
FUNity-opinnäytetyöjärjestelmän käyttöliittymän kehittäminen

HTML5 ja CSS3 Web-kehityksessä



Hämeen Ammattikorkeakoulun opinnäytetyö

Tietojenkäsittelyn koulutusohjelma

Visamäki, kevät 2013

Jani-Markus Nieminen



Visamäki
Tietojenkäsittely
eLearning ja Multimedia

Tekijä	Jani-Markus Nieminen	Vuosi 2013
Työn nimi	FUNity-opinnäytetyöjärjestelmän käyttöliittymän kehittäminen	

TIIVISTELMÄ

Opinnäytetyön toimeksiantaja toimi FUNity Oy, joka tarjosi aiheen opinnäytetyölle järjestelmän ulkoasun kehittämisen parissa. FUNity on järjestelmä opinnäytetöiden ja työharjoitteluiden budjetointiin, työvaiheiden seurantaan sekä ohjaajien ja opiskelijoiden hallinnoimiseen. Opinnäytetyön tavoitteena oli löytää ratkaisu suurimpaan käytettävyysongelmaan, järjestelmän päävalikkorakenteeseen, uusia HTML5- ja CSS3-tekniikoita hyödyntäen. Samalla opinnäytetyössä kartoitetaan mitä HTML5- ja CSS3-tekniikat ovat ja kuinka paljon niiden uusia ominaisuuksia voidaan jo hyödyntää web-kehityksessä.

Lähteenä opinnäytetyössä hyödynnettiin luotettuja ja tunnettuja Internet-sivustoja sekä suomalaista, että ulkomaista tuoretta kirjallisuutta. Useita lähteitä yhdistelemällä vältyttiin tiedon ristiriitaisuuksilta, sillä HTML5 ja CSS3 ovat vielä kehitysasteella sekä muutosalttiita tekniikoita. Käytännön osuuden tulokset, FUNityn navigointimallit, perustuvat Internetissä suosittuihin navigointiratkaisuihin. Eri navigointimallien hyötyjä analysoin itse ja haastattelujen avulla kartoitettiin navigointiratkaisujen hyviä ja huonoja puolia. Lopputuloksena opinnäytetyö on paketti HTML5:n ja CSS3:n käytöstä ja tekniikoiden hyödyntämisestä osana FUNity-opiskelija- ja ohjaajajaportaalin navigointia.

Opinnäytetyön aikana luoduista viidestä erilaisesta navigointiratkaisusta, voidaan jokaista hyödyntää osana FUNitya. Oli tehtävä kompromissi, mille osa-alueelle annetaan eniten painoarvoa: käytettävyys, näyttävyyys, teknisyyys vai rakenne. Jatkossa navigointia ja koko sivuston käytettävyyttä voidaan kehittää opiskelijoiden näkökulmasta, mikäli sille on tarvetta. Myös mobiiliystävällisen sivuston ja sille sopivan navigoinnin tarvetta tulee miettiä erikseen.

Avainsanat HTML5, CSS3, Web-kehitys, käytettävyys

Sivut 43 s.

Visamäki
Degree Programme in Business Information Technology
eLearning and Multimedia

Author	Jani-Markus Nieminen	Year 2013
Subject of Bachelor's thesis	FUNity ethesispass user interface development	

ABSTRACT

The thesis was commissioned by FUNity Oy. to develop system's front-end. FUNity is a system intended for budgeting the thesis process and practical trainings, following operations and managing students and instructors. The main goal of the thesis was to find a solution to the biggest usability problem, main navigation of the portal, using the latest web techniques HTML5 and CSS3. Also to identify in the process what HTML5- and CSS3-techniques are and how many of the new properties can already be used while developing websites.

Well known, trusted websites and Finnish as well as foreign literature were used as a main source of the thesis. Conflicts of information were avoided by combining multiple sources, because HTML5 and CSS3 are still under development and susceptible to changes. The results of the practical contribution of the thesis, navigation templates of FUNity, are based on popular navigation solutions around Internet. Pros and cons of different navigation solutions were analyzed by the author and the interviewees. As a result the thesis there is a guide of the usage of HTML5- and CSS3-techniques in general and as a part of FUNity student and instructor portal navigations.

Every one of the five different navigation solutions developed under the contribution of the thesis can be used as a part of the FUNity system. A compromise must be made based on which sections are given the most stature: usability, appearance, technicality or structure. In the future the navigation and usability of the whole website can be developed from the students' point of view, if it is needed. The need for a mobile friendly website and navigation has to be evaluated separately.

Keywords HTML5, CSS3, front-end development, usability

Pages 43 p.

SISÄLLYS

TERMIT JA LYHENTEET.....	1
1 JOHDANTO.....	2
2 MITÄ OVAT HTML JA CSS.....	3
2.1 HTML-historia.....	3
2.2 CSS-historia.....	4
2.3 HTML ja CSS nykypäivänä.....	4
3 HTML5.....	6
3.1 HTML5 uudet ominaisuudet.....	6
3.1.1 Uudet rakenne-elementit.....	7
3.1.2 Lomakkeet ja syöttökentät.....	8
3.1.3 Mediaelementit.....	9
3.1.4 Canvas piirtotoiminnot.....	10
3.1.5 HTML5 web-muisti.....	11
3.1.6 Paikannus.....	12
3.2 HTML5 ja selaintuki.....	13
4 CSS3.....	15
4.1 CSS3 uudet ominaisuudet.....	15
4.1.1 Media Queries.....	15
4.1.2 Tekstiefektit ja typografiset tyylit.....	17
4.1.3 Reuna- ja varjoefektit.....	18
4.1.4 Useat taustakuvat.....	19
4.1.5 Värit ja läpinäkyvyys.....	20
4.1.6 Siirtymäefektit ja animaatiot.....	21
4.2 CSS3 ja selaintuki.....	22
5 FUNITY ETHESISPASS -JÄRJESTELMÄ.....	24
5.1 Järjestelmän ominaisuudet.....	24
5.2 FUNityn päävalikkorakenne ja sivustolla navigoiminen.....	25
5.3 FUNityn päävalikkorakenteen ongelmat.....	26
6 FUNITYN KÄYTTÖLIITTYMÄN PÄIVITTÄMINEN.....	29
6.1 Esivalmistelut.....	29
6.2 Yläbanneri.....	29
6.3 Opiskelijaportaalin navigointiratkaisut.....	30
6.3.1 Alasvetovalikko.....	30
6.3.2 Suuret painikkeet.....	31
6.3.3 Vaihtoehtoinen sivumenu.....	32
6.4 Ohjaajapuolen navigointiratkaisut.....	33
6.4.1 Megamenu.....	34
6.4.2 Hover-sivumenu.....	35
6.5 HTML5 ja CSS3 osana FUNitya.....	36

7	TULOSTEN ANALYSOINTI	37
7.1	Opiskelijaporttiin menuratkaisujen analysointi	37
7.2	Ohjaajapuolen menuratkaisujen analysointi.....	38
7.3	Jatkokehitys	38
8	YHTEENVETO	40
	LÄHTEET	41

TERMIT JA LYHENTEET

HTML	Hypertekstin merkintäkieli, jonka avulla luodaan Internet-sivustojen rakenne ja sisältö
CSS	Tyyliohjeet, joiden avulla muotoillaan ja luodaan Internet-sivustojen ulkoasu
HTML5	HTML-kielen uusin, vielä kehitysasteella oleva versio
CSS3	CSS-tyylien uusin, vielä kehitysasteella oleva versio
Elementti	Yksittäinen muuttuja HTML:n tai CSS:n muodostamassa tietorakenteessa
Moduuli	Itsenäisesti toimiva osa kokonaisuutta
W3C	WWW-standardeja kehittävä yritysten ja yhteisöjen kansainvälinen yhteenliittymä
WHATWG	Yhteisö, joka ylläpitää ja kehittää HTML kieltä
Portaali	Verkkopalvelu, joka kokoaa yhtenäisen käyttöliittymän useille eri toiminnallisuuksille
JavaScript	Web-ympäristössä käytetty komentosarjakieli, jonka avulla luodaan dynaamista toiminallisuutta sivustoille
Evästeet	Dataa, jota web-palvelimet tallentavat käyttäjien tietokoneelle

1 JOHDANTO

Opinnäytetyö on suunnattu web-sivustojen kehittäjille ja erityisesti opiskelijoille, joille uudet HTML5- ja CSS3-tekniikoiden mahdollisuudet ovat vielä epäselviä. Vaikka uudet standardit ovat vielä kehitysasteella, on suurinta osaa uusista elementeistä ja moduuleista turvallista käyttää Internet-sivuston kehittämisessä. Opinnäytetyön toimeksianto, FUNity-järjestelmän uuden menurakenteen kehittäminen, tarjosi mahdollisuudet päästä käyttämään uusia tekniikoita ulkoasun kehittämisen yhteydessä. FUNity-järjestelmän käyttöliittymä toimii portaalin kautta ja sen päävalikokorakenteessa on selainkohtaisia ongelmia. Opinnäytetyön tavoitteena on kattaa HTML5- ja CSS3-tekniikoiden käyttöä 2013-luvun web-kehityksessä ja selvittää, miten niitä voidaan hyödyntää osana FUNity-järjestelmän ulkoasua.

Tutkimuskysymykset joihin opinnäytetyö vastaa, ovat mitä uusia ja tärkeitä ominaisuuksia HTML5 ja CSS3 tuovat web-kehittäjille? Miten FUNity-opinnäytetyöjärjestelmän portaalin navigointi voisi toimia parhaiten opiskelija- ja ohjaajapuolella? Mitä uusista HTML5- ja CSS3-ominaisuuksista voidaan hyödyntää käytössä olevan web-sivuston parantamiseksi?

Opinnäytetyön ja tutkittavan aiheen laajuudesta johtuen, lukijan tulisi hallita HTML- ja CSS-kielten perusteet, sillä opinnäytetyö kattaa laajemmin uusia ominaisuuksia. Opinnäytetyössä ei myöskään syvennytä JavaScriptin käyttöön, vaikka se on tärkeä osa web-sivujen suunnittelussa. Tarkoituksena on tuoda esiin JavaScriptiä korvaavia HTML5- ja CSS3-ominaisuuksia.

Opinnäytetyön teoria nojaa pitkälti mahdollisimman tuoreeseen kirjallisuuteen sekä luotettujen web-sivustojen tietoon. Muun muassa W3C, joka ylläpitää ja kehittää www:n standardeja, osoittautui luotettavaksi tiedonlähteeksi. Päälähteenä tutkimuksissa oli suomalaisen tietotekniikan ammattilaisen Jukka K. Korpelan suomeksi kirjoittama teos HTML5 käytössä: HTML5 - Uudet ominaisuudet (2011). CSS3-teoriaosuus perustuu pitkälti kirjaan: The book of CSS3 - A developers guide to the future of web design (2011), jonka on kirjoittanut Peter Gasston. Oheiset kirjat tarjoavat tietotekniikan ammattilaisten näkökulmia uusien tekniikoiden hyödyntämiseen. Vaikka HTML5 ja CSS3 ovat muutosalttiita, yhdistelemällä lähteitä ja omia huomioita, ei ristiriitaisuuksista ollut ongelmaa. Uudet tekniikat ovat saaneet paljon huomiota Internetissä ja yhä useammat uudet web-sivustot hyödyntävät uusia ominaisuuksia, tämän vuoksi ajankohtaisen tiedon löytäminen on helppoa.

FUNity-järjestelmän ja sen käytettävyysongelman, pääsivuston navigoinnin, uusimista kartoitettiin luomalla erilaisia navigointimalleja, jotka pohjautuvat Internetissä suosittuihin menuunalleihin. Eri navigointiratkaisujen käytettävyyttä ja ulkoasua pohdin itse teknillisestä näkökulmasta ja uusia näkökulmia saatiin haastatteleamalla lyhyesti asiantuntevia Hämeen ammattikorkeakoulun opettajia. Konkreettisenä tuloksena on viisi erilaista navigointiratkaisua, joita voidaan soveltaa FUNity-portaalin navigoinnissa.

2 MITÄ OVAT HTML JA CSS

HTML ja CSS ovat kaksi web-sivustojen luomiseen käytettyä pääteknologiaa. HTML on avoimen standardin kuvauskieli, jonka avulla on mahdollista esittää tietoa Internetissä. HTML-koodi on kokoelma merkintäkieltä, jonka Internet selaimet tulkitsevat ja kääntävät näkyväksi. (Rouse 2005) HTML:n avulla luodaan Internet-sivustojen rakenne ja sisältö, kuten sivuston teksti, kuvat ja linkit (HTML & CSS 2013). Yhtä itsenäistä osaa merkintäkielestä kutsutaan elementiksi tai puhekielessä HTML-tagiksi. Useita elementtejä käytetään pareittain, jolloin ensimmäinen osa ilmoittaa elementin alun ja seuraava lopun. (Rouse 2005)

CSS eli kaskadisets tyylitiedostoja, joihin kirjoitetaan Internet-sivuston muotoilu. Muotoilu on erotettu HTML-tiedoista, jotta sivuston tyyli ja sisältö pysyisivät erillään. Muotoilutiedosto kertoo Internet selaimille, miten sisältö esitetään. CSS-tyylejä kohdistetaan suoraan HTML-merkintäkielen elementteihin, jolloin voidaan määrittellä kuinka yksittäinen elementti näyttää selaimessa. Yhtä muotoilutiedostoa muuttamalla voidaan muuttaa koko sivuston ulkoasua helposti. Uudella CSS3-versiolla tyylejä voidaan asettaa myös käytettävän laitteen mukaan, jolloin sivusto näyttää erilaisilla eri käyttäjille päätelaitteesta riippuen. (Kyrnin n.d.a)

2.1 HTML-historia

HTML on syntynyt ja kehittynyt varsin lyhyessä, mutta myrskyisessä ajanjaksossa (Korpela 2011, 24). Se syntyi Sveitsissä CERN hiukkasfyysikan laboratoriossa, johon oli kokoontunut fyysikoita ympäri maailmaa. Tim Bernes-Lee, jota kutsutaan webin keksijäksi, halusi löytää keinon, jolla fyysikot voisivat jakaa tietoa globaalisti. Eikä vain ladata tiedostoja tietokoneelle, vaan linkittää tekstiä suoraan tiedostoihin. Toisin sanoen, Bernes-Lee halusi löytää tavan käyttää hypertekstiä, jolloin dokumentteja voitaisiin linkittää yhteen ja niiden välillä voitaisiin siirtyä näytöllä nappeja painamalla. Ensimmäinen web-selaimen prototyyppi syntyi 1990. (Addison 1998)

Vuosien 1990-1994 aikana ei ollut mitään virallisia määrittelyjä HTML-kielelle ja kieli muuttui selaintenkehittäjien toimesta. Tästä varhaisesta HTML-kielestä on käytetty nimitystä HTML 1.0. (Korpela 2011, 24) Vuonna 1994 Dan Connolly kollegoineen keräsi eniten käytetyt HTML-elementit yhteen ja kirjoitti ne ylös. Tämä oli HTML 2 -version alku. Samana vuonna Jim Clark ja Marc Andreessen perustivat yrityksen, jonka tuote oli uusi selain: Netscape. Netscape kehitti HTML-tageja itsenäisesti ja ajoi osaltaan HTML:n kehitystä eteenpäin. Webin standardisoimiseksi Bernes-Leen johdolla perustettiin HTML-standardia ylläpitävä kansainvälisten yritysten ja yhteisöjen yhteenliittymä W3C. (Addison 1998) Vielä tänä päivänäkin W3C kehittää ja ylläpitää HTML-standardia.

