pühendatud puuviljadele!”

Seega võime nüüd kokkuvõtvalt tõdeda, et semantiline HTML aitab meil oma lehekülje sõnumit edasi viia märksa laiemale hulgale kasutajatele kui presentatiivne lähenemine.

Aga ma tahan oma kujundust tagasi!

Ent kas sellepärast, et saada otsingumootorites paremat reitingut tuleb ohverdada kunstiline elamus, mida soovisime pakkuda kõigile moodsate brauseritega külalistele?

Sugugi mitte! See, et elemendi h1 sisu näidatakse enamikes veebilehitsejates suures ja rasvases kirjas, ei tähenda, et kõik esimese astme pealkirjad peavadki sellised välja nägema. Kuna me oma leheküljel pole öelnud, milline h1 peab välja nägema, siis kasutab brauser selle elemendi jaoks lihtsalt vaikimisi kujundust. Mitte miski ei keela meil aga täpselt kirjeldamast seda, kuidas me soovime, et pealkirjad ja kõik muu meie dokumendis välja näeks. Läheb tarvis vaid lühikest stiililehte, mis ütleks järgmist:

```
h1 {
```

font-size: medium;

color: red;

border-bottom: 2px groove gray;

padding-bottom: 0.5em;

}

dl#puuviljad dt {

font-weight: bold;

}

dl#puuviljad dd {

font-style: italic;

font-size: small;

margin: 0;

margin-bottom: 1em;

}

Jättes toodud stiililehe uurimise koduseks ülesandeks, toon järgnevalt ülevaate sellest, kuidas ja milliseid elemente kasutades erinevaid lehekülje osi võiks üles märkida.

Pealkirjad

Esimese astme pealkirja h1 tuleks kasutada leheküljel vaid korra - lehe individuaalse pealkirjana. Teise astme pealkiri h2 sobib kasutamiseks kõikvõimalike alampealkirjade puhul, sealhulgas näiteks pealkirjaks peamenüüle või mõnele muule lehekülje sektsioonile. Kolmanda astme pealkiri h3 sobib alampealkirjadele, mis jäävad h2 alla, h4 läheb h3 alla jne. Peamine on üritada välja selekteerida, milliseid osi tekstist me üldse võime pealkirjadena käsitleda.

Tihtipeale soovime, et meie lehekülje alguses asuv pealkiri oleks kujundatud vastavalt ettevõtte sümboolikale või oleks lihtsalt silmale kena vaadata. Selle asemel, et asendada pealkiri elemendiga img, tasub kaaluda võimalust kasutada pealkirja pildiga asendamiseks CSS-i. Taolistest pildiasendusvõtetest annab mõningase eestikeelse ülevaate lehekülg CSS: Pildiasendustehnikad.

Lõigud ja reavahetused

Tekstilõikude ülesmärkimiseks on teadagi element p. Kahetsusväärsel kombel kasutatakse lõikude eraldamiseks aga tihtipeale hoopis reavahetust br. Üheks peamiseks põhjuseks nõndaviisi toimimisel on asjaolu, et p jätab lõikude vahele tühja rea, aga lehe kujundaja on otsustanud lõigud eraldada taandridadega. See pole aga piisav ettekääne selleks, et loobuda lõikude korrektsest märgistamisest, sest elemendi p välimust saab CSS-ga muuta kergesti just selliseks, nagu on levinud enamikes trükistes:

p {margin: 0; text-indent: 2em}

Ka reavahetusel br on tänapäeva veebis oma koht, aga mitte lõikude eraldajana vaid teksti murdmiseks lõikude sees. Heaks näiteks on luuletused ja aadressid (siinkohal on hea ära märkida, et kahetsusväärselt harva kasutatav element address on mõeldud just kontaktinfo esitamiseks):

<address>

Kohila

Kuuse 9-12

Sander Sass

</address>

Loendid ja navigatsioon

Mõistagi tunnevad kõik järjestatud ja järjestamata loendeid ol ja ul ning oskavad neid ka hariliku teksti sees kasutada. Tihtipeale jääb aga märkamata, et mitmed veebilehtedel kasutatavad komponendid on oma olemuselt lihtsad järjestamata loendid.

