

KARELIA-AMMATTIKORKEAKOULU
Tietojenkäsittelyn koulutusohjelma

Juhani Mikkonen

STANDARDINMUKAISEN WEB-SUUNNITTELUN MUUTOKSIA
VUOSINA 2006-2013

Opinnäytetyö
Huhtikuu 2013



OPINNÄYTETYÖ
Huhtikuu 2013
Tietojenkäsittelyn koulutusohjelma

Karjalankatu 3
80200 JOENSUU
+358 50 260 6800

Tekijä(t)
Juhani Mikkonen

Nimeke
Standardinmukaisen web-suunnittelun muutoksia vuosina 2006–2013

Tiivistelmä

Raportin tarkoituksena on toimia päivityksenä aiempien opintojeni (Pohjois-Karjalan Ammattikorkeakoulun viestinnän koulutusohjelmassa) yhteydessä tekemääni opinnäytetyöhön, jonka nimeke oli ”Standardinmukaisen WWW-suunnittelun tarkastelu Mejä SM 2006 -projektissa”, sekä Karelia-ammattikorkeakoulun tietojenkäsittelyn koulutusohjelman opintoihini liittyvässä aiemmin hankitun osaamisen tunnistamisessa ja tunnustamisessa (AHOT) opinnäytetyön hyväksiluvun prosessissa. Raportissa muun muassa tarkastellaan vuonna 2006 tekemieni ratkaisujen toimivuutta, aiemmassa opinnäytetyössäni esitettyjä tulevaisuusskenaarioita ja niiden tilaa nyt, teknisiä muutoksia vuosien välillä mobiililaitteetkin huomioon ottaen, sekä yleisimpien web-selaimien kehitystä.

World Wide Web Consortiumin suosittelmien puhdasoppisten web-standardien, XHTML Strict-muodon ja CSS2:n korvaajiksi on pikku hiljaa tulossa HTML5 ja CSS3, jotka saattavat osaltaan parantaa web-sivujen rakenteen selkeyttä, sivujen käytettävyyttä ja saavutettavuutta. Uusimmat web-selaimet pyrkivät noudattamaan standardeja, mutta eroavat teknisiltä ominaisuuksiltaan. Teknologian kehityksen myötä laitteiden sekä näyttökokojen kirjon kasvaessa web-sivujen ulkoasun mukautuvuuden suunnittelun merkitys korostuu.

Kieli
suomi

Sivuja 30
Liitteet 6
Liitesivumäärä 9

Asiasanat
AHOT, opinnäytetyö, web-standardit, WWW, XHTML, HTML5, CSS



THESIS
April 2013
Degree Programme in Business
Information Technology
Karjalankatu 3
FI 80200 JOENSUU
FINLAND
+358 50 260 6800

Author(s)
Juhani Mikkonen

Title
Changes in Web Design with Web Standards in the years 2006–2013.

Abstract

The aim of the present paper is to act as an update for my previous thesis called “The examination of Web-standard usage in Mejä SM 2006 – project”, which I wrote during my studies in Degree Programme in Communication at NKUAS. It also acts as a report for the process of compensating the thesis in my current studies in Degree Programme in Business IT through RPL (Identification and recognition of prior learning). In this study I, among other things, evaluate the functioning of solutions I introduced in 2006, the future scenarios presented and the state of those today, the technical changes between the years considering also mobile devices, and the progress of the most common web browsers.

HTML5 and CSS3 are replacing the XHTML Strict and CSS2 web standards recommended by the World Wide Web Consortium (W3C) and they may improve the clarity of web page structures as well as the usability and accessibility of web sites. The newest web browsers aim to follow web standards, but they have different technical characteristics when compared to each other. As the variety of devices and different display sizes increase with the development of technology, the significance of designing adaptable and scalable web pages will be more emphasized in the future.

Language
Finnish

Pages 30
Appendices 6
Pages of Appendices 9

Keywords

RPL, thesis, web standards, WWW, XHTML, HTML5, CSS

Sisältö

1	Johdanto.....	5
2	Vuoden 2006 näkökulma ja tutkimukselliset kysymykset	6
3	Tekniikoiden ja standardien kehitys	7
3.1	Web-standardit	7
3.2	Skaalautuvuuden mahdollisuudet	8
3.3	HTML-kielen kehitys	9
3.5	CSS-kielen kehitys.....	11
3.6	Käytetyimpien web-selaimien kehitys.....	12
3.7	Näyttöresoluutioiden kehitys	14
4	Vuonna 2006 tehtyjen ratkaisujen tarkastelua.....	14
4.1	Rakenne ja taitto.....	14
4.2	Pdf-dokumenttien käyttö	15
4.3	Sivunsisäiset linkit.....	16
4.4	Sivuston skaalautuvuus	17
4.5	Kuvagalleria	18
4.6	Kirjasinlajit ja -koot.....	18
4.7	Kellutukseen perustuvan layoutin ongelma	20
4.8	Mobiililaitteet	21
5	Lopputulokset	24
5.1	Validointi	25
5.2	Simuloituja testauksia	27
6	Pohdinta	27
	Lähteet	30

Liitteet

- Liite 1 Esimerkki web-sivuston ulkoasun skaalautuvuudesta
- Liite 2 SM-Mejä 2006 -sivuston etusivu iPhone 4:llä testattuna
- Liite 3 SM-Mejä 2006 -sivuston etusivun tarkastelu iPad Peek-ohjelmalla
- Liite 4 SM-Mejä 2006 -sivuston tarkastelu Opera Mini-emulaattorilla
- Liite 5 SM-Mejä 2006 -sivuston tarkastelu Android-emulaattorilla
- Liite 6 SM-Mejä 2006 -sivuston tarkastelu nykyajan web-selaimissa

1 Johdanto

Tein vuoden 2006 lopulla Pohjois-Karjalan Ammattikorkeakoulun viestinnän koulutusohjelmassa opiskellessani opinnäytetyöni nimekkeellä ”Standardinmukaisen www-suunnittelun tarkastelu Mejä SM 2006 -projektissa”. Työni koostui Suomen Noutajakoira järjestö ry:lle toimeksiantona tehdystä tiedottavasta web-sivustosta, joka liittyi vuoden 2006 metsästysjäljestäjäkoirien Suomen mestaruuskilpailuihin (SM-Mejä), sekä kirjallisesta raportista työn prosessista.

Tarkemmin ottaen tapahtuma, jonka tiedottamiseen web-sivut tehtiin, oli Suomen Noutajakoira järjestö ry:n (SNJ) yhdessä Kaakon Noutajakoira yhdistys ry:n kanssa järjestämä metsästyskoirien jäljestämiskokeiden voittajaluokan Suomen mestaruuskilpailu. Kilpailu käytiin 23.–24.9.2006 Haminan Pyhällössä, Lintukodossa. Kyseisenlaisia kilpailuja järjestetään yleensä eri vuosina eri rotujärjestön voimin. Niin sanotun mejä-kokeen avulla testataan koiran kykyä seurata haavoitetun riistaeläimen jälkeä. Tällaisten kokeiden tarkoituksena on edistää jäljestävien metsästyskoirien koulutusta, lisätä sekä laajentaa rotukohtaisesti harrastajien määrää, tehdä tunnetuksi jäljestävät koirat ja niiden käyttömahdollisuudet haavoittuneita hirvieläimiä etsittäessä.

Kirjallisessa tuotoksessani lähtökohtana oli standardinmukaisen WWW-suunnittelun tarkastelu. Sivustossa pyrin siihen, että se toimisi siihen aikaan käytetyimmissä web-selaimissa ja validoituisi W3C:n (World Wide Web Consortium) sääntöjen mukaisesti. Samalla pyrin huomioimaan hyvään käytettävyyteen ja saavutettavuuteen liittyviä, hyväksi havaittuja menetelmiä muun muassa ulkoasun skaalautuvuuden kontekstissa. Web-sivuston tekninen toteutus tapahtui XHTML:n Strict-muodossa määritellen ulkoasun tyyli CSS:n (Cascading Style Sheets) keinoin.

Standardinmukaisuuteen liittyy pyrkimys web-suunnittelussa käytettävän kielen yhtenäistämiseen ja web-sivustojen toimivuuteen tarjolla olevissa ja tulevaisuudessa selaimissa ja selainversioissa. Eri web-selaimet eroavat toisistaan muun muassa siinä, millä tavalla ne noudattavat niin sanottuja yhteisiä sääntöjä ja esimer-

kiksi CSS-tuen laajuuden kohdalla. Eri selainohjelmien ja myös eri selainversioiden kanssa on alati painittu yhteensopivuusongelmien kanssa. Muutama vuosi sitten pystyi kenties sanomaan, että standardien noudattamiseen on alettu kiinnittämään yhä enemmän huomiota – ongelmia on ollut siinä, että selaimet ovat muun muassa tulkinneet HTML-kieltä epäjohdonmukaisesti. Webin suuri kasvuvauhti on vaikeuttanut standardointiprosessia. HTML eli Hypertext Markup Language on aikoinaan kehitetty standardi, jonka alkuperäisenä tarkoituksena oli toimia kaikenlaisissa päätelaitteissa tietokoneista matkapuhelimiin. Web-sivustojen standardinmukaisuus ei kuitenkaan korjaa selainohjelmistojen virheiden tuomia ongelmia.

Tämä raportti liittyy opintoihini Karelia-ammattikorkeakoulun (entinen Pohjois-Karjalan ammattikorkeakoulu) tietojenkäsittelyn koulutusohjelmassa ja toimii aiemmin hankitun osaamisen tunnistamisessa ja tunnustamisessa (AHOT) opinnäytetyön hyväksiluvun prosessissa. Raportissa muun muassa tarkastellaan vuonna 2006 tekemiäni ratkaisujen toimivuutta tänä päivänä (vuonna 2013), opinnäytetyössäni esitettyjä tulevaisuusskenaarioita ja niiden tilaa nyt, teknisiä muutoksia vuosien välillä muun muassa mobiililaitteitakin silmällä pitäen sekä yleisimpien web-selaimien kehitystä standardisointiprosessia ajatellen.

2 Vuoden 2006 näkökulma ja tutkimukselliset kysymykset

SM-Mejä 2006 -sivustoja suunnitellessani pyrin standardinmukaisuuden lisäksi ulkoasun skaalautuvuuteen, joka tässä asiayhteydessä tarkoittaa ulkoasun muokautuvuutta suhteellisesti käytössä olevan näytön ja resoluution mukaan. Silloisen työni tutkimuksellisia asioina oli muun muassa tarkastella, rajoittaako standardinmukainen toteutus visuaalisen päämäärän saavuttamista ja näkökulmana oli, minkälainen rooli validoinnilla ja taulukottomalla CSS-suunnittelulla (eli taulukoita käytetään vain niiden alkuperäisessä tarkoituksessaan, ei visuaalisen ulkoasun luonnissa) on standardinmukaisessa WWW-suunnittelussa.

Opinnäytetyöhöni liittyvien web-sivujen suunnittelussa yhtenä huomioitavana asiana oli, että kaikilla käyttäjillä ei välttämättä ole käytössään parhaimmasta päästä olevat laitteet, nopeat nettiyhteydet ja suurikokoiset näytöt. Käytettävyyden kohdalla pyrin huomioimaan hyväksi havaittuja seikkoja ja niihin liittyviä olemassa olevia tutkimustuloksia, erityisesti arvostetun käytettävyysasiantuntijan Jakob Nielsenin sääntöihin pohjautuen. Opinnäytteessäni pyrin myös herättämään ajatuksia parempaan verkkosivusuunnitteluun pyrkimisen kohdalla siten, että lähtökohtana on käytetyimmissä web-selaimissa toimivat verkkosivutoeutukset, painottaen sanaa toimivat, sillä on mahdotonta tehdä kaikissa selaimissa täsmälleen samalta näyttävät sivut.

