



HTML5- ja CSS3-verkko-opas

Matti Rauhala

Opinnäytetyö
Kesäkuu 2012
Tietojenkäsittely
Digimedia

TAMPEREEN AMMATTIKORKEAKOULU
Tampere University of Applied Sciences

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietojenkäsittely
Digimedia

RAUHALA, MATTI:
HTML5- ja CSS3-verkko-opas

Opinnäytetyö 39 sivua
Kesäkuu 2012

HTML5 ja CSS3 ovat Internet-sivujen luomisessa käytettyjen web-standardien kehitysvaiheessa olevia uusia versioita. Standardit ovat osa Tampereen ammattikorkeakoulun tietojenkäsittelyn koulutusohjelmaa lähitulevaisuudessa. Opinnäytteen käytännön osion tuloksena on verkko-opas, jonka tarkoituksena on toimia opiskelijoiden itseopiskelumateriaalina. Opinnäytetyö toteutettiin TAMK:n toimeksiantona, ja sen tavoitteena oli edistää digimediaan suuntautuneiden opiskelijoiden tuntemusta näistä standardeista. Raportissa kartoitetaan, mitkä ovat tekoheikellä parhaiten toimivat ja tärkeimmät ominaisuudet ja selvitetään näiden uusien standardien tarjoamia hyötyjä web-kehityksessä. Raportissa käsitellään myös web-standardien historiaa sekä nykytilannetta ja siihen johdaneita tekijöitä.

Uusien ominaisuuksien toimivuutta rajoittaa tällä hetkellä eniten selainten tuki. Tiettyjä elementtejä tuetaan jo melko laajasti, mutta yhteensopivuus on varmistettava ja tarjottava vaihtoehtoinen ratkaisu. Sekä HTML5:n että CSS3:n tarjoamat ratkaisut korvaavat osittain JavaScriptin tai erillisten liitännäisten käytön. Näin ollen web-kehittäjän ei välttämättä tarvitse turvautua useisiin eri tekniikoihin. Uusista rajapinnoista ja CSS3:n ominaisuuksista on hyötyä erityisesti mobiilikehityksessä. Internetin selaaminen erilaisilla laitteilla lisääntyy jatkuvasti ja kehitetyt HTML5- sekä CSS3-tekniikat antavat paremmat mahdollisuudet toteuttaa laiteriippumattomia mobiiliratkaisuja.

HTML5 ja CSS3 ovat molemmat hyvin laajoja käsitteitä, eikä ole vielä varmaa, kuinka laajoiksi ne lopulta kasvavat. Osaa HTML5:n ja CSS3:n uusista ominaisuuksista on alettu jo käyttää, ja niiden käyttö lisääntyy lähivuosina standardien kehittyessä ja selaintuen parantuessa. Oppaan päivittäminen on olennaista, sillä kaikki tärkeät piirteet eivät mahtuneet mukaan ja niiden määrä tulee kasvamaan tulevaisuudessa.

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Business Information Systems
Digital media

RAUHALA MATTI:
HTML5 and CSS3 Web Guide

Bachelor's thesis 39 pages
June 2012

HTML5 and CSS3 are the new versions of web standards used in the creation of web-pages. In the near future these standards will be a part of the degree programme in Business Information Systems at Tampere University of Applied Sciences. The practical part of this thesis consists of a web guide that is meant for the students as self-learning material. This thesis was assigned by Tampere University of Applied Sciences for the purpose of improving the students' knowledge of the aforementioned standards. The theoretical section gathers the best and most important features and explains the profits that these standards offer for web-developers. The history of the web standards and factors that have led into current situation are also covered.

The functionality of new features is restricted by browser support. Some elements are already widely supported, but compatibility must be ensured and an alternative solution offered. Both HTML5 and CSS3 offer features that partly replace the use of JavaScript or different plug-ins. Thus the web-developer does not need to resort to separate web-technologies. The new interfaces are particularly useful in mobile development. Internet browsing with a wide variety of devices is constantly increasing and HTML5 and CSS3 technologies offer better resources to implement device independent mobile solutions.

HTML5 and CSS3 are both extensive concepts and it is not yet sure how pervasive they will become. Some of the new features in HTML5 and CSS3 are already in use and their utilisation will increase in a few years when the standards evolve and browser support improves. Updating the web-guide is important, because all the essential elements could not be included and the number of elements will increase in the future.

Key words: HTML5, CSS3, web guide.

SISÄLLYS

1	JOHDANTO.....	5
2	MITÄ OVAT HTML JA CSS?	7
2.1	Metakielet ja merkintäkielet	7
2.2	HTML:n synty	8
2.2.1	HTML:n historia	9
2.2.2	HTML 4 ja XHTML	10
2.3	CSS:n historia	10
3	HTML5.....	12
3.1	HTML5:n uudet ominaisuudet.....	12
3.1.1	Canvas	12
3.1.2	Video ja audio	13
3.1.3	Geolocation	14
3.1.4	Web Storage	14
3.1.5	Lomakkeet.....	15
3.1.6	Rakenne-elementit.....	17
3.1.7	Offline-käyttö.....	19
3.1.8	JavaScript	19
3.2	Tie luonnoksesta suositukseksi.....	20
3.3	W3C ja WHATWG	21
3.4	Mobiililaitteet.....	22
3.4.1	Responsive design.....	22
3.4.2	HTML5 vs. Flash	23
3.5	Siirtyminen HTML5:een.....	24
3.6	HTML5:n ongelmat	25
3.7	Web-sovellukset.....	26
4	CSS3-moduulit	29
4.1	Color	29
4.2	Selectors.....	30
4.3	Transforms ja Transitions	31
4.4	Selaimet	32
5	OPPIMATERIAALIN TUOTTAMINEN VERKKOON	34
5.1	Suunnittelu ja toteutus	34
5.2	Typografia.....	34
6	POHDINTA.....	36
	LÄHTEET.....	37

1 JOHDANTO

Opinnäytetyö on ensisijaisesti suunnattu TAMKin tietojenkäsittelyn digimediaan suuntauneille opiskelijoille. Opinnäytetyön tavoitteena on auttaa opiskelijoita omaksumaan uusia verkkostandardeja, jotka ovat näillä näkymin Internetin tulevaisuus. Tavoitteena on myös selkeyttää nykytilanteeseen johtaneita tekijöitä ja perustella miksi HTML5 ja CSS3 ovat tärkeitä tekniikoita. Opinnäytetyön tarkoituksena on tulevia standardeja käsittelevän itseopiskelumateriaalin tuottaminen verkko-oppaan muodossa. Verkkooppaan aiheena ovat kehitteillä olevat HTML5- ja CSS3-standardit. Työssä etusijalla ovat HTML5:n tärkeimmät ominaisuudet, jotka toimivat parhaiten uusissa selaimissa. CSS3:sta käydään läpi suositukset sekä kiinnostavimmat ominaisuudet, jotka ovat vielä keskeneräisiä, mutta toimivat tietyin edellytyksin.

HTML5:n uusien ominaisuuksien toiminnallisuuteen liittyy hyvin vahvasti ohjelmointikieli JavaScript. Aiheen laajuuden johdosta JavaScriptiä ei kuitenkaan käsitellä oppaassa, mutta siinä tarjotaan linkkejä materiaaleihin, joista tutustumisen voi aloittaa. Oppaassa ei käydä myöskään läpi HTML:n ja CSS:n perusasioita, vaan oletuksena on, että ne ovat oppaan kohderyhmälle tuttuja. Aloittelijoille kohdennettuja HTML- ja CSS-oppaita on tehty runsaasti, joten perusasioiden käsittelylle ei nähty tässä työssä tarvetta.

Opinnäytetyöraportissa käydään läpi HTML:n ja CSS:n taustoja, nykytilanteeseen johtaneita tekijöitä sekä seikkoja, jotka tekevät HTML5:stä tärkeän. Uusien ominaisuuksien teoriaa käydään läpi hieman, mutta niihin syvennytään esimerkkien avulla käytännön osuuden tuloksena syntyneessä verkko-oppaassa. Lisäksi raportissa käydään läpi oppimateriaalin tuottamista verkkoon.

Opinnäytetyön toimeksiantajana toimii Tampereen ammattikorkeakoulu TAMK ja yhteyshenkilönä tietojenkäsittelyn koulutusohjelman tuntiopettaja Pasi Pekkanen. Pekkanen kanssa sovitussa tapaamisessa kävi ilmi, että HTML5 on osa tulevaa tietojenkäsittelyn opintosuunnitelmaa ja opiskelumateriaalille olisi tarvetta. Otin toimeksiannon vastaan ja sovimme alustavasti oppaan sisällöstä. Aihe tarkentui nykyiseen muotoonsa sähköpostikeskusteluissa ja tapaamisissa koululla.

Aiheen ollessa hyvin ajankohtainen, ei lähteiden löytäminen ollut ongelmallista. Tutustuessani aiheeseen kävin läpi paljon web-standardeja kehittävän organisaation W3C:n (World Wide Web Consortium) materiaaleja, joista osa on opinnäytetyöraportin lähteenä. Toinen tärkeä verkkolähde oli tietotekniikan uutislehden Tietoviikon Internet-sivut, jotka tarjosivat ajankohtaista ja luotettavaa informaatiota. Opinnäytetyön päälähteenä on käytetty Jukka Korpelan HTML5 - uudet ominaisuudet -kirjaa. Kirja sisältää melko tuoretta ja yksityiskohtaista informaatiota. Korpelan suhtautuminen HTML5:een on hyvällä tapaa varovaisempaa ja kriittisempää verrattuna moniin muihin käyttämiini kirjallisiin lähteisiin. Huolettomampaa lähestymistapaa edustavat maineikkaan web-suunnittelijan Jeffrey Zeldmanin kustantamat HTML5 for Web Designers sekä CSS3 for Web Designers. Monet lähteistä ovat vuosien 2010 ja 2012 välissä julkaistuja. HTML5 on kuitenkin hyvin muutosaltis, joten melko uudetkin lähteet sisältävät joitain vanhentuneita piirteitä. Useita lähteitä vertailemalla ja itse HTML5-spesifikaatiota tulkitsemalla tämä ei johtanut kuitenkaan ongelmatilanteisiin.

2 MITÄ OVAT HTML JA CSS?

HTML (Hypertext Markup Language) eli hypertekstin merkintäkieli ja CSS (Cascading Style Sheets) eli kaskadisoidut tyyliohjeet ovat Internet-sivujen luomiseen käytetyt kaksi pääteknologiaa. HTML:n avulla sivuille luodaan rakenne ja tuotetaan sisältö, CSS:llä puolestaan määritellään sivujen ulkoasu. (HTML & CSS 2012.) HTML:stä tuli 90-luvun alussa tärkeä web-dokumentin rakenteenkuvauskieli. Ennen pitkää HTML-dokumenttien ulkoasua haluttiin muokata ja 90-luvun puolivälissä tähän tarpeeseen kehitettiin CSS. (Addison 1998.) Molemmat tekniikat ovat säilyttäneet asemansa tähän päivään saakka. Sekä HTML:n että CSS:n merkittäviä uudistuksia sisältävät versiot ovat parhaillaan työn alla.

2.1 Metakielet ja merkintäkielet

Metakieli on kieli, jonka avulla voidaan määritellä muita kieliä. Metakielten avulla kuvataan dokumenttityyppejä DTD (Document Type Definition), jotka vastaavat tiettyä merkintäkieltä. DTD:ssä määritellään säännöt siitä, mitä elementtejä mikäkin elementti saa sisältää. Elementtien rakenteen määrittely koostuu aloitustagista, sisällöstä sekä lopetustagista. (On SGML... 2012.)

SGML (Standard Generalized Markup Language) on metakieli, johon HTML-merkintäkieli perustuu versioon 4.01 saakka. SGML kehitettiin jo 60-luvulla ja standardi (ISO 8879) julkaistiin vuonna 1986. (Parviainen 1999.) XML (Extensible Markup Language) on vuonna 1996 W3C:n kehittämä, SGML:stä yksinkertaistettu metakieli, joka on suunniteltu käytettäväksi webissä. XML:ään perustuu merkintäkielestä HTML 4.01 kehittyneempi XHTML. (Extensible Markup... 2012.)

