

Jarmo Heikkinen

## **HTML5 JA SEN TOIMIVUUS SELAIMISSA**

## **HTML5 JA SEN TOIMIVUUS SELAIMISSA**

Jarmo Heikkinen  
Opinnäytetyö  
Kevät 2012  
Tietojenkäsittelyn koulutusohjelma  
Oulun seudun ammattikorkeakoulu

## TIIVISTELMÄ

Oulun seudun ammattikorkeakoulu  
Tietojenkäsittelyn koulutusohjelma.

---

Tekijät: Jarmo Heikkinen

Opinnäytetyön nimi: HTML5 ja sen toimivuus selaimissa

Työn ohjaaja: Teppo Räisänen

Työn valmistumislukukausi ja -vuosi: kevät 2012

Sivumäärä: 48

---

## TIIVISTELMÄ

Sain inspiraation opinnäytetyöhöni ohjaajani Teppo Räisäsen ehdotuksesta miettiessämme mahdollisia opinnäytetyön aiheita. HTML5 vaikutti hyvin mielenkiintoiselta aiheelta, ja opinnäytetyön aihe kelpasi myös harjoittelupaikkani BelleGamesin toimitusjohtajalle Pirjo Ritokangas-Huttuselle. Olimme sopineet harjoittelun aikana mahdollisuudesta tehdä opinnäytetyö yritykseen. Aluksi aiheen tarkastelunäkökulmana oli HTML5:den soveltuvuus pelien ohjelmointiin, mutta aihe tarkentui HTML5:teen ja sen toimivuuteen selaimissa.

Työn tavoite oli saada selville kuinka hyvin nykyiset selaimet tukevat HTML5:den uusia ominaisuuksia, ja verrata selainten tukea keskenään. Internetissä on tarkat tukimäärittelyt, mutta itse ainakin olen kokenut ne hyvin epäselviksi. Määrittelyt muistuttavat enemmän tilastotiedettä kuin helposti ymmärrettävää ohjeistusta, joten katsoin parhaaksi testata HTML5-demoja suoraan selaimilla. Näin saatiin konkreettisesti selville, kuinka hyvin tai kuinka huonosti selaimet pärjäävät.

Tietoperustaksi otin HTML:n historian vuodesta 1989 aina nykyhetkeen saakka, sillä en katsonut tarpeelliseksi perehtyä koko internetin historiaan. HTML:n historia alkaa kuitenkin vasta Tim Berners-Leen CERNin ajoista, joten katsoin tarpeelliseksi aloittaa historian tutkimisen vasta siitä ajanjaksosta. HTML5:den osalta tietoperustaksi otin sen historian, toimintaperiaatteet ja uudet ominaisuudet.

Tutkimuskohteena opinnäytetyössäni oli HTML5-demojen toimivuus selaimissa. Valitsin ensin kolmekymmentä HTML5-demoa tai -sivua, ja sitten arvoinkin tästä otannasta viisi. Jokaista demoa testattiin viidellä eri selaimella ja tulokset kirjattiin taulukkoon vertailua varten. Selaimet olivat Google Chrome, Mozilla Firefox, Apple Safari, Opera ja Windows Internet Explorer.

Tutkimuksen tuloksena Google Chrome ja Mozilla Firefox olivat parhaat selaimet suoriutuessa jokaisesta testauksesta täydellisesti. Huonoiten sijoittuivat Windows Internet Explorer ja yllättäen Opera. Ne suoriutuivat vain kolmesta testauksesta täydellisesti, epäonnistuivat täysin yhdessä ja yhden suorittivat osittain onnistuneesti. Apple Safari sijoittui näiden väliin suoriutumalla neljästä testauksesta täydellisesti ja yhdestä epäonnistuneesti. Testauksessa havaittiin myös pieniä eroja selainten esitystavoissa mutta ne eivät ole merkitseviä relevantteja tässä tutkimuksessa.

Opinnäytetyöni tuloksia voidaan hyödyntää ainakin verkkosivujen ja –sovellusten suunnittelussa, sillä heti työskentelyn alkuvaiheessa on hyvä tietää selain jolle sovellus tehdään. Vaihtoehtoisesti voidaan lähteä heti rakentamaan kaikissa selaimissa toimivaa sovellusta, mutta tässä tapauksessa vastaan voi tulla selaimien HTML5-tuen puute.

---

Asiasanat: HTML, HTML5, CSS, Selain

## ABSTRACT

Oulu University of Applied Sciences  
Degree Program in Business Information Systems

---

Authors: Jarmo Heikkinen

Title of thesis: HTML5 and its functionality in browsers

Supervisor: Teppo Räisänen

Term and year when the thesis was submitted: spring 2012      Number of pages: 48

---

## ABSTRACT

I got an inspiration for my thesis when me and my supervisor Teppo Räisänen talked about possible thesis subjects. HTML5 seemed very interesting subject and my commissioner - Pirjo Ritokangas-Huttunen, CEO of BelleGames - approved it as well. I had completed my training for BelleGames so we agreed that I will do my thesis to them as well. At first the point of view of the thesis was how suitable HTML5 is for game programming but subject was specified to HTML5 and its functionality in browsers.

The goal of this thesis was to find out how well the current browsers support HTML5 and its new features. In addition I compared how the browsers matched against each other. I decided not to use the exact definitions of browser support as I found them too complicated for casual user. HTML5 specifications seemed like science documents to me. So I decided that the best method to test the browsers support was just run the HTML5-demos on each browser. This way I could directly see which browsers are supporting HTML5 and which are not.

For theoretical basis I chose history of HTML from 1989 to the present day because I don't consider it necessary to study the entire history of the Internet. HTML's history goes back to Tim Berners-Lee at CERN so I began my review of history from there. For theoretical basis I also added history of HTML5, its base functionality and the new features.

The target of research was to find out if HTML5 demos worked on selected browsers. At first I chose thirty HTML5 demos and then I randomly took five of them. Each of them was tested on five different browsers and the results were recorded in a table. Selected browsers were Google Chrome, Mozilla Firefox, Apple Safari, Opera and Windows Internet Explorer.

As a result of research Google Chrome and Mozilla Firefox were the best browsers because they manage perfectly of every test. The worst browsers were Windows Internet Explorer and surprisingly Opera as well. They managed to work perfectly only with three of the five tests, failed completely in one and in one test they worked almost successfully. Apple Safari was ranked in the middle as it worked well with four of the tests but failed in one.

The results of my thesis can be used when designing HTML5 web pages and applications because it's good to know which browsers will run your application. You can also start building application which will support all the browser but then you have to think again which features you can use from HTML5.

---

Keywords: HTML, HTML5, CSS, Browser

# SISÄLLYS

1	JOHDANTO .....	7
2	HTML .....	9
	2.1 Historia .....	9
	2.1.1 Ensiaskeleet .....	10
	2.1.2 Virallinen syntymävuosi .....	11
	2.1.3 W3C:n valtakausi alkaa .....	13
	2.1.4 HTML5:den syntyminen .....	14
	2.2 Tekniikat HTML:n rinnalla .....	15
	2.2.1 JavaScript .....	16
	2.2.2 CSS .....	17
	2.2.3 Flash .....	18
	2.2.4 Ajax .....	20
3	HTML5 .....	22
	3.1 Doctypen määrittäminen .....	22
	3.2 HTML5:den uudet ominaisuudet .....	23
	3.2.1 Canvas-elementti .....	24
	3.2.2 Video- ja äänitiedostot .....	25
	3.2.3 Paikkatieto eli geolokaatio .....	25
	3.2.4 Selaimen offline-käyttö .....	27
	3.2.5 CSS3 .....	27
4	SELAIN VERTAILU .....	28
	4.1 Testaus I - Cloth .....	31
	4.1.1 Tulos .....	31
	4.2 Testaus II – Ball Pool .....	33
	4.2.1 Tulos .....	33
	4.3 Testaus III – Canvas Photo .....	35
	4.3.1 Tulos .....	36
	4.4 Testaus IV – Hello Racer .....	38

4.4.1	Tulos.....	38
4.5	Testaus V – Chain Reaction .....	40
4.5.1	Tulos.....	40
4.6	Tulokset.....	42
5	POHDINTA .....	44

# 1 JOHDANTO

*"It could become a World Wild Web.*

*Let us all make it a World Wise Web"*

*-Jean-François Abramatic from W3C, 1997*

Tuntematon valkoinen mies voittaa kautta aikain ensimmäinen Millennium-tekniologiapalkinnon vuonna 2004. Media esittelee meidät ja mies paljastuu Tim Berners-Leeksi, miljoonan euron palkinto myönnetään hänelle WWW:n eli World Wide Webin kehittämisestä. Tuolloin oli ensimmäinen kerta, kun ajatuksen kanssa kuulin kyseisen herran nimen ja saavutukset. Muistan tuolloin ajatelleeni, että tuskin tuo yksi ihminen on yksinään luonut pohjaa nykyiselle internetille. Ajattelin internetin syntyneen tarkoituksella jonkun organisaation kehittämänä, ja Millennium-palkinto luovutettiin varmaankin tuon organisaation johtajalle. Ja kuinka väärässä olinkaan.