SM-Mejä 2006 -sivuston suunnittelun alkuvaiheessa linkkien otsikot ja sivuston aihealueet määräytyivät osittain kilpailuihin aiemmin tehtyjen sivustojen pohjalta yhdistettynä uuden sivuston suunnitteluun liittyvien kehityskeskustelujen kautta esiin tulleisiin tarpeisiin ja vaatimuksiin. Ajateltuja kohderyhmiä sivustolla oli neljä: metsästysjäljestäjäharrastajat (joitakin tuhansia) ja mejä-kokeeseen osallistuvat; kilpailun tuomaristo; organisaatiot (esimerkiksi Suomen Kennelliitto); mediat, jotka saisivat sivuilta materiaalia.

3 Tekniikoiden ja standardien kehitys

3.1 Web-standardit

Web-standardit ovat eräänlaisia määrittelyjä WWW:tä (World Wide Web) ja muita Internetin näkökohtia varten. Nykyään termiä käytetään useimmiten verkkosivujen parhaiden käytäntöjen rakentamisen, web-designin filosofoiden ja web-kehityksen yhteydessä. Päänäkökulmia web-standardien kohdalla ovat web-sivujen ja sivustojen yhteentoimivuus, saatavuus ja käytettävyys. Jos web-sivuston tai web-sivun kuvaillaan noudattavan web-standardeja, tarkoittaa se useimmiten sitä, että sivusto tai sivu sisältää validia (X)HTML-, CSS- ja JavaScript-kieltä. HTML-kielen tulee myös noudattaa tietynlaisia saavutettavuuteen ja semanttisuuteen liittyviä ohjesääntöjä. Tyypillisimmin perustavina tekijöinä web-

standardeista puhuttaessa pidetään muun muassa kansainvälisen yhteisön W3C:n (World Wide Web Consortium) julkaisuja ja suosituksia. (Wikipedia 2013a.)

3.2 Skaalautuvuuden mahdollisuudet

Voidaan ajatella, että ulkoasun skaalautuvuuden keinoin otetaan huomioon esteettömyys tai saavutettavuus, joka tarkoittaa erityisryhmien, kuten heikonäköisten huomioimista suunnittelussa. Ulkoasun skaalautuvuudessa on kyse mittasuhteiden suhteellisuudesta käytössä olevaan näyttökokoon ja resoluutioon, eli esimerkiksi kirjasimet on määritelty suhteellisesti sen sijaan, että niiden kooksi olisi määritelty tietty pikselimäärä. Tämän lisäksi ulkoasun skaalautuvuudella voidaan päästä järkevämpään tilankäyttöön; suurilla näyttöresoluutioilla sivuston reunoihin ei välttämättä jää kamalan paljon tyhjää tilaa. On myös mahdollista suunnitella sivuston toimivuus ulkoasun suhteen siten, että monet sivustolla näkyvät elementit joko muuttavat kokoaan suhteessa johonkin, tai tiettyjen elementtien näkyvyys määräytyy sen mukaan, minkälaisella näytöllä tai laitteella sivustoa katsellaan. Yksi hyvä esimerkki web-sivuston ulkoasun kaikkien elementtien skaalautuvuudesta eri näyttökokojen mukaan löytyy sivustolta www.peecho.com, jossa kyseinen asia on teknisesti melko pitkälle viety (liite 1).

Vuonna 2006 SM-Mejä -sivustoa tehdessäni käytetyin näyttöresoluutio oli 1 024 x 768 pikseliä ja ulkoasun skaalautuvuutta miettiessäni huomioon otettavimpien näyttökokojen kirjo asetui suunnilleen välille 800 x 600 – 1 280 x 1 024 pikseliä. Sivujen ulkoasun optimoimisessa voidaan tarkastella, etteivät tärkeimmän informaation näkyvyys, tekstin luettavuus ja sivun esteettisyys kärsisi.

Tuohon aikaan kuvaputkinäytöt olivat vielä jokseenkin paljon käytössä. Kyseisessä tekniikassa näyttöruudulla värit muodostettiin johtamalla elektroneja kuvaputken pinnan hehkuaineeseen. Lopputuloksena muodostui punaisia, vihreitä ja sinisiä (RGB) kuvapisteitä näytön kuvapinnalle. Nykyisin on käytössä esimerkiksi HD (High Definition) LED LCD -tekniikalla varustettuja litteitä näyttöjä. Lähitulevaisuudessa tietokonenäytöt monipuolistuvat (muun muassa kuvasuhteil-

taan) ja niistä tulee entistä tarkempia. Näyttöjen 4k- eli Ultra High Definition - tarkkuus on tuleva suuntaus. (Taipale 2013.)

Ennen (esimerkiksi vielä 2000-luvun alkupuolella) oli tavanomaista rakentaa web-sivujen layout taulukkoon, vaikka taulukot suunniteltiinkin alun perin taulukkomuotoisen datan esittämiseen. Koska taulukoiden rajat oli mahdollista saada pois näkyvistä, alettiin niitä käyttää kehyksinä tekstile ja kuville. Viestinnän koulutuksessakin (2002–2006) meille opetettiin web-sivujen ulkoasun suunnittelua sijoittamalla graafiset elementit sekä sisältötekstit taulukkoon, vaikka CSS-tyyliohjeet olivat käytettävissä.

Ehkä vielä 2000-luvun alkupuolella saattoi nähdä joillain web-sivustoilla käyttäjälle tarkoitetun ohjeistuksen ”tämä sivu toimii parhaiten selaimen Å versioilla Ö sekä X resoluutiolla”, joka nykyään tosin on jo muinaishistoriaa. Ulkoasun skaalautuvuudella tällaisistakin saavutettavuutta rajoittavista tekijöistä on kuitenkin päästy eroon.

Jos sivusto saadaan skaalautumaan kaikenkokoisiin selainikkunoihin, olisi ennen voitu ajatella, että se rajoittaa sivujen ulkonäön kontrollointia, eli suunnittelijan tulisi valita, tekeekö kiinteän designin vai skaalautuvan. Omasta mielestäni nykyisin suunnittelun lähtökohdat ovat miltei väkisin siinä, että monenlaiset tekniset ympäristöt tulee ottaa huomioon, ja teknisesti on silti mahdollista hallita sitä, miltä sivut näyttävät käyttäjälle; vaikka tehtäisiin ulkoasultaan kiinteät sivut, voitaisiin silti tehdä kiinteät sivut eri laitteille/näytöille, esimerkiksi tekemällä jokaista varten omat tyylimääritykset. Ulkoasultaan skaalautuvat sivut vaativat suunnittelijalta enemmän työtä ja testausta, mutta sen kannattavuus on korkealla käytettävyydenkin kannalta, sillä nettiä selaillaan nykyisin niin monenlaisilla laitteilla ja monen kokoisilla näytöillä.

3.3 HTML-kielen kehitys

SM-Mejä 2006 -sivustoa tehdessäni HTML:n (Hypertext Markup Language) kehitys oli edennyt versioon 4.01 saakka. Yleisnimityksenä HTML:n uusista versi-

oista käytettiin XHTML:ää (eXtensible Hypertext Markup Language). XHTML:illä on ollut tarkoitus hakea yhteensopivuutta muiden XML-sovellusten kanssa. XHTML:n versio 1.0 tarkoittaa periaatteessa samaa kuin HTML 4.01. Rakensin web-sivut käyttäen XHTML 1.0 Strict-versiota, jonka voitiin sanoa olevan puhtasoppisempi kuin esimerkiksi ”väljempi” Transitional-versio. Käytössä oleva versio kerrotaan XHTML-sivustossa jokaisen sivun dokumenttityypin määrittelyissä (DTD eli Doctype).

Yksi standardinmukaisuudessa tärkeä asia on ollut merkityspainotteinen tiedon tallentaminen, eli esimerkiksi pääotsikot, väliotsikot ja kappaleet merkitään semanttisella koodauksella (X)HTML-kielessä H1, H2, P ja niin edelleen. Käytössä ovat olleet samat peruselementit kuin HTML:n ensimmäisessä versiossakin. Tämä on tänäkin päivänä pätevä ja standardinmukainen tapa tiedon esittämiseen HTML:n perusrakenteessa.

HTML:n uusin versio, eli HTML5, tuo merkkaukieleen enemmän semanttisuutta, esimerkiksi rivit

```
<div class="header">  
<h1>Page Title</h1>  
</div>
```

voidaan HTML5:lla merkata seuraavasti:

```
<header>  
<h1>Page Title</h1>  
</header>
```

Kun taannoin taulukkopohjaisesta layout-suunnittelusta siirryttiin <div>-elementtien käyttöön, ollaan nyt siirtymässä yhä selkeämpään rakenteen tekemiseen. Div-elementtienkin käytössä lopputulos voi olla pahimmillaan melkoinen <div>-tagien sekasotku, joten HTML5 voi tehdä merkintäkielestä semanttisesti tarkempaa ja luettavampaa.

HTML5:n määrittely kuvaa tarkasti dokumenttioliomallin (DOM). Määrittely kuvaa HTML-dokumentin ensisijaisesti oliona, jolla on puurakenne ja jota voidaan käsitellä ohjelmallisesti (etenkin JavaScriptillä). Puurakennetta voidaan esittää HTML-tagien avulla tekstimuodossa, jolloin puhutaan sarjallistamisesta. Jos sarjallistetaan XML:n sääntöjen mukaan, käytetään esitystavasta nimitystä XHTML5. Valinta HTML5:n ja XHTML5:n käytön välillä riippuu oikeastaan sisältöjen mediatyypistä (MIME). (Wikipedia 2013b.)

Mobiilisovellusten kehityksen puolella HTML5 on tällä hetkellä siinä asemassa, että täytyy tehdä valinta natiivin (alustakohtainen sovellus) ja HTML5:n välillä. HTML5 lupaa alustariippumattomuutta, mutta natiivisovellus päihittää HTML5:n varsinkin suorituskyvyssä, sillä se voi hyödyntää suoraan käyttöjärjestelmään liittyviä ohjelmointirajapintoja. Sen lisäksi HTML5:n alustariippumattomuuden avain on kyseistä standardia tukeva selain. HTML5:n etuina voivat kuitenkin olla kustannussäästöt verrattuna natiiviin (jota täytyisi tehdä monelle alustalle) sekä nopean päivittämisen mahdollisuus. (Saarelainen 2013.)

3.5 CSS-kielen kehitys

CSS:n (Cascading Style Sheets) avulla määritellään dokumentin ulkoasua koskevat niin sanotut tyyliohjeet. SM-Mejä 2006 -sivuston tekoaikaan XML-kieli alkoi olla yleistymässä, ja samalla myös CSS:n merkitys oli kasvussa, ja nykyisin lähes välttämättömyys web-sivujen ulkoasua tehdessä. Tyyliohjeiden erottaminen sivujen rakenteen kuvauksesta (HTML:llä) on järkevää siinä mielessä, että silloin tekniikoita käytetään siten, kuin niitä on standardinmukaisuudessa tarkoitettukin.