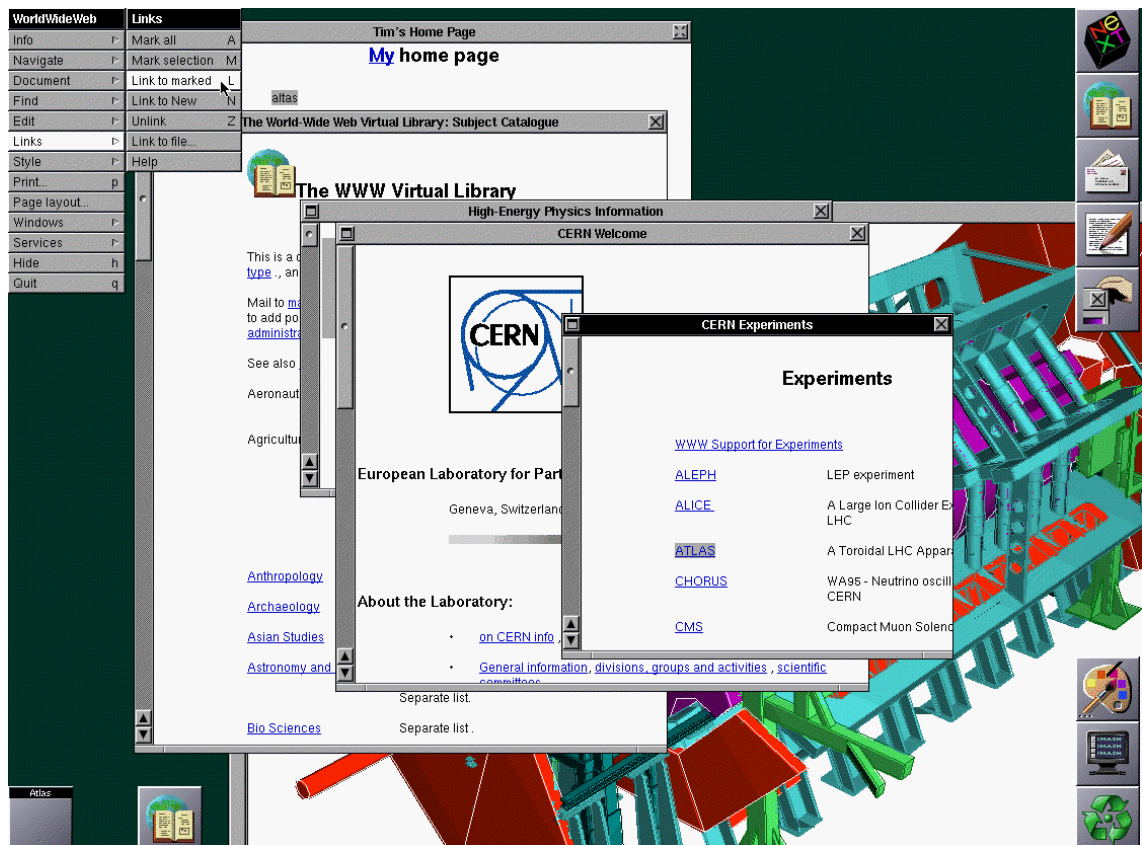
XML-merkkauksen SGML:ää tarkempi muotovaatimus mahdollistaa laajan yhteensopivuuden. XML-dokumentilla saa olla ainoastaan yksi juurielementti. Kaikilla elementeillä pitää olla sekä aloitus- että lopetustagi. Kaikki elementit ovat merkkikokoriippuvaisia (case sensitive). Sisäkkäiset elementit eivät saa mennä ristiin (properly nested). Lisäksi kaikilla attribuuteilla tulee olla arvo, joka esitetään lainausmerkkien sisässä. (XML Syntax... 2012)

HTML5 on määritelty itsenäiseksi merkintäkieleksi. Se on suunniteltu yhteensopivaksi vanhojen HTML- ja XHTML-merkintäkielten kanssa, joten sitä voidaan kirjoittaa joko XML- tai SGML-syntaksin mukaisesti. XML-muotoisen merkkauksen käyttö on suositeltavaa, sillä se mahdollistaa muiden XML-pohjaisten kielten liittämistä suoraan HTML5-dokumenttiin. Paljon käytettyjä XML-pohjaisia kieliä ovat muun muassa matemaattisten kaavojen esittämiseen käytetty kuvauskieli MathML sekä vektorigrafiikan esittämiseen käytetty kuvauskieli SVG. (Korpela 2011, 312–314.)

2.2 HTML:n synty

Vuonna 1989 Tim Bernes-Lee työskenteli Euroopan hiukkasfysiikan tutkimuskeskuksen CERN tietotekniikan osastolla. CERN:ssä tutkittiin atomeihin liittyvää ilmiötä johon osallistui tutkijoita ympäri maailmaa. Työskennellessään Tim sai idean, joka multistaisi informaation jakamisen. Hän alkoi suunnitella tapaa, jolla voidaan jakaa tiedostoja useiden tietokoneiden kesken käyttäen hypertekstiä. Hypertekstillä tarkoitetaan tekstiä, joka sisältää hyperlinkkejä toisiin tiedostoihin. (Addison 1998.)

Sen lisäksi, että erilliset tiedostot olisi mahdollista ladata käyttäjien tietokoneelle, Bernes-Lee halusi, että tutkimusdokumentit sisältäisivät linkkejä toisiin aiheeseen liittyviin dokumentteihin. Näihin dokumentteihin pääsisi käsiksi nopeasti ruudulla olevia nappeja klikkaamalla. Näiden ideoiden pohjalta Bernes-Lee kehitti ensimmäisen selaimen NeXT-tietokoneelle vuonna 1990. Ohjelman nimeksi tuli WorldWideWeb. (Addison 1998.) Samana vuonna Tim pystytti ensimmäisen web-serverin sekä loi ensimmäisen web-sivun (kuva 1) (Bernes-Lee 2012).



KUVA 1. Bernes-Leen ensimmäinen web-sivu (Bernes-Lee 2012)

2.2.1 HTML:n historia

HTML:n alkuvuodet 1990–1994 olivat sekavaa aikaa, sillä mitään virallista määrittelyä ei vielä ollut. Uusia selaimia kehitettiin ja selainvalmistajat lisäsivät HTML-ominaisuuksia mielensä mukaan. (Addison 1998.)

Vuoden 1994 keskivaiheilla web-kehittäjä Dan Connolly kokosi luonnokseen eniten käytetyt HTML-tagit. Luonnos annettiin yleisön kommentoitavaksi ja jatkokehityksessä huomioitiin saatuja ehdotuksia. Tästä luonnoksesta muodostui HTML 2 -spesifikaatio. Saman vuoden aikana Netscape kehitti oman selaimen, joka sai osakseen suuren suosion. Pian Netscape alkoi kehittää HTML:ää omaan suuntaansa ilman muun HTML-yhteisön tukea. Vuoden lopussa Bernes-Lee perusti W3C:n. (Addison 1998.)

Vuonna 1995 julkaistiin HTML 3 -luonnos. Eri selainvalmistajat ottivat käyttöönsä tiettyjä ominaisuuksia tästä luonnoksesta. Sekavuus oli huipussaan, kun selaimiin lisättiin mukaan valmistajien omia uusia ominaisuuksia. HTML 3.2 -spesifikaatioon otettiin

mukaan taulukot, joita käytettiin tekstin ja kuvien asetteluun. Netscape toimi jälleen omin päin lisäämällä kehykset (frames) selaimeensa vuoden lopussa. (Addison 1998.)

Vuoden 1997 alussa W3C sai valmiiksi HTML 3.2 -spesifikaation, joka oli jäsenorganisaatioiden ja selainvalmistajien hyväksymä. W3C oli viimeinkin saanut aikaan yhteisymmärryksen ja onnistunut kehittämään HTML web-standardin. (Addison 1998.)

2.2.2 HTML 4 ja XHTML

HTML 4 julkaistiin vuonna 1999 ja sitä päivitettiin hieman vuonna 2000 versiossa 4.01. HTML 4.01 on HTML:n viimeisin standardi, josta on kolme versiota (Allsopp 2010, 21). Strict-versio sisältää standardissa suositellut ominaisuudet, jotka liittyvät ainoastaan rakenne- ja merkityssuhteiden määrittelyyn. Transitional on sallivampi kuin strict ja se on tarkoitettu HTML:n vanhemmille versioille, jotka sisältävät ulkoasun muotoiluun tarkoitettuja sekä vanhentuneita elementtejä. Kolmas versio frameset on yhtäläinen transitionalin kanssa, mutta se sallii lisäksi selainikkunan jakamisen osiin kehysten avulla. (HTML DOCTYPE... 2012.)

Vuonna 2000 julkaistiin XHTML:n ensimmäinen versio 1.0. XHTML on XML-pohjainen, täysin HTML 4.01:n kanssa yhteensopiva merkintäkieli, joka sisältää myös strict-, transitional- sekä frameset-versiot. Vuonna 2001 ilmestynyt päivitys XHTML 1.1 oli 1.0 Strict -määritelmän mukainen, eikä se sisältänyt eriasteisia versioita. XHTML 2.0 on vuonna 2002 alkunsa saanut, mutta HTML5:n takia vuonna 2009 keskeytetty projekti. (Allsopp 2010, 21.)

2.3 CSS:n historia

HTML:n syntymästä asti ideana oli, että dokumentin asettelu hoidettaisiin käyttämällä erillistä tyyli tiedostoa. Syntaksia tyylien lisäämiselle ei kuitenkaan vielä julkaistu, vaan se jätettiin selaimille. Vuonna 1993 ilmestynyt Mosaic selain mahdollisti tiettyjen värien ja fonttien muokkaamisen. Pian web-suunnittelijat halusivat lisää valtaa muokata HTML-dokumenttien ulkoasua. Tarve tyylien lisäämiseen syntyi muun muassa halusta taittaa web-sivuista sanomalehden kaltaisia. (The CSS saga 2012.)

W3C julkaisi ensimmäisen CSS-spesifikaation vuonna 1996. CSS1 sisälsi perusominaisuudet väreistä, fonteista sekä tyhjän tilan hallinnasta. CSS2 julkaistiin kaksi vuotta myöhemmin, mutta selaintuki siihen pysyi pitkään heikkona. Vuonna 2001 ilmestynyt Internet Explorer 6 oli bugeistaan huolimatta ensimmäisiä selaimia, joissa oli kohtuullinen CSS-tuki. Asiat alkoivat muuttua vuonna 2003, kun Safari julkaisi selaimensa. Seuraavana vuonna julkaistiin Firefox-selaimen ensimmäinen versio. Molemmat selaimet noudattivat standardeja paljon aikaisempia selaimia tarkemmin. (Allsopp 2010, 70–71.)

3 HTML5

HTML5 sisältää aikaisempiin versioihin nähden paljon uusia ominaisuuksia, joiden kehitystä viedään eteenpäin yhtenä spesifikaationa. Spesifikaatiosta on tosin eroteltu ohjelmointirajapintoja (API, Application Programming Interface), joiden toiminnallisuuksiin liittyy vahvasti JavaScript. Tällaisia ohjelmointirajapintoja ovat esimerkiksi paikannukseen käytettävä Geolocation sekä selaimen oma tietovarasto Web Storage. Toisin kuin HTML5, CSS3 on kokonaisuudessaan jaettu useisiin toisistaan irrallisiin spesifikaatioihin. (Weyl, Lazaris & Goldstein 2011, 1–2.)

3.1 HTML5:n uudet ominaisuudet

HTML5 on suuri kokonaisuus, joka tarjoaa jokaisella jotain. Se on laitteistoriippumaton ja vähentää liitännäisten tarvetta omilla natiiveilla eli sisäänrakennetuilla ratkaisullaan. HTML5 tarjoaa myös työkalut grafiikan toteuttamiselle, selkeyttää dokumentin rakennetta sekä yksinkertaistaa muun muassa lomakkeiden toiminnallisuuksia. (Korpela 2011, 13.)