Voidaan sanoa, että jokainen nykyinen verkkosivu on kiitollisuuden velassa Tim Berners-Leelle ja hänen luomalle HTML-kielelle, sillä lähes kaikki nykyiset verkkosivut on tehty HTML:ää käyttäen. Vuosien saatossa HTML:n rinnalle on otettu käyttöön paljon muita tekniikoita, jotka mahdollistavat verkkosivun ulkoasun parantamisen ja interaktiivisuuden luomisen. Kaiken takana toimii kuitenkin vielä puhdas HTML-kieli, sillä automaattisesti koodin luovat kotisivukoneetkin käyttävät HTML:ää.

Opinnäytetyön aihe syntyi tarkoituksen mukaisesti, kun mietimme ohjaajani Teppo Räisäsen kanssa mahdollisia aiheita. Lähdin aluksi tutkimaan HTML5:den soveltuvuutta peliohjelmointiin, mutta tarkempi rajausta tiputti pelinäkökulman kokonaan pois. Tarkennuksen jälkeen opinnäytetyön aiheeksi muodostui HTML5:den ja sen toimivuuden tarkastelu selaimissa. Aihe muuttui omasta mielestäni paljon paremmaksi ja houkuttelevammaksi. Opinnäytetyön

toimeksiantajana on harjoittelupaikkanani minulle tutuksi tullut BelleGames ja yhteyshenkilönä toimii yrityksen toimitusjohtaja Pirjo Ritokangas-Huttunen.

HTML5:tä käsittelevän opinnäytetyön merkittävyys on suuri, sillä HTML5 on mullistanut verkkosivujen ja –sovellusten rakentamisen, ja mahdollistanut aivan uusia kehityssuuntia. Se on tämänkin opinnäytetyön kirjoitusprosessin aikana laajentunut ja kasvanut, jonka myötä yhä useammat yritykset ja mainostajat ovat löytäneet HTML5:den ja sen alati kasvavan potentiaalin. Tästä voidaankin mainita 9.11.2011 päivätty uutinen, jossa Flash-tekniikan luoja Adobe ilmoittaa lopettavansa Flashin kehittämisen mobiililaitteille ja siirtävänsä voimavarat näiltä osin HTML5:den kehittämiseen (Laakso 2011, hakupäivä 15.5.2012.)

Opinnäytetyöni toinen luku käsittelee HTML:n syntyhistoriaa aina vuodesta 1989 nykyhetkeen saakka, sekä muiden olennaisten HTML:n rinnalla toimivien verkkotekniikoiden historiaa ja toimintaperiaatteita. Kolmas luku käsittelee pelkästään HTML5:sta, sen toimintaperiaatteita ja uusia ominaisuuksia. Neljäs luku sisältää itse tutkimuksen, jossa olen satunnaisotannalla valinnut viisi HTML5-demoa, joita analysoin viidellä tärkeimmällä selaimella. Vertailu on toteutettu yksinkertaisesti taulukoimalla eri sivujen tai demojen toimivuus tai toimimattomuus. Internetissä on hyvin tarkkoja selainten tukimäärityksiä, mutta minun mielestäni ne eivät kerro peruskäyttäjälle mitään ollessaan aivan liian monimutkaisia. Siksi tässä tutkimuksessa vertailu toteutettiin taulukoimalla, eikä ohjelmointirivejä tutkimalla.



## 2 HTML

HTML tunnetaan kielenä, jolla lähes kaikki verkkosivut on rakennettu. Yleisestä käsityksestä poiketen HTML ei ole ohjelmointikieli laisinkaan, vaan virallisesti ilmaistuna se on kuvauskieli. HTML-kielen nimi tulee sanoista *Hypertext Markup Language*, eli hypertekstin merkintäkieli, mikä sekoittaa käsitteen selkeyttä entisestään. (W3Schools 2012b, hakupäivä 7.5.2012.)

Kaikessa yksinkertaisuudessaan HTML-kielellä kirjoitetaan tunnisteita (**tags**), jotka kuvaavat miten tunnisteen sisällä oleva teksti tulee esittää verkkosivulla. HTML:llä voidaan muokata tekstin rakennetta eli mikä osa tekstistä lihavoidaan, asetetaan otsikoksi tai jätetään leipätekstiksi. Tunnisteiden sisälle voidaan kuvauskielen nimen mukaisesti asettaa hyperlinkkejä sisältää tekstiä eli hypertekstiä. (Wikipedia 2012a, hakupäivä 7.5.2012.)

Hyperteksti kuulostaa monimutkaiselta, mutta yksinkertaisuudessaan se on vain tekstiä, johon voidaan lisätä linkkejä toisiin tekstisivuihin (W3C 2012, hakupäivä 7.5.2012). Sivujen vaihtamiseen ei siis tarvita uutta osoitetta vaan linkityksen kautta voidaan suoraan siirtyä toiselle sivulle. Tämä kuulostaa hyvinkin alkeelliselta, mutta ennen HTML:n kehittämistä vastaavasta ratkaisusta voitiin vain uneksia.

### 2.1 Historia

Sveitsissä sijaitsevaa Euroopan hiukkasfysiikan tutkimuskeskusta CERNia ei heti uskoisi HTML-kielen syntymäpaikaksi. Kyseisen organisaation sisällä on kuitenkin luotu pohja ja perusrakennusaineet nykyiselle internetille, vaikkei aihe liitykään mitenkään organisaation perimmäisiin tavoitteisiin. HTML luotiin tarpeeseen, mutta tuskin kukaan osasi kuvitella HTML:n kasvavan nykyisiin puitteisiinsa. (Raggett, Lam, Alexander & Kmiec 1998, hakupäivä 21.3.2012.)

CERNissa suoritettavat tutkimukset vaativat ennen ja vaativat vielä nykyisinkin laajaa yhteistyötä useiden eri instituutioiden kanssa ympäri maailmaa, mutta vuonna 1989 yhteistyön hyödyntäminen oli huomattavasti hankalampaa kuin nykyisin. Internettiä käytettiin CERNin ja muiden tiedeinstituutioiden ja -laitosten väliseen tiedonsiirtoon sekä dokumenttien lähettämiseen, mutta toiminta oli periaatteessa vain tekstitiedostoihin osoitettujen linkkien lähettämistä. Linkin kautta toinen osapuoli pystyi lataamaan tekstitiedoston omalle tietokoneelle tarkastelua varten. (Raggett ym. 1998, hakupäivä 21.3.2012.)

### **2.1.1 Ensiaskeleet**

Tim Berners-Lee työskenteli vuonna 1989 CERNin it-osastolla, kun hän aloitti uudenlaisen konseptin hahmotteleminen. Hänen ideanaan oli tarjota maailman kaikille tutkijoille mahdollisuus yhdistää tietonsa yhteen paikkaan, ja näin ollen myös organisoida tätä tietoa. Tieto organisoitaisiin niin, että luettuaan yhden dokumentin tutkija voisi suoraan siirtyä toiseen dokumenttiin, jossa olisi ensimmäiseen dokumenttiin liittyvää relevanttia tietoa. Näin ollen tiedosta muodostuisi eräänlainen verkko (**web**), joka pitäisi sähköisessä muodossa tiedon sisällään ja olisi saatavilla mistä tahansa tietokoneesta mistä tahansa päin maailmaa. (Raggett ym. 1998, hakupäivä 21.3.2012.)