1995 ja 1997-lukujen välillä kehitettiin ja määriteltiin HTML 2.0, HTML 3.2 ja HTML 4.0 -versioita sekä vakiinnutettiin HTML:n asemaa. Vuodesta 1997 vuoteen 2003 saakka HTML:n kehitys oli pysähtynyttä. Päivityk-

siä tuotiin ainoastaan HTML 4.0:aan, josta viimeiseksi versioksi jäi 4.01, joka sitten muovattiin XML-pohjaiseksi XHTML 1.0-versioksi. (Korpela 2011, 24) XML on HTML:n kaltainen merkintäkieli, joka on suunniteltu kuljettamaan dataa (W3Schools 2013a).

HTML5:n kehitys alkoi vuosina 2004-2006. Applen, Mozillan ja Operan perustama WHATWG alkoi kehittää ajatusta kokonaan uudesta HTML-versiosta, joka rakentuisi vanhojen HTML-versioiden pohjalta. Tällöin kaavailtiin muun muassa lomaketoiminnallisuuden laajennusta, joka myöhemmin tuli osaksi HTML5:ta. Vuodesta 2007 eteenpäin HTML5:n kehitys jatkui WHATWG ja W3C:n yhteistyönä sen suosion kasvaessa. HTML5:n kehitys on saanut alkunsa ajatuksesta, että HTML ja selaimet kehittyisivät rinnakkain selainten valmistajien tehden yhteistyötä. Sillä sellaisia uusia piirteitä, joita ei ole tarkkaan määritelty ja kirjattu, ei ole toteutettu ollenkaan tai ne ovat toteutuneet hitaasti ja epäyhtenäisesti. (Korpela 2011, 24).

2.2 CSS-historia

CSS:n tarina alkoi 1994-luvulla. CERN:n tutkimuslaitoksella työskennellyt Håkon Wium Lie näki puutoksen aikansa web-sivustoissa, sillä sivustoja ei voitu muotoilla esimerkiksi sanomalehden muotoon. Myös monet aikansa web-sivustojen kirjoittajat valittivat, ettei heillä ollut tarpeeksi vaikutusvaltaa, miltä heidän sivustonsa näyttivät. Kolme päivää ennen Netscapen julkistamista, Håkon julkaisi ensimmäisen luonnoksen CSS kaskadisista tyyli tiedostoista. CSS ei ollut aikansa ainoa tyyli ehdokas. Kamppailua käytiin siitä, kuinka paljon sivustojen luojilla olisi valtaa sivustojen ulkoasuun suhteessa käyttäjiin. CSS voitti tyyli ehdokkaiden kamppailun asettumalla kehittäjien ja käyttäjien väliin. Sivustoja ei voitaisi suunnitella ilman toista osapuolta, vaan kummankin toiveet olisi kuultava. (Addison 1999)

CSS:ta tuli viimein W3C:n suositus joulukuussa 1996, kun Microsoft ilmoitti lisäävänsä CSS-tuen selaimiinsa. Pian Netscape seurasi Microsoftin jalanjalkia, koska oli tärkeää etteivät selaimet tukisi eri määrittelyjä. (Addison 1999) Nykyisinkin vielä käytössä oleva CSS 2 -versio julkaistiin 1997. Nyt puhutaan "uudesta" CSS3-versiosta, mutta itse asiassa CSS3 ei ole kovin uusi, sillä sen kehitys aloitettiin jo vuonna 1998. Koska selainten CSS 2 toteutus oli hidasta ja sekavaa, W3C päätti siirtää työmäärää CSS3-versiosta takaisin CSS 2 kehitykseen. Vasta vuonna 2005 CSS3-moduulit otettiin takaisin luonnos-asteelle ja niiden käsittely alkoi alusta. (Gasston 2011)

2.3 HTML ja CSS nykypäivänä

Nykyisin web-sivustojen luomisessa käytetään vielä vahvasti HTML 4 ja CSS 2.1 -versioiden ominaisuuksia. Uudet versiot HTML5 ja CSS3 ovat nostaneet suosiotaan, mutta uusia ominaisuuksia käytetään harkiten, sillä selaintuki on vielä vajavaista. Käytännössä uudet versiot tuovat mukanaan

paljon uusia työkaluja web-kehittäjille, muuttamatta sivustojen käytännön-toteutusta. HTML5:sta on alettu kutsua webin tulevaisuudeksi, mutta sen sisältö on vielä keskeneräinen ja kiistanalainen. (Korpela 2011, 19) CSS3 tuo uusia ominaisuuksia vanhan version tueksi, eikä se mullista CSS:n perusrakennetta (Korpela 2011, 285).

Uusien tekniikoiden kehitystyö jatkuu WHATWG:ssä, kun W3C hyväksyy tuloksia ja vahvistaa ne suositukseksi (Korpela 2011, 58). Siksi vielä tässä vaiheessa on oltava kriittinen uusia ominaisuuksia opiskellessa, sillä suositukset saattavat vielä muuttua. Koska kehitys on nopeaa, kannattaa ilmaisu HTML5 ymmärtää muodossa 'nykyinen HTML5-luonnos'. (Korpela 2011, 25)

Selaimet päivittyvät tukemaan uusia ominaisuuksia jatkuvasti. Tärkeintä on tässä vaiheessa opetella HTML5- ja CSS3-ominaisuuksia ja pyrkiä kohti uutta. Mitään erityistä siirtymistä ei tarvitse tehdä, vaan olemassa oleviin sivustoihin voidaan ruveta lisäämään uusia HTML5- ja CSS3-ominaisuuksia. Selainten toimivuuteen voidaan varautua eri keinoin, mikäli täyttä tukea ei vielä ole selaintenkehittäjien toimesta. (Korpela 2011, 15) Web-kehittäjän ei tarvitse valita työkaluikseen HTML5 ja CSS3:ta, sillä HTML 4 ja 5 sekä CSS 2 ja 3 -versiot sekoittuvat helposti yhteen. Uusista ominaisuuksista voidaan valita käyttöön sopivimmat tilanteen mukaan.

3 HTML5

Verkkosivustojen luomisen peruskieleksi on muodostumassa HTML5. Se poikkeaa kuitenkin suurelta osin HTML:n aiemmista versioista. HTML5 on myös vaikeasti määriteltävä käsite. Yleensä HTML5:stä puhuttaessa tarkoitetaan uusia käsitteitä, jotka eivät kuitenkaan välttämättä sisälly HTML5-luonnoksiin. Kuitenkin puhuttaessa HTML5:stä tulkitaan se kokonaisuudeksi, joka käsittelee myös ohjelmoinnin ja sovellusten puolelle meneviä asioita. HTML5 on siis oikeasti yleisnimitys työlle, jota tehdään HTML-kielen laajentamiseksi ja määrittelemiseksi. Ei vielä tiedetä mitä HTML5 tulee loppujen lopuksi olemaan ja joidenkin arvioiden mukaan kielen virallinen määrittely tehdään vasta 2020-luvulla. Kuitenkin kaikki merkittävä HTML ja Webbiin liittyvä kehitystyö, jota voidaan väljästi se-
littää kuuluvaksi HTML:ään, on HTML5:sta. (Korpela 2011, 18-19)

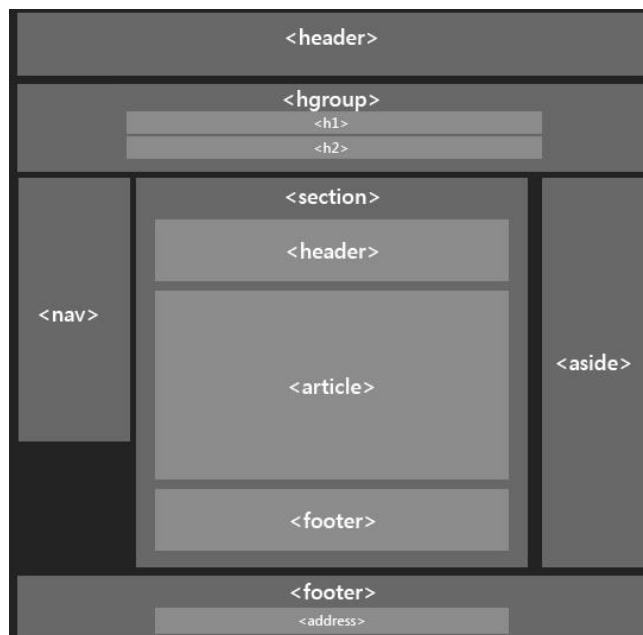
3.1 HTML5 uudet ominaisuudet

HTML5:n uudet ominaisuudet liittyvät vahvasti JavaScript kieleen ja sen käyttöön (Korpela 2011, 71). HTML5:een luettuja ominaisuuksia on runsaasti. Canvas-elementti on piirustusalueesta 2D piirustuksiin. Elementit median toistoon ja median upottamiseen web-sivustolle, video- ja audio-elementit, ovat uudet vastineet nykyisin vielä suosituille Adoben kehittämälle Flash-kehitysympäristölle. Paikkatietojen avulla voidaan selvittää käyttäjän sijainti ja tarjota sisältöä kohdennetusti sijainnin mukaan. Selaimessa toimiva web-muisti tuo uuden tavan tiedon tallettamiselle. Lomakkeiden uudet piirteet, kuten lomakekentän tarkastaminen ilman tarvetta JavaScriptille, ovat myös uutta HTML5:ssä. HTML5 tarjoaa myös kavi-
vattuja uusia rakennetyyppejä HTML-dokumentin jäsentelyyn. Uudet rakenne-elementit vähentävät muotoilussa div-elementin käyttöä. (Korpela 2011, 13)

HTML5 on vielä keskeneräinen luonnos, eivätkä kaikki piirteet vielä toimi kokonaan. On huomioitava eri selainten HTML5-tuki ja arvioitava itse kokeilemalla mitä piirteitä kannattaa jo käyttää. Suurta osaa uusista ominaisuuksista voidaan käyttää parantamaan sivustojen esitystapaa uusilla selaimilla, kuitenkin säilyttäen toiminnallisuuden vanhempien selainversioiden välillä. (Korpela 2011, 19-20) Internetistä löytyy myös käytännöllisiä ja kokoajan päivittyviä ohjeita HTML5-elementtien turvalliseen käyttöön.

3.1.1 Uudet rakenne-elementit

HTML-kielen rakenne-elementtejä käytetään HTML-dokumentin sisäisen rakenteen kuvaamiseen. HTML 4 -kielessä rakenne-elementtien määrä on melko suppea, joka on johtanut kehittäjät korvaamaan yleisesti käytettyjä rakenteita div- ja span-elementeillä. Jatkuva div-elementtien käyttö on sekavaa ja epäjärjestyksellistä sekä saattaa haitata kehittäjää koodinpuolella. Esimerkiksi usein web-sivun alalaitaan luodaan osio, johon sijoitetaan yhteistiedot, sivuston tekijätiedot tai linkkejä sivun muuhun sisältöön. Tätä aluetta kuvataan usein tavalla `<div class="footer">`. HTML5-kielessä yksinkertaisesti riittää pelkkä `<footer>`. (Korpela 2011, 86)



Kuva 1. Uudet HTML5 rakenne-elementit kuvaamassa hyvin tyyppillisen verkkosivun rakennetta käytännössä (Steve 2010).

Selvitetään yllä olevan kuvan (kuva 1) HTML5 web-sivun rakennetta. Header ylätunniste asettuu sivuston yläpään kuvaamaan sivuston "yläbanneria". Tämä osio sisältää usein h1-otsikkotagin, logon tai navigointiosion. Kyseessä ei ole HTML-dokumentin alussa olevan head-elementin korvike, vaan header-elementti on tarkoitettu sivuston ryhmittelyyn sivuston näkyvässä osassa. Otsikoiden jäsentelyyn on luotu ryhmittely elementti `hgroup`. Tämän avulla voidaan liittää otsikoita yhteen, kun käytetään esimerkiksi alaotsikkoa täydentämään pääotsikkoa, sen käytön hyöty on kuitenkin kyseenalaista. Seuraava maininnan arvoinen tärkeä uusi elementti `nav` merkitsee sivuston navigoinninpaikkaa. Navigointiosa sisältää linkkejä sivuston muihin osiin, jotka kirjoitetaan vanhaan tapaan usein listaelementtejä `ul` ja `li` käyttäen. (Korpela 2011, 94-97)

Sisällön ryhmittelyyn uusia elementtejä ovat muun muassa `article`, `aside`, `figure` ja `section`. `Article` kuvaa sivustolla tekstinpaikkaa, tyyppillisesti blogimerkintää tai uutisartikkelia. `Aside`-elementti koostuu pienestä reuna-merkinnästä, joka sisältää sisältöä sivuavaa tai tarkentavaa tietoa. Kuvia ja tekstiä voidaan koota kokonaisuudeksi `figure`-elementin avulla. `Figure`-elementti kertoo, että kyseessä on kuvakokonaisuus ja kuvien alla mahdol-

lisesti kuvateksti. Section-elementtiä käytetään, kun halutaan ryhmitellä dokumenttia aiheenmukaisesti osiin. Esimerkkinä sivuston pääsivun paloittelu aihekokonaisuuksiin, mihin ennen on käytetty div-elementtiä (Korpela 2011, 92-98)

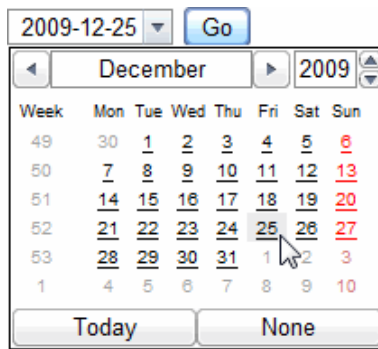
Uusien rakenne-elementtien käyttö ei vaikuta sivuston ulkoasuun. Käyttö tuo kuitenkin selkeyttä kehittäjälle CSS-tyylin ja JavaScript-koodin puolelle. Näin ollen uudenlainen merkkaustapa on hyödyksi selkeytensä vuoksi monikäyttöisen div elementin käytön vähetessä. Selaimet eivät huomioi rakenne-elementtejä, paitsi sivuston CSS-muotoilun kannalta. Onkin tärkeää huomata tyyli-tiedostoa luotaessa uusien rakenne-elementtien merkkaustavat. Kun div-elementille annetaan tyyli-luokka `style="nav"`, siihen viitataan CSS-tiedostossa `.nav`. HTML5 nav-elementtiin viittaaminen tapahtuu ilman pistettä edessä. Käytännössä tämä sotkee sivuston muotoilun vanhoilla IE selaimilla, sillä ne eivät tunnista uutta HTML5 nav-elementtiä, vaan sivuuttavat sen rakenteessa kokonaan. (Korpela 2011, 87-89) Tämä on kuitenkin korjattavissa pienellä JavaScript koodilla, jonka avulla vanhoille selainversioille voidaan opettaa uudet HTML5 rakenne-elementit.