3.1.1 Canvas

HTML5:n tärkeimpiin ominaisuuksiin kuuluu canvas-elementti, jonka avulla on mahdollista toteuttaa dynaaminen piirtoalusta. Canvas-elementin käyttöhaarukka on yksinkertaisten muotojen, kuten ympyröiden ja neliöiden, esittämisestä selaimessa toimiviin peleihin ja piirto-ohjelmiin (kuva 2). Canvas-elementti ei itsessään tee mitään, vaan toiminnallisuus hoidetaan API:en kuten Canvas 2D Context avulla käyttäen JavaScriptiä (Keith 2010, 23–24).



KUVA 2. Canvas-elementillä toteutettu piirto-ohjelma Sketchpad (MUDCUBE 2012)

3.1.2 Video ja audio

Tärkeimpiin uutuuksiin kuuluu myös natiivi videon ja äänen esitystapa. Aikaisemmin mediaupotukseen on tarvittu erillisiä liitännäisiä kuten Flash (Korpela 2011, 177). Videopalvelu YouTube on vuodesta 2010 asti tarjonnut mahdollisuuden katsella videoita Flashin sijaan HTML5-videona (kuva 3). Yksi video-elementtiin liittyvistä ongelmista on selainten tukemat videoformaatit. HTML5-video tukee MP4-, WebM- sekä Ogg-formaatteja. Kuitenkaan mikään edellä mainituista ei toimi kaikissa selaimissa, joten videotiedosto on esitettävä esimerkiksi sekä MP4- että Ogg-muodossa. Sama ongelma liittyy audio-elementin käyttöön, missä äänitiedosto on esitettävä sekä Ogg- että MP3-muodossa. (Sani 2010.)

YouTube HTML5 Video Player

This is an opt-in trial of HTML5 video on YouTube. If you are using a supported browser, you can choose to use the HTML5 player instead of the Flash player for most videos. Your comments will help us improve and perfect the mixtures we're working on. So jump in, play around, and send your [feedback](#) directly to the brains behind the scenes.

Supported Browsers

We support browsers that support both the video tag in HTML5 and either the h.264 video codec or the WebM format (with VP8 codec). These include:

- Firefox 4 ([WebM, Available here](#))
- Google Chrome ([WebM](#))
- Opera 10.6+ ([WebM, Available here](#))
- Apple Safari (h.264, version 4+)
- Microsoft Internet Explorer 9 (h.264, [Available here](#), [WebM support available here](#))
- Microsoft Internet Explorer 6, 7, or 8 with Google Chrome Frame installed ([Get Google Chrome Frame](#))

Notes

- Fullscreen support is partially implemented. Pressing the fullscreen button will expand the player to fill your browser. If your browser supports a fullscreen option, you can then use that to truly fill the screen
- If you want to find videos with WebM formats available, you can use the Advanced Search options to look for them (or just add &webm=1 to any search URL)

Additional Restrictions (we are working on these!)

- Videos with ads are not supported (they will play in the Flash player)
- On Firefox and Opera, only videos with WebM transcodes will play in HTML5
- If you've opted in to other testtube experiments, you may not get the HTML5 player (Feather is supported, though)

What does my browser support?

Video tag h.264 WebM

You are not currently in the HTML5 trial.
[Join the HTML5 Trial](#)

KUVA 3. Videopalvelu YouTuben HTML5-videokokeilu (YouTube 2012)

3.1.3 Geolocation

Geolocation on HTML5-spesifikaatiosta eroteltu paikannukseen tarkoitettu API. Geolocationin avulla voidaan hankkia tietoja käyttäjän maantieteellisestä sijainnista ja tarjota olennaista tietoa, kuten lähimmät nähtävyydet tai ravintolat. Selain pyytää aina käyttäjältä lupaa sijaintitietojen käyttämiseen. Tästä huolimatta monet pitävät paikannusta yksityisyydensuojaa vaarantavana. Paikannuksen tarkkuus riippuu käytettävästä laitteesta sekä käytössä olevasta Internet-yhteydestä. Esimerkiksi käyttäjäkohtaisia säätietoja esitettäessä tieto kaupungista riittää varsin hyvin. Mobiililaitteen GPS:n avulla paikannus on paljon täsmällisempi ja mahdollistaa esimerkiksi reittiohjeiden antamisen. (Korpela 2011, 237–238.) Facebookissa paikannusta käytetään Paikat-toiminnoissa, jossa sijaintitietoja voi jakaa ystävien kesken (Facebook 2012).

3.1.4 Web Storage

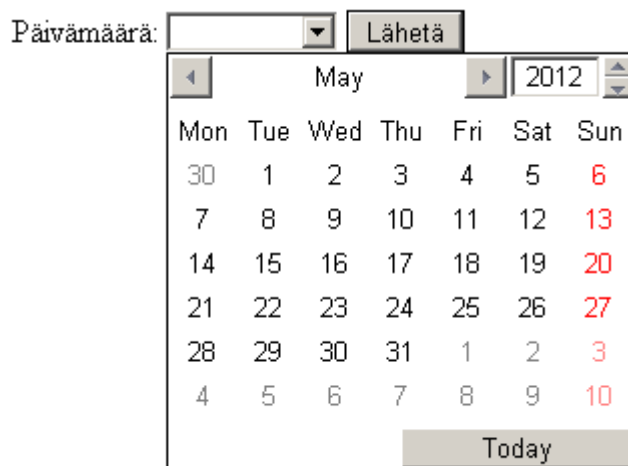
Web Storage API on käyttäjän selaimessa sijaitseva tietovarasto. Web Storagea käytetään evästeiden (cookies) tapaan käyttäjän henkilökohtaisen näkymän esittämiseen, mutta se on evästeitä kehittyneempi, turvallisempi ja nopeampi tekniikka. Web Storage tallentaa tietoja käyttäjän omalle tietokoneelle, eikä se kuormita sivustoa. Suorituskykyä

parantaa myös se, että tietoja ei lähetetä turhaan jokaisen palvelinpyynnön yhteydessä vaan ainoastaan niitä pyydettyä. (HTML5 Web... 2012.)

Web Storage sisältää kaksi oliota: localStorage ja sessionStorage. Istuntonmuisti sessionStorage tallentaa tietoja vain istunnon ajaksi ja muisti tyhjenetään käyttäjän poistuessa sivulta. Paikallismuisti localStorage on tarkoitettu tiedon pitkäaikaiseen varastointiin, eikä muistille ole asetettu päättymisaikaa. Istuntonmuisti sopii esimerkiksi säilyttämään käyttäjän lisäämiä tuotteita verkkokaupan ostoskorissa. Paikallismuistia voidaan käyttää hyödyksi esimerkiksi tallentamalla muistiin käyttäjän selaamia tuotteita ja tarjoamalla näitä kiinnostuksen kohteita myöhemmin. (Korpela 2011, 229, 232.)

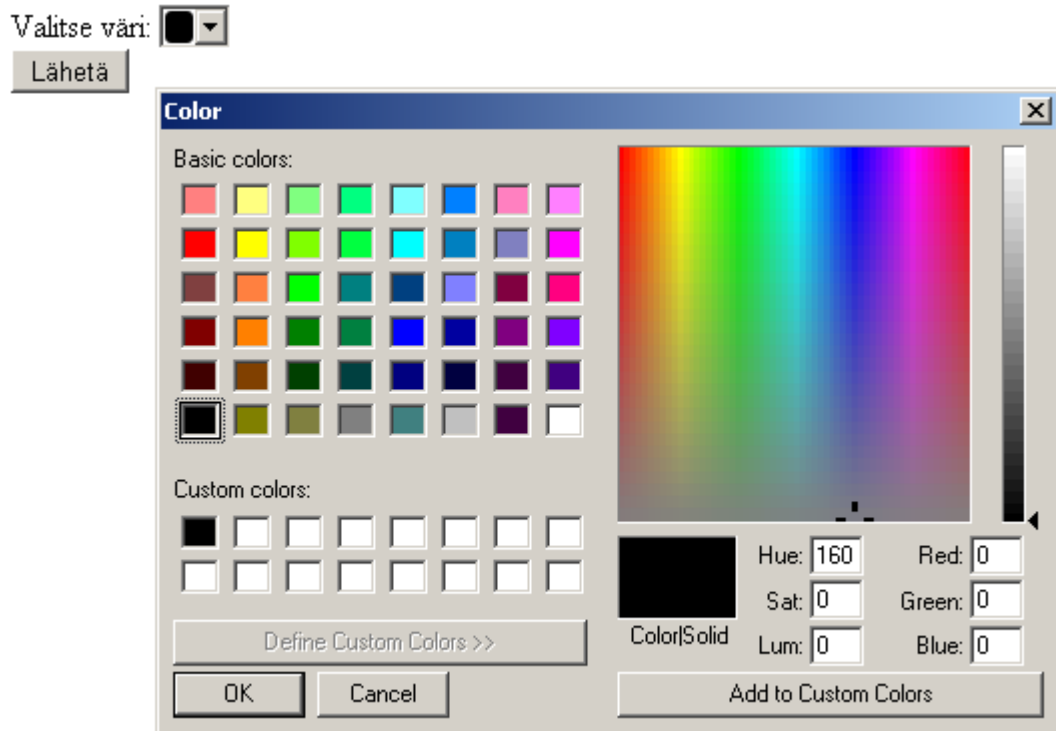
3.1.5 Lomakkeet

HTML5 tuo lomakkeisiin paljon uudistuksia. Uusissa ajanilmauksiin liittyvissä syöte-tyypeissä selaimen on tarkoitus luoda graafinen käyttöliittymä, josta haluttu ajankohta valitaan. Esimerkiksi paljon käytetty kalenterinäköymä, josta päivämäärä valitaan, voidaan toteuttaa yksinkertaisesti date-syötetyyppiä käyttämällä (kuva 4). Muita ajanilmaukseen liittyviä syötetyyppejä ovat datetime, month, week ja time. (Korpela 2011, 143–145, 151.)



KUVA 4. Päivämäärän valinta Opera-selaimessa date-syötetyyppiä käytettäessä

Käytettäessä syötetyyppiä color, käyttäjä voi valita värin RGB-arvon selaimen luomasta graafisesta käyttöliittymästä (kuva 5). Kuvankäsittelyohjelmista tutusta käyttöliittymästä väri on helppo valita tietämättä mitään itse värikoodeista. (Korpela 2011, 149.)



KUVA 5. Värin valinta Opera-selaimessa

Yhteystiedon, kuten email, tel ja url, tarkoituksena syötetyypeissä ei ole luoda erillistä käyttöliittymää, vaan syötekentät pysyvät tavallisina tekstikenttinä. Yhteystieto syötetyyppien tarkoituksena on käyttäjän syötteen tarkistaminen, joka on ennen suoritettu käyttämällä JavaScriptiä. (Korpela 2011, 144–145.)

Muita uusia syötetyyppejä ovat search, range ja number. Search-syötetyypin tarkoituksena on, että selain voi asettaa omaa hakutoimintoa vastaavan muotoilun, jolla se voidaan erottaa tavallisesta tekstikentästä. Syötetyyppi number lisää syötekenttään painikkeet, joiden avulla arvo voidaan helposti valita (kuva 6). Syötetyyppiin voidaan lisätä määritteet min ja max, jotka määrittävät lukuvälin, määritteellä step säädetään askelväli. (Korpela 2011, 146–147, 150.)



KUVA 6. Numeron valinta Opera-selaimessa käytettäessä number-syötetyyppiä

Syötetyyppi range luo syötekentästä liukusäätimen, jolla arvo voidaan valita (kuva 7). Käytössä ovat samat määritteet kuin number-syötetyypissä. Syötetyyppi range on käte-

vä tilanteissa, joissa tarkan arvon mittaaminen ei ole tärkeintä, vaan havainnollisuus ja helppokäyttöisyys. (Korpela 2011, 148.)