Kyseisen konseptin erikseen hahmotteleminen tuntuu varmasti turhalta, mutta täytyy muistaa mitä aikaa tuolloin elettiin. Vuonna 1989 tietotekniikka oli vielä hyvinkin alkeellista, henkilökohtaisia tietokoneita oli harvalla ja tietokoneita käytettiin lähinnä tutkimus- ja kehitystyössä (COM271 2012, hakupäivä 21.3.2012). Raggetin (1998, hakupäivä 21.3.2012) mukaan muutamat verkkoviestinnän kehityssuunnat loivat tarpeen jollekin World Wide Webin kaltaiselle. Hyperteksti oli yleistymässä, internetin käyttäjämäärät olivat nopeassa kasvussa ja uusi internetin nimipalvelujärjestelmä DNS mahdollisti jokaisen tietokoneen helpon yksilöimisen internetissä (Raggett ym. 1998, hakupäivä 21.3.2012).

Tim Berners-Lee ja hänen ideansa osuivat siis juuri sopivaan murroskohtaan internetin historiassa. Kaiken lisäksi Berners-Lee oli ennen CERNiin tuloaan työskennellyt tekstinkäsittelyn ja dokumenttien tuottamisen parissa, sekä hän oli kehittänyt Enquire-nimisen hypertekstijärjestelmän henkilökohtaiseen käyttöönsä jo vuonna 1980 (Raggett ym. 1998, hakupäivä 21.3.2012.) Kaikkien palasten lokshtaessa yhteen Berners-Lee kirjoitti vuonna 1989 muistion, jossa hän ehdotti esimiehelleen internet-pohjaisen hypertekstijärjestelmän kehittämistä (Newbury 2012, hakupäivä 12.5.2012.)

Vuodesta 1989 tuli näin ollen HTML:n ja samalla WWW:n virallinen syntymävuosi, vaikkei kumpaakaan vielä edes nimetty. Vuoden 1990 aikana Tim Berners-Lee jatkoi konseptin kehitystä, ja sai avukseen Robert Cailliau -nimisen miehen CERNin seinien sisältä (CERN 2008, hakupäivä 13.5.2012). He saivat vuoden aikana valmiiksi HTML:n määrittelyn ja prototyypit selain- sekä serverisovelluksista (Wikipedia 2012a, hakupäivä 7.5.2012).

Ensimmäinen julkinen määrittely HTML:lle oli vuoden 1991 loppupuolella Tim Berners-Leen internetissä julkaisema dokumentti nimeltään HTML Tags, joka sisälsi määrittelmät kahdeksalletoista HTML-elementille (Wikipedia 2012a, hakupäivä 7.5.2012.) Berners-Leen kehittäjätaustan mukaisesti hän halusi keskustella avoimesti kenen tahansa asiasta kiinnostuneen kanssa, joten hän aloitti avoimen keskustelun internetissä sähköpostilistan avulla (Raggett ym. 1998, hakupäivä 21.3.2012.) Raggettin (1998, hakupäivä 21.3.2012) mukaan tämä niin sanottu elektroninen keskusteluryhmä herätti muutamien akateemikkojen ja kehittäjien mielenkiinnon Berners-Leen ja Cailliaun luomusta kohtaan, ja keskustelu jatkuikin aktiivisena julkaisusta lähtien.

### **2.1.2 Virallinen syntymävuosi**

Vuoden 1993 kesäkuussa internet-protokollien standardoinnista vastaava IETF-organisaatio julkaisi ensimmäinen virallinen spesifikaation HTML:lle, joka oli nimetty "Hypertext Markup Language (HTML)" -internetvedokseksi (Newbury

2012, hakupäivä 12.5.2012). Kontion (2001, 9) mielestä kyseinen julkaisu tarkoitti myös World Wide Webin ensijulkaisua, sillä rakentuuhan WWW:n hypertekstijärjestelmä HTML:stä. Saman vuoden toukokuussa CERN oli jo julkaissut World Wide Webin lähdekoodin julkisessa verkossa, jota kuka tahansa voisi hyödyntää (Francis 2008, hakupäivä 13.5.2012). Joka tapauksessa HTML:n ja WWW:n julkaisu mullisti tavan varastoida, esittää ja hakea tietoa internetissä (Kontio ym. 2001, 9).

HTML:n myötä internetin käyttö helpottui suuresti, sillä toisista koneista ei enää tarvinnut tietää muuta kuin osoite eli URL. Osoitettakaan ei välttämättä tarvittu, sillä sivuilta toisille pystyi helposti siirtymään linkkien kautta ja näin löytämään nopeasti lisää mielenkiintoisia sivustoja. Uusi tapa käyttää internetiä houkutteli nopeasti paljon uusia käyttäjiä, ja kannusti myös luomaan uutta sisältöä internetiin. Pelkästään vuoden 1993 aikana internetin koko kasvoi huimat 340 000%, ja kasvua vauhdittivat ensimmäiset graafiset selaimet Mosaic ja vuonna 1994 julkaistu Netscape (Kontio ym. 2001, 9.)

Korpela (2011,24) pitää vuosia 1990–1994 HTML:n varhaishistorian vuosina, sillä selaimet ja niiden kehittäjät määräsivät miten kieli kehittyi. Jokaiselle kehittäjällä oli omat ideansa HTML:stä ja ne lisättiin julkaistaviin selaimiin, ja siitä syntyikin lähes hallitsematon kaaos (Raggett ym. 1998, hakupäivä 21.3.2012). Varhaisesta HTML:stä ja sen monivaiheisesta alkutaipaleesta on totuttu käyttämään HTML 1 -nimitystä, vaikkei sitä koskaan virallisesti määriteltykään (Korpela 2011, 24).

Kaaoksen keskellä Tim Berners-Lee oivalsi, että taatakseen HTML:n yleismaallisen ja jatkuvan toimivuuden tarvittaisiin jokin riippumaton standardeja määrittelevä taho (Living Internet 2012, hakupäivä 13.5.2012). Berners-Lee perusti vuonna 1994 oivalluksensa pohjalta World Wide Web Consortiumin eli W3C:n, ja organisaation lähtökohtaiseksi visioksi muodostui WWW:n tekniikoiden ja käytäntöjen yhtenäistäminen (Francis 2008, hakupäivä 13.5.2012). W3C:llä meni kuitenkin muutama vuosi vakiinnuttaa oma asemansa

HTML:n määrittelijänä, joten seuraavana vuonna julkaistu HTML 2.0 tapahtui vielä IETF-organisaation toimesta (Newbury 2012, hakupäivä 12.5.2012).

### **2.1.3 W3C:n valtakausi alkaa**

HTML:n kehitys jatkui hyvinkin vauhdikkaana vuoteen 1997 asti, jolloin määriteltiin peräti kaksi uutta versiota HTML-kielestä (Korpela 2011, 24). Vuoden 1996 aikana HTML:n määrittelyvastuu oli siirretty virallisesti W3C:lle, ja ensimmäinen heidän virallinen HTML-määrittelynsä oli HTML 3.2 (Wikipedia 2012a, hakupäivä 7.5.2012). Tämä versio ei saanut koskaan suurta suosiota johtuen massiivisesta 150-sivuisesta määrittelydokumentistaan ja selainkehittäjien pettymyksestä kehityksen hitauteen (Newbury 2012, hakupäivä 12.5.2012).

Vuoden 1997 joulukuussa julkaistiin HTML 4.0, mutta se ei ehtinyt vanhentua edes kahta kuukautta kun W3C julkaisi toisen työryhmän aikaansaaman XML 1.0 -määrittelyn (Pilgrim 2012, hakupäivä 13.5.2012). Samoihin aikoihin niin kutsuttu "selainsota" oli päättynyt Microsoftin Internet Explorerin voittoon Netscapesta, ja IE kahmaisi selainmarkkinat lähes kokonaan (Fabritius 2010, hakupäivä 13.5.2012). Kilpailun kuollessa myös selainten kehitys pysähtyi lähes tyystin, ja HTML:n kehittämistä ei pidetty enää tarpeellisena (Fabritius 2010, hakupäivä 13.5.2012). Korpela (2011, 24) toteaaakin HTML:n kehityksen lähes pysähtyneen seuraaviksi neljäksi vuodeksi, jonka aikana julkaistiin vain muutamia vähäisiä päivityksiä ja lähes saman tien unohduksiin vaipunut ISO-HTML-standardi.