W3C:n mielestä on hyvä opetella HTML 4- sekä HTML5-rakenne-elementit ja käyttää HTML5-elementtejä aina, kun mahdollista. HTML 4 on suotavampaa, mikäli sivuston käyttäjäkunnalla ei ole välttämättä JavaScript tukea käytössä, joka luo tarvittavan turvaverkon HTML5-elementtien käytölle. (W3C n.d.b)

3.1.2 Lomakkeet ja syöttökentät

HTML-lomakkeiden käsitettä on päivitetty uudella kattauksella tietotyypejä, jotka mahdollistavat muun muassa entistä paremman tiedon tarkistuksen selaimissa. Lomakekenttiin, tarkemmin input-elementtiin, on HTML5:ssa lisätty paljon uusia type-määritteitä vanhojen arvojen tueksi. Nämä lomakekenttien arvot kuvaavat kentän sisällöllistä merkitystä. Vanhoja arvoja ovat esimerkiksi text (teksti), button (painike), checkbox (valintaboksi), image (kuva), reset (tyhjennä), submit (lähetä) ja password (salasana). Uusia HTML5 mukana tulleita syötetyyppejä ovat color (väri), date (päivämäärä), email (sähköpostiosoite), number (numeerinen arvo), search (hakusanat) ja aikaa tarkemmin ilmaisevat week, month, datetime, datetime-local ja time. (Korpela 2011, 144)

Uudet tietotyypit parantavat HTML-koodin luettavuutta ja kunhan selaintuki paranee, uudet syötetyypit tuovat lomakkeiden kenttiin parannettuja käyttöliittymiä (Korpela 2011, 144). Esimerkiksi kentässä, jonka arvo on `input type="date"`, käyttäjä asettaa päivämäärän kenttään käyttäen apunaan visuaalista kalenteria. On kuitenkin mietittävä, onko tämä tapa optimaalisin, sillä kalenterikuukausien selaaminen syntymäaikaa asettaessa voi olla aikaa vievää. Tällöin tekstikenttä sopinee tarkoitukseen paremmin. Date-arvoa ei siis tule oletuksena laittaa jokaiseen päivämäärää tiedustelemaan kenttään, vaan on mietittävä käyttötarkoitusta. (Korpela 2011, 144-145)



Kuva 2. Input type="date" päivämäärän valitsin Opera selaimessa.

HTML5 type-arvot voidaan ottaa käyttöön heti. W3C:n ohjesivulla mainitaan, että mikäli selain ei tue uutta tietotyyppiä, se näyttyy tavallisena tekstikenttänä. (W3Schools 2013b) Tosin kehittäjän on mietittävä mahdolliset seuraukset käyttäjän vapaalle tiedonsyötölle tekstikenttään. Esimerkiksi jos päivämäärä halutaan tallettaa tietystä muodosta tietokantaan, lomakkeen syötekentän täyttö ohjeistettava käyttäjälle ja lomakekentän lähetys tarkastettava käyttäen JavaScriptiä.

3.1.3 Mediaelementit

Odotetuimpia uutuuksia HTML5:n mukana ovat uudet median upotuselementit ja niiden vaikutus tulevaisuuden mediantoistoon selaimessa. Uusia upotuselementtejä ovat video ja audio, joiden avulla HTML5-dokumenttiin voidaan upottaa videota ja ääntä tavallisella linkillä. Linkin avulla upotettu video näyttyy selaimen mukaan soittimena, josta löytyy painikkeet videon hallintaan. Videonupotus on helppoa, mutta videomaattien tuki aiheuttaa vielä ongelmia. (Korpela 2011, 178-181)

Browser/Device	Market Share	HTML5 Video	Flash Video
Chrome	32%	✓	✓
Firefox	18%	✓	✓
Internet Explorer 9/10	16%	✓	✓
Internet Explorer 6/7/8	10%	✗	✓
Safari	4%	✓	✓
Opera	1%	✓	✓
iOS	7%	✓	✗
Android	5%	✓	✗
Other (feature phones)	7%	✗	✗

Kuva 3. Vuonna 2013 selainten markkinaosuus ja niiden mediatuki (Longtail, 2013).

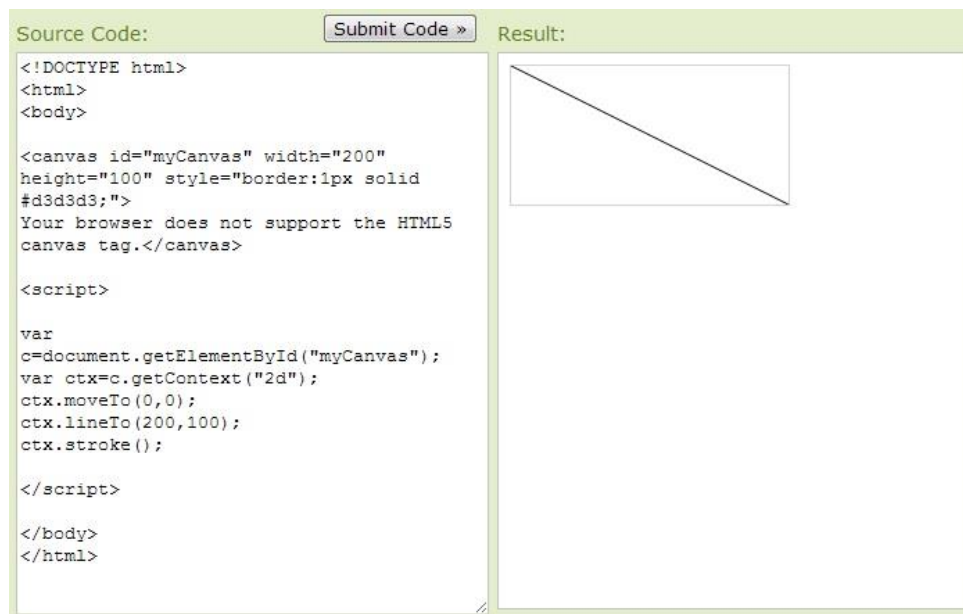
Adoben kehittämä Flash-ympäristö on hallinnut Internetissä jo pitkään muun muassa videoiden, animaatioiden ja pelien esitystä lukuisilla sivuilla. Flash-pohjaista videon esitystapaa on kuitenkin moitittu, sillä Flash on

valmistajakohtainen tekniikka. HTML5 on selaimiin integroitu tekniikka, joka mahdollistaa videoiden kytkemisen web-sivuston tapahtumiin ja median toisto tapahtuu ilman ylimääräisiä liitännäisiä. (Korpela 2011, 180) Applen edesmennyt johtaja Steve Jobs ei ikinä pitänyt Flash kehitysympäristöä kannattavana. Hänen mukaansa Adobe Flash on suljettu ympäristö, kun avoinstandardi HTML5 antaa kehittäjien luoda edistynyttä grafiikkaa ilman tarvetta liitännäisille. Jobbsin mukaan Adobe Flash on pc-aikakauden tuotos, eikä sovellu mobiililaitteisiin. (Jobs 2010) Myöhemmin vuonna 2012 Adobe ilmoitti lopettavansa Flash Player:n kehittämisen myös Android mobiilialustalle (Adobe 2012). Tämä päätös vahvistaa Flash:n tappiota HTML5:lle.

Flash on kuitenkin vielä vuonna 2013 tarpeellinen videoiden esitykseen Internet Explorer 6-8 versioiden käyttäjille (Korpela 2011, 189). Ongelmana HTML5 video-elementin kanssa on, että sen toimiminen vaatii videon olevan vähintään kahdessa eri formaatissa: MP4 ja WebM (Longtail 2013). HTML5 on kuitenkin välttämätön, kun halutaan esittää videota Applen mobiili-laitetta käyttäville (Jobs, S. 2010). Suositut videopalvelut, kuten YouTube, ovat alkaneet jo esittää videoita HTML5-muodossa, mutta kokeilu on vielä testivaiheessa. Käyttäjä voi itse aktivoida YouTube:n HTML5-videot käyttöön käyttäjäasetuksista. (Youtube 2013)

3.1.4 Canvas piirtotoiminnot

Canvas-elementti tuo HTML5:n mukana piirtoalustan graafisille esityksille. Elementti on itsessään yksinkertainen, mutta mahdollisuudet käyttää graafisia piirtokäskyjä tekevät sen käytöstä mielenkiintoista. Käyttäessä Canvas-elementtiä, sen voi sijoittaa HTML-dokumenttiin ja asettaa sille leveys, korkeus ja id-arvo, johon viitata JavaScriptillä. Canvas-alueelle piirtäminen tapahtuu JavaScript-koodilla Canvas API -sovellusliittymän avulla. Viivoja voidaan piirtää koordinaateilla Canvasin sisällä, vasemman yläkulman ollessa koordinaatiston origo, eli piste (0,0). Canvas-elementin koon määrittäminen on tehtävä nimenomaan HTML-elementtiin, sillä nämä määritteet luovat käytettävän koordinaatiston. CSS:llä asetettu koko aiheuttaa sen, että Canvas ei skaalaudu oikein. (Korpela 2011, 200-209)



Kuva 4. HTML5 Canvas-elementti ja sille piirtäminen JavaScript-koodilla. Vasemalla oleva koodi on suoritettu ja tulos näkyy oikealla. (W3Schools 2013c)

Piirtotoiminnon käytön hyöty verrattuna liitettyyn kuvaan on sen joustavampi muokkaaminen. Esimerkiksi piirretyn viivan värin tai leveyden muokkaaminen tapahtuu helposti JavaScript koodia muokkaamalla. JavaScript tarjoaa myös työkalut piirrosten animointiin. Canvasen muotoiluun ei käytetä CSS-tyylejä, sillä viivat ja alueet eivät ole HTML-elementtejä. CSS:n avulla voidaan asettaa ainoastaan piirtoalueen taustaväri tai -kuva, muu muotoilu tehdään JavaScriptillä. (Korpela 2011, 203)

Canvas-elementtiä voidaan käyttää jo kaikilla uusilla selaimilla. Internet Explorer 8 ja vanhempien tukemiseksi käyttöön voidaan ottaa ExplorerCanvas niminen JavaScript kirjasto, joka opettaa Canvas-elementin selaimelle. Kehittäjän on kuitenkin huomioitava, että erillisen ExplorerCanvas kirjaston käyttö ei korvaa alkuperäistä Canvas-toimintoa. Yksinkertaisia kuvioita luotaessa vanhemmille selaimille ExplorerCanvas on kuitenkin riittävä apukeino. (html5please 2013)

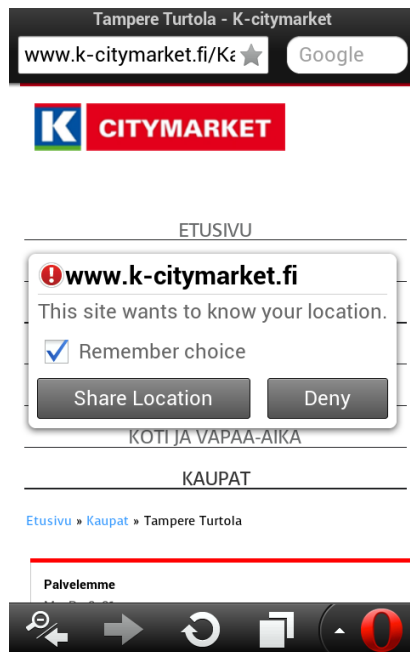
3.1.5 HTML5 web-muisti

Internet selailussa palvelin ja selain pitävät kirjaa tapahtumista. Tietojen tallennusta tapahtuu palvelimen pyynnöstä muun muassa evästeiden avulla käyttäjän tietokoneelle. Palvelin sitten tarvittaessa pyytää tallennetut tiedot käyttöönsä. HTML5 tuo evästeet korvaavan, kehittyneemmän ratkaisun tiedon tallennukselle: web-muistin. Web-muisti on paikallinen tietovarasto käyttäjän tietokoneella, johon verkkosivu voi tallentaa tietoa ja lukea sitä. Tämä on paitsi tehokkaampaa, myös tietosuoja on parempi. Evästeiden kanssa tietoja lähetetään eteenpäin siltä varalta, jos niitä satuttaisiin käyttämään. Web-muistin kanssa sivusto erikseen pyytää tarvitsemiaan tietoja. Web-muistia voidaan hyödyntää esimerkiksi verkkokaupassa, jossa käyttäjä voi valita häntä kiinnostavat tuotteet ja verkkokauppa muistaa käyttäjän valitsemat tuotteet tulevaisuudessa. (Korpela 2011, 230-232)

Teknisesti web-muistia käytetään JavaScriptin avulla. JavaScript kutsuu funktiota, joka tallettaa halutun tiedon nimellä, tai funktiota, joka hakee aiemmin talletetun tiedon. (Korpela 2011, 231) Tietoa voidaan tallettaa istuntokohtaisesti (sessionStorage), jolloin tieto katoaa käyttäjän sulkiessa selaimen, tai lokaalisti (localStorage) ilman tiedon vanhenemista. Web-muistia voidaan hyödyntää kaikilla selaimilla, paitsi Internet Explorer 7 ja vanhemmilla selainversioilla. (W3Schools 2013d)

3.1.6 Paikannus

Geolocation paikannus on HTML5:seen lisätty tapa käyttäjän maantieteellisen sijainnin selvittämiseen. Tästä on paljon hyötyä markkinoinnin sekä julkisten palvelujen suunnittelun näkökulmasta. Paikannuksen tarkkuudesta riippuen voidaan käyttäjän olinpaikkaa käyttää esimerkiksi lähellä sijaitsevan myymälän aukioloaikojen tai paikallissään esittämiseen. Oman olinpaikan paikannusta voidaan kuitenkin pitää yksityissuojaa ja turvallisuutta vaarantavana, sillä taskussa kulkevan puhelimen GPS voidaan paikantaa jopa metrien tarkkuudella. Tämän vuoksi käyttäjä saa itse päättää kenelle jakaa olinpaikkansa. (Korpela 2011, 238)



Kuva 5. Puhelimen Opera Mobile -selain pyytää lupaa jakaa sijainti K-Citymarket -sivuston kanssa.

Geolocation paikannus on yksi esimerkki HTML5 merkityksen laajuudesta, sillä paikannuksella ei sinänsä ole mitään tekemistä HTML-merkkäuskielen kanssa. Itse paikannus tapahtuu JavaScript API sovellusliittymän avulla. HTML-dokumenttiin on liitettävä JavaScript-koodia, jonka avulla kysytään käyttäjän sijaintitietoja. Selaimen aukeaa kysymys, mikäli käyttäjä haluaa jakaa olinpaikkansa. Henkilön olinpaikan kanssa voidaan käyttää Google Maps karttapalvelua, jonka avulla saadaan näkyviin kartta henkilön lähiympäristöstä.

3.2 HTML5 ja selaintuki

HTML5-selaintukea on vaikeaa kartoittaa tarkasti. Eri selaimet kääntävät HTML-dokumentin näkyväksi esitykseksi erilailla. Itse kääntäminen tapahtuu selaimen taittokomponentin avulla. Eri elementeillä on erilainen oletusesitystapa riippuen selaimesta. (Korpela 2011, 33) Karkeata kuvaa HTML5-tuesta eri selainten välillä saadaan esimerkiksi sivuston html5test avulla. Html5test antaa pisteluvun, joka kuvaa käytössä olevan selaimen HTML5-tuen laajuutta. Mitä korkeampi pistemäärä, sitä laajempi HTML5-tuki selaimessa on.



	Score	Bonus
Maxthon 4.0 »	476	15
Chrome 26 »	468	13
Opera 12.10 »	419	9
Firefox 20 »	394	10
Safari 6.0 »	378	8
Internet Explorer 10 »	320	6

Microsoft Surface and others

Kuva 6. Html5test.com sivuston antama pisteytys HTML5-tuesta uusien selainten välillä huhtikuussa 2013.

Ongelmallisin selainversio HTML5:n käytön kannalta on Internet Explorer 8, sillä se ei tue monia HTML5-ominaisuuksia, mutta yhä 10% maailman Internet käyttäjistä surffaa Internet Explorer 8:lla (Richter 2013). Selaimen käytön suosio johtuu Windows XP käyttöjärjestelmästä, sillä Internet Explorer 8 on viimeisin asennettavissa oleva selainversio XP käyttöjärjestelmälle (Hunt 2013). Windows XP:n tuki loppuu huhtikuussa 2014, joten sitä ennen sitä vielä käyttävät yritykset eivät välttämättä näe hyötyä siirtymään uuteen käyttöjärjestelmäversioon (Microsoft 2013).