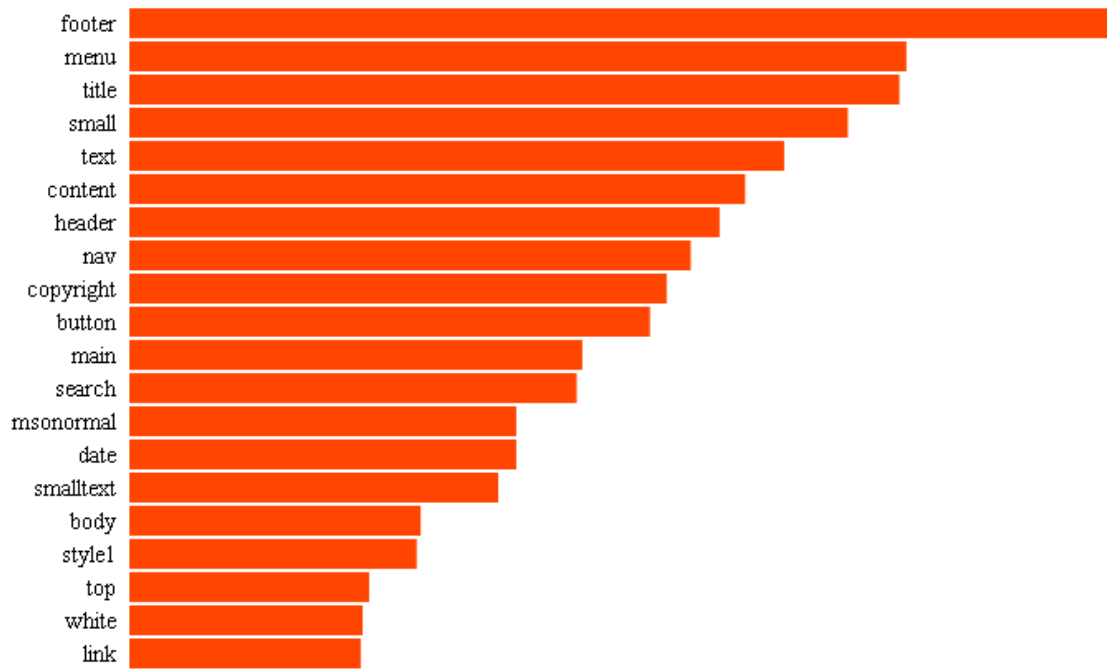


KUVA 7. Pisteiden valinta Opera-selaimessa range-syötetyyppeä käytettäessä

HTML5 tuo mukanaan helppokäyttöisiä syötetyyppejä, joiden toiminnallisuus on hoidettu ennen JavaScriptillä. Uusien syötetyyppien tuki selaimissa on vielä heikko ja monet uudistuksista toimivat ainoastaan Operan selaimessa. Mikäli selain ei tunnista uusia syötetyyppejä, se tulkitsee ne normaaleiksi tekstikentiksi. Siirtyminen uusiin syötetyyppeihin ilman vaihtoehtoista sisältöä on vielä huono idea. Syötetyyppejä käytettäessä kannattaa ottaa huomioon vanhat selaimet ja ohjeistaa käyttäjää syöttämään arvo manuaalisesti. Monissa tapauksissa lomakkeen toiminnallisuus on syytä hoitaa HTML5:sta tukemattomille selaimille vanhaan tapaan käyttäen JavaScriptiä. Muita lomakkeiden uudistuksia ovat väistynyt aputeksti, kentän asettaminen pakolliseksi, automaattinen täydennys, fokusointi sekä hiirellä vetäminen (Drag and drop). (Keith 2010, 54–55.)

3.1.6 Rakenne-elementit

Uudet rakenne-elementit ovat peruja div-elementin eniten käytetyistä class-määritteistä (kuva 8). Uusia elementtejä ovat article, aside, figcaption, figure, footer, header, hgroup, nav sekä section. (Google 2012.) Uudet elementit on tarkoitettu puhtaasti semanttisiksi eli merkitystä kuvaaviksi. Kaikki elementit on suunniteltu siten, että niiden tulee sisältää tiettyä informaatiota täyttääkseen käyttövaatimukset. Mikäli tarkoituksena on ainoastaan sisällön asettelu, tulee käyttää uusien elementtien sijasta div-elementtiä. (Keith 2010, 69–70.) Uusista rakenne-elementeistä saattaa tulevaisuudessa olla suuri hyöty hakukoneissa, mutta tällä hetkellä ne tekevät koodista vain hieman lyhyempää ja siistimpää. (Korpela 2011, 87)



KUVIO 8. Google Web Statistics Käytetyimmät class-määrittet (Google 2012)

Article-elementin tarkoituksena on muodostaa itsenäinen kokonaisuus, joka olisi mahdollista kopioida toiselle sivulle. Tyypillisimpiä esiintymiä ovat blogimerkintä, uutinen tai arvostelu. Aside-elementti on tarkoitettu sivuhuomautukseksi, joka liittyy pääaiheeseen, mutta voitaisiin jättää poisikin. Hyvä esimerkki asiden käytöstä on artikkelista erotettu lainaus tai uutisen yhteydessä tietoisu aiheena olevasta maasta. Figuren tarkoituksena on koota yhteen kuva ja kuvateksti eli figcaption. Figure-elementtiä tulee käyttää informaatiota sisältäviin valokuviiin, piirroksiin, kuvaajiin ja grafiikoihin, mutta ei koristeukuviin. Footer-elementtiä voidaan käyttää joko koko sivun tai pelkän artikkelin alätunnisteena. Molemmissa tapauksissa footerin tulee sisältää esimerkiksi tekijätietoja, päiväys sekä tekijänoikeusmerkintä. Header voi olla joko koko sivun ylätunniste tai artikkelin otsikko. Koko sivun ylätunnisteena toimiva header voi sisältää tietoa kuten logo, sisällysluettelo ja hakukenttä. Hgroup-elementin avulla voidaan ryhmittää eri tason otsikoita siten, että ryhmä vastaa yhtä otsikkoa. Hyvänä esimerkkinä on varsinainen otsikko ja sitä seuraava täydentävä alaotsikko, jotka voidaan mieltää yhdeksi otsikoksi. Nav-elementti on tarkoitettu määrittämään sivuston päänavigointia. Section-elementti vaikuttaa hyvin samantlaiselta kuin div-elementti. Erona on, että div-elementti ei kerro käyttäjälle mitään sisällöstään. Section-elementti on tarkoitettu tiettyyn aiheeseen liittyvän sisällön ryhmittelyyn. (Korpela 2011, 90–98.)

3.1.7 Offline-käyttö

Etenkin mobiililaitteiden kannalta tärkeä ominaisuus on mahdollisuus offline-käyttöön. Mobiililaitteiden Internet-yhteydet saattavat pätkiä ja nopeudet vaihdella. Offline-käyttö on mahdollista sovellusvälimuistin avulla. Sovellusvälimuistia ei ole tarkoitettu kaikkien sivustojen käyttöön, vaan ennen kaikkea sovelluksille, jotka sisältävät paljon informaatiota ja joita päivitetään harvoin. Sovellusvälimuistin toiminta perustuu välimuisti-ilmoitukseen, jossa selaimelle kerrotaan, mitkä tiedostot sen tulee ladata muistiin. (Korpela 2011, 267–270, 274.)

3.1.8 JavaScript

Monien HTML5-ominaisuuksien toiminnallisuuteen liittyy vahvasti JavaScript. Parhaimpana esimerkkinä tästä on uusi canvas-elementti. Itsessään canvas-elementti on hyvin yksikertainen pätkä HTML-koodia, jossa määritellään piirtoalustan koko ja annetaan sille tunniste. Tätä enempää ei voida itse HTML-koodilla saavuttaa, vaan kaikki toiminnallisuus hoidetaan JavaScriptillä, jossa viitataan canvas-elementin tunnisteeseen. (Keith 2010, 23.)

HTML5 sisältää myös ominaisuuksia, jotka eivät ole canvas-elementin tavoin yhtä riippuvaisia JavaScriptistä. Audio- ja videosoitinien luominen HTML5:llä on hyvin yksinkertaista. Mikäli selaimen luoman natiivisoittimen ulkoasu tai ominaisuudet eivät ole käyttötarkoitukseen sopivat, voi oman soittimen luoda JavaScriptillä (Keith 2010, 31–32).

Yksi JavaScriptin suurimpia käyttötarkoituksia oli aikaisemmin lomakkeiden muokkaaminen, kuten kentän asettaminen pakolliseksi tai alta väistyvän aputekstin lisääminen. HTML5 pyrkii korvaamaan tämänkaltaisia suosittuja ominaisuuksia tekemällä niistä helppokäyttöisiä ja natiiveja. Kaikki uudet selaimet eivät kuitenkaan tue uusia syöte-tyyppejä, vanhoista puhumattakaan. Selaimet, jotka eivät tue uusia lomakeominaisuuksia, näyttävät kentät normaaleina tekstikenttinä. Monille sivustoille esimerkiksi päivämäärän valitseminen kalenterista on lomakkeen tärkeä ominaisuus, eikä tietojen manuaalinen syöttäminen tekstikenttään riitä. Vaikka HTML5 korvaa JavaScriptiä lomakkeissa, ei sitä vielä sovi unohtaa. (Keith 2010, 40–45, 54.) Vanhat selaimet kannattaa

ottaa huomioon tärkeissä tilanteissa, kuten kalenterinäkömman esittämisessä. Tämänkaltaisissa tilanteissa on syytä tehdä tarkistus ja tarjota sisältö vanhoille selaimille käyttäen JavaScriptiä.

3.2 Tie luonnoksesta suositukseksi

W3C:n suositus-status on huolellisen työn ja pitkän suosituspolun läpäisemisen takana. Ensimmäinen vaihe kohti suositus-statusta käynnistyy, kun työryhmä laatii aiheestaan teknisen dokumentin. W3C voi julkaista tämän dokumentin statuksella WD (Working Draft) yleisön arvosteltavaksi ja kommentoitavaksi. Työryhmä on yleensä noin 10 hengen joukko W3C-jäsenorganisaatioiden nimeämiä työntekijöitä. (Nykänen 2003.)

Työryhmän ollessa tyytyväinen dokumenttiin, siitä pyydetään vielä lopullista palautetta (Last Call Announcement). Last Call on osoitus siitä, että työryhmän mukaan WD on saanut hyväksyntää ja ryhmä on valmis jatkamaan dokumentin kehitystä seuraavalle tasolle. Palautteen huomioimisen jälkeen voidaan edetä seuraavaan vaiheeseen CR (Candidate Recommendation). W3C uskoo, että dokumentti on saanut laajalti huomiota ja se tyydyttää työryhmän vaatimuksia. Tarkoituksena on kerätä palautetta ja toteutuskokemuksia myös W3C:n ulkopuolelta. (Nykänen 2003.)

CR-vaiheen jälkeen dokumentti julkaistaan statuksella PR (Proposed Recommendation), jolloin W3C-ohjausryhmä ottaa dokumentin viimeiseen käsittelyyn. Mikäli PR menee läpi, saa dokumentti W3C:n hyväksynnän eli suositus-statusen REC (Recommendation). Dokumentti voidaan myös palauttaa WD-vaiheeseen tai keskeyttää kokonaan missä tahansa suosituspolun vaiheessa. (Nykänen 2003.)

HTML5:n ensimmäinen luonnos julkaistiin 22.1.2008. W3C:n 25.5.2011 julkaistussa tiedotteessa ilmoitettiin, että HTML5 on edennyt Last Call -vaiheeseen. Samassa tiedotteessa vahvistettiin, että spesifikaatio on pysynyt aikataulussa ja se saa suositus-statusen vuoden 2014 aikana. (W3C Invites... 2011.) Monien mielestä 2014 on liian optimistinen arvio. Toiset ovat sitä mieltä, että spesifikaatio on kokonaisuudessaan valmis vasta vuonna 2022, ellei myöhemmin. Vuoden 2012 aikana nähdään, pysyykö HTML5 aikataulussa, sillä CR-vaiheeseen siirtymisen on ennustettu tapahtuvan kyseisenä vuonna.