SM-Mejä 2006 -sivuston tekoaikaan CSS oli edennyt versioon 2, ja level 3 oli jo kehitteillä. CSS2:n uudet ominaisuudet verrattuna CSS1:een olivat muun muassa kuvien ja tekstin asemointi minne vain, elementtien laittaminen päällekkäin, sisällön ylivuodon käsittely ja taulukoiden ulkoasun parempi muotoilu. CSS3 on tuonut mukanaan paljon uudenlaisia ominaisuuksia, ja osalla niistä korvataan jopa Flashin ja Java appletien ominaisuuksia. Tänä päivänä CSS3:n selaintuki

on vieläkin melko pieni, etenkin Internet Explorerissa. CSS3:n uusiin ominaisuuksiin kuuluvat esimerkiksi animointi, elementin kääntäminen, laatikoiden muuntaminen, fontin venytys, tekstin kelaaminen, tietyt rivien ominaisuudet, puheominaisuudet puheselaimille sekä kolmen eri akselin suuntainen käänteleminen. (Wikipedia 2013c.)

3.6 Käytetyimpien web-selaimien kehitys

SM-Mejä 2006 -sivustoa suunnitellessani otin eniten huomioon standardeja kaikkein pisimmälle noudattavat selaimet Firefox 1.5, Internet Explorer 6 ja Opera 8.53. IE:n kohdalla siihen aikaan otettiin yleensä websivujen suunnittelussa pääasiassa huomioon versio 5.x ja siitä ylöspäin. Muiden kuin IE:n selainten kohdalla on ollut mahdollista testata suunnittelemaansa sivuja asentamalla samaankin koneeseen eri selainten eri versioita. IE:n versio on aina ollut kytköksissä Windows-käyttöjärjestelmän versioon, joten kyseisiä selaimia ei ole voitu asentella rinnakkain samalle koneelle, paitsi jos oli esimerkiksi jakanut kovalevyn erillisiin osioihin. Mac-maailmassa Safari-selaimesta oli siihen aikaan käytössä versio 2.0.4.

Nykyisin virtualisointi mahdollistaa uudenlaisia keinoja vaikkapa kehitteillä olevien web-sivujen testaamiseen erilaisissa ympäristöissä. Virtualisoinnissa tietokoneelle asennetaan virtualisointiohjelmisto (esimerkiksi VirtualBox), jossa luodaan yksinkertaistettu malli PC-laitteistosta sisältäen muun muassa näytönohjaimen, kiintolevyn, näppäimistön, CD/DVD-aseman, näytön ja prosessorin. Virtuaalikoneelle asennettu käyttöjärjestelmä näkee tämän yksinkertaistetun laitekokoonpanon, mutta käyttää todellisuudessa niin sanotun isäntäkoneen fyysistä laitteistoa. Täten yhdelle ja samalle koneelle voitaisiin virtualisoinnin keinoin asentella esimerkiksi monia eri Windows-versioita ja sitä kautta eri Internet Explorer-versioita.

Vuonna 2006 käytetyin verkkoselain oli IE (versio 6) ja toiseksi käytetyin Firefox. Muita käytetyimpiä olivat Mozilla, Netscape ja Opera. Netscapen kehitys päättyi vuonna 2007. Mozillalla tarkoitettiin tarkemmin ottaen Mozilla Application

Suitea, jonka kehitys loppui vuonna 2006. Sen jälkeen Mozilla Foundation keskittyi Firefoxin ja Thunderbirdin kehittämiseen. Vuoden 2013 alkupuolella käytetyin selain oli Google Chrome ja toiseksi käytetyin Mozilla Firefox, joiden perässä tulivat Internet Explorer, Safari ja Opera. Tiedot ovat peräisin W3Schools.com -sivuston tilastoista. (W3Schools 2013a.)

Maailmanlaajuiset markkinaosuudet johtavien web-selaimien keskuudessa jakautuu maaliskuun 2013 tilanteen mukaan niin, että Chrome 25.0 on kärjessä (32.28 %). Sen jälkeen tulevat IE 9.0 (15.81 %), Firefox 19.0 (15.11 %), IE 8.0 (10.29 %), Safari iPad (3.83 %), IE 10.0 (2.26 %), Safari 6.0 (2.25 %), Safari 5.1 (1.67 %), Chrome 26.0 (1.42 %), Chrome 24.0 (1.19 %), Opera 12.1 (0.83 %) ja Firefox 18.0 (0.78 %). (StatCounter 2013.)

Tämän raportin kirjoitushetkellä suosituimpien selaimien viimeisimmät versiot ovat seuraavat: Chrome 26, Firefox 20, Internet Explorer 10, Safari 6.0 ja Opera 12.15. Kaikki nämä selaimet noudattavat alan yleisiä standardeja. Kaikissa selaimissa on mukana tärkeimmät ominaisuudet esimerkiksi svg-vektorigrafiikasta ja HTML5-standardista. Kaikista selaimista löytyy samoja ominaisuuksia, kuten yksityisyystila, näytön ja tekstin zoomaus, koko ruudun selaus ja laajennusosat. Eroja näkyy muun muassa muistinkulutuksen ja JavaScript-nopeuden kohdalla. WWW-sivujen tekijät eivät nähtävästi kuitenkaan tänä päivänä testaa sivujaan niin paljon IE:llä kuin muilla selaimilla, kuten Firefoxilla ja Chromella. Tästä syystä saatetaan monien suomalaistenkin WWW-sivujen kohdalla törmätä sivustojen toimivuusongelmiin IE:tä käytettäessä. (Järvinen 2012.)

Sivustolla html5test.com voidaan tarkastella oman, käytössään olevan selaimen sekä muiden selaimien HTML5 tuen monipuolisuutta. Jos tarkastellaan, mitkä selaimet tukevat HTML5:sta parhaiten, on pöytäasemaselainten listan kärjessä tällä hetkellä Maxthon 4.0. Toisena on Chrome, sitten Opera, jonka jälkeen tulevat Firefox, Safari ja Internet Explorer. (Leenheer 2013.)

3.7 Näyttöresoluutioiden kehitys

Vuonna 2006 käytetyimpänä näyttöresoluutiona oli 1 024 x 768. Sitä suurempien näyttöjen käyttäjämäärä oli kasvussa. Jonkin verran oli vielä myös kokoa 800 x 600 käytössä. Vuoteen 2013 tullessa resoluutio 1 024 x 768 oli käytössä vain 9%:lla, ja 90%:lla tavanomaisista internet-käyttäjistä oli käytössään suurempi kuin edellä mainittu resoluutio. Näiden 90%:n keskuudessa käytetyin resoluutio oli 1 366 x 768, seuraavana muun muassa 1 920 x 1 080 ja muut suuret resoluutiot. Tiedot ovat peräisin W3Schools.com -sivuston tilastoista. (W3Schools 2013b.)

4 Vuonna 2006 tehtyjen ratkaisujen tarkastelua

SM-Mejä 2006 ei ollut tämän raportin kirjoitushetkellä enää virallisesti julkaistuna verkossa, mutta latsin tekemäni sivuston testailua varten oman internetliittymäni palveluntarjoajan kotisivutilaan. Näin ollen pystyin katselemaan sivustoa eri paikoissa erilaisilla välineillä sekä tutkimaan niiden toimivuutta erilaisissa ympäristöissä.

4.1 Rakenne ja taitto

Koska työni SM-Mejä 2006 -sivuston parissa liittyi pääasiassa vain ulkoasullisiin ja sivun koontiin liittyviin asioihin, tarkastelen tässä raportissa vain tämänkaltaisia teknisiä asioita. Järjestelin navigointilinkit listaelementiksi, ja tarkemmin ottaen järjestämättömäksi (unordered) listaksi, jonka muotoilua on mahdollista säädellä CSS-kielellä. Rakenteen ryhmittelyyn käytin lohkotason elementtejä (div), joilla voidaan muodostaa tietynlaisia kokonaisuuksia dokumentin eri osista. CSS-tyylitiedostoihin tulivat kaikenlaiset väri-, tekstikoon, reunaviivojen, taustakuvien, leveyksien, kellutusten, marginaalien, sisennysten ja näkyvyyksien määrittelyt.

Sivuston oikeaan laitaan sekä alalaitaan jäi ohut reunusalue, joka tulee esille selaimessa siksi, ettei dokumentin body-elementille oltu määritelty marginaalien nollausta niihin kohtiin. Tähän ei oikeastaan ollut mitään muuta syytä, kuin se, että reunoihin sain näkyviin ääriviivat, jotka omalla tavallaan kehystivät kokonaisuutta. Nyttemmin ottaisin tämän reunamarginaalin pois, koska se on tarkemmin ajateltuna melko turha seikka.

Sivuston taitto on melko perinteinen, jakautuen pääpiirteittäin ylätunnistealueeseen, kolmeen palstaan ja alatunnistealueeseen. Ylimmän tason navigointilinkit sijoitettiin ylätunnistelohkon alaosaan riviksi. Alemman tason linkit sijoitettiin omaksi rivikseen ylemmän tason linkkien alapuolelle sekä alimman tason linkit sivuston vasempaan palstaan. Keskipalsta oli tarkoitettu pääasialliselle sisällölle ja oikean puolen palsta lähinnä yhteistyökumppaneiden logoille. Palstat on rakennettu sisällytetyistä ja kellutetuista lohkotason elementeistä. Navigoinnissa oli hieman otettu mallia silloisesta näkymästä sivustossa <http://www.stopdesign.com>, joka myös perustui XHTML:n Strict-muotoon. Sen lisäksi koin, että valitsemani ratkaisu navigoinnissa ja ulkoasullisessa kaavassa (banner, kolme palsta ja footer-alueet) mukailee monien muidenkin sivustojen käytäntöjä ja yleisiä käyttöliittymäkonventioita.

4.2 Pdf-dokumenttien käyttö

Joskus web-sivuilla törmää siihen, että jokin linkki aukaiseekin esimerkiksi Adobe Reader-dokumentin (pdf) toisen web-sivun sijaan. Tämä on tietenkin käytettävyyden kannalta hyvä ilmoittaa käyttäjälle vaikkapa sijoittamalla linkin viereen vihje aukeavan dokumentin muodosta. SM-Mejä 2006 -sivuston etusivulle tuli jossain vaiheessa linkitys tiedotettavan tapahtuman käsiohjelmaan, joka oli julkaistu pdf-muotoisena. Siihen aikaan pdf-dokumentit avautuivat aina omaan ohjelmaikkunaansa (aiemmin ohjelma oli saatavilla nimellä Acrobat Reader). Tämän raportin kirjoittamisen aikaan Adobe Reader on edennyt versioon 11 ja nykyisin pdf-dokumentit on mahdollista ladata ja aukaista suoraan selainikkunaan katseltavaksi (ja tallennettavaksi sitä kautta). Selainikkunaan avaamisen voi kuitenkin ottaa halutessaan pois päältä Adobe Readerin asetuksista. Käytet-

tävyyden kannalta suoraan selainikkunaan aukeaminen on mielestäni hyvä käytäntö – se tapahtuu nopeammin kuin käynnistämällä lukuohjelma ja selaimessa näkyy myös latautumisen edistyminen graafisena palkkina ja numeraalisena tietona.

4.3 Sivunsisäiset linkit

Tietyillä sivuilla SM-Mejä 2006 -sivustolla käytin sisällysluettelomaisesti otsikko-linkkejä saman sivun sisällä liikkumiseen, eli linkit oli kohdennettu niin sanottuihin ankkureihin. Esimerkiksi käytettävyydsiantuntija Jakob Nielsenin suositukseen kuuluu sivunsisäisten linkkien käytön välttäminen, koska yleensä hyperlinkien odotetaan aukaisevan uuden sivun ja täten sivunsisäiset linkit voisivat aiheuttaa hämmennystä käyttäjässä (Nielsen 2006).