HTML 4 oli vuoteen 1998 mennessä monien mielestä lähes valmis, ja W3C päätti järjestää asian tiimoilta työpajan pohtimaan WWW:n tulevaisuutta (Mills 2011, hakupäivä 13.5.2012). Työpaja tuli enemmistön päätöksellä siihen tulokseen, että WWW:n tulevaisuus olisi XML:ssä ja XHTML:ssä (Mills 2011, hakupäivä 13.5.2012). XHTML:n tarkoituksena olisi luoda uusi rakenteellinen

pohja verkkosivustoille ja määritellä tarkemmin ohjelmoinnin sääntöjä (Fabritius 2010, hakupäivä 13.5.2012).

W3C päätti näin ollen hylätä HTML:n kehittämisen ja keskittyä tulevaisuudessa vain XML- ja XHTML-kielten kehittämiseen. HTML jäädettiin virallisesti versioon 4.01 ja W3C:n seuraava julkaisu oli HTML:n pohjalta rakennettu XHTML-spesifikaatio. Tämä oli siis XML-versio juuri haudatusta HTML-kielestä, ja se noudatti XML-kielen syntaksisääntöjä. (Lawson & Sharp 2011, xi.)

Hyvin pian W3C aloitti työskentelemisen XHTML 2.0 –nimisen spesifikaation parissa, ja sille asetettiin mahtipontisia tavoitteita. Uuden kuvauskielen tarkoitus oli aiheuttaa vallankumous olemalla loogisempi ja paremmin suunniteltu kuin mikään aikaisempi kuvauskieli. Tämä aiheuttaisi luonnollisesti sen, ettei mikään vanhemmilla kuvauskielillä rakennettu verkkosivu toimisi täydellisesti, vaan ne jouduttaisiin kääntämään uudelle kielelle toiminnan takaamiseksi. (Lawson & Sharp 2011, xi.)

Tämä WWW:n kehityssuunta ei miellyttänyt kaikkia, ja moni halusikin palata takaisin HTML:n pariin. Vuonna 2004 julkaistiin Firefox 1.0 -selain, joka oli vihdoinkin kilpailukykyinen vastustaja pitkään selainmarkkinoita hallinneelle Microsoftin IE:lle. Firefoxin tuomien raikkaiden uudistustuulten mukana heräsi myös HTML:n kipinä, kun Mozilla, Opera ja Apple perustivat Web Hypertext Application Technology Working Groupin, eli WHATWGN. (Fabritius 2010, hakupäivä 13.5.2012.)

#### **2.1.4 HTML5:den syntyminen**

WHATWG:llä oli lähes vastakohtainen näkemys World Wide Webin kehityssuunnasta kuin W3C:llä, joka halusi mullistaa verkkosivut XML-kielensä avulla (Fabritius 2010, hakupäivä 13.5.2012.) WHATWG halusi säilyttää vanhojen HTML-dokumenttien sopivuuden myös uusissa selaimissa, mutta tarjota samalla käyttäjille uusia tapoja kehittää verkkosivuja eteenpäin (Mills 2011, hakupäivä 13.5.2012.) Tavoitteena oli myös päästä eroon asennettavista

kaupallisista lisäosista, joilla oli jo vuosien ajan tuotu HTML-dokumentteihin toimivuutta ja interaktiivisuutta jota itse kuvauskieli ei kyennyt toteuttamaan (Fabritius 2010, hakupäivä 13.5.2012.) Fabritiuksen (2010, hakupäivä 13.5.2012) mukaan HTML:ään haluttiin vihdoinkin tuoda normaaleiksi verkkosivujen osiksi muodostuneita elementtejä, kuten videota ja ääntä, mutta ilman edellä mainittuja kaupallisia tuotteita kuten Flashia tai Silverlightia.

HTML:n kehitys aloitettiin siis uudestaan vuonna 2004 WHATWG:n toimesta, mutta W3C:llä meni vielä muutama vuosi ennen kuin he näkivät tässä kehityssuunnassa potentiaalia (Fabritius 2010, hakupäivä 13.5.2012.) Näiden muutaman vuoden aikana WHATWG ehti julkaista Web Applications 1.0 -nimisen määrittelyn, jossa dokumentoitiin selainten yhteisiä toimintaperiaatteita ja ominaisuuksia, sekä WWW:n uusia toimintasääntöjä (Mills 2011, hakupäivä 13.5.2012.) Vuonna 2007 W3C käynnisti uudelleen oman HTML-kehityksensä ottaen samalla suojiinsa WHATWG:n kehitysprojektin, ja yhtenä ensimmäisistä päätöksistä oli ottaa käyttöön WHATWG:n Web Applications 1.0 -määrittely ja alkaa kutsumaan uutta HTML-kehitystä HTML5:deksi (Mills 2011, hakupäivä 13.5.2012.)

## **2.2 Tekniikat HTML:n rinnalla**

Alun perin HTML-kuvaskieli kehitettiin ainoastaan tiedon esittämiseen, joten varsin pian kehittäjät aloittivat haaveilun interaktiivisista ja toiminnallisista verkkosivuista (Kontio ym. 2001, 180.) Interaktiivisuudella tarkoitetaan vuorovaikutteisuutta, eli kuinka HTML-dokumentti reagoi käyttäjän tekemiin valintoihin ja tarvittaessa muokkaa itseään käyttäjän haluamalla tavalla (Kontio ym. 2001, 180.) Nykyisiin HTML-dokumentteihin voidaan jo lisätä videokuvaa tai ääntä toiminnallisuuden saamiseksi, ja skriptikielillä puolestaan luodaan vuorovaikutteisuutta ja toiminnallisuutta verkkosivuille (Peltomäki 2000, 2.)

Tässä kappaleessa perehdytään tärkeimpiin tekniikoihin ja niiden luomis- sekä syntyhistorioihin. Jokaisesta tekniikasta kerrotaan myös toimintaperiaate, ja mihin tai mitä varten sitä käytetään HTML-kielen rinnalla.

### 2.2.1 JavaScript

JavaScript on tulkettava komentosarjakieli, eli komentosarjan tulkkauksen ja suorituksen suoritus suoritetaan samanaikaisesti jolloin ympäristön uudelleen lataamiselta vältytään. HTML-kielen yhteydessä tämä tarkoittaa sitä, ettei verkkosivua tarvitse ladata uudelleen jos jokin toiminto suoritetaan. JavaScript on varta vasten luotu tuomaan interaktiivisuutta HTML-dokumentteihin, sillä se tarjoaa kehittäjille ohjelmointityökalun selaimen sisällä. (W3Schools 2012a, hakupäivä 7.5.2012.)

Netscape julkaisi JavaScript-kielen ensimmäisen version Navigator 2 –selaimen ohessa vuonna 1995, ja se oli ensimmäinen laatuaan tarjoamalla vihdoinkin kehittäjille mahdollisuuden interaktiivisten verkkosivujen luomiseen. JavaScript vakiinnutti paikkansa hyvin nopeasti, koska sen avulla voitiin toteuttaa ohjelmia, jotka latautuivat nopeasti ja ne oli ne helppo upottaa HTML-dokumenttiin. Julkaisunsa jälkeen kieltä on kehitetty jatkuvasti, ja sen pohjalta on määritelty kaksi pääsuuntausta; Microsoftin kehittämä JScript ja yleisessä käytössä oleva EcmaScript. (Peltomäki 2000, 2–3)

Korpelan (2011, 47) mukaan JavaScriptistä on muodostunut ainoa käytettävä selainohjelmoinnin kieli, eli kieli, jolla ohjelmoidaan itse selainta suorittamaan haluttuja toimintoja. Se soveltuu parhaiten usein toistettaviin, lyhyisiin ja yksinkertaisiin ohjelmiin (Peltomäki 200, 6.) Ohessa esimerkki, jossa HTML-dokumentin sisällä on yksinkertainen JavaScript-funktiokutsu.

```
<html>
  <head>
    <title>JavaScript-esimerkki</title>
  </head>
  <body>
    <script>
```



```
        document.write("Hello World!");
    </script>
</body>
</html>
```

## 2.2.2 CSS

CSS eli Cascading Style Sheets, on erityisesti HTML-dokumenttien muotoiluun käytettävä tyyli tiedosto. Tyyli tiedosto määrittelee sen, miltä HTML-dokumentin eri elementit näyttävät selaimessa. CSS on erillinen tiedosto, mikä mahdollistaa helposti tyylin siirtämisen dokumentista toiseen ja vaivattoman tavan muokata kerralla kaikkia samaa tyyli tiedostoa käyttäviä verkkosivuja. (W3Schools 2012c, hakupäivä 14.5.2012.)