3.3 W3C ja WHATWG

W3C eli World Wide Web Consortium on Bernes-Leen johtama kansainvälinen web-standardeja kehittävä organisaatio. Jäseniin kuuluvat muun muassa Apple, Microsoft, Mozilla, Opera, Adobe, Google, Oracle, IBM sekä Nokia (Current Members 2012). W3C:n työn näkyvimpänä tuloksena ovat suositukset (W3C Recommendation). Suositukset ovat suosituspolun läpäisseitä ehdotuksia. Vuosien aikana W3C on standardoinut useita web-tekniikoita kuten HTML, CSS, XML, SVG, MathML ja DOM (Standards 2012).

W3C:n tavoitteena on taata kaikille käyttäjille mahdollisimman yhtenäinen ja toimiva Internet. Tekninen toteutus on toimivan webin perusta. Noudattamalla suosituksissa määriteltyjä pelisääntöjä edesautetaan tätä tavoitetta. Suositusten noudattaminen on ensiaskel suuntaan, jossa sivut toimivat eri selaimissa ja käyttökokemus on samankaltainen. (Nykänen 2003.) Virheellisesti merkatut sivut voivat näkyä kehittäjän selaimella hienosti, mutta tämä ei tarkoita, että toiset käyttäjät eri selaimella tai saman selaimen vanhemmalla versiolla ovat samaa mieltä. Pelkästään standardien noudattaminen ei kuitenkaan riitä. Internetin tila tänä päivänä ei ole niin yksinkertainen ja se aiheuttaaakin harmaita hiuksia web-kehittäjille ympäri maailmaa. Tilanne olisi huomattavasti kaoottisempi ilman minkäänlaisia yhteisiä pelisääntöjä.

HTML:n kehitys oli hidasta vuosina 1998–2003, jolloin kieleen tuli vain pieniä päivityksiä. Vuonna 2002 W3C aloitti mahtipontisen projektin, joka kulki nimellä XHTML 2.0. XHTML 2.0:n oli tarkoitus olla täysin uusi kieli, joka ei olisi yhteensopiva vanhojen HTML- ja XHTML-kielten kanssa. (Allsopp 2010, 21)

Selainyhtiöt Mozilla ja Opera halusivat viedä HTML:n kehitystä eteenpäin, mutta ehdotus tyrmättiin W3C:n kokouksessa vuonna 2004. Enemmistö kannatti HTML:n kehittämistä XHTML 2.0 pohjalta. Mozilla ja Opera päättivät kuitenkin jatkaa HTML:n kehitystä omalla tavallaan. Apple liittyi mukaan kuvioihin ja näin syntyi järjestö WHATWG (Web Hypertext Application Technology Working Group). (Korpela 2011, 58)

HTML5:n kehityksen lähtökohtana voidaan pitää vuotta 2004, jolloin WHATWG alkoi kehittää laajennettua lomakekäsitettä Web Forms 2.0 sekä muita pienempiä laajennuksia. XHTML 2.0:n kehitys ei ollut niin sujuvaa kuin aluksi odotettiin ja vuonna 2006

W3C alkoikin kiinnostua WHATWG:in työstä, joka oli ollut tuottoisampaa. W3C alkoi tehdä yhteistyötä WHATWG:in kanssa jatkaen samalla XHTML 2.0:n kehitystä. Vuonna 2007 W3C perusti HTML5-työryhmän, joka julkaisi ensimmäisen HTML5-luonnoksen vuonna 2008. (Korpela 2011, 58) Vuotta myöhemmin XHTML 2.0:n kehitys lopetettiin tuloksettomana (Allsopp 2010, 21).

3.4 Mobiililaitteet

HTML5 on mobiililaitteiden kannalta varsin tervetullut uutuus. Se on laitteistoriippumaton kuvauskieli, joka tekee Internet-sivuista sovelluksia, jotka eivät tarvitse erillisiä liitännäisiä. (Sulkko 2011.)

Tärkeintä liitännäistä eli Flashia kehittänyt Adobe ilmoitti vuonna 2011 luopuvansa sen kehittämisestä mobiililaitteisiin ja aikovansa panostaa HTML5:een. Flash on ollut hallitseva tekniikka erilaisen median upottamiseen Internet-sivuilla jo pitkään. (Laakso 2011.) Tällainen päätös Adoben taholta kertoo siitä, että HTML5 on saavuttamassa loistavaa asemaa mobiililaitteissa.

Mobiililaitteiden määrän kasvu on ollut huimaa viime vuosina. Tietoliikenneyritys Cisco on ennustanut Internetiä käyttävien mobiililaitteiden määrän ylittävän maapallon asukkaiden väkiluvun vuoden 2012 loppuun mennessä. Arvion mukaan vuonna 2016 maailmassa olisi noin 10 miljardia Internet-yhteydellä varustettua mobiililaitetta. (Luotola 2012.) Utissivusto TechCrunchin arvion mukaan HTML5-puhelimia olisi noin miljardi vuoteen 2013 mennessä (Perez 2011a).

3.4.1 Responsive design

Yhä useammalla Internet-selaajalla on käytössään mobiililaitte perinteisen pöytäkoneen sijasta. Mobiililaitteiden määrän on arvioitu syrjäyttävän pöytäkoneet Internet-käytössä vuoden 2013 aikana. (Walsh 2010.) Tämä aiheuttaa haastetta web-suunnittelijoille, sillä tavallisille pöytäkoneille ja suurille näytöille suunnitellut sivut eivät välttämättä toimi hyvin mobiililaitteissa. Monesti mobiililaitteita varten joudutaan tekemään sivuista kokonaan oma versio. Responsive design on termi, jolla tarkoitetaan sivuston joustavaa

suunnittelua, joka pystyy vastaamaan käyttäjien erilaisten laitteiden tarpeisiin. (Andrew 2012, 366–371.)

Media Queries eli mediatyypit on tällä hetkellä W3C:n PR-statusen saanut CSS3-ominaisuus. Mediatyyppien avulla sivuston ulkoasu voidaan määrittellä mediakohtaisesti. CSS-koodiin on mahdollista tehdä omat määrytykset pöytäkoneelle, tabletille sekä älypuhelimelle. Hyvä esimerkki tästä on kolmipalstaiseksi aseteltu sivusto, joka ei tuota ongelmia pöytäkoneen suurelta näyttöpäätteeltä selattaessa. Käyttäjän vieraillessa kyseisellä sivustolla älypuhelimella, on asia kuitenkin toinen. Käytettäessä mediatyyppejä sivuston asettelua voidaan muokata mobiililaitteille sopivammaksi esimerkiksi tarjoamalla erilainen asettelu tietyille resoluutioille. Mediatyypeillä voidaan myös poistaa sivuston palstoitus siinä tapauksessa, kun selainikkuna on maksimissaan esimerkiksi 480 pikseliä leveä. Tässä tapauksessa pöytäkoneen kolmipalstainen näkymä muuttuu älypuhelimelle yksipalstaiseksi, jossa palstat ovat allekkain (kuva 9). Mediatyypeillä on myös mahdollista osoittaa eri medioille kokonaan omat tyylitiedostonsa (Andrew 2012, 371–374.)



KUVA 9. Responsive design sivujen muokkaamisessa eri medioille (Dabrowski 2012)

3.4.2 HTML5 vs. Flash

9.11.2011 Adobe vahvisti huhut ja ilmoitti lopettavansa Flashin kehittämisen mobiililaitteisiin. Adobe on hyväksynyt tappionsa HTML5:lle mobiililaitteiden osalta, mutta se

ei tarkoita vielä koko Flashin hautaamista. Adobe jatkaa Flashin kehittämistä PC:lle, mutta sovelluksissa sekä mobiililaitteissa kääntyy HTML5:n puoleen. (Winokur 2011.)

Applen edesmennyt toimitusjohtaja Steve Jobs oli aina hyvin vastahakoinen Flashin suhteen, ja Apple ilmoittikin vuonna 2007, ettei se aio tukea Flashia mobiililaitteissaan (Laakso 2011). Vuonna 2010 Jobs kirjoitti avoimen kirjeen ”Thoughts on Flash”, jossa hän esitti syitä Flashin sopimattomuuteen mobiililaitteissa. Jobsin mukaan Flash on PC-aikakauden tuotos, joka on suunniteltu käytettäväksi tietokoneella ja hiirellä. Mobiililaitteiden aikakauden tunnusmerkkejä ovat kuitenkin vähäinen virrankulutus, kosketusnäytöt sekä avoimet web-standardit, joissa kaikissa Flash Jobsin mukaan epäonnistuu. Näiden lisäksi Flashin suuri ongelma on ollut sen vajavainen tietoturva. (Jobs 2010.)

Vaikka monet uskovatkin vahvasti koko Flashin kaatumiseen, on eräs seikka, joka saattaa pitää sen pelissä mukana. Adobe:n Wallaby ja Googlen Swiffy ovat ohjelmia, joiden avulla Flashin swf-tiedostot voidaan kääntää HTML5-koodiksi. Ohjelmat ovat kuitenkin vielä kokeellisia ja ne mahdollistavat vain yksinkertaisten Flash-animaatioiden kääntämisen. (Perez 2011b.) Tehokas Flash-kääntäjä saattaisi olla aivan uusi käänne, joka vahvistaisi Adobe:n asemaa uusien web-standardien rinnalla.

3.5 Siirtyminen HTML5:een

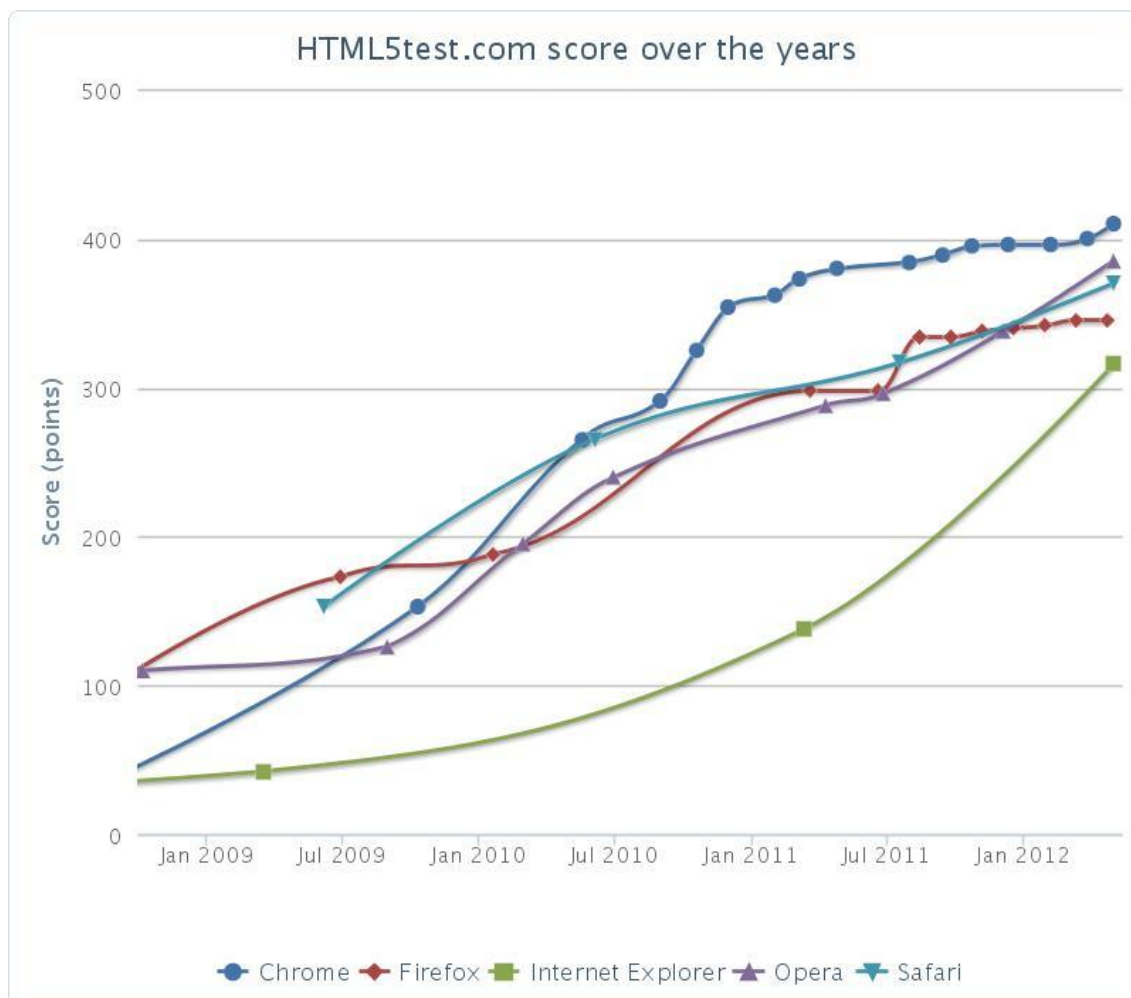
Tutkimusyhtiö Evan’s Datan selvityksen mukaan noin puolet web-kehittäjistä on siirtynyt käyttämään HTML5-spesifikaation mukaista koodia (Rinta 2012). Määrä on kasvussa ja HTML5 tulee olemaan osa koodaajien arkipäivää lähitulevaisuudessa (Vänskä 2011).