Ongelmia voi aiheuttaa esimerkiksi se, ettei hahmoteta, minne linkistä hypättiin (mitä sisältöjen sisällä pitäisi oikeastaan tarkastella) tai sitten se, että käyttäjä luulee ladanneensa uuden sivun, jolloin klikatessaan ”takaisin”-painiketta käyttäjä pääsee vain edelliseen kohtaan samalla sivulla edellisen sivun sijaan. Yksi tekijä tämänkaltaisissa käytettävyysongelmissa voi olla se, ettei käyttäjä näe itse sivun vierittymistä tai huomaa että vierityspalkki on liikkunut – mikäli tätä vahvennetaan visuaalisesti, eikö silloin ”takaisin”-painikkeen käyttäytyminenkin ole odotetumpaa? (Lawson 2007.)

CSS-kielen tämänhetkisellä versiolla (CSS3) pystytään visuaalisesti vahventamaan tarkasteltavia kohde-elementtejä pseudoluokkien avulla ja täten tuomaan paremmin esille sivunsisäisten linkkien kohteita käyttäjälle. Kyseinen kohteita käsittelevä pseudoluokka on nimeltään `”:target”`. Käytännössä tämä voi toimia esimerkiksi niin, että käyttäjän klikatessa sivunsisäistä linkkiä, kohteena oleva tekstikappale muuttuu taustaväriältään.

Sivun sisällä liikkumisen huomaamattomuuteen, eli selaimen ”pikahyppäykseen” on olemassa ratkaisu JavaScriptin keinoin – jo vuonna 2003 Stuart Langridge kirjoitti skriptin nimeltä Smoothscroll, jonka avulla linkin ja kohteen välistä

”hyppäystä” voidaan niin sanotusti pehmentää ja tehdä näkyvämmäksi käyttäjälle. Edellä esitetyt käytettävyyden ehostuskeinot vaativat selaintukea CSS3:lle ja JavaScriptille, mutta niiden tuen puuttuminen ei kuitenkaan sen enempää heikennä sivujen katselukokemusta siinä mielessä, mihin aiemmin on totuttu. (Lawson 2007.)

4.4 Sivuston skaalautuvuus

SM-Mejä 2006 -sivuston palstojen leveyksiin määriteltiin prosentuaalisten arvojen lisäksi tietty maksimileveys. Hain tällä ratkaisulla sitä, että vaikka palstan leveys olisikin alustavasti mukautuva selaimen kokovaihteluissa, säilyisi tekstipalstojen luettavuus hyvänä maksimiarvojen avulla. Tänä päivänä jättäisin kuitenkin tällaiset maksimileveyksien pikseliarvot kokonaan pois suunnittelusta, ja tekisin palstojen leveydet täysin prosenttiarvojen avulla, sillä kappaleleveyksiin vaikuttaa varsin moni asia – fonttikoko, ympäröivät marginaalit, sisennykset ja vierekkäiset muut (lohko)elementit. Jotakuinkin SM-Mejä 2006 -sivustosta tulikin oikeastaan ulkoasullisesti skaalautuvan ja kiinteän suunnittelun sekoitus, joten pyrkisin nykyisin parempaan skaalautuvuuteen ja tekisin enemmän töitä mobiililaitteiden huomiointin kannalta. Mobiililaitteista puhutaan myöhemmin tässä raportissa.

Vielä 2000-luvun alkupuolella voitiin pitää tärkeänä, että web-sivuja pystyttäisiin lukemaan ymmärrettävästi myös ilman tyylitiedostoja ja täten huomioiden vanhojen selainversioiden käyttäjät, tai tilanteet jolloin tyylitiedostot otettaisiin pois käytöstä. Otin jonkinasteisesti huomioon sivujen luettavuuden ilman tyylitiedostoja SM-Mejä 2006 -sivustoa tehdessä ja myös testailin sivuja ottaen tyylitiedostojen linkityksen pois päältä. Nykyisin olisi kuitenkin tärkeämpää toteuttaa sivusto teknisesti niin, että suunniteltaisiin ulkoasun skaalautuvuus sopivaksi ja elementit mukautuvaksi joko kooltaan tai näkyvyydeltään eri laitteita ja näyttökokoja varten.

Mukautuvaa verkkosivusuunnittelua tuetaan nykyisin myös ohjelmallisesti; esimerkiksi Adobe Edge Reflow-ohjelma on tarkoitettu erikokoisille näytöille opti-

moitujen verkkosivujen tekemiseen hyödyntäen HTML5:ttä ja CSS-tyylejä. Ennen julkaisua tekijä voi tarkastella, miltä sivut näyttävät ohjelman lisäosalla nimeltä Adobe Edge Inspect. Adoben uudet ohjelmat on saatavilla yksinoikeudella vain Adobe Creative Cloud-pilvipalvelun jäsenille. (Stubin 2013.)

4.5 Kuvagalleria

SM-Mejä 2006 -sivustolle tuli kuvagalleria, jossa hyödynsin JavaScriptiä toteutuksessa, jossa pieniä esikatselukuvia klikkaamalla suurempi kuva avautui niin sanottuun popup-ikkunaan, joka on käytännössä käyttöliittymäelementtien osalta karsittu ja kooltaan tiettyyn kokoon määritelty uusi selainikkuna. Tämä hie- man vanhanaikainen ratkaisu vaati sen, että jokainen normaalikokoinen kuva tarvitsi oman HTML-sivunsa, jota klikatessa kutsuttiin näkyviin. Nykyisin on saatavilla paljon mukavampia JavaScript-toteutustapoja sellaisen gallerian tekoon, jossa pieniä kuvia klikkaamalla suurempi versio aukeaa näkyville. Yksi tällainen on helposti käyttöön otettava, Lokesh Dhakaran kirjoittama Lightbox, josta on tämän raportin kirjoitushetkellä ladattavissa versio 2.51. Lightbox hyödyntää tekniikassaan jQuery-rakennetta, joka on monipuolinen JavaScript-kirjasto. Lightboxia käyttäen ei tarvitse tehdä ylimääräisiä HTML-sivuja ja tiedostonhallinta on selkeämpää.

4.6 Kirjasinlajit ja -koot

Aiemmin käyttäjän työasemaan asennetut fontit saattoivat vaikuttaa enemmänkin yhtenä web-sivujen ulkoasuun vaikuttavina tekijöinä; tyylitiedostoihin määriteltiin pieneksi listaksi muutama sopiva fontti, jotka katsottiin sopivan ainakin muutama mahdolliseen tilanteeseen (listalla ensimmäinen on ensisijaisesti käytössä, mutta jos sitä ei ole käyttäjällä asennettuna, otetaan seuraava käyttöön ja niin edelleen). Nykyään on yhä enemmän havaittavissa käytäntö, jossa palvelimelle viedään halutut fontit ja niin sanotusti sisällytetään ne CSS-tyyliohjeistuksiin mukaan, jolloin web-sivujen suunnittelu on kirjasimien kohdalla vähemmän rajoitettua. Käyttöön otettavia kirjasimia löytyy myös ”pilvestä” eli

niitä voidaan ottaa suoraan käyttöön eri palvelun tarjoajilta. Esimerkiksi Google tarjoaa vapaasti käytettäviä web-fontteja web-käyttöön optimoituna osoitteessa <http://www.google.com/fonts/>.

Tyylitiedostoon annettiin entisaikaan muutamia vaihtoehtoja kirjasinlajiksi, jolloin selain valitsi luettelosta ensimmäisen sellaisen, joka järjestelmään oli asennettu. Jos jotkut tekstinpätkät haluttiin esittää jollain erikoisemmalla fontilla, yksi keino sen näkyvyyden varmistamiseksi oli laittaa se kuvana. Tällainen keino ei kuitenkaan tue semanttisuutta eikä ulkoasun skaalautuvuutta. Sisällytettyjen fonttien käyttö on tätä nykyä auttanut suunnittelijaa säilyttämään halutunlaiset kirjasinlajit, jotka ladataan siis palvelimelta, eivätkä ne korvaudu tai riipu käyttäjän koneelle asennetuista fonteista.

Skaalautuvan ulkoasun suunnittelussa tulisi huomioida se, että jos käyttäjä suurentaa tai pienentää valitun kirjasimen kokoa tehdäkseen siitä itselleen luettavuudeltaan paremman, taitossa kokonaisuus tulisi säilyä kasassa. Entisaikaan fonttikoko määriteltiin aina pikseleissä, koska sen avulla suunnittelijat pystyivät määrittämään tarkan koon. Sittemmin alettiin kiinnittää enemmän huomiota saavutettavuuteen, jolloin suhteelliset tekstikoot (määriteltynä prosentein tai em-arvoilla) tulivat suosioon. Sen jälkeen niin heikkonäköiset kuin kuka tahansa muukin pystyi muuttamaan selaimessaan oletustekstikoon, tai lennossa käyttäen näppäinkomentoa Ctrl+ / Ctrl- (Windows) ja Command+ / Command- (Mac).

Vielä muutamia vuosia sitten edellä mainitulla keinolla teksti suureni tai pieneni siten, että sivun muotoilu säilyi ennallaan. Nykyään kyseisen toiminnon kautta käytetyimpien selaimien oletusarvoinen toiminta on sivun – ei tekstin – suuren-taminen/pienentäminen, jolloin siis koko sivu muotoiluineen ja graafisine elementteineen suurenee tai pienenee. Täten tekstikoon määrittäminen joko kiinteäksi tai suhteelliseksi on käyttäjän toimesta mitätöitävissä. Voisiko siis olla, että valinta suhteellisten yksiköiden ja kiinteiden arvojen välillä on vain mukavuuskysymys? Kenties valinta tulee tehdä projektikohtaisesti, ja näiden vaihtoehtojen käytännöllisyydestä voidaan myös olla kahta eri mieltä. Sivun suuren-tamis- ja pienentämistoiminnon huono puoli on se, että se miltei aina vaatii vaakasuorassa tapahtuvaa vierittämistä selaimessa, joka puolestaan heikentää

sivujen käytettävyyttä. Halutessaan esimerkiksi nykyaikaisessa Mozilla Firefox-selaimessa tämä oletusarvoinen sivun zoomaus voidaan vaihtaa tekstin zoomaukseksi selaimen omista asetuksista. (Moll 2009.)

Joskus suhteellisten mittayksiköiden käyttö kirjasinten kokomäärittelyissä voi olla suunnittelijan kannalta hyvä idea varsinkin silloin, jos kokojen suhteellisuus toisiinsa nähden on tärkeää. Teknisessä mielessä suhteellisten kokojen käyttö antaa joustavuutta muutoksille; jos esimerkiksi designissa otsikko H1 on kolme kertaa suurempi kuin runkoteksti, kiinteitä arvoja käytettäessä tämä voisi olla runkotekstin kohdalla 12px ja H1:sen kohdalla 36px. Jos myöhemmin runkotekstiä tulisi suurentaa vaikkapa yhden pikselin verran (13px), otsikko H1 ei olisi sikaan enää kolme kertaa suurempi, jolloin sitä pitäisi muuttaa myös (39px). Tällöin yksi pieni muutos vaatisi monia uusintalaskelmia. Tähän verrattuna, asettaessa rungon tekstille esimerkiksi koko 0.8em (noin 12px useimmissa pöytäkoneselaimissa) ja H1:lle 300%, H2:lle 200% ja niin edelleen, runkotekstiä muuttaessa designissa säilyy eheys suhteellisia kokoja käytettäessä. (McLellan 2009.)

4.7 Kellutukseen perustuvan layoutin ongelma

Silloin, kun lohkotason (div) elementtejä kellutetaan (float), sijoitetaan ne monesti säiliömäisesti yhden container-elementin sisälle, joka myös itsessään on div-elementti. Tällöin kellutettujen elementtien ympärille voidaan tehdä vaikkapa reunustusviivat määrittelemällä containeriin border-ominaisuudet, tai esimerkiksi taustaväri. Yksi ongelma tässä on ollut se, että selaimia on täytynyt käskytää "venyttämään" kyseinen container-elementti siten, että se ylittää kellutettujen elementtien ympärille.