HTML-elementtien ominaisuuksia voidaan siis muokata CSS-tyyli tiedoston avulla, ja itse tyyli tiedosto koostuu elementtejä muokkaavista säännöistä. Esimerkiksi HTML:n *body*-elementille voidaan määrittää CSS-tiedostossa haluttu eriävä fontti, ja kun HTML-dokumentti käännetään verkkosivuksi, niin *body*-elementin teksti on erilaista kuin muu verkkosivun teksti. On myös mahdollista sisällyttää CSS-tyyli tiedosto HTML-dokumenttiin, mutta tämä ei ole kovin joustava tapa soveltuen tosin hyvin testaamiseen. Alla olevassa esimerkissä *body*-elementille määritetään fontti CSS-tyyli tiedoston ollessa sisällytettyinä. (Korpela 2011, 46.)

```
<!DOCTYPE html>
  <title>
    Testi
  </title>
  <style>
    body { font-family: Cambria }
  </style>
  <p>
    Hello World!
  </p>
```

CSS:n tarina alkaa vuodesta 1994, kun HTML-kieltä käytettiin yhä enemmän elektronisten julkaisujen alustana. Kielestä kuitenkin puuttui täysin kaikki ulkoasun muokkaamiseen tarvittavat elementit. HTML-kuvauskieltä ei alun alkaenkaan rakennettu niin, että sen avulla voitaisiin vaikuttaa verkkosivun ulkoasuun. Tim Berners-Lee oli kuitenkin kehitellyt tapaa jolla dokumentin rakenne ja ulkoasu erotettaisiin erillisillä tiedostoilla. Hän ei koskaan kuitenkaan julkaissut niitä, vaan halusi antaa selainten valmistajille vapaat kädet dokumenttien ulkoasujen muokkaamiseen. Selainvalmistajat eivät kuitenkaan lähteneet hyödyntämään tätä ja verkkosivut pysyivät hyvinkin samankaltaisina. (Lie & Bos 1999, hakupäivä 14.5.2012.)

Vuonna 1994 useat eri tahot työskentelivät tyylitiedostojen parissa, ja ensimmäiset luonnokset ja vedokset julkaistiin samana vuonna. Kehitys jatkui kiivaana, kunnes vuoden 1996 joulukuuhun mennessä CSS 1:stä oli jo tullut World Wide Web Consortiumin eli W3C:n määrittely. Ensimmäinen kaupallinen selain, joka tuki edes osittain CSS:n käyttöä, oli vuoden 1996 elokuussa julkaistu Microsoft's Internet Explorer 3 ja Netscape otti tuen mukaansa heti seuraavaan selainversioonsa. (Lie & Bos 1999, hakupäivä 14.5.2012.)

Vuonna 1998 julkaistu CSS 2 toi paljon uudistuksia, kuten fonttien varjostuksen ja elementtien tarkemman asemoimisen. Nykyisin käytössä oleva CSS-versio on julkaistu jo vuonna 1999, mutta CSS 3:sta kehitetään edelleen ja se onkin tärkeä osa useita HTML5:den tuomia uusia ominaisuuksia. W3C on aloittanut CSS 4:n kehityksen vuonna 2009 mutta mikään nykyinen selain ei vielä tue sitä. (Wikipedia 2012c, hakupäivä 15.5.2012.)

### **2.2.3 Flash**

Flash on kehitysympäristö, jolla voidaan tuottaa erilaisia multimediaesityksiä niin verkkosivuille kuin mobiililaitteisiinkin (Wikipedia 2012b, hakupäivä 15.5.2012.) Se voi sisältää videokuvaa, tekstiä, äänitiedostoja, kuvia tai pelkkiä komentoja, tai vaikka kaikkia näitä yhdessä (Paananen 2008, 7.) Flashin

tuotoksilla luodaan verkkosivuille toimivuutta ja interaktiivisuutta, sillä se taipuu mm. pelien ohjelmoimiseen ja animaatioiden luomiseen (Wikipedia 2012b, hakupäivä 15.5.2012.)

Flashin juuret ulottuvat aina vuoteen 1993, jolloin Jonathan Gay –niminen herrasmies päätti muutaman ystävänsä kanssa luoda piirto-ohjelman, jolla voitaisiin piirtää kynällä suoraan tietokoneen näytölle ja tämän tulisi olla helpompaa kuin perinteinen kynällä paperille –tyyli. Ohjelmisto sai nimekseen SmartSketc, ja se saikin paljon positiivista palautetta, mutta usein palautteessa kehoitettiin muuttamaan ohjelma animaatioiden luomiseen kykeneväksi. Gay ja hänen yhtiökumppaninsa olivat alkaneet kuulemaan ihmisten puhuvan internetistä ja WWW:stä, ja ne vaikuttivat lupaavilta markkinoilta. WWW:stä voisi kehittyä niin suosittu, että ihmiset haluaisivat luoda ja lähettää graafisia kuvia sekä animaatiota sen kautta toisilleen. Näin ollen vuonna 1995 SmartSketc-ohjelmaan alettiin tuomaan animaation luomistyökaluja. (Gay 2012, hakupäivä 15.5.2012.)

Itse Gayn (2012, hakupäivä 15.5.2012) sanojen mukaan SmartSketchin nimi vaihdettiin seuraavana vuonna FutureSplash Animatoriksi, koska animaatiosta oli kasvanut ohjelman tärkein osa. Ohjelma sai vuoden 1996 aikana tärkeää näkyvyyttä ja arvokasta mainetta muutaman suuren yrityksen, Microsoftin ja Disneyn, käyttäessä ohjelmaa, ja vuoden loppuun mennessä Gayn ja kumppaneiden yritys myytiin Macromedialle ja FutureSplash Animatorista tuli uuden yrityksen käsissä Macromedia Flash 1.0 (Gay 2012, hakupäivä 15.5.2012.) Macromedia myytiin Adobe Systemsille vuonna 2005, ja näin ollen myös Flash muokkaantui meidän kaikkien tuntemaan muotoonsa eli Adobe Flashiksi (Wikipedia 2012b, hakupäivä 15.5.2012.) Flashista onkin tullut lähes synonyymi animaatiolle ja sitä kehitetään edelleen (Gay 2012, hakupäivä 15.5.2012.)