Vaikka HTML5 onkin vielä keskeneräinen, voi uusia ominaisuuksia käyttää jo tietyin ehdoin. Siirryttäessä HTML5:n mukaiseen merkkaukseen suuria muutoksia ei välttämättä tarvita. HTML5 on yhteensopiva vanhojen HTML- ja XHTML-versioiden kanssa. Vanhentuneiden elementtien käyttöä ei suositella, mutta tuki niille on silti olemassa (Allsopp 2010, 271). Vanhojen sivujen muuntaminen HTML5-sivuiksi onnistuu vaihtamalla dokumenttityyppi HTML5:n mukaiseksi. Muita uudistuksia, kuten rakennelementtejä, voi ottaa käyttöön melko huoletta. (Korpela 2011, 15.) Täysin moitteetto-

masti toimivien vanhojen sivujen kohdalla kannattaa miettiä, onko uudistusten käyttöönotto järkevää. Kokonaan uusia sivuja suunniteltaessa uusien ominaisuuksien kokeileminen on paljon suotavampaa. Monet uusista ominaisuuksista ovat hyvin nopeasti ja helposti käytettävissä. Uusia ominaisuuksia käytettäessä kannattaa miettiä sivujen kohderyhmää. Vanhat selaimet on syytä ottaa huomioon sivuilla, joiden kävijöistä suurella osalla sellainen on käytössä. Näissä tilanteissa vaihtoehtoinen sisältö on syytä tarjota JavaScriptillä. (Keith 2010, 82). Mikäli HTML5:n uudistuksissa on tarkoitukseen sopivia ratkaisuja ja vanhat selaimet pystytään huomioimaan, ei käytölle ole esteitä.

3.6 HTML5:n ongelmat

HTML5:n ollessa vielä luonnosasteella, on sen toimivuus rajoittunutta. The HTML5 test -sivuston mukaan tällä hetkellä parhaiten HTML5-ominaisuuksia tukeva selain on Maxthon 3.3.8 pisteillä 437. Toiseksi sijoittuu pistemäärällä 402 Chrome 19, joka on W3C:n selaintilastojen mukaan maailman käytetyin selain (Browser Statistics 2012). The HTML5 test on palvelu Internetissä, joka antaa pisteitä käytetylle selaimelle sen mukaan, kuinka hyvin se tunnistaa HTML5:n uusia ominaisuuksia (kuvio 10). Heikointen testissä suoriutuu Internet Explorer 9. (The HTML5 test 2012)



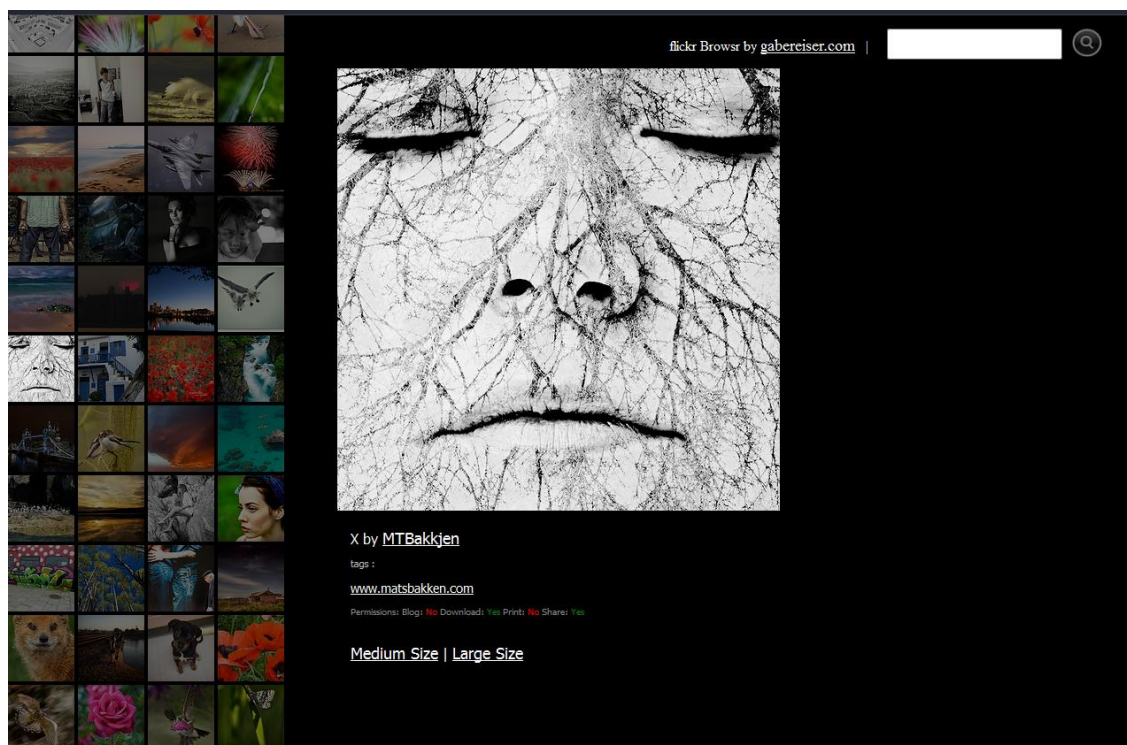
KUVA 10. HTML5-ominaisuuksia tukevat selaimet (The HTML5 test 2012)

HTML5 karsii pois vielä hetki sitten tärkeitä liittännäisiä kuten Flash. Flash on saanut osakseen kiritiikkiä liittyen sen vaarallisiin tietoturva-aukkoihin. Ilman riskialttiita liittännäisiä tietoturva kohenee, mutta HTML5 sisältää itsessään uudentyyppisiä tietoturva-vauhkia. Ennen selaimen kautta pyrittiin murtautumaan käyttäjän tietokoneelle. Web Storaagen myötä käyttäjän selaimen syntyvä tietovarasto tekee itse selaimesta hyökkäyskohteen. HTML5 tulee korjaamaan monta tietoturvaongelmaa ja sen tulevaisuus näyttää valoisaalta. HTML5:n ollessa nuori kieli, tulee ongelmiin kuitenkin alussa varautua. (Kolehmainen 2011.)

3.7 Web-sovellukset

Web Applications eli web-sovellukset ovat tietokoneella käytettävien ohjelmien kaltaisia, mutta ne toimivat käyttäjän selaimessa. HTML5:n myötä Internet-sivujen luonne on muuttumassa. Uusia ominaisuuksia käyttävät sivut eivät välttämättä näytä enää tavalli-

silta sivuilta vaan pikemminkin sovelluksilta, joita on totuttu aikaisemmin näkemään vain tietokoneiden ruudulla. Kuvassa 11 on Flickr-kuvapalvelun selaamiseen tehty HTML5-sovellus, jossa kuvien selaaminen on tehty omaperäisellä tavalla (kuva 11).



KUVA 11. Flickrin HTML5-sovellus (Reiser 2010)

Vanhanaikaiset staattiset web-sivut kuten Wikipedia ovat kokoelma dokumentteja. Modernien web-sovellusten kuten YouTube ja Facebook tunnusmerkkejä ovat vuorovaikuteisuus. Osa web-sivuista, kuten Gmail sähköposti ja toimisto-ohjelmisto Google Docs, ovat korvanneet jo osaltaan perinteisiä tietokoneohjelmia. (Kessin 2011, 1–2.)

HTML5 tarjoaa työkalut, jotka mahdollistavat webin käyttämisen yhä monipuolisempaan sovellusalustana. JavaScript ja sen avulla toimivat ohjelmointirajapinnat sekä JavaScript-kirjastot kuten jQuery mahdollistavat sovellusten toiminnallisuudet. Ominaisuudet kuten sovellusvälimuisti, offline-käyttö, paikallinen SQL-tietokanta sekä pääsy käsiksi paikallisiin tiedostoihin tekevät web-sovelluksista kilpailukykyisiä verrattaessa perinteisiin tietokoneohjelmiin. (Kessin 2011, 1–2.)

Sovelluksiin, jotka toimivat selaimessa, liittyy merkittäviä hyötyjä verrattuna tietokoneohjelmiin. Käyttäjän ei tarvitse asentaa kiintolevytilaa käyttäviä ohjelmia saati päivityksiä, mikä tekee sovelluksen kokeilemisesta helppoa ja nopeaa. Lisäksi käyttäjällä on

aina käytössään sovelluksen uusin versio. Web-sovellukset toimivat eri laitteissa ja käyttöjärjestelmissä, kunhan käytössä on uusia standardeja tukeva selain. Sovelluskehittäjien työmäärä pienenee huomattavasti, sillä sovelluksesta ei tarvitse tehdä versioita jokaiselle käyttöjärjestelmälle. (Kessin 2011, 3.)

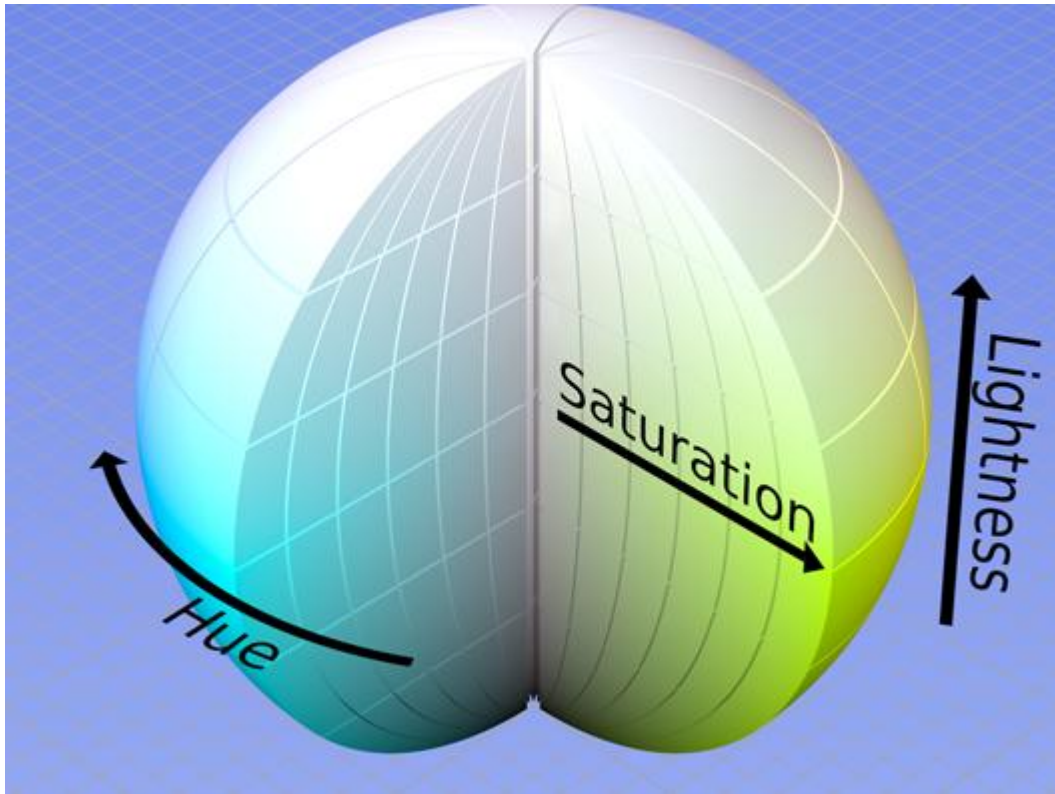
On mahdollista, että web-sovellukset korvaavat tietokoneohjelmia tulevaisuudessa. Vielä ei voida puhua kuitenkaan kuvankäsittelyohjelma Photoshopin kaltaisista raskaista sovelluksista, jotka sisältävät turhan tehokkaita työkaluja toimiakseen selaimessa (Robinson 2011.)