JavaScriptillä on luotu HTML5- ja CSS3-elementtien toimivuuden varmistamiseksi selaimille koodin pätkiä sekä lisäosia, joita kutsutaan englanniksi nimellä polyfills. Polyfills:t tarjoavat selaimen tekniikan HTML5-ominaisuudelle, vaikka selain ei sitä oletuksena tukisi. Näitä JavaScript kirjastoja kannattaa hyödyntää, mikäli käytettävä HTML5-ominaisuus on olennainen sivuston toiminnan kannalta. Muussa tapauksessa useita HTML5- ja CSS3-elementtejä voidaan käyttää hyödyksi parantamaan sivuston ulkoasua tuoreilla selaimilla, mutta toiminnallisuuden säilyminen myös vanhemmilla selainversioilla. (html5please 2013)

HTML5:ta on mahdollista ruveta käyttämään jo melko laajasti. Aloittaesasi kehittämään uutta sivustoa, on pohjaksi hyvä ottaa HTML5-Boilerplate. Se on Paul Irishin ja Divya Manianin kehittämä alusta, joka tarjoaa ratkaisut useimpiin selainten HTML5-toteutusten ongelmiin. Boilerplate sisältää myös valmiita CSS-luokkia, jotka auttavat sivuston muotoilussa vanhoille selaimille. HTML5-Boilerplate on ilmainen vapaanlähdekoodin paketti, jonka voi ladata ja käyttää ilmaiseksi. Projektia ylläpitää

yhteisö H5BP, johon on liittynyt kymmeniä kehittäjiä. Valmis pohja sisältää myös saman kehittäjätiimin tuotoksen nimeltä Modernizr. Sen JavaScript kirjaston avulla voidaan selvittää käyttäjän selaimen HTML5- ja CSS3-tukea. Näin kehittäjä voi luoda vaihtoehtoista sisältöä verkkosivulle riippuen käyttäjän selaimesta. Näin uusien tekniikoiden mahdollistamia muutoksia voidaan turvallisesti tuoda vanhemmillekin sivustoille vaihe kerrallaan. (html5boilerplate 2013)

4 CSS3

CSS3 ei ole sama kuin HTML5, vaikka näin on osa erehtynyt luulemaan. HTML5 on merkkauskieli ja CSS on merkkauskielisten dokumenttien maailman käytetyin muotoilukieli. Koska CSS-kieltä kehittävä W3C tiesi uuden CSS-version toteutumisen vievän aikaa, he tekivät CSS3-kehityksestä modulaarista. Näin sen eri osa-alueita voitaisiin kehittää rinnakkain ja suosituimmat ominaisuudet saataisiin käyttöön nopeammin. Osa uusista CSS3-moduuleista päivittää CSS 2.1 -version ominaisuuksia, kun osa moduuleista on kokonaan uusia. Uudet selaimet päivittyvät jatkuvasti tukemaan uusia CSS3-moduuleita, joten CSS3:n käyttöä voidaan suositella varauksella. Samoin kuin HTML5:n kanssa, kaikkien ominaisuuksien toimivuuden odotus voi viedä vielä useita vuosia. (Gasston 2011, 2-4)

Uudet moduulit ovat muutosalttiita ja niiden status muuttuu W3C:n mukaan luonnoksesta suositukseksi. Usein tie W3C:n suositukseksi menee seuraavasti, kun moduuli saavuttaa toimivan luonnoksen (working draft) tilan, se siirtyy selainten kehittäjien ja työryhmien arvosteltavaksi. Kun dokumentointi ja teknilliset vaatimukset ovat täytetty, voi moduuli siirtyä luonnosehdokkaaksi (candidate recommendation), tai jos lisätyötä tarvitaan, palata takaisin työvaiheeseen. Luonnosehdokasvaiheessa selaimet voivat lisätä moduulin tuen selaimen ja kerätä palautetta maailmalta. Kun vähintään kaksi selainta on ottanut moduulin käyttöön, eikä teknisiä ongelmia ole esiintynyt, voi kehitys jatkua tilaan ehdotettusuositus (proposed recommendation). Tämän jälkeen W3C tiedotuskomitea hyväksyy moduulin suositukseksi. Prosessi luonnoksesta suositukseksi ei kuitenkaan aina ole näin suoraviivainen. (Gasston 2011, 3)

4.1 CSS3 uudet ominaisuudet

CSS3 mukana tulleita uusia ominaisuuksia ovat monenlaiset visuaalista näyttävyyttä ja sivuston muotoilua edesauttavat moduulit. Osa uusista ominaisuuksista korvaa osittain Flash:in ja tarpeen JavaScriptille. Uusia ominaisuuksia ovat tekstin muotoilu ääriiviivoilla ja varjostuksilla, taustakuvien luonti sekä niiden sijainnin ja koon muuttaminen lennosta. Myös elementtien animointi ja erilaiset siirtymäefektit sivuston tapahtumiin onnistuu uusilla CSS3-ominaisuuksilla. Sivuston skaalautuminen erikokoisille laitteille ja näytöille voidaan saavuttaa myös CSS3:n avulla. (Gasston 2011, 4)

4.1.1 Media Queries

Internetiä ei enää selata ainoastaan kotikoneen näytöltä, vaan myös älypuhelimilla, tablet-laitteilla ja pelikonsoleilla. Näyttöjen koot ja tarkkuudet ovat myös kehittyneet, joten Internet sivuston sisällön tarjoaminen samassa muodossa kaikille käyttäjille ei ole enää kannattavaa. CSS3 tuo moduulin nimeltä media queries, joka tuo uusia mahdollisuuksia vanhan media tyyppin määrittelyyn. Web-kehittäjät saavat vapauden luoda sivustoja, jotka

tarjoavat käyttäjälle parhaan mahdollisen käyttökokemuksen laitteesta riippumatta. (Gasston 2011, 9-10)



Kuva 7. CSS3 media query:n käyttö mahdollistaa sivuston ulkoasun muuttamisen näyttökoolle sopivaksi. (Nutil n.d)

Media query:n kannattavuutta korostaa sen käytön helppous. Ennen mobiiliystävällisen sivuston rakentamiseen käytettiin omaa sub-domainia sekä JavaScript koodia tarkastamaan, onko käyttäjä selailemassa sivustoa älypuhelimella. CSS3 media query osaa tunnistaa käyttäjän päätelaitteen, sen ruudun resoluution ja jopa laitteen asennon. Näin kehittäjä voi helposti tarjota eri CSS-muotoiluja riippuen käyttäjän laitteesta. (Gasston 2011, 10)

```
<!--Media Query CSS tiedostot-->
<link rel="stylesheet" href="allstyles.css" />
<link rel="stylesheet" media="screen and (max-device-width: 480px)" href="narrowstyles.css"/>
<link rel="stylesheet" media="screen and (min-width: 621px)" href="widestyles.css"/>
<link rel="stylesheet" media="screen and (max-width: 620px)" href="narrowstyles.css" />
```

Kuva 8. Esimerkki CSS3 media query:n käytöstä

CSS3 media query:n avulla voidaan ottaa käyttöön oikea tyylitiedosto tilanteen mukaan. Ylhäällä olevassa esimerkissä (kuva 8) allstyles.css tiedostoa käytetään aina kaikissa tapauksissa. Kun selainikkunan koko on vähintään 621 pikseliä leveys suunnassa (min-width), ladataan käyttöön widestyles.css tiedosto. Kun selainikkunan koko laskee alle 620 pikselin (max-width), tai jos käytössä olevan laitteen ruudun maksimi leveys on 480 pikseliä (max-device-width), ladataan käyttöön narrowstyles.css tiedosto. Widestyles.css tiedosto pitää sisällään sivuston muotoilun suurille näytöille, kun narrowstyles.css on luotu mobiilikäyttäjää ajatellen.

Vaihtoehtoisia tapoja media query:n käytölle ovat media sääntöjen kirjoittaminen css-tyylitiedoston sisään. Tyylitiedostot voidaan ladata myös ulkoiselta palvelimelta käyttämällä import-komentoa. CSS-tyylejä voidaan asettaa myös korkeuden tai käytettävän laitteen asennon mukaan, ei ainoastaan leveyden. Suunnitellessa sivuston tyylitiedostoja on tärkeää huomioida mobiilikäyttäjien kaistanleveys. Suurten sivustojen suunnittelussa on kannattavaa luoda mobiilityyli ensiksi ja ladata sivun raskas sisältö vasta, kun käyttäjän tunnistetaan käyttävän suurempaa näytönkoko. (Gasston 2011, 11-17)

4.1.2 Tekstiefektit ja typografiset tyyli

CSS tekstimoduulia on päivitetty viimeksi vuonna 2007 ja kehittäjillä ei ole ollut paljoa mahdollisuuksia tekstin muotoiluun CSS:n avulla. Nyt CSS3 tuo mukanaan paljon uusia työkaluja, joiden avulla voidaan luoda visuaalisesti näyttäviä otsikoita, mutta myös parantaa osaltaan tekstin luettavuutta. (Gasston P. 2011, 65)

Vuonna 2005 Safarin mukana esiteltiin tekstin varjostus efekti, mutta sitä ei toteutettu CSS 2.1 -versiossa. Nyt CSS3 on tuonut ominaisuuden takaisin. Ennen tekstin varjostuksen käyttöä on tarpeen opetella koordinaation käyttö, sillä tekstille asetetaan varjostus käyttäen x ja y koordinaatteja. Koordinaatti-arvot voivat olla myös negatiivisia. Useita varjoefektejä voidaan yhdistää samaan tekstiin, jolloin saavutetaan kolmiulotteinen vaikutelma. (Gasston 2011, 66) Varjostuksia voidaan käyttää sivustoilla, sillä ominaisuus on hyvin tuettu uusissa selaimissa, eikä efektin puuttuminen osaltaan vaikuta sivuston käytettävyyteen (Gasston 2011, 80).

TEXT Shadow

```
<h1 style="text-shadow:1px 1px 2px #0000FF;">TEXT Shadow</h1>
```

Kuva 9. Tekstin varjostus asetetaan x ja y-koordinaattien avulla..

Toinen CSS3:n mukana tullut tekstiefekti mahdollistaa tekstin ääriiviivan luomisen ilman varjostusta. Text-outline ja text-stroke ominaisuuksia käyttämällä voidaan luoda tekstille reunus, joka eroaa tekstin väristä. Ääriiviivan liian suuria arvoja tulee varoa, sillä tekstistä tulee vaikealukuista. (Gasston 2011, 73)

TEXT Stroke

```
<h1 style=" -webkit-text-fill-color: #00FF00;
-webkit-text-stroke: 2px #000;">
TEXT Stroke</h1>
```

Kuva 10. Tekstin ääriviivat piirtyvät text-stroke ja väri text-fill-color ominaisuudella.

Muut CSS3-tekstimoduulin ominaisuudet eivät ole kovin visuaalisia, mutta voivat parantaa osaltaan tekstin luettavuutta. Esimerkiksi tekstiä voidaan pätkiä käyttämällä clip ja ellipsis arvoja, jotta pitkä lause ei ylittäisi sille asetettua aluetta. Word-wrap mahdollistaa pitkien sanojen automaattisen pilkkomisen. Resize ominaisuuden avulla taas voidaan luoda tekstialue, jota käyttäjä voi itse laajentaa hiiren avulla. (Gasston 2011, 74-76) Tekstin skaalaamiseen käytettävien ominaisuuksien kohdalla on kuitenkin

oltava vielä varovainen, sillä osa niistä ei vielä toimi laajasti. Puuttuessaan ne voivat vaikeuttaa tekstin näkyvyyttä. (Gasston P. 2011, 80)

4.1.3 Reuna- ja varjoefektit

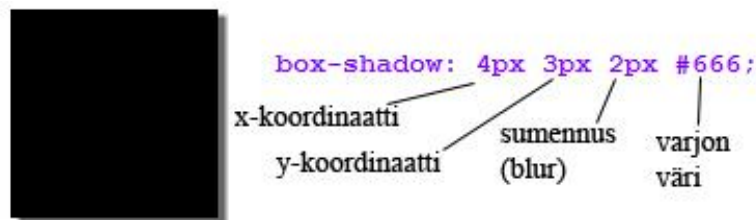
Paljon käytetty CSS borders-ominaisuus reunojen luomiseen saa CSS3:n mukana monien kaipaamia uusia visuaalisia ominaisuuksia. Tärkeimmät niistä ovat reunojen pyöristys ja varjostuksien luominen, minkä tekeminen aikaisemmin on ollut hankalaa. Kyseisiä uusia ominaisuuksia käytettäessä on taas ymmärrettävä x- ja y-koordinaatiston toimintaperiaate.



Kuva 11. Div-elementtiin on luotu reunojen pyöristys CSS3:n avulla. Oikealla vieressä suurennos oikeasta yläreunasta, josta nähdään kuinka y- ja x-akselin pisteiden välille piirtyy käyrä, joka luo pyöristyksen.

Reunojen pyöristys onnistuu asettamalla jokaiselle nurkalle arvo, joka määrittää reunan pyöristyksen piirtämällä käyrän x- ja y-akselin välille. Esimerkissä (kuva 11) on x- ja y-akselin arvo jokaisessa kulmassa 20px, jolloin pyöristys tapahtuu y-akselin pisteestä 20 x-akselin pisteeseen 20. (Gasston 2011, 108)

Reunoille voidaan pyöristyksen lisäksi asettaa kuva ominaisuudella border-image, joka mahdollistaa koristeellisten reunojen luomisen kuvien avulla. Reunaa voidaan koristella myös käyttämällä useita erivärisiä viivoja border-color ominaisuuden avulla. Varjostuksien luominen elementeille onnistuu tekstin varjostuksen tavoin box-shadow ominaisuudella (kuva 12). Käyttöperiaate on sama; asetetaan x- ja y-koordinaatit, sumennus ja väri.



Kuva 12. CSS3 box-shadow asetettuna div-elementtiin.

4.1.4 Useat taustakuvat

Sivuston näytävyyttä parantavat useat graafiset elementit, kuten kuvat. Kuvien asettelun helpottamiseksi CSS3 tuo kehittäjille tapoja asettaa useita taustakuvia samaan elementtiin sekä helpottaa kuvien sijoittamista. Taustakuva-mooduulin suosiosta johtuen useat selaimet ovat ottaneet sen jo käyttöön. (Gasston 2011, 94)



Kuva 13. Samaa div elementtiin on CSS3:n avulla asetettu useita kuvia.

Käyttäkseen useaa taustakuvaa samanaikaisesti yhdessä elementissä riittää, että eri kuvat erotetaan pilkun avulla. Ensimmäinen kuva listassa on aina päällimmäisenä, jolloin viimeinen listassa jää taustalle. Kuvia voidaan sijoittaa käyttämällä background-position ja kokoa muuttaa background-size ominaisuuksilla. Esimerkissä (kuva 13) kuvien sijainti ja toistuvuus (background-repeat) ovat määritetty suoraan background-ominaisuuteen. Ensimmäisen kuvan sijainti on 55% vasemmalta ja 65% ylhäältä sen isäntä elementin mukaan. Sijainti 0% 0% sijoittaisi kuvan vasempaan yläreunaan, kun taas 100% 100% oikeaan alareunaan. Ominai-

suus background-color taustaväriin asettamiseen ei kuitenkaan hyväksy useita arvoja. (Gasston 2011, 94-95)

4.1.5 Värit ja läpinäkyvyys

Värien käyttö on CSS 2 -ajoista asti ollut punaisen, vihreän ja sinisen värien yhdistelemistä oikean sävyn saavuttamiseksi (RGB) ja varsinkin haa-lemman sävyn saavuttaminen on ollut vaikeaa. CSS3 väri-moduuli helpottaa oikean värisävyn löytämisessä, kun läpinäkyvyyttä voidaan asettaa uuden opacity-ominaisuuden avulla. Opacity hyväksyy arvon nollan ja ykkösen väliltä, nollan ollessa täysin läpinäkyvä. Läpinäkyvyys voidaan myös asettaa suoraan värimäärittelyyn, jolloin väriin loppuun lisätään uusi arvo. Tällöin puhutaan RGBA (red, green, blue, alpha) arvosta. Alpha arvo asetetaan samalla tavalla kuin opacity. (Gasston 2011, 120-123)



```
background-color: rgba(0,0,0,0.5);  
background-color: black; opacity: 0.5;
```

Kuva 14. Kahdella eri merkitsemistavalla saavutetaan sama lopputulos: 50% läpinäkyvyyttä mustalle värille.