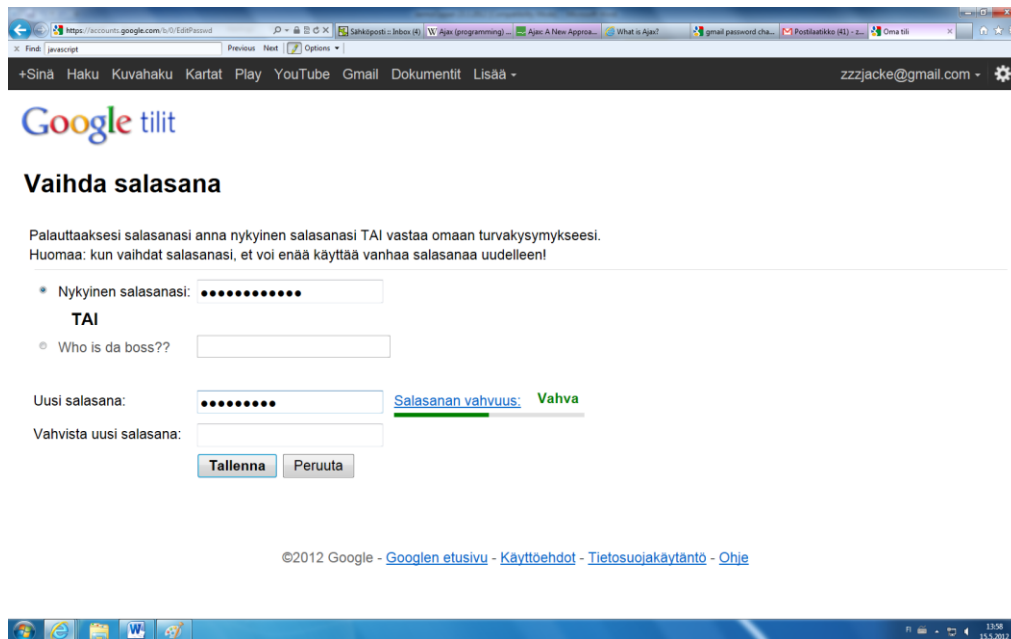
## 2.2.4 Ajax

Ajax koostuu useasta eri teknologista, jotka mahdollistavat selaimen ja palvelimen säännöllisen viestinnän ilman, että käyttäjä edes huomaa sitä. Ilman Ajaxia selain joutuu ottamaan yhteyden palvelimeen aina kun käyttäjä tekee jonkin muutoksen tai antaa komennon, ja tämän jälkeen lataamaan koko sivun uudelleen. Tämä on hyvinkin turhauttavaa käyttäjän näkökulmasta. Käytettäessä Ajaxia selain ja palvelin vaihtavat dataa taustaprosessin kaltaisesti, jolloin sivua ei tarvitse ladata aina muutoksia tehdessä. Tälle periaatteelle rakentuu Ajax ja sen avulla luodaan verkkosovelluksille interaktiivisuutta, käytettävyyttä ja nopeutta. (Garrett 2005, hakupäivä 14.5.2012.)

HTML5:llä ja Ajaxilla on ollut kummankin syntymästä saakka yksi yhteinen ideologia, eli pyrkimys on ollut luoda käyttäjille toimivia ja laajoja sovelluksia ilman erikseen asennettavia lisäosia tai asennuspaketteja (Asleson & Schutta 2007, 14). Teknologiat joita Ajax käyttää, ovat kaikki olleet olemassa pitkään ja löytyvätkin lähes kaikista nykyaikaisista selaimista, joten ylimääräisiltä asenteluilta usein vältytään (Asleson & Schutta 2007, 14). Ajax koostuu informaation esittämiseen ja muotoiluun käytettävistä XHTML:stä ja CSS:stä, interaktiivisuuden ja dynaamisuuden aikaansaamiseksi käytettävästä DOMista, datan siirtämiseen käytettävästä XML:stä, asynkronisen tiedonhaun toimimiseksi käytettävästä XMLHttpRequestista ja kaikki punotaan yhteen käyttämällä JavaScriptiä (Garrett 2005, hakupäivä 14.5.2012).

Äskeinen kappale kaivannee lisäselitystä. Ajax on lyhenne sanoista Asynchronous JavaScript and XML, eli asynkroninen JavaScript ja XML (Garrett 2005, hakupäivä 14.5.2012). Asynkroninen sarjaliikenne tarvitsee vain yhden signaalin toimiakseen kumpaankin suuntaan, mikä tässä tapauksessa tarkoittaa sitä, että juuri XMLHttpRequestin avulla päästään eroon verkkosivujen uudelleenlataamisesta (Kyrnin 2012, hakupäivä 15.5.2012). Google-tilien salasananatarkistus (kuviokuva 1.) toimii esimerkkinä, jossa verkkosivu tarkastaa

asynkronisesti salasanan sen kirjoittamisen aikana, ja ilmoittaa onko salasana tarpeeksi vahva ja pituudeltaan sopiva.



*KUVIO 1. Salasanatarkistus*

Ajaxin historia on nimensä puolesta lyhyt mutta toimintaperiaatteeltaan pitkä, sillä Ajaxin kaltaiseen ratkaisuun on pyritty jo pitkään. Yksi ensimmäisistä ratkaisuista oli Java-appletin käyttö, mutta se kärsi Javan hitaudesta ja huonosta yhteensopivuudesta selainten kanssa (Raymond 2006, 6).

## 3 HTML5

Ensinnäkin, HTML5 tarkoittaa paljon muutakin kuin HTML-kuvauskielen uusinta versiota. Siitä on tullut ikään kuin yleisilmaus kaikille verkkosivujen uusille ja tuleville ominaisuuksille, ja monet HTML5-nimille markkinoitavat ominaisuudet tarvitsevat paljon muutakin toimiakseen kuin itse kuvauskielen. En itsekkään lähde erittelemään tai arvostelemaan eri ominaisuuksien nimeämispolitiikkaa, vaan tässäkin opinnäytetyössä HTML5 tulee ymmärtää merkityssä "nykyinen HTML5-luonnos". Ja luonnoksen sijaan HTML5 on pikemminkin laaja kokoelma luonnoksia, mutta termi-HTML5 on vakiintunut käyttöön ja niin sitä käytetään tässäkin opinnäytetyössä (Korpela 2011, 25.)

Ensimmäinen vedos HTML5:stä julkaistiin vuoden 2008 tammikuussa ja sitä on hiljalleen otettu käyttöön. HTML5 on rakennettu tukemaan vanhentuneita elementtejä jotta siirtymävaihe versioiden välillä sujuisi mahdollisimman sulavasti. Tällä hetkellä uusien ominaisuuksien yleistyminen on täysin selaimien valmistajien varassa, eli miten selainvalmistajat päättävät sisällyttää uudet ominaisuudet selainten uusiin versioihin.

HTML5:den tuomien ominaisuuksien opiskelussa ja testaamisessa on mahdotonta olla täysin ajan tasalla, koska selainten tuet uusille ominaisuuksille muuttuvat koko ajan. Harvemmin tietenkään tukea joltain ominaisuudelta poistetaan, mutta selainpäivitys voi muuttaa ominaisuuden toimintaa joko yksin tai muiden ominaisuuksien ohessa.

### 3.1 Doctypen määrittely

Doctype on HTML-kielen kielioppimäärittely, eli se kertoo selaimelle mitä elementtejä ja elementtien attribuutteja HTML-dokumentissa on käytetty. Tämän avulla selain osaa näyttää HTML-dokumentin tekijän tarkoittamalla tavalla. HTML 4.0:ssa on peräti kolme eri kielioppimäärittelyä.

HTML 4.01 Strict on määrittelyistä tiukin, ja jättääkin paljon elementtejä ja attribuutteja pois määrittelystä. Kehykset eivät myöskään kuulu tähän, ja tyylitiedoston käyttäminen onkin kehoitettavaa (Kontio ym. 2001, 17)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
    "http://www.w3.org/TR/html4/strict.dtd">
```

HTML 4.01 Transitional –määrittelyssä on myös kaikki tiukan määrittelyn ominaisuudet, ja lisäksi Strict-määrittelystä pois jätetyt elementit sekä attribuutit. Kehykset eivät ole sallittuja tässäkään määrittelyssä. (W3Schools 2012d, hakupäivä 15.5.2012.)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01  
Transitional//EN"  
    "http://www.w3.org/TR/html4/loose.dtd">
```

HTML 4.01 Frameset on ominaisuuksiltaan laajin määrittely, sillä siinä on kaikki edellisissä määrittelyissä olevat piirteet sekä kehykset.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"  
    "http://www.w3.org/TR/html4/frameset.dtd">
```

HTML5:ssä on vain yksi tapa määrittää doctype, ja verrattuna aikaisempiin määrittelyihin HTML:den määrittelyn voi sentään muistaa ulkoa (Lawson & Sharp 2011, 2). Se voidaan myös kirjoittaa muotoon `<!doctype html>`, koska määrittelyssä isoilla ja pienillä kirjaimilla ei ole eroa (Korpela 2011, 28).