Jos container-elementille määriteltäisiin korkeusarvon, pitäisi silloin myös tietää, kuinka korkeat sen sisällä olevat elementit tulevat olemaan, ja silloinhan tämä johtaisi kiinteään suunnitteluun. Yksi vanha ratkaisu on ollut sellainen, että kellutettujen elementtien lisäksi containeriin on sisällytetty vielä yksi ylimääräinen elementti, johon on määriteltynä "clear: both"-ominaisuudet, jolloin containeriin

on siis tullut yksi kelluttamaton elementti ja container venyttänyt itsensä sen mukaisesti. (Koch 2013.)

SM-Mejä 2006 -sivustoa suunnitellessa itselläni oli myös tämänkaltainen ongelma containerin kanssa, ja ratkaisin sen sijoittamalla yhden ylimääräisen elementin "clear: both"-ominaisuuksin. HTML:ssä tämä sijoitettiin containerin loppuun. Myös kyseisessä sivustossa containerin tarkoituksena oli pitää sisällään kellutetut, palstoiksi tarkoitetut elementit, näyttää oikeassa ja alalaidassa reunaviiva sekä näyttää myös tietty taustaväri. Ilman ylimääräistä kelluttamatonta elementtiä container saattoi niin sanotusti loppua kesken arvaamattoman korkeuksena.

Uudempi ratkaisu tähän ongelmaan on lisätä containeriin pari lisäominaisuutta; "overflow" sekä "width". Overflow-määrite voi olla "auto", "hidden" tai "scroll". Jälkimmäisin tuo esille vierityspalkit, vaikkei niitä tarvittaisikaan. Selaimista ainakin Internet Explorer ja Opera tarvitsevat välttämättä joko leveys- tai korkeusmääritteen. Laittamalla leveyden arvoksi 100 % on hyvä lähtökohta, mutta monimutkaisemmat layoutit voivat tarvita muunlaisia arvoja. Mac-koneiden Exploreria varten "overflow"-ominaisuudelle voidaan laittaa määrite "hidden", sillä muuten se näyttää aina vierityspalkit. Muissakin tapauksissa varmistaakseen, ettei vierityspalkkeja näytetä, voidaan lähtökohtaisesti laittaa ominaisuudelle "overflow" arvo "hidden". Kuvatun ratkaisun avulla ulkoasua varten ei enää tarvita ylimääräisiä elementtejä. (Koch 2013.)

4.8 Mobiililaitteet

Sivustoja suunnitellessa voidaan CSS:llä tehdä monia eri tyyliohjeistustiedostoja eri medioita varten, kuten tulostin ja mobiililaitteet. SM-Mejä 2006 -sivustolle määrittelin oman, handheld-medioille sopivan CSS-tyyliohjeistuksen, mutta tänä päivänä kyseinen handheld-median rajausta toimii vain vanhojen puhelinten nettiselaimissa. Sivuston suunnitteluajanaan kyseeseen saattoi tulla esimerkiksi sen ajan versio Opera Mini-selaimesta. Eri tyyliohjeistuksien linkitys sivustolle ja määrittelyt tiettyä mediaa varten tapahtuu HTML-sivun HEAD-osiossa.

Eri medioille tarkoitetut tyyliohjeet voidaan määritellä myös yhteen ja samaan CSS-tiedostoonkin @media-säännöillä (esimerkiksi "@media print { // elementit ja ominaisuudet }"). CSS3 mahdollistaa myös sisäkkäiset @media-säännöt. Kaikki suurimmat selainohjelmat käyttävät "screen" mediaa oletusarvoisesti ja suurin osa niistä tukee myös "print"-mediamäärittä silloin kun sivu tulostetaan. "Handheld"-mediamäärite on käytössä vain joissakin kannettavissa laitteissa ja pienikokoisissa näytöissä. Sitä tukee muun muassa Opera ja osittain Pocket Internet Explorer, mutta käyttäjä saattaa joutua ottamaan se manuaalisesti käyttöön, sillä selaimien oletuksena voi silti olla "screen" media. (Wilton-Jones 2011.)

Tänä päivänä mobiililaitteiden kohdalla puhutaan enemmänkin älypuhelimista ja tablet-tietokoneista, joten CSS-suunnittelussa tarvitaankin hieman laajempia käytäntöjä silloin, kun halutaan ottaa huomioon esimerkiksi iPhone, iPad, Blackberry-puhelimet, Blackberry Playbook ja muut tabletit, Android-puhelimet ja Android-tabletit kuten Motorola Xoom ja Samsung Galaxy.

SM-Mejä 2006 -sivuston kannettavien laitteiden tyyliohjeistus oli määritelty pelkästään "handheld"-mediamääritteellä, joten varsinkin nykyisin suurin osa mobiililaitteiden selaimista jättää sen täysin huomiotta, ja esimerkiksi testatessa niin sanotulla peruspuhelimella kuten Nokian C2-01 ja älypuhelimella, kuten iPhone 4:lla (liite 2) selaimet näyttävät sivuston "normaalisti", pöytäkoneselaimien tyyliohjeilla (ainakin oletusasetuksilla). Näin ollen lukeakseen sivuston sisältöjä, voi käyttäjä joutua suurentamaan selainäkymää tarkasteltavien kohteiden osalta. Jos opinnäytteenä tekemäni sivusto pitäisi suunnitella tänä päivänä, ottaisin mobiililaitteet paremmin huomioon jo siksi, että niiden käyttö netin selaamisessa on yleistynyt niin paljon. Mobiililaitteiden kohdalla erilaisia näyttökojoja ja resoluutioita on aikamoinen määrä, ja siksi ulkoasun skaalautuvuus olisiikin yhä tärkeämmässä asemassa.

Ulkoasultaan skaalautuvan web-sivuston tekemisessä ensimmäisiä askeleita on lisätä sivun HEAD-osioon seuraavankaltainen rivi:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=2.0, user-scalable=yes" />
```

Tämän tagin avulla (mobiili)laitteelle voidaan saada tiettyjä näyttötiloja aikaiseksi. Seuraava askel on lisätä linkitykset tyyliohjeisiin eli CSS-tiedostoihin:

```
<link media="Screen" href="styles.css" type="text/css"
rel="stylesheet" />
```

ja

```
<link media ="handheld, only screen and (max-width: 480px),
only screen and (max-device-width: 480px)"
href="mobile.css" type="text/css" rel="stylesheet" />
```

sekä lisäksi