Kahden tai useamman väriin asteittaista yhdistämistä toisiinsa kutsutaan CSS-kielessä nimellä gradient. Sen avulla luodaan häivytyksen efekti värien välillä ja ennen tätä on luotu käyttäen kuvatiedostoja. CSS3:n avulla voidaan värejä yhdistää suoraan koodissa, mutta merkitseminen eri selainten välillä on vielä ongelmallista. Eri selainten kanssa on käytettävä sopivia etuliitteitä ja värien tulkitseminen on eri selainten välillä erilaista. Esimerkiksi Mozilla Firefox selain tarvitsee etuliitteen -moz- ja väriin suunta voidaan määrittää kulman (angle) mukaan. Google Chrome selain käyttää taas etuliitettä -webkit- eikä väriin suuntaa voida asettaa kulman mukaan. (Gasston 2011, 132-133) Eri selainten väliset syntaksit tulevat toivottavasti ajan myötä muuttumaan. Tällä hetkellä Internet Explorer 10 on ainoa selain, joka tukee gradient-värejä ilman etuliitettä (Gervais 2012).

```
/* Mozilla Firefox */
background-image: -moz-linear-gradient(top, #CEE980 0%, #EEF6D9 100%);
/* Opera */
background-image: -o-linear-gradient(top, #CEE980 0%, #EEF6D9 100%);
/* Safari/Chrome */
background-image: -webkit-gradient(linear, left top, left bottom,
    color-stop(0, #CEE980), color-stop(1, #EEF6D9));
/* Chrome 11+ */
background-image: -webkit-linear-gradient(top, #CEE980 0%, #EEF6D9 100%);
/* W3C, IE10 Release Preview */
background-image: linear-gradient(to bottom, #CEE980 0%, #EEF6D9 100%);
```

Kuva 15. Esimerkki määrittely CSS3 gradient värille eri selaimille.

Webkit tarjoaa eniten ominaisuuksia gradient värien muotoilemiseen, mutta niiden käyttäminen on monimutkaista. Firefoxin syntaksi on helpoin ymmärtää, mutta se ei tarjoa niin paljo ominaisuuksia värien hallintaan. Helpoimmalla päästään vielä käyttämällä kuvatiedostoa, sillä gradients määrittely on vielä käymässä läpi siirtymävaihetta. (Gasston 2011, 145)

4.1.6 Siirtymäefektit ja animaatiot

CSS:n avulla ei juuri ole hallittu Internet sivustojen käyttäytymistä, vaan käyttäjän toimintoja muun muassa animoidaan JavaScriptin avulla. Transitions-siirtymäefektit ja animations-animaatiot ovat CSS3-moduuleita, joiden avulla voidaan luoda liikettä sivustoille ilman JavaScript koodia. (Gasston 2011, 163)

Kun elementille asetetaan uusi arvo CSS:n avulla, esimerkiksi elementin värin vaihtaminen pseudo-luokan :hover avulla, muutos tapahtuu välittömästi. Siirtymäefekteillä voidaan animoida näitä muutoksia. Kun elementille asetetaan transition-ominaisuus, tapahtuu siirtyminen uuteen arvoon sulavasti. Transition tarvitsee toimiakseen animoitavan ominaisuuden, kuten taustavärin. Siirtymisen aika määritetään sekunneissa ja siirtymisen tyyliä voidaan muuttaa transition-timing-function -arvoa muuttamalla. Samaan elementtiin voidaan asettaa useita siirtymäefektejä, esimerkiksi elementin sijaintia, kokoa tai tyyliä voidaan muuttaa. (Gasston 2011, 170-171)



Kuva 16. 3-kuvan sarjassa nähdään div-elementille asetettu transition-efekti: 4 sekunnin aikana kuva muuttaa väriä ja sijaintia, kun kursori viedään elementin ylle.

Siirtymäefektit ovat hyödyllisiä, mutta rajoittuneita. Animaatio-moduulin syntaksi mahdollistaa monimutkaisempien animaatioiden luomisen (Gastton 2011, 172). Toisin kuin siirtymäefektit, ensimmäiseksi luodaan animaatio, jonka jälkeen se sidotaan haluttuun elementtiin. Voit antaa animaatiolle sitä kuvaavan nimen ja käyttää samaa animaatiota useita kertoja saman sivun sisällä. CSS3-animaatiot luodaan käyttäen @keyframes sääntöä selaimissa Firefox, Internet Explorer 10 ja Opera. Google Chrome ja Safari selaimet käyttävät muotoa -webkit-keyframes. Animaatioilla on useita mahdollisia arvoja. Voit muun muassa määrittää sen keston, liikkeen suunnan, värin ja viiveen. Animaation eri vaiheet merkitään käyttämällä prosentteja, 0% kohdalla animaatio alkaa ja 100% tarkoittaa loppua. Näin esimerkiksi animaation keskivaiheen haluttu muutos merkitään 50%:n kohdalle. (W3schools n.d.e)

4.2 CSS3 ja selaintuki

CSS3 syntaksi on osittain erilainen CSS 2.1 version kanssa. Eri selainvalmistajat tarjoilevat uusia CSS3-ominaisuuksia käyttäen etuliitteitä. Tulevaisuudessa etuliitteet tulevat poistumaan, kun kaikki selaimet tukevat uutta ominaisuutta samalla komennolla. Sitä ennen on käytettävä etuliitteitä useiden ominaisuuksien merkitsemisen kohdalla. (Storey 2013)

```
.box_rotate {
  -webkit-transform: rotate(7.5deg); /* Chrome, Safari 3.1+ */
  -moz-transform: rotate(7.5deg); /* Firefox 3.5-15 */
  -ms-transform: rotate(7.5deg); /* IE 9 */
  -o-transform: rotate(7.5deg); /* Opera 10.50-12.00 */
  transform: rotate(7.5deg); /* Firefox 16+, IE 10+, Opera 12.10+ */
}
```

Kuva 17. Transform-ominaisuus tarvitsee toimiakseen selaimen etuliitteen vielä toistaiseksi.

Kuten yllä olevasta kuvasta (kuva 17) voimme nähdä, tarvitsee eri selaimille asettaa niitä vastaava etuliite. Etuliitteet ovat aina selainkohtaiset, mutta tulevat vähitellen poistumaan käytöstä. Kun ominaisuudet tulevat W3C suositukseksi, tarvitsee transform-ominaisuus asettaa koodiin ilman etuliitteitä. Etuliitteetön merkintä on hyvä lisätä koodiin jo tässä vaiheessa, jotta muotoilu toimii myös tulevaisuudessa. Selaimet ohittavat niitä koskemattomat etuliitteet, joten ylimääräisistä etuliitteistä ei ole haittaa, mikäli et ole varma tarvitaanko selaimelle etuliitettä vai ei. (Kyrnin n.d.b)

CSS3:n käytön säännöksi voidaan luetella samat kuin HTML5:n: suurin osa CSS3-ominaisuuksista toimii suosituimpien selainten uusilla versioilla, mutta vanhat Internet Explorer selaimet aiheuttavat ongelmia. Visuaalisuutta voidaan kuitenkin toteuttaa sivustolle huoletta, mikäli sen puuttuminen ei vaikeuta sivuston käyttöä. Ohessa suuntaa antava tilasto(kuva 18), joka kuvaa selainten suosituimpien visuaalisten CSS3-ominaisuuksien tukea huhtikuussa 2013.

Browser Rendering Engine	 Firefox Gecko	 Safari Webkit			 Chrome Webkit	
Version	20	4	5	5.1+	26	
Total CSS3 Support *	97.2%	83.4%	88.9%	97.2%	94.4%	
Browser Rendering Engine	 Internet Explorer Trident			 Opera Presto		
Version	6 – 8	9	10	11.1	11.6	12.1
Total CSS3 Support *	8.3%	55.6%	94.4%	80.6%	83.3%	88.9%

Kuva 18. CSS3 suosituimpien ominaisuuksien selaintuki huhtikuussa vuonna 2013 (Norman 2013).

5 FUNITY ETHESISPASS -JÄRJESTELMÄ

FUNity eThesisPass on järjestelmä, jonka toiminnallisuuksia käytetään portaalin kautta. Järjestelmä on tarkoitettu korkeakoulujen opinnäytetöiden ja työharjoitteluprosessien ohjaamiseen ja seuraamiseen. Järjestelmän nimi FUNity tulee sanoista "fun in unity" ja järjestelmän tavoitteena on vähentää työmäärää automatisoiden opinnäytetöiden ja työharjoittelujen seurantaa, erityisesti opettajien näkökulmasta. Automatisoidun järjestelmän avulla päästään paperittomaan opinnäyte-, loppu-, päättö- ja insinöörityöiden lausuntojen kirjaamiseen. FUNity tarjoaa myös kanavan opiskelijoiden ja ohjaajien sähköpostimaiselle kommunikoinnille. Järjestelmä on integroitu koulun sähköpostiin, jolloin viestit ja ilmoitukset toimivat myös sähköpostin kautta. (Seppänen 2013)

FUNity-järjestelmä on toimiva ja käyttöönotettu kahdessa Suomen ammattikorkeakoulussa. Opinnäytetyötä tehdessä tarkoituksena ei ollut puuttua järjestelmän toiminnallisiin tai päivittää jo olemassa olevia ominaisuuksia. Portaalissa on pieniä kosmeettisia vikoja, muun muassa sivuston päävalikkorakenne aiheuttaa selainkohtaisia käytettävyysongelmia. Opinnäytetyön toimeksiantajan ja järjestelmän kehittäjän, Lasse Seppäsen, asettama tavoite oli ulkoasun päivittäminen ottaen työkaluiksi uudet HTML5- ja CSS3-tekniikat.

5.1 Järjestelmän ominaisuudet

FUNity tarjoaa sekä ohjaajille, että opiskelijoille oman kirjautumispuolen koulutusohjelmakohtaisesti. Koulutusohjelmakohtaisesta portaalista on hyötyä ohjaajille, sillä näkyvissä on vain tarpeellinen tieto ja järjestelmä on nopea käyttää. FUNity-portaaleja voidaan muokata käyttäjän tarpeisiin, esimerkiksi koulutusohjelmalla voi olla erikseen omat portaalit eri töiden ja projektien hallinnoimiseen. Käytössä voi olla myös eri kieliversioita suomenkielisen portaalin lisäksi. (Seppänen 2013)

Ohjaajienpuolelta yksi järjestelmän ominaisuuksista on budjetointi. Sen avulla yhden opiskelijatyön viemä aika voidaan budjetoida ohjaajien välille. Budjetoinnilla kukin ohjaaja saa määritellyn vuosittaisen ohjaus- ja kommentointimäärän, joista pidetään kirjaa. Ohjaajapuoli tarjoaa ohjaajille työkalut kaikkiin opinnäytetyön seurantavaiheisiin. Opiskelija voi lähettää työnsä tarkastettavaksi kielellisesti, jolloin ohjaaja, jonka roolina on suomenkielentarkastaminen, saa työn luettavaksi. (Seppänen 2013)

Ohjaajille voidaan jakaa eri rooleja, mikä helpottaa työvaiheiden jakoa. Ohjaajat voivat järjestelmässä muun muassa kirjata lausuntoja, kommentoida opiskelijan työvaiheita, lähettää ryhmäsähköpostia opinnäytetyöntekijöille tai rajatusti tietyille käyttäjäryhmälle. Kuitenkin esimerkiksi loppuarvosanan antamisen oikeus voidaan rajata vain tietyille ohjaajille. Uusien opiskelijoiden ja seminaarien lisääminen järjestelmään, kuin myös järjestelmän ylläpito-oikeudet ovat rajatut. (Seppänen 2013)

Portaaliin kirjautuminen sallitaan ohjaajapuolelta harjoitteluun tai opinnäytetyön tekijäksi ilmoittautuneelle opiskelijalle. Kun opiskelija on saa-

nut kirjaututtua opiskelijaportaaliin, hän pääsee kirjaamaan harjoittelun vaiheita ja myöhemmin opinnäytetyönsä tekemistä. FUNity tarjoaa väylän blogipohjaiseen vaiheiden tallettamiseen ja kommunikointi ohjaajien kanssa onnistuu helposti. Järjestelmä tallettaa viestit myöhempää tarkastelua varten ja kummatkin osapuolet saavat sähköpostiin viestin uudesta kirjauksesta järjestelmässä. Kun koko ohjaushistoria tallentuu samaan paikkaan, on sen lukeminen myöhemmin saumatonta. (Seppänen 2013)

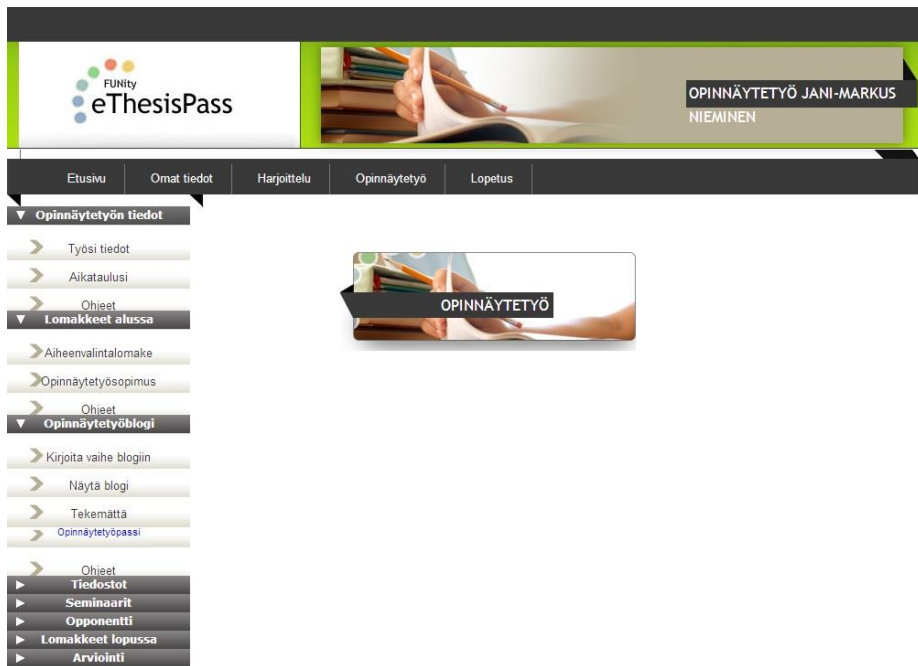
Opiskelija saa portaalin kautta tietoa tulevista seminaaripäivistä ja voi ilmoittautua niihin haluamaan ajankohtana. Oma työ on mahdollista tallettaa järjestelmään versiohallitusti Open Office tai Microsoft Wordin luomissa tiedostomuodoissa. Omien työvaiheiden seuraaminen on tehty helpoksi ja opiskelija näkee, mikä on seuraava työvaihe. (Seppänen 2013)

5.2 FUNityn päävalikkorakenne ja sivustolla navigoiminen

Selkeän nettisivuston navigointi koostuu yleisesti kahdesta valikkokokonaisuudesta: pää- ja sivumenusta. Päämenu löytyy usein sivuston yläosasta ja on kokoajan näkyvissä sivulla liikuttaessa. Päämenu edustaa sisältöä, joka kiinnostaa käyttäjiä eniten. Sen tulisi olla lyhyt ja ytimekäs, sisältää sivuston tärkein sisältö. Käyttäjän tulisi löytää ensisilmäyksellä sivustolle siirtyessään päämenun kautta etsimänsä tieto. Päämenun ei tulisi sisältää yli 10:tä painiketta selkeyden vuoksi. (Antharia 2011)

Sivumenu sisältää linkkejä sisältöön, joka ei vastaa sivuston päätavoitetta, mutta se saattaa kiinnostaa käyttäjää (Knight 2011). Suosittu tapa on piilottaa alamenu päämenun painikkeisiin, jolloin viedessä hiiren kursori painikkeen päälle, saadaan sivumenu linkit näkyviin. Sivuston sisällön kasvaessa saatetaan tarvita vielä kolmas alamenu, mutta sen syvemmälle valikkorakenteessa menemistä tulisi välttää. Jos käytössä on laajempi kuin kolmeportainen navigointi, tulisi web-sivuston menurakenne miettiä uudestaan. (Antharia 2011)

FUNity-portaalin navigoiminen toimii kahden menuvalikon kautta, sekä järjestelmän ohjaaja-, että opiskelijapuolella. Sivuston päämenu sijaitsee ylälaidassa ja ohjaa käyttäjän halutun aihekokonaisuuden äärelle. Selkeästi ilmaistut painikkeet ovat käyttäjälle yksiselitteisiä, esimerkiksi opinnäytetyö-painike ohjaa käyttäjän opinnäytetyötä käsittelevälle sivulle. Päämenusta linkkiä painaessa sivuston vasempaan reunaan aukeaa sivumenu aihekokonaisuuden ympärille. Sivumenu on jaettu otsikoin helpottamaan käyttäjää löytämään tarvitsemansa tieto. Linkkejä pystyy piilottamaan näkyvistä klikkaamalla otsikkoa ja järjestelmä muistaa piilotetut otsikot, vaikka käyttäjä kirjautuisi välillä portaalista ulos. Piilotetut otsikot saadaan takaisin näkyviin klikkaamalla otsikkoa uudestaan.