4 CSS3-moduulit

CSS:n viimeisin suositus CSS 2.1 julkaistiin 7.6.2011. Kehitteillä olevan CSS3:n työstäminen aloitettiin jo vuonna 1998. CSS3 ei ole enää CSS 2:n tapaan yksi laaja spesifikaatio, vaan CSS 2.1 on jaettu pienempiin osakokonaisuuksiin, moduuleihin. Spesifikaatioon on myös lisätty kokonaan uusia moduuleja, jotka ovat toisistaan riippumattomia, ja niillä on omat W3C-statukset. Tällä hetkellä kolme moduulia, Color, Selectors sekä Namespaces, ovat saaneet suositus-statusen. Color-moduuli esittelee uusia tapoja värien määrittämiseen. Selectors puolestaan laajentaa selektoreiden joukkoa ja mahdollistaa yhä monipuolisemman viittauksen elementteihin. Namespaces-moduuli määrittää syntaksin, jonka mukaan nimiavaruuksia voidaan käyttää CSS:ssä. Suositus-statusen saaneiden moduuleiden lisäksi monien moduuleiden ominaisuuksia on lisätty selaimiin. (Gasston 2011, 1–3.)

4.1 Color

CSS3 Color -moduulin suurimpana uutuutena on värien esittäminen läpinäkyvänä opasiteetin ja alfa-kanavan avulla. RGBA-muodon (red, green, blue, alpha) lisäksi värejä voidaan esittää HSL- ja HSLA-muodon avulla. HSL (hue, saturation, lightness) avulla värien säätäminen onnistuu RGB-muotoa kätevämmiin (kuva 12). Haluttaessa vaihtaa RGB-muodossa esitetyn värin sävyä, on kaikkia kolmea arvoa muutettava. HSL-muodossa voidaan pitää sama värisävy (hue) ja muuttaa hieman kylläisyyttä (saturation) tai valoisuutta (lightness). HSL-muotoon voidaan myös lisätä läpinäkyvyyttä alfa-kanavan avulla käyttäen HSLA-muotoa. Color-moduulissa on myös mukana uusia nimettyjä värejä kuten azure, coral ja darkred. (Weyl ym. 2011, 131–133.)



KUVA 12. HSL-muotoa havainnollistava värikartta (Wikipedia 2009)

4.2 Selectors

Selektoreita on ollut CSS:ssä alusta alkaen. Selektorien avulla voidaan valita HTML-dokumentista elementtejä tai niiden osia joihin tyyliä lisätään. CSS3 Selectors täydentää selektorien joukkoa ja mahdollistaa elementtien valitsemisen yhä kattavammin. (Weyl ym. 2011, 120.) Selektorit voidaan jakaa kahteen luokkaan, DOM (Document Object Model) -selektoreihin ja pseudo-selektoreihin. DOM-selektoreilla viitataan suoraan elementteihin. Esimerkissä käytetään selektoria `h3~ul` (element1~element2) asettamaan taustaväri listaelementeille (ul), joita edeltää kolmannen tason otsikkoelementti (h3) (kuva 13). (Gasston 2011, 23.)

Pseudo-luokan selektoreilla tarkoitetaan selektoreita, joilla viitataan elementin osaan tai tilaan. Esimerkissä vaihdetaan kappaleen ensimmäisen kirjaimen väri ja suurennetaan kokoa käyttämällä selektoria `:first-letter` (kuva 14). (Gasston 2011, 33.)

Lista 1

- elementti 1
- elementti 2
- elementti 3

Lista 2

- elementti 1
- elementti 2
- elementti 3

Lista 3

- elementti 1
- elementti 2
- elementti 3

KUVA 13. DOM-selektori element1~element2

Käyttämällä pseudo selektoria `:first-letter` voidaan valita kappaleen ensimmäinen kirjain

KUVA 14. Pseudo-selektori `:first-letter`

4.3 Transforms ja Transitions

CSS3 Transformista on kaksi eri spesifikaatiota, 2D Transforms sekä 3D Transforms. Molemmat ovat WD-tason luonnoksia, mutta 2D transformien tuki selaimissa on parempi. Transforms mahdollistaa elementin muodon manipuloinnin kuvankäsittelyohjelmista tutuilla funktiolla. Elementtiä on mahdollista skaalata (scale), siirtää (translate), pyörittää (rotate) ja asettaa viistoon (skew). (CSS3 transform... 2012)

CSS3 Transitions -moduuli on WD-statuksen spesifikaatio, jonka avulla CSS-ominaisuuksia voidaan muuntaa toiseen tietyn ajan kuluessa siirtymäanimaation avulla. Animaatio voidaan käynnistää elementtiä hiirellä klikkaamalla tai erilaisilla tilamääritteillä kuten kohdistettu, aktivoitu tai fokusoitu. Yleensä sivustojen navigointipalkin linkeille on toteutettu jokin korostustoiminto niitä kohdistettaessa hiirellä. Tämä voi olla esimerkiksi linkin värin muuttaminen toiseen tai fontin koon suurentaminen, joiden toteutuksessa on käytetty selektoria `:hover`. Käytettäessä ainoastaan hoveria, siirtymästä

tulee kuitenkin melko karkea. Transition avulla linkin kohdistukselle voidaan lisätä siirtymäanimaatio määrittelemällä sille kesto ja ajoitusfunktio. Transitiolla saadaan helposti aikaan pieniä animaatioita, jotka tekevät siirtymästä pehmeän ja sulavan. Aikaisemmin vastaavaan on pystytty JavaScriptin ja Flashin avulla. (Cederholm 2010, 17–19.) Transitiota voidaan käyttää yhdessä transformin kanssa. Hyvä esimerkki on elementin kääntäminen rotaten avulla, joka yhdessä transition kanssa aiheuttaa sulavan pyörimisliikkeen.

4.4 Selaimet

Webin kannalta ihanteellisessa tilanteessa luotaisiin yhteiset pelisäännöt, joita kaikki selainvalmistajat alkaisivat noudattaa. Kuten historia on osoittanut, on asia kuitenkin usein päinvastoin. Netscape piti pitkään pintansa, mutta myöhemmin webin kehitystä alkoi dominoida Microsoft. Internet Explorer 6:n lisääminen uuteen Windows XP:hen vakiinnutti aseman suosituimpana selaimena vuosiksi (David 2010, 4–6).

CSS3 Transitiot ovat hyvä esimerkki siitä, miten selainvalmistajat nykyään tuovat spesifikaation omia ominaisuuksiaan. Transitiot olivat nimittäin alun perin WebKitin kehittämä ominaisuus ainoastaan Safari-selaimeseen. WebKit on selainmoottori, jota käyttävät Applen Safari sekä Googlen Chrome. Muut selaimet Internet Exploreria lukuun ottamatta ovat lisänneet tuen transitiolle myöhemmin. Transitioiden tapaan transformit ovat myös WebKitin kehittämä ominaisuus, joka on johtanut kahteen eri W3C luonnokseen. (Cederholm 2010, 16–17, 53.)

WebKitin aikaansaannos on alkujaan myös animations, jonka W3C on myöhemmin ottanut luonnokseksi. WebKit-selainten lisäksi Firefox on lisännyt tuen animationseille. Nämä uudet ominaisuudet ovat saaneet hyvän vastaanoton, joka on pakottanut muut selaimet tulemaan perässä. (Cederholm 2010, 110.)

Mikään selain ei vielä varsinaisesti tue näitä uusia ominaisuuksia, vaan puhuttaessa tuesta tarkoitetaan prefixien avulla lisättyä testitukea. Prefixit ovat CSS-koodissa ominaisuuksien eteen sijoitettuja etuliitteitä. Kaikilla selaimilla on omat etuliitteensä; Mozillalla -moz-, Safarilla sekä Chromella -webkit-, Operalla -o- ja Internet Explorerilla -ms-. Selainvalmistajat lisäävät prefixejä, jotta uusia ominaisuuksia olisi mahdollista kokeilla

jo käytännössä ennen kuin ne otetaan osaksi spesifikaatiota. Monet ominaisuudet toimivat prefixien avulla jo varsin hyvin. Prefixeistä on myös se hyöty, että uusien ominaisuuksien testaaminen tuo esiin ongelmakohtia, joita voidaan korjata itse spesifikaatioon. (Cederholm 2010, 10.)

5 OPPIMATERIAALIN TUOTTAMINEN VERKKOON

Oppimateriaalit ovat tärkeä osa oppimista. Oppimateriaaleihin kuuluu kaikki aineisto, jota opiskelija käyttää oppimisprosessin aikana. Tuotantoprosessina verkkooppimateriaalin tuottaminen ei eroa muusta sisällöntuotannosta. Prosessi alkaa suunnittelusta ja etenee siitä toteutukseen, testaukseen ja lopulta jakeluun. (Keränen & Penttinen 2007, 148.) Tuotettaessa materiaalia verkkoon on myös hyvä tietää, mitä rajoituksia käytettävä media tuo mukanaan.

5.1 Suunnittelu ja toteutus

Ennen oppaan tekemisen aloittamista täytyi kerätä laajalti informaatiota uusimmista kirjoista ja alan tärkeimmistä web-lähteistä. Aiheeseen perehtyminen auttoi määrittelemään keskeisimmät asiat, joihin tuli perehtyä tarkemmin. Aihevalinta oli hyvin laaja joten sen käsittely kokonaisuudessaan olisi ollut liian haastava tehtävä. Perustietojen keräämisen jälkeen HTML5:n ja CSS3:n tärkeimmät piirteet oli helppo valita oppaaseen ja rajata se toteuttamiskelpoiseksi. Samankaltaisten oppaiden vertailu auttoi myös rajauksessa sekä sivun suunnittelussa.

Opinnäytetyön käytännön osuus, HTML5 & CSS3 sinne ja takaisin -verkko-opas on toteutettu käyttäen kyseisiä tekniikoita. Työ koostuu kuudesta pääosiesta: HTML5 & CSS3 sinne ja takaisin, HTML5-dokumentin rakenne, HTML5 ja selaimet, lomakkeet, Uudet ominaisuudet ja CSS3. Osiot alkavat aina teoriaosuudella, jota seuraa koodiesimerkkejä, jotka on tehty hyvin yksinkertaisiksi ja selkeiksi. Jokaista ominaisuutta seuraa taulukko, josta ilmenee selaintuki. Oppaan pääosassa ovat tekstit ja koodiesimerkit ja se on suunniteltu hyvän typografian mukaiseksi, jotta se olisi helposti luettavaa.