Näiteks leidub enamikul lehtedest menüü, mille abil saab lehe erinevate osade vahel navigeerida. Oma olemuselt pole tegu aga millegi muuga, kui linkide loendiga. (Ka rõhtsalt paiknev menüü on loend, sest horisontaalne paigutus on kõigest üks võimalikest viisidest loendi kuvamiseks.) Kuid selle asemel, et märkida menüü üles kui järjestamata loend, eraldatakse menüü lingid tihtipeale reavahetustega

või paigutatakse hoopis tabeli lahtritesse.

Selleks, et anda menüüle kena välimust pole seda sugugi tarvis ümbritseda mittesemantilise müraga - piisab vaid oskuslikust stiililehtede kasutamisest. Suurepärased õpetused CSS-ga menüüde kujundamiseks on leitavad lehel Listutorial.

Üleüldse on mitmesugused navigatsioonielemendid kirjeldatavad loenditena. Näiteks lingid, mis viitavad eelmisele ja järgmisele leheküljele:

Eelmine lk

Järgmine lk

Loendeid tuleb vaid osata näha: loendi elementideks võivad olla artikli kohta postitatud kommentaarid, veebilehel teostatud otsingu tulemused, pildigalerii pildid jpm.

Ka definitsioonide loendil dl on palju kasutusvõimalusi. Lisaks erinevate terminite kirjeldamisele võib neid kasutada ka mitmel pool mujal. Näiteks võib sellise loendi abil kirja panna dialoogi, kus dt märgib kõnelejat ja dd lausutavat teksti, või jagada hoopis lehekülje selle abil alamosadeks. Võimalusi on palju.

Tabelid

Ehkki paljud veebistandardite propageerijad räägivad, et tabelid on saatanast - mida nad võibolla puhtpresentatiivse kasutamise puhul ka võivad olla - on tabelid siiski suurepärane vahend mitut sorti informatsiooni esitamiseks. Hoolitseda tuleb aga selle eest, et tabelid saaksid korrektselt üles tähendatud.

Igale leheküljel olevale tabelile tuleks määrata kirjeldus kasutades atribuuti summary. Kirjeldus ei pea olema pikk, aga piisav selleks, et mittevisuaalseid brausereid kasutavad külastajad saaksid aimu tabeli funktsioonist. Kui tabel on kasutusel vaid kujunduslikul eesmärgil, siis tuleks kasutada tühja kokkuvõtet:

<table summary=””>

Kui on soov anda tabelile pealkiri, pole tarvis tekitada eraldi hx pealkirja, sest tabeli algusesse saab lisada spetsiaalse elemendi caption. Ülejäänud tabeli võib hilisema mugavama kujundamise eesmärgil jagada päiseks thead, kehaks tbody ja jaluseks tfoot.

Paljudel lehekülgedel on tabeli tulpade pealkirjad märgitud järgnevalt:

<td>Tulp 1</td>

Kuid see on märk teadmatusest, sest HTML pakub nii tulpade kui veergude pealkirjastamiseks spetsiaalset elementi th, mida brauserid juba vaikimisi rasvasena kuvavad (ning mõistagi võib sellegi elemendi välimust CSS-ga muuta):

<th>Tulp 1</th>

Rasvane ja kaldkiri

Kaks olulist tüpograafilist vahendit veebis on rasvane ja kaldkiri (allajoonitud teksti kasutamine mujal kui linkides tekitab lehe kasutajates üksnes segadust). Traditsioonilised elemendid vastavate efektide loomiseks on b ja i, kuid need märgendid ei sisalda vähimatki semantilist tähendust.