```
<!--[if IEMobile]>
<link rel="stylesheet" type="text/css" href="mobile.css"
media="screen" />
<![endif]-->
```

Ensimmäiset rivit liittyvät oletusarvoiseen tyyliohjeistukseen pöytäasemien selaimille. Toisena olevien rivien sisällä määritellyssä "mobile.css"-tiedostossa ylikirjoitetaan tarvittavat asetukset ja määrittelyt, jotka ovat oletusarvoisessa tyyliohjeistuksessa. Vaikka komennoissa on näkyvillä maksimileveys 480px, toimii tämä tyyliohjeistus silti esimerkiksi iPhone 4-puhelimessa, jonka todellinen leveys on 640px. "Max-device-width"-ominaisuuden lisäksi käytetään "max-width"-ominaisuutta, koska jotkut Android-laitteet eivät lataa mobiilityylejä pelkästään ensimmäistä komentoa käyttäen. (Baker 2011.)

Oletusarvoisen tyyliohjeistuksen lisukkeeksi tehtävän toisen tyyliohjeistuksen ("mobile.css") sijaan voidaan myös sisällyttää tyyliohjeet yhteen ja samaan tiedostoon käyttäen @media-sääntöjä. Kolmas lisättävä rivi, joka alkaa if-lauseella, on

tarkoitettu Windows Phone (7)-laitteiden Internet Exploreria varten, joka ei tue CSS3:n ”max-width” ja ”max-device-width”-ominaisuuksia. (Baker 2011.)

On ennustettu, että kuluvan vuosikymmenen aikana mobiilien laajakaistaliittymien määrä kymmenkertaistuu. Samalla yhden käyttäjän siirtämä tietomäärä satakertaistuu. Vuoteen 2020 mennessä mobiilin tietoliikenteen määrä tuhatkertaistuu yltäen gigatavuun päivässä käyttäjää kohti. Tällä hetkellä internetliikenteestä neljä prosenttia liikkuu matkapuhelinverkossa, vuonna 2020 yli puolet. (Leidenius 2012.)

Vuodesta 2005 lähtien mobiilin tietoliikenteen käyttö on kaksinkertaistunut vuosittain (nykyisin ollen noin eksatavun verran vuodessa). Vuoteen 2020 mennessä mobiilin dataliikenteen osuus koko internetin tuhannen eksatavun liikenteestä on 60 prosenttia. Täten operaattoreilta vaaditaan nopeampia verkkoja, älykkäämpää tekniikkaa ja mahdollisuutta priorisoida eri maksuluokan asiakkaita. (Leidenius 2012.)

5 Lopputulokset

Näyttäisi siltä, että vaikka tämän hetken suosituimmat selaimet noudattavatkin yleisimpiä standardeja, joudutaan silti ottamaan huomioon, miten eri selaimet ja selainversiot tukevat kutakin ominaisuutta (vai tukevatko ollenkaan) – varsinkin uusien tekniikoiden ilmestyessä täytyy tarkkaan miettiä, mitä ominaisuuksia kannattaa ottaa suunnitteluun mukaan, sillä jokainen vähänkin erikoisempi ratkaisu varmasti vaatii eri selainten ja selainversioiden huomioimista ja sen kautta monenlaisia toimivuuden lisävarmistuksia.

Tällä hetkellä voidaan vielä tehdä ihan hyvin samoihin standardeihin pohjautuvilla säännöillä kuin vuonna 2006, eli XHTML Strict (1.0 tai siitä ylöspäin) ja CSS(2), ainakin niin sanotusti normaaleja web-sivuja. HTML5 ja CSS3 voivat tulla kyseeseen siinä vaiheessa, kun halutaan tehdä käyttöjärjestelmäriippumattomia ja selainpohjaisia sovelluksia vaikkapa mobiilimaailmaan. SM-Mejä 2006 -

sivuston tekeminen silloisilla web-standardeilla varmisti sen, että sivut toimivat tänäkin päivänä pääasiassa samalla tavalla kuin aikoinaan ja näyttävät pitkälti samalta kaikissa käytetyimmissä selaimissa (liite 6).

Kohti HTML:n versioon 5 ja CSS:n versioon 3 mennessämme joku suunnittelija on saattanut jopa ottaa askeleen taaksepäin tehden sivuja HTML 4.01 Strict-dokumenttityypillä, sillä vaikka XHTML:stä povattiin webin tulevaisuutta ja siten XML:n pitäisi olla tätä päivää, näytti ainakin vielä XHTML version 2 kehityksen aikaan siltä, etteivät XHTML:n uudet versiot oikeastaan edenneet mihinkään kunnolla julkaistaviksi web-standardeiksi. (Moll 2009.)

5.1 Validointi

Testasin SM-Mejä 2006 -sivuston etusivun standardinmukaisuuden validoimista W3C-organisaation nettisivuilla olevilla työkaluilla CSS2:n kohdalla sekä XHTML Strict 1.0 –merkkauskielen osalta. Työkalut löytyvät osoitteista <http://jigsaw.w3.org/css-validator/> ja <http://validator.w3.org/>. Tämän lisäksi ajoin sivun W3C:n ”mobileOK Checker”-palvelun läpi, joka tarkistaa, onko web-sivut mobiiliystävällisiä. Tämän pystyi tekemään osoitteessa validator.w3.org/mobile.

Validoinnissa tuli joitain virheitä esille: CSS:n osalta yksi Internet Explorer-selainta varten tehty yksittäinen ilmaisu ja sen lisäksi tulostusta varten tehdyssä tyyliohjeistuksessa oleva pilkku desimaaliluvussa (tulisi olla piste). XHTML-merkkauskielen osalta yksi virhe esiintyi sivun aloittavassa html-tagissa, josta puuttui ”xmlns”-attribuutti, joka on pakollinen kaikissa dokumenttityypeissä, jotka perustuvat XML:ään.

W3C:n ”mobileOK Checker” antoi pitkähkön listan epäkohdista tarkistettaessa SM-Mejä 2006 -sivuston etusivun mobiiliystävällisyyttä. Vakavuusasteen mukaisesti jaoteltuja virheitä oli seuraavasti: 1 kriittinen, 3 vakavaa, 3 keskivakavaa ja 3 matalaa. Kriittisin virhe oli ladattavien tiedostojen yhteiskoko: 257.3KB (HTML-sivut, kuvatiedostot ja tyylitiedostot). Ohjelman ilmoittamista vakavista virheistä ensimmäinen oli se, että dokumentti ei ole kunnollisesti kirjoitettu UTF-

8 merkistökoodauksella – tämä ei kuitenkaan ole virhe, sillä dokumenttiin oli määritelty ISO-8859-1 -merkistökoodaus. Toinen vakava virhe oli se, että sivulta löytyneessä taulukossa oli vähemmän kuin kaksi <tr>-elementtiä (taulukkoriviä) – täten taulukkoelementtiä on käytetty ulkoasuun eikä taulukkomaisesti esitettävän tiedon esittämiseen. Ohjelma varoittaa, että mobiiliselaimissa on paljon eroavaisuuksia taulukoiden renderöimisessä, joten niitä ei voi luotettavasti käyttää ulkoasun muotoiluissa. Olin käyttänyt sisällöissä yksirivistä taulukkoa yksittäisen kuvan ja tekstikappaleen asettamisessa vierekkäin. Kolmas vakavista virheistä oli se, että CSS-tyyliohjeistuksesta löytyi epävalidia kieltä.