```
<!DOCTYPE html>
```

### 3.2 HTML5:den uudet ominaisuudet

Uusien ominaisuuksien yleistyminen on tällä hetkellä kiinni ainoastaan selainvalmistajista, sillä mitä pikemmin he ottavat uusien ominaisuuksien tuen mukaan versioihinsa, sitä pikemmin ne yleistyvät. On myös kehitetty erinäisiä

oikopolkuja vanhemmille selaimille jotta uudet elementit ja ominaisuudet toimisivat edes jotenkin niissä. Näissä ei tietenkään päästä täydelliseen toimivuuteen mutta jonkin asteinen tuki on parempi kuin ei tukea ollenkaan.

### 3.2.1 Canvas-elementti

Canvas-elementtiä pidetään yhtenä HTML5:den mielenkiintoisimmasta uudistuksista, sillä se tarjoaa mahdollisuuden käsitellä että luoda dynaamisesti grafiikkaa, eli voit canvas-pohjaisella piirto-ohjelmalla tehdä kuvan ja sen jälkeen tallentaa sen tietokoneellesi.

Verkkosivuilla kuvia ei voi suoraan muokata vaan ne ladataan aina jostain erillisestä hakemistosta, ja halutessasi muokata kuvaa sinun täytyy muokata kuvan alkuperäistä versiota ja sen jälkeen päivittää kuvan muutokset verkkosivulle. Canvas mahdollistaa kuvan käsittelyn suoraan selaimessa, ja sen muita sovellustapoja ovat mm. verkkosivuun upotettu piirtoalusta joka toimii samalla tavalla kuin mikä muu tahansa piirto-ohjelma.

Kuten useat muutkin HTML5:den tuomat uudet elementit, niin myös canvas, on helppo kutsua ja perustoiminnaltaan yksinkertainen. Pelkkä metodikutsu riittää tuomaan elementin sivulle ja oletuskokona on automaattisesti 300 pikseliä leveä ja 150 pikseliä korkea canvas. (Lawson & Sharp 2011, 118)