Kui tahame oma teksti esitada kaldus või rasvasena peaksime kõigepealt mõtlema, miks me seda teha soovime. Võib-olla me soovime sõna kirjutada kaldkirjas selleks, et seda rõhutada? Sellisel juhul peaksime kasutama elementi em (emphasis). Või soovime mõne fraasi kirjutada rasvaselt, et seda tugevasti rõhutada? Sellisel puhul tuleks valida element strong (strong emphasis). Või soovime oma lugejatele tutvustada uut mõistet ning kirjutada see muust tekstist erinevalt seal, kus ta meie dokumendis esmakordselt esineb? Miks mitte kasutada elementi dfn (definition). Esitades veebis mõnda teadustööd, on reeglina kombeks kirjutada muutujanimed kursiivis. Selleks puhuks on olemas element var (variable).

Kui me soovime aga teksti kirjutada kaldkirjas või rasvaselt üksnes kujunduslikul eesmärgil, nagu meie esimeses koodinäites, siis peaksime kasutama CSS-i.

Muidugi ei suuda eelnevalt kirjeldatud semantilised elemendid kirjeldada kõiki situatsioone, kus me soovime teistsuguse kirjapildiga midagi edasi anda. Näiteks kirjutatakse liikide ladinakeelsed nimetused (nt Vulpes vulpes - rebane) reeglina kaldkirjas. Samuti kirjutame ka muud võõrkeelsed sõnad tihtipeale kaldkirjas.

Matthew Thomas soovitab oma blogis (Matthew Thomase blog), et sellistel puhkudel oleks mõttekas kasutada siiski elemente b ja i, mis ei lisa küll tekstile semantikat, kuid säilitavad oma erinevuse muust tekstist ka siis, kui stiililehti pole saadaval.

Oluline on siinkohal märkida, et pole tarvis tormata uisapäisa oma lehtedel kõiki i-sid ja b-sid em-de ja strong-dega asendama. Kui pole kavas iga kaldus või rasvase kirja kasutamise põhjust läbi mõelda, siis tasub jääda i ja b juurde, sest kui tekstis on nii võõrkeelsed sõnad

kui rõhutatud sõnad märgistatud i-ga kõlab lehekülg kõnesüntesaatoris lihtsalt igavalt, kui aga kõik on märgitud em-ga kõlavad leheküljel olevad võõrkeelsed väljendid lausa tobedalt.

Tsiteerimine

Tsiteerimise ja viitamise jaoks on HTML-is mitmeid elemente. Kui me soovime viidata mõnele allikale, saame kasutada elementi cite:

<cite>Sipsik</cite> on kuulus raamat Eno Raualt.

Kui soovime kedagi tsiteerida, siis on meil kasutada element blockquote, millel on valikuline atribuut cite (mitte segamini ajada samanimelise elemendiga), millega saab anda veebiaadressi, kust tsiteeritav tekst on võetud. Kuna enamik brausereid nimetatud atribuudi väärtust aga ei näita, siis tuleb praktikas viidatav aadress eraldi välja tuua:

 Juku kirjutas pühapäeval

oma blogis:

<blockquote cite=”http:// Juku .ee/2005/07/10”>

Järgmine aasta õpin ainult viitele!

</blockquote>

Lisaks tsitaatidele on olemas ka element q reasiseste tsiteerimiste tarbeks. Kahjuks pole see element aga brauserites kuigivõrd toetatud (q-ga ümbritsetud teksti ei panda automaatselt jutumärkidesse), mistõttu tuleks seni veel kasutada harilikke jutumärke.

Lühendid

Tekstis kasutatavaid lühendeid võib üles märkida kasutades kas elementi abbr (abbrevation) või acronym. Viimase eeliseks on asjaolu, et ta töötab ka MS Internet Exploreriga, esimene on jällegi semantiliselt korrektsem, sest mitte kõik lühendid pole akronüümid. Mõlema elemendi kasutamine käib koos atribuudiga title, mille väärtust kuvatakse brauseris, kui lühendist kursoriga üle libistada:

<abbr title=” Microsoft ”>MS</abbr>

Koodinäited veebis

Veebis programmide koodinäidete esitamiseks on juba ammu kasutatud elementi pre, mis on tegelikult mõeldud eelvormindatud teksti märgendamiseks. Kuid lisaks mainitule leidub HTML-is ka spetsiifilisemaid elemente.