Keskivakavien virheiden kohdalla ohjelma varoitti seuraavanlaisista asioista: dokumentissa ei ole välimuistitustietoja ("Expires" tai "Cache-Control"-otsikkotietoja). Välimuistituksen kontrolli voi vähentää mobiiliselainten verkossa käyntikertoja tehden sivujen lataamisesta nopeampaa ja halvempaa. Välimuistitukseen voidaan kuitenkin parhaiten vaikuttaa webserverin (esimerkiksi Apache, Microsoft IIS) HTTP headerin asetuksilla tai palvelinpuolen skriptauksilla (esimerkiksi PHP, CGI). HTTP header-tiedot ovat ydinosa HTTP-protokollassa selaimen ja palvelimen välisissä pyyntö- ja vastausviesteissä. HTML-dokumentin <head>-osioon voidaan sijoitella välimuistitusta koskevia ohjeita <meta>-elementin avulla, mutta tämä ei ole kovin tehokas keino, sillä kaikki välityspalvelimet sivuston ja käyttäjän välillä jättävät ohjeet huomiotta. Kahden muun vakavan virheen kohdalla testiohjelma herjasi HTTP headerin puutteellisista tai virheellisistä merkistökoodaukseen liittyvistä tiedoista. (Nottingham 2012.)

Matalan tason mobiiliystävällisyyden virheistä ensimmäinen oli, että absoluuttisten arvojen käyttäminen CSS:ssä johtaa layouteihin, jotka eivät mukaudu näyttökokoihin. Joidenkin asioiden, kuten maksimileveyksien määrittelyihin olin käyttänyt tarkkoja pikseliarvoja, jotka oikeastaan vähentävät sivuston todellista ulkoasun skaalautuvuutta. Toinen matalan tason virhe oli, että dokumentti ei validoidu XHTML Basic 1.1:sen tai MP 1.2:sen sääntöjen perusteella. XHTML Mobile Profile (MP) on Open Mobile Alliance-järjestön määrittelemä standardi mobiililaitteita varten. Sen on kuitenkin korvannut XHTML Basic 1.1, josta tuli yksi W3C:n virallisista suosituksista vuonna 2008. Kolmantena virheenä ohjelma kertoi, että kaikista kuvista ei löydy leveys- tai korkeusattribuutteja. Kaikkien

näiden edellä kuvattujen epäkohtien lisäksi testiohjelma listasi muutamia muita, pienempiä varoituksia ja suosituksia. (Wikipedia 2013.)

5.2 Simuloituja testauksia

Voidaan sanoa, että SM-Mejä 2006 -sivustoa ei ole optimoitu mobiililaitteita varten. Ongelmia voi tulla varsinkin pienikokoisilla näytöillä varustetuissa laitteissa. Tableteilla sivuston renderöityminen ei pahemmin kärsi, sillä näytön koko on riittävän suuri koko sivualan renderöitymisen kohdalla. Testasin sivuston näkyyttä virtuaalisella iPadilla osoitteessa ipadpeek.com (liite 3).

Nykyisin esimerkiksi Opera Mini-selaimessa käyttäjä voi vaihtaa asetuksista matkapuhelinnäkymän päälle, jolloin web-sivu kapenee ruudun asettamaan leveysrajoitukseen. Testasin SM-Mejä 2006 -sivustoa Opera Mini-simulaattorilla osoitteessa www.opera.com/fi/developer/opera-mini-simulator (liite 4) niin, että matkapuhelinnäkymä oli päällä. Tästä huomasin sen, että sisällöt olivat kyllä kohtuullisen hyvin luettavissa, vaikka graafinen ulkoasu ei mukaudu kovinkaan järkevästi.

Lopuksi kokeilin sivuston toimivuutta Android-emulaattorilla sivulla www.manyomo.com. Valitulla resoluutiolla (1 024 x 600, 160 dpi) ja Androidin versiolla 4.2 sivusto renderöityi onnistuneesti ja näkymä oli verrattavissa pöytä-koneiden selainnäkyymiin sivuston toimivuuden kohdalla (liite 5).

6 Pohdinta

Opinnäytetyössäni SM Mejä 2006 -projektin parissa kiinnitin sekä tutkimuksesani että sivuston tekemisessä huomiota muun muassa taulukottomaan layout-suunnitteluun, ulkoasun skaalautuvuuteen, XHTML Strict-muotoiseen standardiin ja rakenteesta eriytetyn CSS-tyyliohjeen käyttöön, vasteaikoihin, saavutettavuuteen ja käytettävyyteen, validoitumiseen ja toimivuuteen erilaisissa tarjolla

olevissa selaimissa. Tällaisten asioiden huomioimisen kautta esitin päästävän yleisesti ottaen kustannustehokkaaseen suunnitteluun, työmäärän vähentämiseen, nopeisiin latausaikoihin, erilaisten nettiselainten ja näyttökokojen käyttäjien tukemiseen, päivitysten tekemisen helpottamiseen sekä uudelleensuunnittelun helpottamiseen.

HTML-koodaushan vaikuttaa pääosin siihen, miten web-sivujen sisältämiä asioita näytetään käyttäjän koneella olevassa selaimessa, eli se toimii tavallaan ohjeistuksena selaimille. Täten HTML-sivut toimivat web-sovelluksen front-end -puolella, kun taas palvelimella voi olla sovelluksen back-end -puoli, mikäli ei ole kyseessä pelkistetyt ja staattiset HTML-sivut. Monimutkaisemmissa web-sovelluksissa latausaikoihin voi vaikuttaa verkon ja palvelimen teknisten ominaisuuksien lisäksi valitut tekniikat back-end -puolella. Tällöin niin sanotusti asiakaspuolen HTML-sivun rakenteen selkeys on vain yksi monista tekijöistä vasteajan kohdalla.

Standardinmukaisuus on hyvä opastus verkkosivusuunnittelua tekeville yhä edelleen. Se auttaa myös tekemään sivuista saavutettavat, jolloin esimerkiksi tekstin ääneen lukemisen tekniikkaa käyttävät henkilöt pystyvät hyödyntämään sisältöjä. Sen lisäksi se helpottaa hakukoneoptimointia, sivujen renderöitymistä eri selaimissa ja sivujen uudistamista uudenlaisten web-teknologioiden tiimoilla.

Selkeys on yksi asia, jota standardien käytön kohdalla yritetään parantaa tulevaisuudessa, ja tässä kohtaa varsinkin HTML5:n uudet ominaisuudet korostuvat. Viime vuosien saatossa vakaamman pohjan saavuttanut <div>-tagien käyttö HTML-sivun elementtien erottelussa saattaa tulevaisuudessa muuttua yhä kuvaavampaan ja rakennetta selkiyttävämpään merkintätapaan HTML-koodauksessa.

Vaikka SM-Mejä 2006 -projektiin liittyvä sivusto ei ollut täysin virheettömästi suunniteltu, tuli todettua, että sivut kuitenkin toimivat tänäkin päivänä suosituimmissa selaimissa testattuna oikein hyvin ja näyttävät vertailullisesti samalta niissä kaikissa (liite 6). Jos kyseiset sivut pitäisi tehdä tänä päivänä, eivät muu-

tokset ja lisäykset kohdistuisi niinkään standardeihin ja koodaustapaan, vaan enemmänkin toteutustapoihin ja mobiilituen parantamiseen.

Vaikka suosituimpien selaimien keskuudessa löytyy toisiinsa nähden samanlaisia selainominaisuuksia ja uusimpien standardien tukea, ovat ne kuitenkin teknisesti eroavaisia. Suunnitteluja tehdään nykyisin paljolti Chromea ja Firefoxia silmällä pitäen – Internet Explorerin käyttö ja suosio on selvästi tippunut muuttaman viime vuoden aikana. Selainohjelmistojen kilpailullisuus ei edelleenkään poista kokonaan niin sanottujen bugikorjailuiden tekemistä web-sivujen suunnittelussa, vaikka tehtäisiin W3C:n kehotuksien mukaisesti standardeja noudattaen.

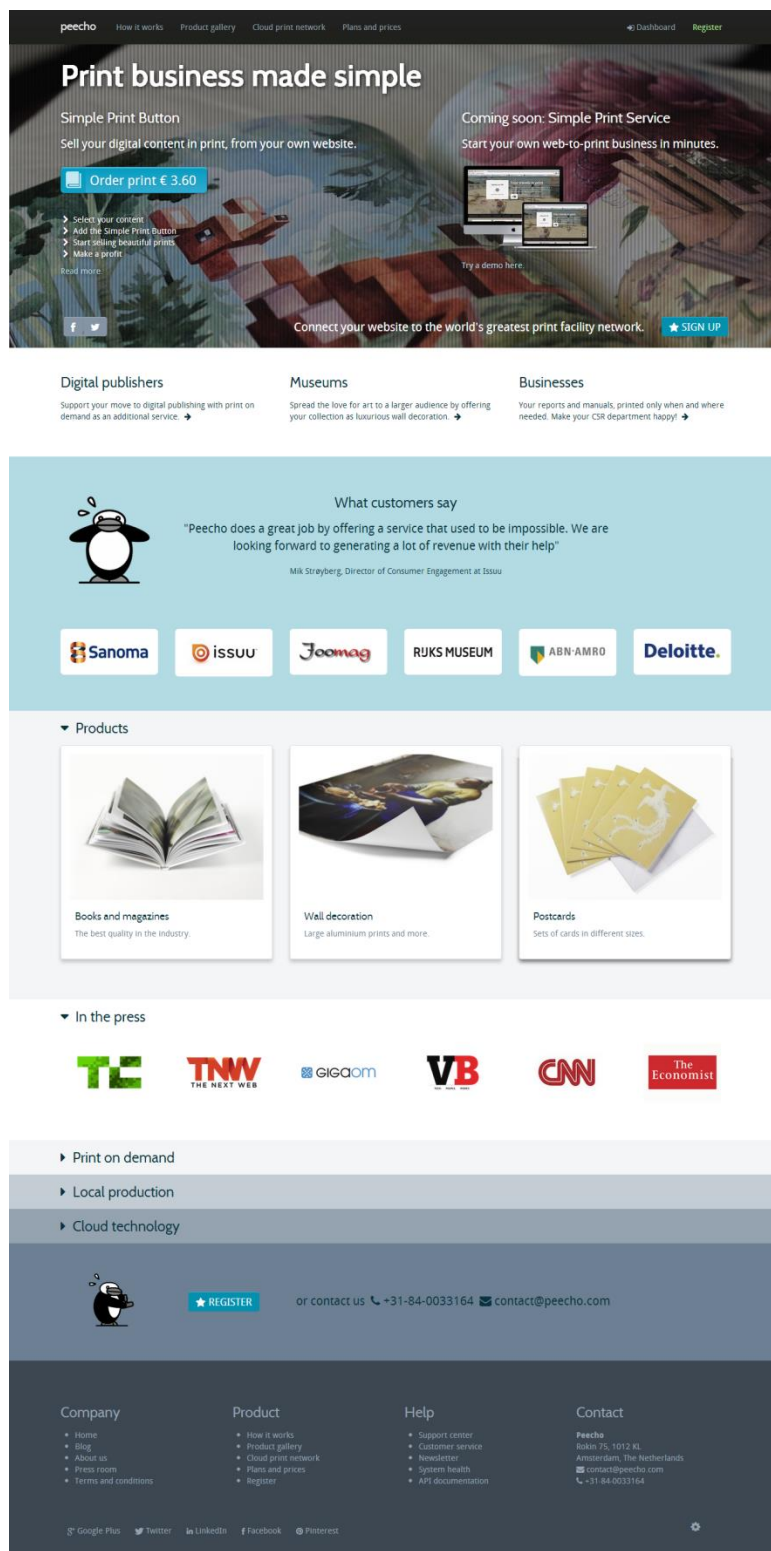