Kuva 19. Näkymä FUNity opiskelijaportaalin opinnäytetyö-sivusta.

Ohjaajapuolella sisältöä on enemmän, joten päämenu sisältää enemmän painikkeita. Käyttöliittymä on kuitenkin sama. Otsikoiden piilottamisen hyöty korostuu ohjaajapuolella, sillä sisältöä on enemmän ja sivumenu ulottuu selainikkunan alalaitaan saakka.

5.3 FUNityn päävalikkorakenteen ongelmat

FUNityssa navigoiminen toimii käytettävyyden näkökulmasta kiitettävästi. Tämän hetkisen menun toiminnallisuus on toteutettu JavaScript-kielellä. Linkki jäsentely on selkeää ja käyttäjän on helppo löytää haluamansa tieto menujen kautta. Sisältöä ei olisi järkeä jakaa kolmanteen alamenuun. Menurakenteessa on kuitenkin useampia selainkohtaisia ongelmia. Nämä tiedossa olevat ongelmat ovat HTML- ja CSS-pikkuvikoja ja osa ongelmista tulee vastaan ainoastaan Microsoftin Internet Explorer selainta käytettäessä. Kyseiset ongelmat vaikuttavat osaltaan käytettävyyteen sekä laskevat sivuston visuaalista näyttävyyttä.

Ohjaajapuolen ongelma on päämenussa, joka näyttää oudolta käyttäjän tuodessa hiiren päämenun linkin päälle. Avautuvan linkin muotoilu ei näytä hyvältä ja lisäksi vanhemmalla Internet Explorer selaimella avautuvaa linkkiä ei pysty painamaan (kuva 20). Painikkeen sijainti on myös arveluttava. Järjestelmään lisätään uusia opiskelijoita ohjaajan toimesta, joten painike uuden opiskelijan lisäämiseen voisi olla suoraan päämenussa. Nyt painike on piilossa painikkeen alla, eikä käyttäjä välttämättä ymmärrä katsoa oikeasta paikasta.

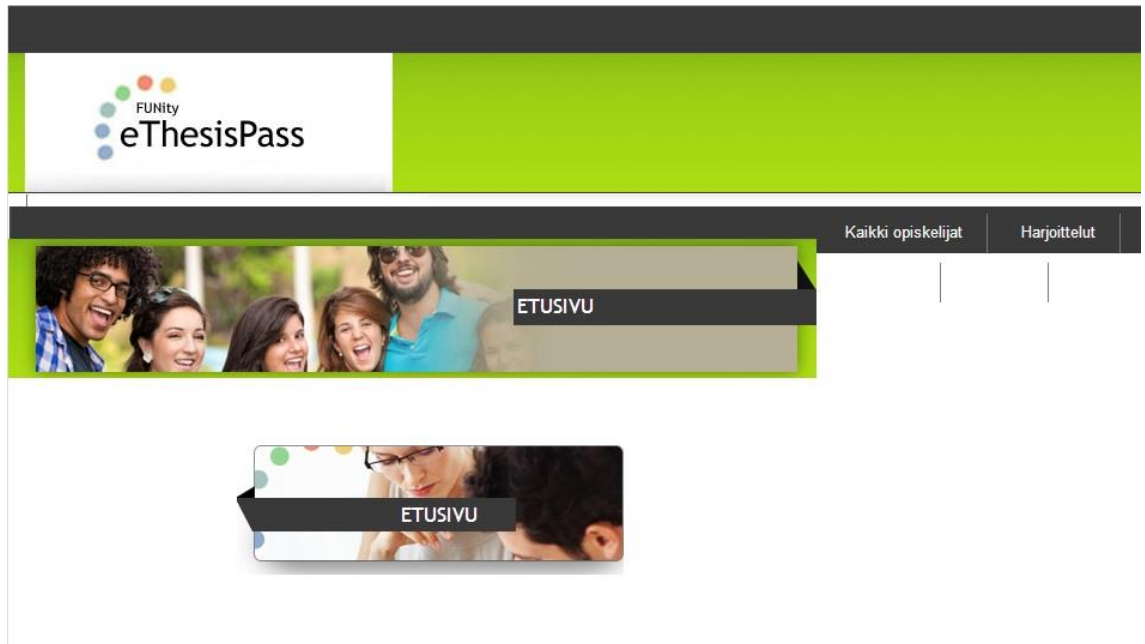


Kuva 20. Vasemmalla ohjaajapuolen päämenu Google Chrome selaimella, hiiren kurssi on "Kaikki Opiskelijat" -painikkeen päällä. Oikealla sama päämenu auki Internet Explorer selaimella, jolla linkki avautuu väärään paikkaan.

Toinen vastaan tullut ongelma löytyi, kun sivustoa käytetään kapeammalla selainikkunan koolla sekä opiskelija-, että ohjaajapuolella. Kun selainikkunan koko muuttuu liian pieneksi, hyppää ylhäällä oleva kuva pois paikaltaan. Tämä vaikeuttaa käyttöä tilanteissa, joissa selainikkunaa ei voida levittää koko ruudulle. Sisältö skaalaa kuvan pudotessa sen oikealle puolelle, jolloin selainikkunan kokoa joudutaan kasvattamaan tai sivua tarvitsee rullata oikealle nähdäkseen sisällön.



Kuva 21. FUNity ohjaajapuoli, selainikkuna koko ruudulla



Kuva 22. FUNity ohjaajapuoli, selainikkuna on pienennetty alle puoleen.

Kyseisiin ongelmiin on tavoitteena löytää ratkaisu HTML5- ja CSS3-ominaisuuksia hyödyntäen. Portaalissa navigoiminen onnistuu, mutta sen ulkoasua voitaisiin yhtenäistää muun sivuston kanssa. Uuden navigoinnin avulla päästäisiin samalla eroon käytettävyysongelmista.

6 FUNITYN KÄYTTÖLIITTYMÄN PÄIVITTÄMINEN

Opinnäytetyön tuloksena FUNity-järjestelmän opiskelija- ja ohjaajapuolelle luotiin muutamia erilaisia navigointimalleja. Erilaiset menuvaihtoehdot pohjautuvat suosituimpiin Internetissä käytettyihin navigointiratkaisuihin. Navigointimallien avulla etsittiin ratkaisua FUNity-portaalin pikkuvikoihin sekä mietittiin kuinka uusia web-tekniikoita voidaan hyödyntää verkkosivustojen ulkoasun ehostamiseen. Tämän tutkimuksen tiimoilta luodut navigointiratkaisut ovat suunniteltu käytettäväksi tietokoneen ruudulta, vaikka voivat taipua myös älypuhelin näytöille. Kokonaisen mobiiliystävällisen sivuston laatiminen ei sisälly tämän opinnäytetyön tutkimukseen.

6.1 Esivalmistelut

2013-luvun web-kehityksen trendeihin liittyy vahvasti sivustojen muotoilu mobiili- ja tablet-laitteille. CSS3:n avulla elementeille luodaan läpinäkyvyys, varjostukset ja siirtymäefektit. FUNityn uutta navigointia miettiessä täytyi tutkia millaisia navigointiratkaisuja sivustot ovat uusien ominaisuuksien varassa luoneet. Navigointityypeistä vertikaaliset sivumenut sekä suuret pudotusvalikot ovat yleistyneet (Rocheleau 2013).

FUNity-portaali on jaettavissa kolmeen osioon: yläbanneri, navigointi sekä sisältöosa. Yläbanneri sisältää FUNityn logon sekä aihealueittain vaihtuvan kuva-elementin. Alueen HTML rakenne-elementit tuli miettiä uudestaan vastaamaan HTML5-standardeja. Navigointimallit kehitettiin tyhjästä, sillä FUNityn oma CSS-kirjasto on erittäin pitkä, eikä tarkoituksena ollut puuttua sivuston sisällön muotoiluun tai graafisten elementtien sijainteihin. Työkaluina kehityksessä hyödynsin XAMPP apache http serveriä näyttämään web-sivustot paikallisesti omalla tietokoneella ja Notepad++ editoria koodin kirjoittamiseen.

6.2 Yläbanneri

FUNityn tapauksessa yläbanneri on alue sivustolla, joka pitää sisällään sivuston logon ja kuva-elementin. Ongelmana FUNityn yläbannerissa on kuva-elementin putoaminen alas sille asetetulta paikalta, kun selaimenikkunan koko muuttuu liian pieneksi. Kuvan putoaminen johtuu sen isäntä-elementille asetetusta leveydestä ja div-elementeille asetetusta float-arvosta.

```

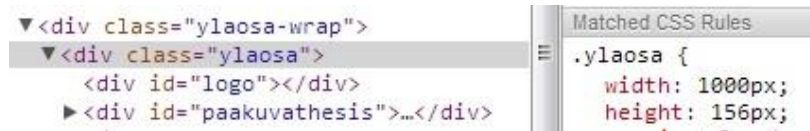
▼ <body>
  ▼ <div id="ylaosa">
    <div id="logo">
    </div>
    ▶ <div id="paakuvaowndata">...</div>
  #ylaosa {
    width: 100%;
    height: 156px;
    background: ▶
      url(../img/header/paakuvabg.gif)
  }
  
```

Kuva 23. FUNity yläbannerin HTML- ja CSS-koodi tarkasteltaessa Google Chrome selaimella.

Tarkasteltaessa FUNity-portaalin HTML- ja CSS-asettelua yläbannerille Google Chrome selaimella (kuva 23) nähdään, että div-elementti

id="ylaosa" sitoo div-elementit logolle ja kuva-elementille. Yläosan leveydelle on asetettu arvo width:100%, joka tarkoittaa, että leveys on aina yhtä suuri kuin selainikkunan koko. Tämän avulla taustaväri värittää koko yläbannerin riippumatta millä resoluutiolla sivustoa selataan. Logo- ja kuvaelementille on asetettu arvo float:left, jolloin elementit hakeutuvat mahdollisimman vasemmalle ja asettuvat näin vierekkäin. Isäntäelementille asetettu leveys prosentuaalisesti on kuitenkin ongelmaksi, sillä selainikkunan leveys voi jäädä pienemmäksi kuin logo- ja kuvaelementin yhteisleveys. Näin tapahtuessa kuvaelementti ei mahdu isäntäelementin ylaosa sisään ja putoaa alas vasemmalle.

Ratkaistaksemme ongelman, on div-elementille ylaosa asetettava kiinteä leveys. FUNityn yläbannerin HTML-dokumenttiin sijoitetaan uusi div-elementti, joka pitää kuva- ja logoelementit paikoillaan. Yhdelle div-elementille on kuitenkin asetettava leveydeksi 100%, sillä näin yläbannerissa pysyy taustaväri selainikkunan koosta riippumatta.



Kuva 24. FUNity yläbannerin korjattu muotoilu estää kuvaelementin putoamisen.

6.3 Opiskelijaportaalin navigointiratkaisut

FUNityn opiskelijapuolelle luotiin kolme erilaista menumallia, joissa haettiin eri näkökulmia kuinka FUNityn navigointi voisi toimia. Menumallit hyödyntävät osaa uusista CSS3-ominaisuuksista, joiden käyttöä voidaan perustellusti suositella. Koska sivuston navigoimiseen tarkoitettut menut ovat suhteellisen yksinkertaisia ja rajattuja kokonaisuuksia, ei uusille HTML5-ominaisuuksille ollut juuri tarvetta. Seuraavat navigointimallit ovat suunniteltu ainoastaan FUNityn opiskelijaportaalin käyttöön.

6.3.1 Alasvetovalikko

Ensimmäinen opiskelijaportaalin navigointimalli perustuu piilotettuihin painikkeisiin. Alasvetovalikot menuissa ovat erittäin yleisiä useilla Internet sivustoilla. CSS:n avulla piilotetaan menun toinen ja kolmas taso asettamalla listaelementille arvo display: hidden. Pseudo-luokan :hover avulla painikkeen elementin tyyli muutetaan muotoon display: block. Näin listaelementti ilmestyy näkyviin ja avautuvia linkkejä selataan hiiren kursorin avulla.



Kuva 25. FUNity opiskelijaportaalin ensimmäinen navigointimalli koostuu alasveto painikkeista, jotka saadaan näkyviin hiiren kursorin avulla.

Navigointiosan merkitsemiseen on käytetty HTML5 rakenne-elementtiä `nav` viittaamaan navigointiosan paikkaa. Linkkien rakenne on järjestetty HTML-dokumentissa `ul` ja `li`-elementtien avulla. Muotoilu ja menun toiminnallisuus ovat toteutettu täysin CSS-tyyliin avulla. Päälinkkien ympärille muodostuvat pyöristetyt reunat valinnan mukaan, mutta myös osoittamaan millä sivulla käyttäjä on sillä hetkellä. Reunojen pyöristys tapahtuu CSS3 `border-radius` ominaisuuden avulla. Menun toisen ja kolmannen tason linkeistä ainoastaan ensimmäisen ja viimeisen linkin reunat ovat pyöristetty. Erienevä muotoilu ensimmäiseen linkkiin on toteutettu CSS3 pseudo-luokan ominaisuudella `first-child` ja listan viimeiseen linkkiin viitataan arvolla `last-child`.

Alas vedettävän menun vahvuus on tilan säästämässä. Moniportainen menu saadaan tiivistettyä pieneen tilaan. FUNityn tapauksessa päästään eroon ylimääräisen sivumenu tarpeesta, jolloin sisällölle jää enemmän tilaa. Ongelmana alavetomenussa on toisaalta painikkeiden näkyvyys: käyttäjä ei välittömästi näe etsimäänsä sivua ja joutuu selaamaan linkkien sisältöä. Tällöin korostuu linkkien järjestely ja kuinka osuvasti ne on nimetty. FUNityn opiskelijaportaalin käyttöön menu toimisi hyvin, sillä aihekokonaisuudet ovat jäsennellyt ymmärrettävästi eikä menupainikkeita ole liikaa. Menun toimivuus on todettu selainversioilla Google Chrome 26, Mozilla Firefox 20 ja Internet Explorer 8-11, eikä ongelmia havaittu.