Element code on mõeldud lähtekoodi tähistamiseks. Elemendiga samp (sample) saab tähistada programmide näidisväljundeid ning elemendiga kbd (keyboard) kasutaja poolt sisestamiseks mõeldud teksti. Näiteks:

Et näha kataloogi sisu, kirjuta <kbd>ls -l</kbd>.

Vormid

Vormid on dünaamiliste lehtede alustala, ent üksnes vormi nuppudel on automaatselt küljes neid kirjeldav tekst, ülejäänud vormi väljadele tuleb see määrata käsitsi - soovitavalt sildiga label. Kui vormi väljaga on seotud ka silt, siis piisab vormi elemendi aktiveerimiseks ka sildil klikkimisest, mis on eriti kasulik radio ja checkbox tüüpi väljade puhul (kahjuks ei toimi see aga MS Internet Exploreris). Silti saab siduda kas eraldi for-atribuuti kasutades:

<input type=”checkbox” name=”soov” id=”soov”>

<label for=”soov”>Ei soovi reklaami.</label>

või paigutades vormi välja elemendi label sisse:

<label><input type=”checkbox” name=”soov”>

Ei soovi reklaami.</label>

Pikemate vormide puhul on mõistlik koondada temaatiliselt seotud sildid ja väljad gruppidesse, kasutades selleks elementi fieldset koos tema kaaslasega legend, mis määrab grupi pealkirja:

<fieldset>

<legend>Kontaktinfo</legend>

<label>Nimi:

<input type=”text” name=”nimi”></label>

<label>E-post:

<input type=”text” name=”epost”></label></p>

</fieldset>

Pildid

Veebilehtedel leiduva pildimaterjali võib laias laastus jaotada kaheks: sisulist väärtust omavad ja vaid kujunduslikku eesmärki kandvad. Esimeste hulka kuuluvad skeemid, diagrammid, kaardid, illustratsioonid, fotod jne. Ka näiteks asutuse logo kannab reeglina sisulist väärtust. Kõik sedasorti pildid tuleks kanda lehele hariliku img elemendiga, millele on mõistagi määratud sobilik alternatiivtekst atribuudiga alt.

Kujunduslike piltide lisamiseks tuleks kasutada CSS-i atribuuti background-image, või kui siiski soovitakse kujundamiseks kasutada elementi img, siis tuleks talle määrata tühi alternatiivtekst, et mittegraafilised brauserid ei näitaks pildi koha peal failinime:

<img src=”vasaknurk.jpg” alt=””>

Klasside ja identifikaatorite nimed

Muidugi ei leidu HTML-is kõiki elemente, mida meil tarvis võib minna, mistõttu tuleb võtta appi atribuudid class ja id. Et aga meie lehekülge ei nakatuks haiguste classitis ja divitis kätte, peame enne nende atribuutide kasutamist olema veendunud, et HTML-is tõesti ei leidu meie jaoks sobivat elementi. Seejärel tuleks vaadata, kas me ei saaks võtta mõnda olemasolevatest semantilistest elementidest ja täpsustada seda lisades klassi või ID. Alles siis, kui ka see variant ei sobi, peaksime pöörduma elementide div ja span poole.

Kuid ka klasside ja ID-de nimed tuleks valida nõnda, et nood kirjeldaksid semantikat. Näiteks kui soovime kuvada punast pealkirja ei peaks me määrama pealkirjale mitte klassi punane vaid hoopis näiteks viga:

<h2 class=”viga”>Viga andmete sisestamisel!</h2>

Eelpoolkirjeldatud on vaid üldised nõuanded, mida ei pea ilmtingimata järgima. See, kus ja kuidas üht või teist elementi kasutada, võib olla erinevate lehekülgede puhul üksjagu erinev. Samuti ei tasu uskuma jääda ühe autori arvamust - asjast huvitatud otsivad kindlasti ise võrgust endale materjali juurde.

Lingid samal teemal:

CSS: Pildiasendustehnikad
Listutorial
Matthew Thomase blog
Dave Shea, Markup Guide
Shirley E. Kaiser, Semantics, HTML, XHTML, and Structure
W3C, HTML 4.01 Specification

- Lahendused
- Tarkvara