Älypuhelinien ja tablettien käyttö on tätä nykyä yhä kasvamassa, joten näkisin WWW-sivustojen toimivuuden parantamisen erilaisissa ympäristöissä tärkeäksi asiaksi. Tässä asiassa yhtenä suurena edesauttavana tekijänä on ulkoasun skaalautuvuuden suunnittelu, joka voi tuottaa enemmän suunnittelutyötä kuin kiinteiden sivujen. Samalla kun HTML5 voi lupaila alustariippumattomuuden kautta ajan ja rahan säästöä (etenkin mobiilisovelluksissa), voitaneen web-sovellusten eri näyttö- ja selainkokoihin sekä laiteympäristöihin skaalautumiseen panostamisessa tarvita periaatteessa enemmän aikaa ja rahaa kuin ennen. Koska mobiililaitteiden laajan kirjon lisäksi pöytäkoneiden näyttöjen kohdalla ollaan menossa kohti erikoisempien ja vaihtoehtoiltaan rikkaampien kuvaushteiden käyttöön, tulee ulkoasun skaalautuvuus yhä merkityksellisemmäksi web-sivujen suunnittelussa.

Lähteet

- Baker, Jeffrey. 2011. Mobile CSS.
<http://www.seabreezecomputers.com/tips/mobile-css.htm>. 10.4.2013.
- Järvinen, Petteri. 2012. Chrome ja Firefox kirivät, IE hyytyy. Tietokone 4, 57.
- Koch, Peter-Paul. 2013. Clearing floats.
<http://www.quirksmode.org/css/clearing.html>. 10.4.2013.
- Lawson, Bruce. 2007. Improving the usability of within-page links.
<http://dev.opera.com/articles/view/improving-the-usability-of-within-page-l/>. 3.4.2013.
- Leenheer, Niels. 2013. The HTML5 test. Sights.
<http://html5test.com/results/desktop.html>. 10.4.2013.
- Leidenius, Kim. 2012. Internetistä tulee liikkuva. Tietokone 4, 22.
- McLellan, Drew. 2009. The Fallacy of Page Zooming.
<http://allinthehead.com/retro/343/the-fallacy-of-page-zooming>. 10.4.2013.
- Moll, Cameron. 2009. Coding like it's 1999. 10.4.2013.
http://cameronmoll.com/archives/2009/06/coding_like_its_1999/.
- Nielsen, Jakob. 2006. Avoid Within-Page Links.
<http://www.nngroup.com/articles/avoid-within-page-links/>. 3.4.2013.
- Nottingham, Mark. 2012. Caching Tutorial. 10.4.2013.
http://www.mnot.net/cache_docs/.
- Saarelainen, Ari. 2013. Taistelu jatkuu: HTML5 vs. muu maailma. Tietokone 3, 28-31.
- StatCounter. 2013. Global market share held by the leading web browser versions as of March 2013.
<http://www.statista.com/statistics/158095/most-popular-internet-browsers/>. 17.4.2013.
- Stubin, Toni. 2013. Uudet tuotteet. Tietokone 3, 16.
- Taipale, Kai. 2013. 10 tulevan vuoden suuntausta. Tietokone 2, 16.
- W3Schools. 2013a. Browser Statistics.
http://www.w3schools.com/browsers/browsers_stats.asp. 10.4.2013.
- W3Schools. 2013b. Browser Display Statistics.
http://www.w3schools.com/browsers/browsers_display.asp. 10.4.2013.
- Wikipedia. 2013a. Web Standards.
http://en.wikipedia.org/wiki/Web_standards. 2.4.2013.
- Wikipedia. 2013b. HTML5.
<http://en.wikipedia.org/wiki/HTML5>. 2.4.2013.
- Wikipedia. 2013c. CSS.
<http://fi.wikipedia.org/wiki/CSS>. 2.4.2013.
- Wikipedia. 2013. XHTML Mobile Profile.
https://en.wikipedia.org/wiki/XHTML_Mobile_Profile. 10.4.2013.
- Wilton-Jones, Mark. 2011. CSS tutorial - Media types.
<http://www.howtcreate.co.uk/tutorials/css/mediatypes>. 10.4.2013.

Esimerkki web-sivuston ulkoasun skaalautuvuudesta

Kuvassa www.peecho.com-sivuston etusivu resoluutiolla 1 366 x 768 px.



Esimerkki web-sivuston ulkoasun skaalautuvuudesta

Kuva www.peecho.com-sivuston etusivun yläosasta resoluutiolla 480 x 800 px.



Digital publishers

Support your move to digital publishing with print on demand as an additional service. →

Museums

Spread the love for art to a larger audience by offering your collection as luxurious wall decoration. →

Businesses

Your reports and manuals, printed only when and where needed. Make your CSR department happy! →

What customers say

"Peecho does a great job by offering a service that used to be impossible. We are looking forward to generating a lot of revenue with their help"

Mik Strøyberg, Director of Consumer Engagement at Issuu

▼ Products



Books and magazines

The best quality in the industry.



Wall decoration

Large aluminium prints and more.



Postcards

Sets of cards in different sizes.

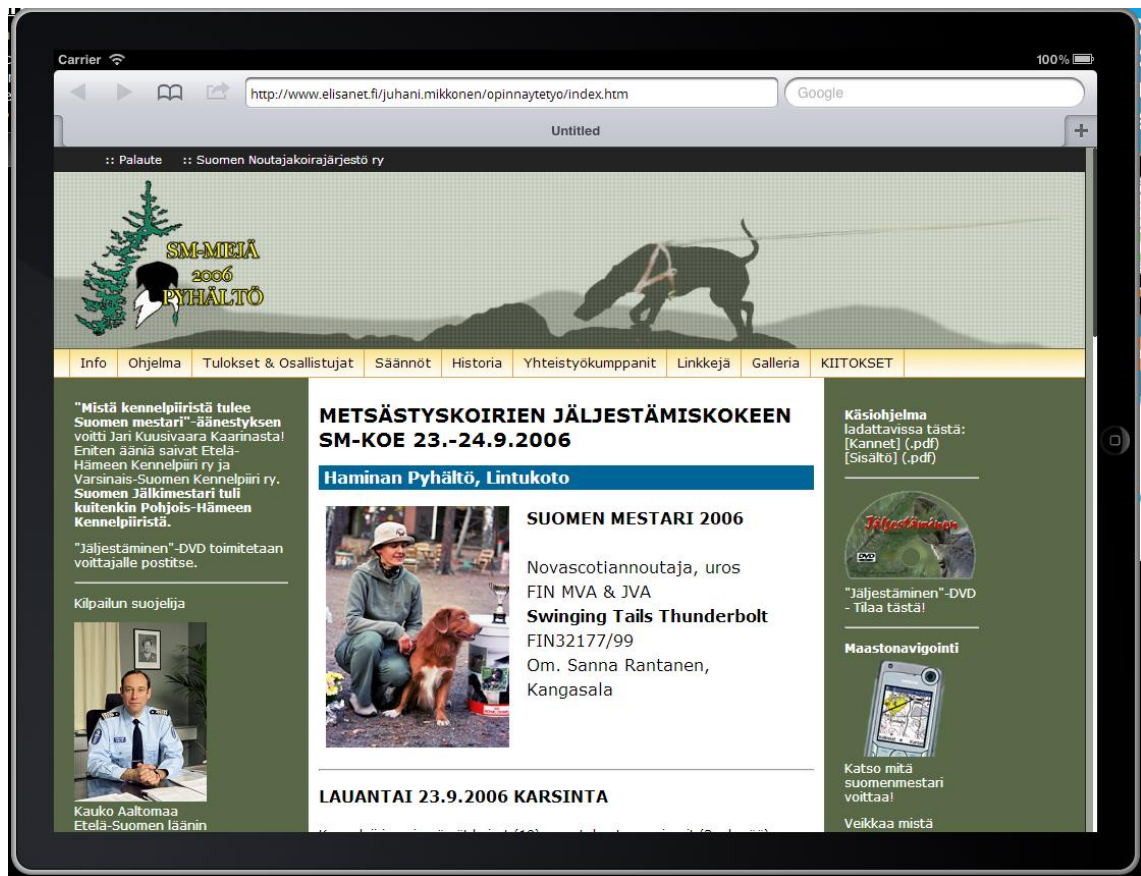
SM-Mejä 2006 -sivuston etusivu iPhone 4:llä testattuna

Laite: iPhone 4 (16GB kovalevy), malli MC603KS.

Näyttö: LED-taustavalaistu TFT LCD kosketusnäyttö, 326 ppi:n pikselitiheys (pixels per inch), 3.5 tuumaa, resoluutio 960 x 640.



SM-Mejä 2006 -sivuston etusivun tarkastelu iPad Peek-ohjelmalla



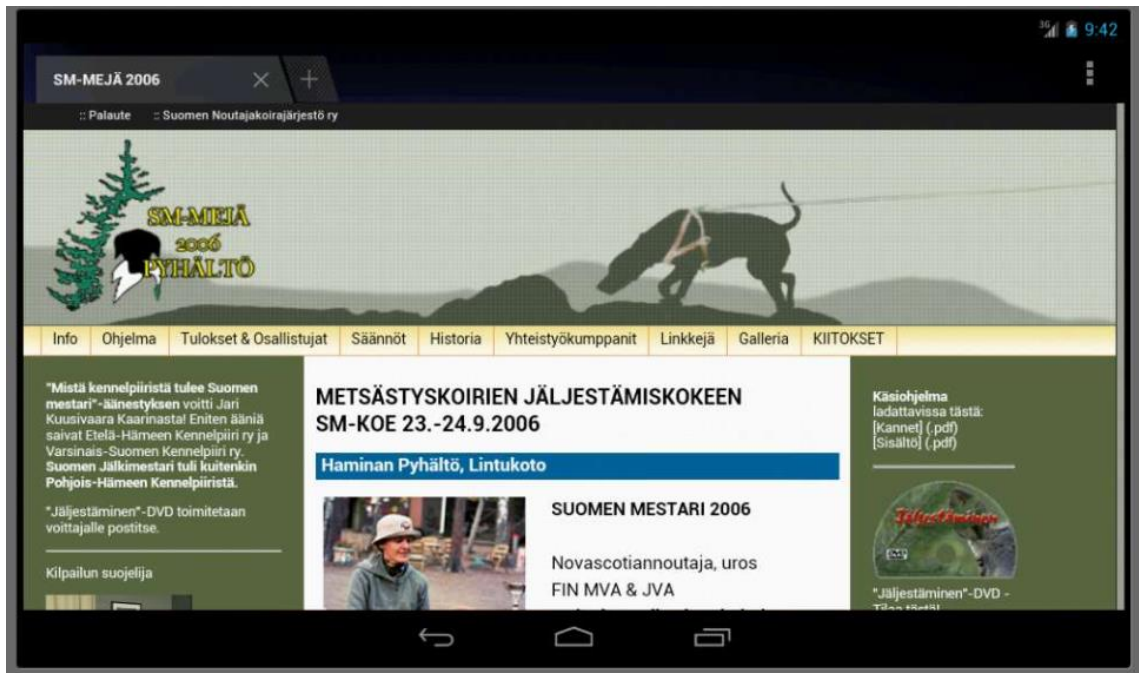
SM-Mejä 2006 -sivuston tarkastelu Opera Mini-emulaattorilla



SM-Mejä 2006 -sivuston tarkastelu Android-emulaattorilla

Valittu resoluutio: 1 024 x 600 (160 dpi)

Valittu Android-versio: 4.2

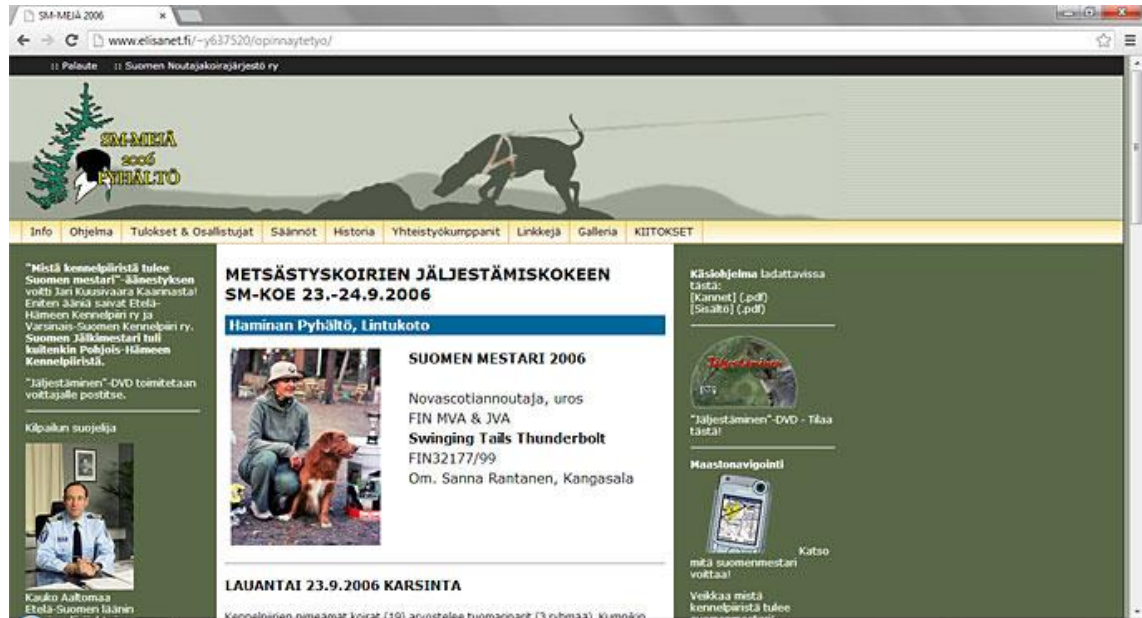


SM-Mejä 2006 -sivuston tarkastelu nykyajan web-selaimissa

Tekninen ympäristö: Acer Aspire V3-571G (kannettava), Windows 7 64-bit
Näyttöresoluutio: 1 366 x 768 px.

Selaimet on lueteltu kuvakohtaisesti jokaisen kuvan yläpuolella.

Chrome 26.0

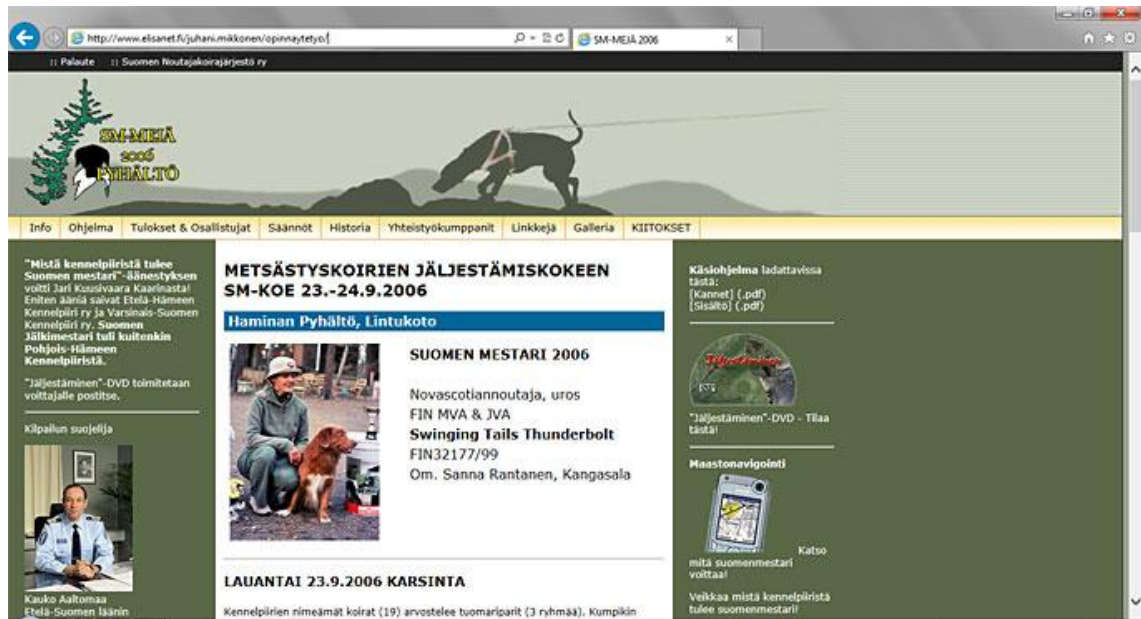


Firefox 20.0

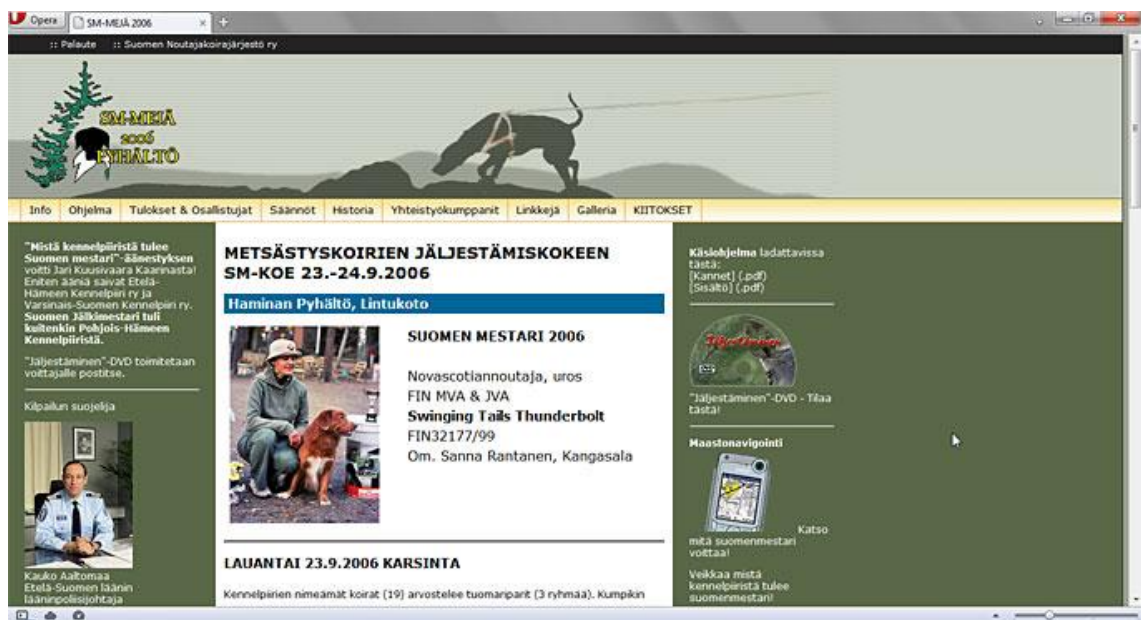


SM-Mejä 2006 -sivuston tarkastelu nykyajan web-selaimissa

Internet Explorer 10



Opera 12.15



SM-Mejä 2006 -sivuston tarkastelu nykyajan web-selaimissa

Safari 5.1.7

SM-MEJÄ 2006

Apple Yahoo! Google Maps YouTube Wikipedia Uutiset Suositut

Palautte Suomen Noutajakoirajärjestö ry

SM-MEJÄ 2006 PYHÄLTÖ

Info Ohjelma Tulokset & Osallistujat Säännöt Historia Yhteistyökumppanit Linkkejä Galleria KIITOKSET

"Mistä kennelpiiristä tulee Suomen mestari?" -kyselyyn voitti Jani Kuusivaara Kaarnasta. Ennen äänin saivat Eelis Hameen Kennelipiiri ry ja Varsinais-Suomen Kennelipiiri ry. Suomen Jätkämestari tuli kuitenkin Pohjois-Hämeestä Kennelpiiristä.

"Jäljestäminen" DVD toimitetaan voittajalle postitse.

Käpaikan suojelija

Katri Aaltonen
Etelä-Suomen läänin läänipoliisijohtaja

METSÄSTYSKOIRIEN JÄLJESTÄMISKOKEEN SM-KOE 23.-24.9.2006

Haminan Pyhältö, Lintukoto

SUOMEN MESTARI 2006

Novascotiannoutaja, uros
FIN MVA & JVA
Swinging Tails Thunderbolt
FIN32177/99
Om. Sanna Rantanen, Kangasala

LAUANTAI 23.9.2006 KARSINTA

Kennelpienin nimeämät koirat (19) arvostele tuomariparit (3 ryhmää), Kumpikin

Käsiobjektia ladattavissa laissa:
[Kannet] (.pdf)
[Sisältö] (.pdf)

"Jäljestäminen" DVD -Tilaa tässä!

Maastonavigointi

Katso mitä suomenmestari voittaa!

Voikkaa mistä kennelpiiristä tulee suomenmestari!