6.3.2 Suuret painikkeet

Seuraava lähestymistapa FUNityn navigoimiseen oli luoda suuret ja visuaaliset painikkeet, joiden avulla siirtyä aihekokonaisuudesta toiseen. Suuret painikkeet voitaisiin sijoittaa sivuston yläosaan tai sivuun, jolloin toisen tason linkit avautuisivat näkyviin sivun ylä- tai alalaitaan.



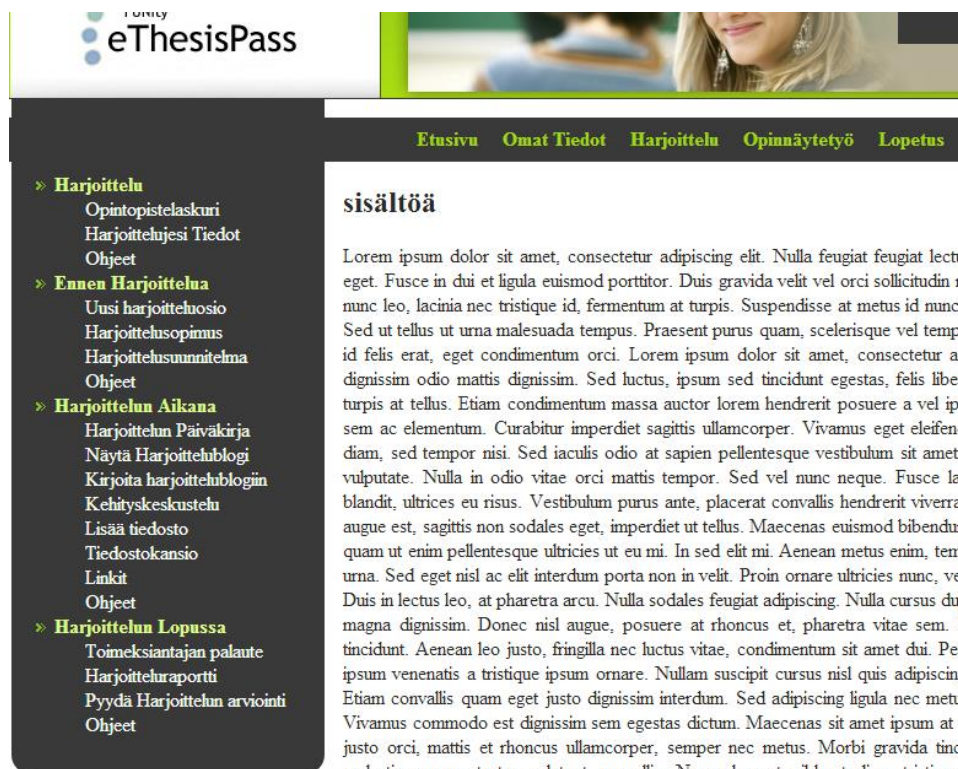
Kuva 26. Opiskelijaportaalin navigoinnin luonnos, jossa haetaan visuaalisempaa ilmettä suurten painikkeiden avulla

Menu on toteutettu CSS3:n avulla ja painikkeet ovat linkkejä aihekokonaisuuksien välillä siirtymiseen. Käyttäjän viedessä cursorin linkin päälle painike suurenee ja teksti vaihtaa väriä. Efekti on saavutettu CSS3 siirtymäefektin arvolla scale ja animaatiolla smallToBig. Symbolin väri vaihtuu samalla haaleammaksi sille asetetun läpinäkyvyyden vuoksi. Symbolit ovat kuvakkeita, jotka ovat luotu käyttämällä erillistä kirjasinpakettia. CSS-muotoilussa ominaisuus @font-face mahdollistaa kirjaintyypin käyttämisen, jotka sijaitsevat palvelimella. Käyttämäni web-kirjasinpaketti on nimeltään websymbols-regular ja se sisältää erilaisia symboleita, joita sijoitetaan sivulle tekstin tavoin. Näin säästytään luomasta erilisiä kuvastodostoja, jotka ovat sidottuja niille asetettuun resoluutioon.

Näin suuret painikkeet ovat selkeitä ja herättävät huomiota, mutta eivät ole kaikista käytännöllisimmät järjestelmän navigoimiseen. Ongelmaksi muodostuu myös alamenujen sijoittaminen. Yksi mahdollisuus olisi sijoittaa ne sivuston ylälaitaan ja hyödyntää mallissa yksi (kuva 25) esittelemääni alavetovalikkoa. Selaintuki näin monen CSS3-ominaisuuden kanssa rajaa visuaalisimman lopputuloksen uusimpiin selainversioihin, mutta käytettävyys säilyy myös vanhoilla Internet Explorer selainversiolla aina versioon 8:n saakka.

6.3.3 Vaihtoehtoinen sivumenu

FUNity hyödyntää tällä hetkellä sivumenua sivuston vasemmassa reunassa. Sivumenuun hyöty on, että kaikki linkit ovat aihekokonaisuuden mukaan kokoajan näkyvissä. Sivumenu rajaa sisältöä, joka sijoittuu menun viereen. Sivumenu on toteutettu hyödyntäen FUNityn teeman värimaailmaa ja ilman JavaScriptiä.



Kuva 27. Opiskelijapuolen kolmas menuehdotus on vanha sivumenu tehty uudestaan yhdistellen teeman värimaailmaa.

JavaScriptin puutteen vuoksi menuja ei voida enää piilottaa hiirellä klikkaamalla, mutta kaikki linkit ovat näkyvissä aihekokonaisuuden mukaan. Menun muotoilussa on käytetty maltillisesti CSS3-ominaisuuksia. Käyttäjän viedessä hiiren linkin ylle linkin väri vaihtuu asteittain, joka on saavutettu siirtymäefektin avulla. Menualueen alimmat nurkat ovat pyöristetty border-radius ominaisuudella. CSS3 media query:n avulla sisältöosa siirtyy sivumenuun alapuolelle, jolloin käytettävyys säilyy kapeammallakin ruudulla.

Menun käytettävyys on hyvällä tasolla. Koska linkkejä ei ole piilotettu CSS:n tai JavaScriptin avulla, onnistuu navigoiminen mobiili-laitteillakin. Linkit ovat jäsenneilty värien avulla, jolloin hahmottaminen on selkeää. Pitkien listojen käyttö ei ole aina hyväksi, sillä käyttäjä joutuu selaamaan listaa läpi löytääkseen hakemansa. FUNityn tapauksessa opiskelijapuolella linkkejä ei ole liikaa, joten sivumenua voidaan käyttää huoletta. Menun toteutus on myös lähellä FUNityn käytössä olevaa navigointiratkaisua, joten käyttöönotto on helppoa. Järjestelmän vanhat käyttäjät ovat tottuneet käyttämään kyseistä mallia.

6.4 Ohjaajapuolen navigointiratkaisut

Ohjaajapuoli on käytettävyydeltään samanlainen kuin opiskelijaportaali, mutta sisältöä on enemmän. Ohjaajaportaalin linkkien määrää vaihtelee muun muassa järjestelmässä ohjattavien opiskelijoiden määrän mukaan. Tämän vuoksi navigointiratkaisun tuli olla erilainen kuin opiskelijapuolella, jotta suuri määrä linkkejä saataisiin jäsenneiltyä järkevästi. Päämenuun

tuli yhteensä kahdeksan painiketta, joista neljä sisältää alalinkkejä. Lisää opiskelija-painike siirtyi osaksi päämenua, sillä mielestäni painikkeen kannattaa olla näkyvässä käytettävyyden parantamiseksi. Muuten linkkirakenne on selkeä, eikä tarvitse muutoksia.

6.4.1 Megamenu

Englanniksi suuria alasvetovalikoita kutsutaan usein nimellä mega drop-down menu. Megamenu on toimintaperiaatteeltaan sama kuin opiskelija-puolen ensimmäinen menuvaihtoehto. Sen tarkoituksena on jäsentää CSS:n avulla suuri määrä linkkejä painikkeen alle piiloon.



Kuva 28. Ohjaaja-portaalin megamenu

Megamenun HTML-dokumentissa ei riitä pelkkien listaelementtien ul ja li käyttö, vaan linkit ovat sidottu div-elementtien sisään. Näin div-elementit voidaan jakaa sarakkeisiin muuttamalla yhden elementin leveyttä verrattaessa menun kokonaisleveyteen. Visuaalinen puoli on rakennettu CSS3:n avulla. Taustaväri on CSS3 gradient ja reunojenpyöritykset ovat luotu border-radius ominaisuudella. Avautuva menualue on sidottu sulavasti painikkeeseen poistamalla painikkeesta alareunan reunojenpyöritykset ja lisäämällä arvoa padding-bottom, jolloin painike venyy alaspäin. Linkkien eteen on asetettu pieni kuvatiedosto erottamaan linkkien sijainnit, sillä pidemmät linkit jakaantuvat väkisin kahdelle riville, joka vaikeuttaa luettavuutta. Tilan säästämiseksi ylläpito-menu avautuu poikkeuksellisesti vasemmalle, jotta nettisivustoa voidaan käyttää kapeammalla näytöllä ja selainikkunalla.

Megamenuun mahtuu suuri määrä sisältöä, jolloin tilaa säästyy. Linkkejä ei myöskään tarvitse piilottaa useaan tasoon. Käytettävyys on hyvä ja hiirellä on helppo liikkua linkkien välillä. Ongelmaksi muodostuu kuitenkin ruudunkorkeus, sillä kahdeksan pitkän sarakkeen sijoittamisen avuksi tulisi ottaa JavaScript, jotta linkkejä voitaisiin tarvittaessa piilottaa hiirellä klikkaamalla. Menu toimii uusimmilla selainversioilla, mutta gradient

taustaväri katoaa vanhemmilla. Tämä vaikuttaa kuitenkin vain menun ulkoasuun.

6.4.2 Hover-sivumenu

Toinen lähestymistapa ohjaajapuolen navigointiin oli ottaa vertikaalinen sivumenu takaisin, mutta yhdistää siihen pudotusvalikon toiminnallisuus. Näin saadaan suurempikin linkkimäärä kätevästi piiloon, eikä sivumenu ulotu sivuston alareunaan saakka, kuten tällä hetkellä ohjaajapuolella tapahtuu.



Kuva 29. Ohjaajamenun toinen navigointimalli sisältää vertikaalisen sivumenuun, joka sisältää piilotettuja linkkejä.

Toisen tason linkit tulevat näkyviin siirryttäessä ylämenun painikkeista halutun aihekokonaisuuden alueelle. Kuvassa (kuva 29) on auki harjoittelut - sivusto, jolloin vasemmalle ilmestyy sivuston aihealueet. Sivumenuun otsikot aukeavat hiiren kosketuksesta, jolloin nähdään piilotetut linkit. Menuussa on käytössä samoja CSS3-ominaisuuksia kuin aikaisemmin, muun muassa painikkeiden taustaväri on CSS3 gradient ja reunoja on pyöristetty border-radius ominaisuudella.

Käytettäessä piilotettuja linkkejä, vältetään liian pitkältä listalta. FUNityn ohjaajapuolella linkkejä voi olla suuri määrä. Liian pitkiä listoja on hyvä välttää, joten tämänkaltainen ratkaisu toimisi hyvin selkeyden ja käytettävyyden näkökulmasta. Käyttäjä joutuu kuitenkin selaamaan otsikoita löytääkseen hakemansa linkin. Uusien ohjaajien on työläämpää löytää oikea linkki, verrattuna ensimmäiseen ohjaajaportaalin navigointimalliin. Myös vanhemmilla mobiiliselaimilla linkkien käyttö tuottaa ongelmia ja käytettävyys ei ole optimaalisin. Tämä voitaisiin korjata toteuttamalla menu vaihtoehtoisesti mobiiliselaimille JavaScriptillä CSS:n sijasta.

6.5 HTML5 ja CSS3 osana FUNitya

HTML5 web-sivusto ilmoitetaan selaimille HTML-dokumentin alussa muodossa `<!DOCTYPE html>`. Kyseinen julistus HTML-dokumentin alussa kertoo selaimelle, että sivusto käyttää HTML5-standardeja ja näin selain osaa kääntää sivuston oikein. (W3Schools 2013f) FUNityn HTML rakenne-elementit ovat kuitenkin HTML 4.1 -version mukaisia, jolloin kyseinen määrittely ei ole tarpeen. Rakenne-elementit auttavat hahmottamaan koodia paremmin. HTML-dokumentin päivittämisen hyöty vanhalla sivustolla on kyseenalaista, sillä se ei muuta sivuston ulkoasua tai toimintaperiaatetta. Uusista rakenne-elementeistä voitaisiin FUNity -portaalissa hyödyntää header-elementtiä ilmaisemaan yläbanneria, nav-elementtiä ilmoittamaan navigointiosaa sekä section-elementtiä osoittamaan sivuston sisällön paikka.

HTML5 kannattaa ottaa käyttöön uutta sivustoa suunniteltaessa tai suuren päivituksen yhteydessä. FUNityn tapauksessa HTML-elementtien muuttaminen aiheuttaa paljon työtä muutoksen hyötyyn nähden, sillä JavaScript ja CSS-koodissa viitataan usein HTML-elementeille annettuihin luokkiin. Rakenne-elementtien uusiminen ei siis ole mielestäni kannattavaa, ellei järjestelmään tehdä samalla suuria muutoksia koodinpuolella.

CSS3-ominaisuudet ovat huomattavasti helpompi sijoittaa osaksi vanhaa sivustoa, koska niiden puute vanhoilla selaimilla ei usein tuhoa sivuston käytettävyyttä. Uusien FUNity navigointimallien luomisessa CSS3 säästi selvää aikaa, kun graafista sisältöä voidaan luoda ilman ylimääräisiä kuvatiedostoja. Hyödyllisimpiä ja selainyhteensopivimpia ominaisuuksia ovat muun muassa reunojenpyöritykset, varjostuksien luominen, siirtymäefektit ja gradient-taustaväri. Niitä helppo käyttää ja ne tuovat uutta graafista ilmettä.

7 TULOSTEN ANALYSOINTI

Opiskelija- ja ohjaajapuolen navigointi voidaan toteuttaa monella eri tavalla. Esittelemäni navigointimallit ovat luonnoksia ja niiden jatkokehitys on mahdollista. Esimerkiksi navigointimallien ulkoasu on helposti muutettavissa. Malleja tulee analysoida erityisesti käytettävyyden, lähestyttävyyden sekä teknisen toteutuksen näkökulmasta. Kysymykselle, mikä navigointimalli on paras, on vaikeaa löytää yksimielistä vastausta. Tämän vuoksi uusia näkökulmia navigointiratkaisujen hyvistä ja huonoista puolista kartoitin haastattelemalla kahta asiantuntevaa Hämeen ammattikorkeakoulun tietojenkäsittelyn opettajaa. Haastattelutilanteessa opettajat käyttivät jokaista navigointimallia ja keskustelimme jokaisen mallin käytettävyydestä sekä miten niillä luonnistuu FUNity-portaalin selaaminen. Pohdimme haastateltavien kanssa myös, miten menuja voitaisiin jatkossa kehittää ja mihin osa-alueisiin investointi olisi kannattavaa.

Navigointimallien kartoituksessa ei suoritettu laajempaa käyttäjätutkimusta. Syynä tähän oli, että laajemmalle tutkimukselle ei riittänyt ajallisia resursseja. Navigointi on myös pieni osa portaalin käytettävyyden kokonaisuutta, mitä voidaan tehostaa muillakin tavoilla. Uskon FUNityn käyttäjien myös hallitsevan esittelemieni navigointiratkaisujen käytön, sillä navigoinnit pohjautuvat tuttuihin, useilla Internet-sivustoilla käytettyihin ratkaisuihin.