5.2 Typografia

Typografia on tekstin ulkoasun suunnittelua. Se on tärkeä visuaalinen elementti, jonka tavoitteena selkeys ja tyylikkyys. Tekstin ulkoasuun liittyvät kirjasinkoko, kirjasintyyli sekä kirjasinleikkaus. Kirjasimet voidaan jakaa kahteen pääryhmään, päätteellisiin an-

tiikvoihin ja päätteettömiin groteskeihin. Tunnettuja antiikvoja ovat esimerkiksi Times ja Courier. Käytettyjä groteskeja puolestaan ovat Arial ja Helvetica. Yhdessä julkaisussa ei tule käyttää liikaa eri kirjasimia, vaan maksimimääränä pidetään yleensä kolmea. (Keränen & Penttinen 2007, 170.)

Web-julkaisuun liittyy tiettyjä rajoitteita kirjasinten käytössä. Painotuotteiden pitkiin leipätekstiosuuksiin sopii päätteellinen antiikva, mutta verkkojulkaisuissa asia on toinen. Näytöltä tekstiä luettaessa päätteettömät groteskit ovat selkeämpiä kuin antiikvat, jotka ovat varsinkin pienillä resoluutioilla vaikeaselkoisia. (Sinkkonen, Kuoppala, Parkkinen & Vastamäki 2006, 124.)

Tekstin lukemiseen näytöltä liittyy keskittymistä häiritseviä tekijöitä kuten heijastukset, monitorin etäisyys sekä tekstin vierittäminen. Näistä syistä tekstin lukeminen ja ymmärtäminen on näytöltä luettaessa hitaampaa verrattuna painotuotteisiin. Näytöltä lukemista voidaan helpottaa käyttämällä lyhyitä alle 10 rivin kappaleita sekä selkeää kappalejakoja ja otsikointia. Teksti on oppimateriaalia tuotettaessa tärkeässä roolissa. Tekstin päätarcoitus on viestittää lukijoille haluttuja asioita ja mielikuvia. Asioista voidaan viestiä monella eri tavalla riippuen kirjoitustyylistä. (Keränen & Penttinen 2007, 170–171.)

6 POHDINTA

Opinnäytetyöraporttiin sekä käytännön osuuden tuloksena syntyneeseen oppaaseen on sisällytetty suunnitellut ominaisuudet ja olen lopputulokseen tyytyväinen. Oppaan sisältö on ajankohtainen ja kattava ja sen avulla saa kokonaiskuvan uusien verkkostandardien perusteista. Raportti täyttää sille määritellyt tavoitteet. Siinä käydään läpi ominaisuudet, joiden perusteella uusien standardien keskeinen asema Internet-tekniikoiden joukossa on perusteltu. Opas ei ole kuitenkaan vielä päätynyt loppukäyttäjille, joten on vaikea arvioida, täytyvätkö sille määritellyt tavoitteet. Toivon, että opiskelijat huomioivat työni ja siitä on apua monille. Mikäli opas saa hyvän vastaanoton, voi sen jatkokehityksessä olla ainekset toisen opiskelijan opinnäytetyölle tulevaisuudessa. Opinnäytetyössä haastavaa oli lähdekritiikin kirjoittaminen sillä suuri osa käytetyistä lähteistä oli standardeja kehittäväältä organisaatiolta tai siihen liittyviltä tahoilta.

Opinnäytetyön käytännön osuuden työstämisen aikana ilmeni, että työ on melko laaja, eikä siinä ole tilaa kaikille kiinnostaville ominaisuuksille. Oppaan laatimisen aikana kiinnostus moneen ominaisuuteen syveni. Aiheista responsive design tai canvas-elementti olisi varmasti riittänyt asiaa omiksi töikseen. HTML5- ja CSS3-tekniikat olivat ennestään hieman tuttuja, mutta tämän hetkisen tietämyksen perusteella olisin todennäköisesti päätenyt syventymään toiseen edellä mainituista aiheista.

Oppaan elinkaaren kannalta jatkokehitys on tärkeää. Arvion mukaan HTML5-standardi on valmis vuonna 2014. Vuoteen 2014 mennessä spesifikaatio voi muuttua ja monet oppaan ulkopuolelle jääneet ominaisuudet tulevat olemaan tärkeässä asemassa. Vielä ei tiedetä varmaksi, kuinka laaja kokonaisuus HTML5 tulee olemaan. Vuonna 2014 oppaan päivittäminen olisi ajankohtaista. Opasta voisi myös laajentaa kattamaan JavaScriptin, sillä se on niin monen ominaisuuden toiminnallisuuden takana. Esimerkkien interaktiivisuutta voitaisiin parantaa mahdollistamalla koodin muokkaaminen käyttäjille.

LÄHTEET

Addison, P. 1998. A history of HTML. Luettu 10.2.2012.
<http://www.w3.org/People/Raggett/book4/ch02.html>

Allsopp, J. 2010. Developing with Web Standards. Berkeley: New Riders.

Andrew, S. 2012. The CSS3 Anthology. Collingwood: SitePoint.

Bernes-Lee, T 2012. The WorldWideWeb browser. Luettu 17.3.2012.
<http://www.w3.org/People/Berners-Lee/WorldWideWeb.html>

Browser Statistics 2012. Luettu 1.5.2012.
http://www.w3schools.com/browsers/browsers_stats.asp

Cederholm, D. 2010. CSS3 For Web Designers. New York: A Book Apart.

CSS3 transform Property 2012. Luettu 24.1.2012.
http://www.w3schools.com/cssref/css3_pr_transform.asp

Current Members 2012. Luettu 23.5.2012
<http://www.w3.org/Consortium/Member/List>

Dabrowski, M. 2012. Responsive Design and Its Impact on Mobile. Luettu 28.5.2012.
http://www.systemsalliance.com/who-we-are/insights/SAI%20Blog/Responsive_Design_and_Its_Impact_on_Mobile

David, M. 2010. HTML5: Designing Rich Internet Applications. Amsterdam: Focal Press

Extensible Markup Language (XML) 2012. Luettu 13.2.2012.
<http://www.w3.org/XML/>

Facebook 2012. Luettu 15.5.2012.
<http://developers.facebook.com/html5/build/features/location/>

Gasston, P. 2011. The Book of CSS3. San Francisco: No Starch Press.

Google 2012. Luettu 15.5.2012.
<https://developers.google.com/webmasters/state-of-the-web/2005/classes?hl=fi-FI>

HTML & CSS 2012. Luettu 10.2.2012.
<http://www.w3.org/standards/webdesign/htmlcss>

HTML DOCTYPE Declaration 2012. Luettu 20.3.2012
http://www.w3schools.com/tags/tag_doctype.asp

HTML5 Web Storage 2012. Luettu 10.1.2012.
http://www.w3schools.com/html5/html5_webstorage.asp

- Jobs, S. 2010. Thoughts on Flash. Luettu 18.2.2012
<http://www.apple.com/hotnews/thoughts-on-flash/>
- Keith, J. 2010. HTML5 For Web Designers. New York: A Book Apart.
- Keränen, V. & Penttinen, J. 2007. Verkko-oppimateriaalin tuottajan opas. Helsinki: WSOYpro/Docendo.
- Kessin, Z. 2011. Programming HTML5 Applications. California: O'Reilly Media.
- Kolehmainen, A. 2011. Html5 mahdollistaa verkkohyökkäykset suoraan selaimesi. Luettu 5.5.2011.
http://www.mikropc.net/kaikki_uutiset/html5+mahdollistaa+verkkohyokkaykset+suoraan+selaimesi/a742973
- Korpela, J. 2011. HTML5 - uudet ominaisuudet. Jyväskylä: WSOY.
- Laakso, H. 2011. Adobe lopettaa Flashin mobiilista. Luettu 1.2.2012.
http://www.tietoviikko.fi/kaikki_uutiset/adobe+lopettaa+flashin+mobiilista++keskittyy+html5een/a719758
- Luotola, J. 2012. Mobiililaitteita on kohta enemmän kuin ihmisiä. Luettu 18.2.2012.
http://www.tietoviikko.fi/kaikki_uutiset/mobiililaitteita+on+kohta+enemman+kuin+ihmisia/a779136
- MUDCUBE 2012. Luettu 24.3.2012
<http://mudcu.be/software/>
- Nykänen, O. 2003. W3C pähkinänkuoressa. Luettu 20.4.2012.
<http://www.w3c.tut.fi/reports/2003/0113aboutw3c/index.html>
- On SGML and HTML 2012. Luettu 14.2.2012.
<http://www.w3.org/TR/html4/intro/sgmltut.html>
- Parviainen, J. 1999. SGML-, HTML-, ja XML-kielten eroja ja yhtäläisyyksiä. Luettu 13.2.012.
<http://www.tml.tkk.fi/Studies/Tik-110.300/1998/Essays/merkkauselet.html>
- Perez, S. 2011a. Forecast: 1 Billion HTML5 Phones By 2013. Luettu 18.2.2012.
<http://techcrunch.com/2011/12/07/forecast-1-billion-html5-phones-by-2013/>
- Perez, S. 2011b. Google Swiffy Converts Flash to HTML5. Luettu 1.6.2012.
<http://www.readwriteweb.com/mobile/2011/06/google-swiffy-converts-flash-to-html5.php>
- Rinta, N. 2012. Kehittäjät siirtyvät html5:een. Luettu 1.5.2012.
http://www.tietoviikko.fi/kaikki_uutiset/kehittajat+siirtyvat+html5een/a755915
- Reiser, G. 2010. Flickr Browsr. Luettu 1.6.2012.
<http://www.gabereiser.com/articles/web-development/2010-08-10/flickr-browsr>
- Robinson, D. 2011. HTML5 web apps to replace native applications. Luettu 1.6.2012.

<http://www.v3.co.uk/v3-uk/interview/2045976/html5-web-apps-replace-native-applications/page/3>

Sani, I. 2010. YouTube testaa Flashista luopumista. Luettu 14.3.2011.
<http://www.tietoviikko.fi/kehittaja/youtube+testaa+flashista+luopumista+ndash+tilalle+html5/a366859>

Sinkkonen, I., Kuoppala, H., Parkkinen, J. & Vastamäki, R. 2006. Käytettävyyden psykologia. 3. painos. Helsinki: Edita Publishing Oy.

Standards 2012. Luettu 23.5.2012.
<http://www.w3.org/standards/>

Sulkko, S. 2011. Html5 - mikä se on ja mitä se tarkoittaa? Luettu 1.2.2012.
<http://www.tietoviikko.fi/msareena/msblogit/bittiviidakko/html5+ndash+mika+se+on+ja+mita+se+tarkoittaa/a688822>

The CSS saga 2012. Luettu 28.3.2012.
<http://www.w3.org/Style/LieBos2e/history/>

The HTML5 test 2012. Luettu 1.1.2012.
<http://html5test.com/results/desktop.html>

Vänskä, O. 2011. Html5:stä tulee koodaajan arkipäivää. Luettu 1.5.2012.
http://www.mikropc.net/kaikki_uutiset/html5sta+tulee+koodaajan+arkipaivaa/a740441?s=l&wtm=mikropc/-15122011&

W3C Invites Broad Review of HTML5 2011. Luettu 26.03.2012.
<http://www.w3.org/2011/05/html5lc-pr.html.en>

Walsh, M. 2010. Mobile To Outpace Desktop Web By 2013. Luettu 23.4.2012.
<http://www.mediapost.com/publications/article/120590/>

Weyl, E., Lazaris, L. & Goldstein, A. 2011. HTML5 & CSS3 For The Real World. Collingwood: SitePoint.

Wikipedia 2009. Luettu 4.3.2012.
http://en.wikipedia.org/wiki/File:HSL_color_solid_sphere_munsell.png

Winokur, D. 2011. Flash to Focus on PC Browsing and Mobile Apps; Adobe to More Aggressively Contribute to HTML5. Luettu 18.2.2012.
<http://blogs.adobe.com/conversations/2011/11/flash-focus.html>

XML Syntax Rules 2012. Luettu 14.2.2012.
http://www.w3schools.com/xml/xml_syntax.asp

YouTube 2012. Luettu 27.3.2012.
<http://www.youtube.com/html5>