7.1 Opiskelijaportaalin menuratkaisujen analysointi

FUNity-järjestelmän menurakenteen kohdalla on tehtävä kompromisseja. Suhteellisen varma ja käytännönläheinen vaihtoehto opiskelijaportaalin navigoinnille olisi alavetovalikon tai sivumenuun käyttö. Alavetovalikko on selkeä kokonaisuus ja säästää tilaa. Se on myös hyvä käyttää, sillä linkkien sijoittuminen keskelle ruutua helpottaa niiden lukemista. Sivumenu on myös selkeä lukea, mutta pitkän listan selaaminen voi olla työlästä, siksi piilotetut painikkeet ovat parempi lähestymistapa. (Joukainen, haastattelu 31.05.2013) Alavetovalikon linkit eivät kuitenkaan ole käyttäjälle välittömästi näkyvissä, jolloin niiden välillä joutuu uusi käyttäjä selailemaan etsiessään hakemaansa tietoa. Tämä on käytettävyyden näkökulmasta huonompi vaihtoehto verrattuna sivumenuun käyttöön. Sivumenu näyttää linkit käyttäjälle selkeästi listassa ja sen linkit ovat myös helpommin muokattavissa mobiilistävälliseksi CSS:n avulla.

Järjestelmään saataisiin sen kaipaamaa visuaalista ja nykyaikaista ilmettä, mikäli pääpaino menurakenteessa asetettaisiin sen ulkoasuun. Näin järjestelmä olisi helpommin lähestyttävä. (Saksa, haastattelu 22.05.2013) Opiskelijapuolen suuria painikkeita hyödyntävä menukokonaisuus on menu-malleista visuaalisin. Suuret painikkeet ovat luontevia myös mobiiliselaimelle. Ne ovat selkeitä käyttää, mutta kuitenkin tilaa vieviä (Joukainen, haastattelu 31.05.2013). Visuaalista näyttävyyttä voitaisiin hakea käyttämällä CSS3 transition- ja animaatio-ominaisuuksia. Näitä ominaisuuksia ei tarvitse sijoittaa suuriin painikkeisiin, vaan niitä voitaisiin soveltaa esimerkiksi sivumenuun. Visuaalisuutta painottava lähestymistapa olisi

muokata sivumenun otsikot näyttävimmiksi ja animoida linkkien esiin siirtyminen.

Opiskelijaportaalin navigointia ja sen ulkoasua pohtiessa, on tärkeää huomioida opiskelijan mahdollinen puna-vihersokeus (Joukainen, haastattelu 31.05.2013). Koska järjestelmän portaalin tämänhetkinen väriteema yhdistelee paljon vihreää ja mustaa väriä, on vihreää väriä käytetty paljon eri navigointimalleissa. Lopulliseen versioon tekstin värinä tulisi välttää vihreää, mutta taustassa vaaleanvihreän käyttö ei tuota ongelmia (Terveyskirjasto 2012).

7.2 Ohjaajapuolen menuratkaisujen analysointi

Visuaalisia animaatioita ei tarvitse tuoda ohjaajapuolelle, sillä sitä käyttävät pääosin opettajat ja järjestelmän ylläpitäjät. Ohjaajapuolella tärkeintä on portaalin helppo navigoiminen, nopea siirtyminen linkkien välillä sekä selkeys. Ohjaajapuolella ei ole myöskään käyttäjiä niin paljon kuin opiskelijapuolella. Linkit näyttäytyvät erilailla eri ohjaajille, riippuen heille asetetuista oikeuksista. Linkkien rakenne muuttuu myös muun muassa ohjattavien opiskelijoiden määrän mukaisesti.

Suuren alavetovalikon käyttö sai kahdesta vaihtoehdosta eniten kannatusta. Se on selkeä käyttää ja linkkien välillä on helppo siirtyä, jota edesauttaa linkkien aukeaminen keskelle ruutua (Joukainen, haastattelu 31.05.2013). Linkkien piiloutuessa täysin yläosaan jää sisällölle enemmän tilaa. Alavetovalikon linkkejä tulisi kuitenkin jäsentää järkevämmiin, jotta luettavuus paranee. Esimerkiksi neljän samanmittaisen sarakkeen asettaminen vierekkäin on hyvä, sillä selkeä linja säilyy. Avautuvasta valikosta tulisi ottaa pääotsikko pois, sillä aihekokonaisuus ilmoitetaan painikkeessa. Näin välttyään turhalta toistolta ja säästetään tilaa. (Saksa, haastattelu 22.05.2013) Ylläpitovalikko näyttäytyy vain muutamille ja harvoille sivuston ohjaajille. Koska se sisältää huomattavan määrän linkkejä, ylläpitovalikon jakamista kahteen erilliseen osioon voisi harkita (Joukainen, haastattelu 31.05.2013).

Sivumenumallin ongelmana on, että siitä puuttuu opastusteksti. Käyttäjälle ei ilmoiteta missään selkeästi, että minkä sivuston alla hän sillä hetkellä on. Sivumenumun piilotetut painikkeet myös aiheuttavat käyttäjälle huomattavasti enemmän selailua linkkien välillä. (Saksa, haastattelu 22.05.2013) FUNityn ohjaajaportaalin tämänhetkinen sivumenu on todella pitkä, mutta siinä korostuu otsikoittain piilotettavien linkkien piilotuksen hyöty. CSS:n avulla piilotettu sivumenu ei välttämättä ole paras lähestymistapa ohjaajapuolelle, linkkien suuren ja vaihtelevan määrän vuoksi.

7.3 Jatkokehitys

Navigointimallien ulkoasua tulisi pohtia, jotta se istuu järjestelmän kokonaisilmeeseen. Alavetovalikoiden taustavärinä haaleat värit saivat eniten kannatusta, sillä tummalla taustalla valkoisen tekstin lukeminen on raskeampaa (Joukainen, haastattelu 31.05.2013). Menujen toiminnallisuutta

voidaan vaihtoehtoisesti parantaa korvaamalla CSS-ominaisuuksia JavaScript-ominaisuuksin, muun muassa mobiililaitteille.

Mikäli mobiilikäyttö on tärkeä kriteeri, on järkevintä käyttää kahta erilaista navigointiratkaisua, joista oikean valitseminen tapahtuu CSS3 media queryn avulla. Harkinnan arvoista on myös tehdä sivustosta kerralla mobiiliystävällinen. Kysymys on, kuinka paljon mobiiliystävällisen ulkoasun laatumiseen halutaan käyttää resursseja. Tulisi selvittää, onko mobiilisivustolle tarvetta opiskelijoiden tai ohjaajien puolesta. Mikäli tarkoituksena on päästä ainoastaan selailemaan FUNity portaalia, voidaan tämä saavuttaa käyttämällä navigoinnissa mobiiliystävällisiä elementtejä. Tällöin CSS:n avulla piilotetut painikkeet eivät ole sopivin ratkaisu, vaan sivumenun tai suurempien painikkeiden käyttö on suotavampaa. Esimerkiksi sivumenu-mallin linkit voidaan helposti sitoa painikkeen taakse piiloon ja CSS:n avulla muotoilla linkit tarpeeksi suuriksi kosketusnäyttöjen kannalta.

8 YHTEENVETO

Opinnäytetyön tavoitteena oli tutkia uusien web-tekniikoiden HTML5- ja CSS3-ominaisuuksia sekä hyödyntää niitä osana FUNity-opinnäytetyöjärjestelmän navigointia. Molemmat tekniikat ovat erittäin laajoja käsitteitä, eikä kaikkia uusia ominaisuuksia saatu käsiteltyä opinnäytetyön aikana. Teoriaosuus on rajattu lähteinä käytetyn kirjallisuuden mukaan tärkeimpiin osa-alueisiin. Käytännönsuuden navigointimallit hyödyntävät uusia CSS3-ominaisuuksia, jotka ovat erittäin käyttökelpoisia tuomaan visuaalista ilmettä osana vanhempaakin sivustoa. 2013-luvun web-kehityksen trendeihin liittyvät vahvasti CSS3-efektien käyttö, joten niiden opetteleminen on suotavaa. Teknisestä näkökulmasta HTML5 ei tuonut mitään hyötyä näin pienen osan päivitysprosessiin, mutta tarjoaa huomattavan määrän uusia ominaisuuksia web-kehittäjille varsinkin uutta sivustoa suunniteltaessa.

Tutkimuskysymykseen, mikä on paras tapa FUNity opiskelija- ja ohjaaja-portaalin navigoimiseen, löytyi useampia käyttökelpoisia ratkaisuja. Parhaan ratkaisun löytämiseksi tulee asiaa harkita useasta eri näkökulmasta. Opiskelijaportaalin navigoimiseen sopisi parhaiten yksinkertaiset alasve-tovalikot tai sivumenu. Ohjaajaportaalin menuratkaisuksi suuri pudotusva-likko vaikutti käytettävyydeltä sekä selkeydensä vuoksi parhaimmalta rat-kaisulta. Kun käytössä on tuttuja teknisiä ominaisuuksia ja samanlaisia menuratkaisuja käytetään usealla Internet-sivustolla, on niiden käyttöön siirtyminen helppoa. Uudet tekniikat mahdollistavat visuaalisen lopputu-loksen ja yhdistelemällä CSS3-transition ja animaatio-efektejä saavutetaan helposti lähestyttävä kokonaisuus. Lopullisen navigoinnin käyttöönoton yhteydessä on värimaailmaa ja ulkoasua mahdollista muuttaa pienellä vai-valla niiden omasta CSS-tyylistä.

CSS3 media queryn käytön kannalta FUNity-opinnäytetyöjärjestelmän portaalin navigoimiseen voitaisiin helposti luoda oma mobiililaitteille suunnattu menuratkaisu. Sopivia menuratkaisuja olisivat suuret painikkeet sekä sivumenujen käyttö ilman CSS:n avulla tapahtuvaa painikkeiden pii-lotusta. Tarve mobiilistävälliselle sivustolle tulisi kartoittaa erikseen ja miettiä, kuinka suurta osaa sivustosta on tarve käyttää mobiilisti.

LÄHTEET

- Addison, W. 1998a. 2- A history of HTML.
Luettu 18.04.2013.
<http://www.w3.org/People/Raggett/book4/ch02.html>
- Addison, W. 1999b. Cascading Style Sheets, designing for the Web (2. painos). Luettu 20.04.2013
<http://www.w3.org/Style/LieBos2e/history/Overview.html>
- Adobe 2012. An Update on Flash Player and Android. Luettu: 27.04.2013
<http://blogs.adobe.com/flashplayer/2012/06/flash-player-and-android-update.html>
- Antharia n.d. Understanding Site Navigation: Key Terms (Part 1).
Luettu 24.04.2013
<http://www.antharia.com/resources/details.php?id=86>
- Gasston, P. 2011. The Book of CSS3: A developer's guide to the future of web design. San Fransisco: William Pollock.
- Gervais, B. 2012. CSS Gradients with Internet Explorer (IE6-10). Luettu: 09.05.2013
<http://www.briangervais.com/blog/css-gradients-with-internet-explorer-ie6-10>
- html5boilerplate 2013. The web's most popular front-end template. Luettu: 27.04.2013
<http://html5boilerplate.com/>
- html5please 2013. HTML5 Please. Luettu: 27.04.2013
<http://html5please.com/>
- Hunt, T. 2013. The impending crisis that is Windows XP and IE 8. Luettu: 27.04.2013
<http://www.troyhunt.com/2013/01/the-impending-crisis-that-is-windows-xp.html>
- Joukainen, P. 2013. Tietojenkäsittely opettaja. Hämeen ammattikorkeakoulu. Haastattelu 31.05.2013
- Knight, K. 2011. Planning And Implementing Website Navigation. Luettu 24.04.2013
<http://www.smashingmagazine.com/2011/06/06/planning-and-implementing-website-navigation/>
- Korpela, J. 2011. HTML5 – Uudet ominaisuudet. Jyväskylä: WSOY.
- Kyrnin, J. n.d.a. What is CSS? Luettu: 27.05.2013
<http://webdesign.about.com/od/beginningcss/a/aa021607.htm>

Kyrnin, J. n.d.b. CSS Vendor Prefixes - What Are They and Why You Should Use Them. Luettu: 15.05.2013.

<http://webdesign.about.com/od/css/a/css-vendor-prefixes.htm>

Longtail 2013. The State Of HTML5 Video.

Luettu: 27.04.2013

<http://www.longtailvideo.com/html5/>

Margaret, R. 2005. HTML (Hypertext Markup Language). Luettu: 27.05.2013

<http://searchsoa.techtarget.com/definition/HTML>

Microsoft 2013. Windows XP SP3 and Office 2003 Support Ends April 8, 2014. Luettu: 27.04.2013

<http://www.microsoft.com/en-us/windows/endofsupport.aspx>

Norman 2013, Browser Support Checklist CSS3. Luettu: 15.05.2013

<http://www.normansblog.de/demos/browser-support-checklist-css3/>

Nutil, P. n.d. Luettu: 15.05.2013

<http://www.shutterstock.com/pic-110942165/stock-vector-responsive-web-design-elements-are-displayed-on-different-devices.html>

Richter, J. n.d. The Internet Explorer 8 Countdown. Luettu: 27.04.2013

<http://theie8countdown.com/>

Rocheleau, J. 2013, Web Design: 20 Hottest Trends To Watch Out For In 2013. Luettu: 16.05.2013

<http://www.hongkiat.com/blog/web-design-trend-2013/>

Saksa, T. 2013. Tietojenkäsittely opettaja. Hämeen ammattikorkeakoulu. Haastattelu 22.05.2013

Seppänen, L. 2013. FUNity. Luettu 15.04.2013.

<http://www.FUNity.fi/>

Steve, 2010. HTML5 Tutorial: New Structural Elements!. Luettu 26.04.2013

<http://www.mrc-productivity.com/techblog/?p=724>

Steve, J. 2010. Thoughts on Flash. Luettu: 27.04.2013

<http://www.apple.com/hotnews/thoughts-on-flash/>

Storey, D. 2013. Public service announcement: time to update your CSS3. Luettu: 29.04.2013

<http://generatedcontent.org/post/37949105556/updateyourcss3>

Terveyskirjasto 2012. Värisokeus ja poikkeava värinäkö. Luettu: 31.05.2013

http://www.terveyskirjasto.fi/terveyskirjasto/tk.koti?p_osio=100&p_artikkeli=dlk00347&p_teos=dlk&p_selaus=7728

W3C n.d.a. HTML & CSS. Luettu 20.04.2013.

<http://www.w3.org/standards/webdesign/htmlcss.html>

W3C n.d.b. HTML structural elements. Luettu 26.04.2013

http://www.w3.org/wiki/HTML_structural_elements

W3Schools n.d.a. Introduction to XML. Luettu 18.04.2013.

http://www.w3schools.com/xml/xml_what_is.asp

W3Schools n.d.b. HTML5 Input Types. Luettu 26.04.2013

http://www.w3schools.com/html/html5_form_input_types.asp

W3School n.d.c. HTML5 Canvas. Luettu: 27.04.2013

http://www.w3schools.com/html/html5_canvas.asp

W3School n.d.d. HTML5 Web Storage. Luettu: 27.04.2013

http://www.w3schools.com/html/html5_webstorage.asp

W3School n.d.e. CSS3 Animations. Luettu: 15.05.2013

http://www.w3schools.com/css3/css3_animations.asp

W3Schools n.d.f. HTML <!DOCTYPE> Declaration. Luettu: 21.05.2013

http://www.w3schools.com/tags/tag_doctype.asp

Youtube n.d. YouTube HTML5 Video Player. Luettu: 27.04.2013

<http://www.youtube.com/html5>