```
<canvas id="Canvas"></canvas>
```

Piirtäminen tapahtuu käyttämällä Canvas API –sovellusliittymää JavaScript-koodilla ja piirtäminen perustuu siihen, että vasen yläkulma on origo, eli piste (0,0).

Eri selaimien tuki canvas-elementille on varsin hyvä, suosituimmista selaimista ainoastaan Internet Explorerin muutamat vanhemmat versiot eivät suoraan tue



elementtiä tai sen piirto-ominaisuuksia. Canvas-elementin saa kuitenkin toimimaan kahdella eri tavalla, joko käyttämällä Silverlightin html5canvas-kirjastoa tai kääntämällä canvas-API:n Microsoftin VML:ksi. Kumpikaan ratkaisutavoista ei kuitenkaan tue täysin kaikkia canvas-elementin ominaisuuksia mutta yleisimmät piirto- ja käsittelymetodit onnistuvat. (Lawson & Sharp 2011, 118)

### **3.2.2 Video- ja äänitiedostot**

HTML5 lisää mediatiedostot HTML-dokumenttiin upottamalla ne siihen. Vanhemmissa HTML-versioissa median lisääminen on tapahtunut aina tavallisella linkillä, ja linkkiä aukaistaessa selain aukaisee, varsinkin videoiden tapauksessa, erillisen mediasoittimen. Upotettu mediatiedosto ilmestyy verkkosivulle omalle alueelleen ja on soittavissa suoraan siitä. Median hallintotyökalut ovat yleensä upotetun alueen alalaidassa, ja tähänkin voidaan vaikuttaa muuttamalla elementin määrittelyä. (Korpela 2011, 178.)

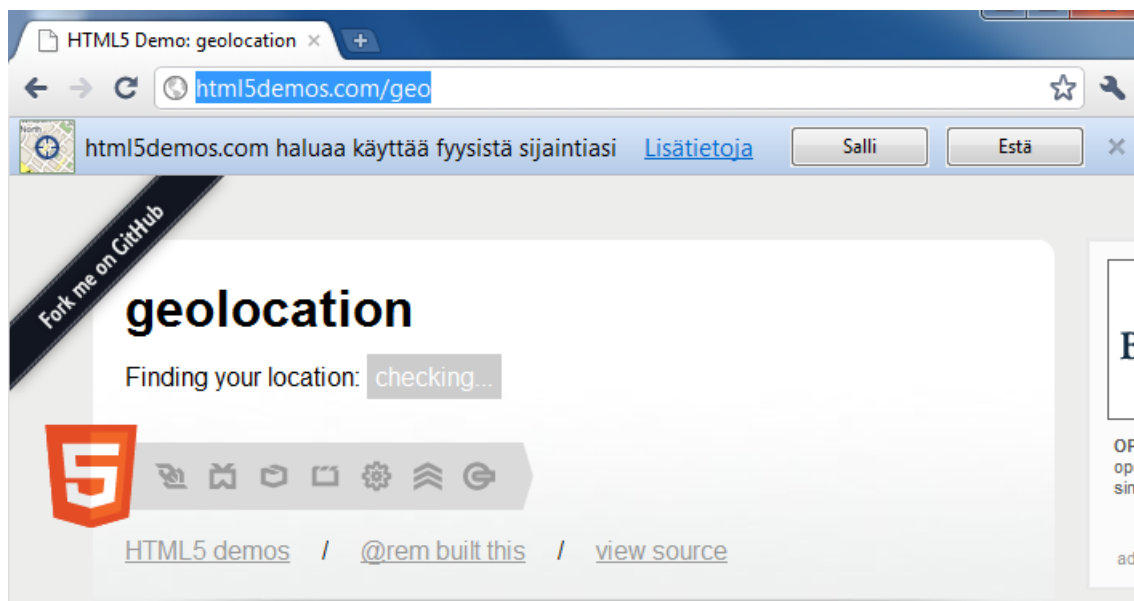
Upotettavan videon tiedostomuodosta on kiistely jo pitkään, sillä HTML5 ei määrittele tätä erikseen. Tällä hetkellä tasapainoillaan kolmen eri vaihtoehdon kanssa. H.264 on käytetyin ja sitä kannattavat ainakin Microsoft, Adobe ja Apple, mutta patenttisääntöjen takia avoimen lähdekoodin Firefox ei voi sitä käyttää. Google tarjoaa tähän ratkaisuksi ilmaiskäyttöön annetun WebM-nimistä videotekniikkaa, joka toimiikin jo ainakin Firefoxissa ja Googlen Chromessa. (Sani 2010, 32–35).

### **3.2.3 Paikkatieto eli geolokaatio**

Käyttäjän maantieteellisen sijainnin selvittämiseksi HTML5:ssä on geolocation API eli paikannus-ohjelmointirajapinta. Paikannustiedon selvittäminen antaa mahdollisuuksia eritoten tuotteiden markkinointiin ja mainostamiseen, sillä jos selain tietää asuinalueesi niin se voi kohdistaa lähialueen yritysten mainontaa juuri sinulle. Yksityisen henkilön näkökulmasta hyödyiksi nousevat taatusti

erilaisten reitti- tai ajo-ohjeiden saaminen. Mielenkiintoisia sovellusmahdollisuuksia on tietenkin paljon, mainitsemisen arvoisena sovellus joka paikallistaa verkkosivujesi käyttäjät kartalta. (Korpela 2011, 237–238.)

Paikannusta voitaisiin halutessa käyttää myös väärin tarkoituksiin, kuten selvittämään ihmisten kotiosoitteita tai seuraamaan ihmisten liikeitä. Kuten kuvio 2 huomataan, paikannussovellus kysyy käyttäjältä lupaa paikantaa käyttäjä. Lupa on sivustokohtainen, joten käyttäjä voi itse päättää erikseen jokaisen palvelun ja sivuston paikkatietojen käytöstä. (Korpela 2011, 238–239)



*KUVIO 2. Geolocation-demo (HTML5 demos 2012, hakupäivä 15.5.2012)*

Geolokaation selaintuki on varsin hyvä, sillä Firefoxin, Chromen, Safarin ja Operan uusimpien versioiden lisäksi useat eri mobiiliselaimet tukevat sitä (Lawson & Sharp 2011, 188.) Internet Exploreriin paikannustuki tuli vasta 9-version myötä, ja sitä vanhemmat versiot eivät tue ohjelmointirajapintaa lainkaan (Korpela 2011, 243.) Korpelan (2011,243) mukaan on kuitenkin olemassa Gears-niminen liitännäinen, jonka avulla selaimen saa tukemaan paikannuksen keskeisimpiä toimintoja.

### **3.2.4 Selaimen offline-käyttö**

Internet-yhteyksien ollessa nykyisiä paljon hitaampia ja huomattavasti kalliimpia voitiin vain toivoa mahdollisuudesta selata verkkosivuja ilman varsinaisen internet-yhteyden päällä pitämistä. Käyttömaksu perustui käytettyyn aikaan joten säästeliäimmät katkaisivat yhteyden sivuja selatessaan. (Korpela 2011, 268)

Verkkosivujen käyttäminen ja selaaminen on mahdollista jos itse sivut ja niiden käyttämät tiedostot ovat tallentuneet selaimen välimuistiin (Korpela 2011, 268.) HTML5:den myötä onkin tullut mahdolliseksi rakentaa sovellus, joka toimii täysin itsenäisesti offline-tilassa ja internet-yhteyden ollessa päällä tietoa voidaan siirtää verkkoon.

### **3.2.5 CSS3**

On muodostunut lähes tavaksi lukea aivan oma kokonaisuutensa CSS3 HTML5-käsitteen alle, mutta tässäkin noudatetaan juurtunutta tapaa luetella kaikki suhteellisen uusi HTML5:teen kuuluvaksi (Korpela 2011, 1, 285.)

Yksi mielenkiintoisimmista CSS3:n uusista ominaisuuksista on media-kyselyt, joiden avulla verkkosivu saadaan näyttämään päätelaitteelle sopivassa muodossa (Uskali 2011, hakupäivä 16.5.2012.) Lawsonin ja Sharpin (2011, 102) mukaan toimintaperiaate on kyselyiden lähettämistä selaimelle, jonka vastausten perusteella otetaan käyttöön esimerkiksi tietyn kokoiselle näytölle tarkoitettuja tiedostoja. Tämä tarkoittaa kaikessa hienoudessaan sitä, että samaa verkkosivustoa voidaan esittää sopivan näköisenä niin mobiililaitteessa kuin pöytäkoneessa ainoastaan CSS-tyylitiedostoa ja media-kyselyjä käyttämällä (Uskali 2011, hakupäivä 16.5.2012).

## 4 SELAIN VERTAILU

Tutkimuksen tarkoituksena on pääpiirteittäin tuoda ilmi eri selainten tuki HTML5:delle ja sen ominaisuuksille. Valitsin ensimmäiseksi kolmekymmentä erilaista näyttävää tai muuten HTML5:den uusia ominaisuuksia hyödyntävää demoa, joista sitten arvoitin internetistä löytämälläni numerogeneraattorilla viisi. Pyrin tällä siihen, etteivät omat mielipiteeni pääsisivät vaikuttamaan valittaviin demoihin taikka niiden saamaan vertailutulokseen.

Selaimista on valittu testattavaksi viisi eniten käytettyä: Google Chrome, Mozilla Firefox, Windows Internet Explorer, Apple Safari ja Opera. Ohessa lista selainten tarkoista versionumeroista, koska selainten tuki HTML5:delle voi muuttua uuden version tai pienenkin päivityksen myötä.



Firefox 12.0



Chrome 18.0.1025.168



Opera 11.64



Safari 5.1.7



Internet Explorer 9.0.8112.16421

Testattavat HTML5-demot ja HTML5:den uusia ominaisuuksia hyödyntävät sivut on pyritty valitsemaan useasta eri kategoriasta, ja niiden etsimiseksi käytin Googlen hakukonetta. Suoritin useita hakuja mm. "the best HTML5 sites", "HTML5 demos" ja "awesome HTML5 sites", mutta useissa kokoelmasivustoissa löytyi samoja demoja tai sivuja. Testasin kaikkien saatavuuden ennen valitsemista, sillä internetin luonteen mukaisesti osoitteet ja sovellukset saattavat vaihtaa paikkaa hyvinkin nopeasti. Tässäkin tapauksessa muutaman mielenkiintoisen demon osoitteet eivät johtaneet mihinkään.

Lisäsin omasta mielenkiinnosta joukkoon myös muutaman HTML5-versioinnin vanhoista tietokonepeleistä, ihan vaan väriä tuomaan. Loin demoista ja sivuista numeroidun listan arvonnin helpottamiseksi, ja ne ovat siihen otettu suhteellisen satunnaisessa järjestyksessä mikä parantaa edelleen testauksen luotettavuutta.

Tutkimuksessa suoritettiin viisi erillistä vertailua, ja jokaisessa vertailussa testattiin kunkin selaimen suoriutuminen sivun tai demon esittämisestä. Jokaisen vertailun alussa on ilmoitettu testattavan demon tai sivun tarkka verkko-osoite, kuvakaappaus testattavasta ja mahdollisimman tarkka kuvaus kuinka demon tai sivun tulisi toimia oikein. Näiden perusteella on arvioitu selaimen suoriutuminen demosta, ja lyhyesti kommentoitu mahdollisesta toimimattomuudesta ja mahdolliset syyt siihen. Jos selain suoriutui ilman ongelmia, niin kommentointiin on ilmoitettu myös se. Jokaisesta testauksesta on kirjoitettu lyhyt tulosten arviointi ja lopuksi kaikkien vertailujen tulokset tiivistetty yhteen erilliseen Tulokset-kappaleeseen.

Arvontaprosessi arpoi seuraavat numerot: 4 9 12 1 21.

**Käytetyt sivut:**

<http://devstand.com/design/impressive-3d-html5/>

<http://www.hongkiat.com/blog/48-excellent-html5-demos/>

<http://www.1stwebdesigner.com/freebies/html5-demos/>

<http://blog.templatemonster.com/2011/04/14/jquery-css3-photo-gallery-tutorials-html5-website/>

<http://www.nihilogic.dk/labs/chess/>

<http://www.adityaravishankar.com/projects/games/command-and-conquer/>

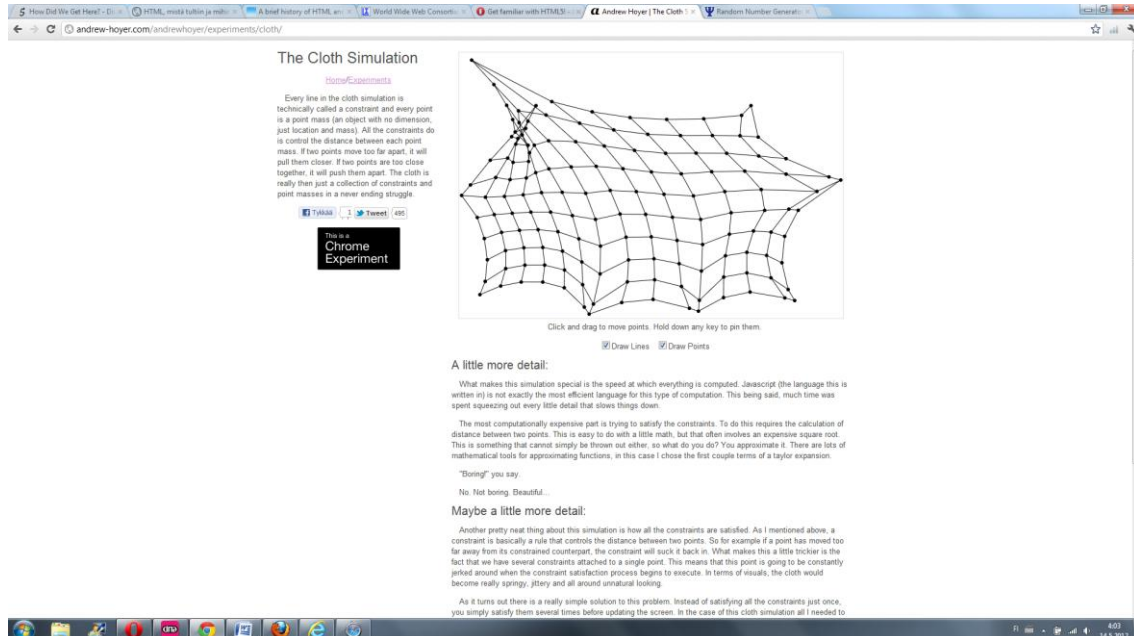
<http://wolfenstein.bethsoft.com/>

**Lista:**

1. Hello Racer
2. Video Destruction
3. 3D Canvas
4. The Cloth Simulation
5. Galactic
6. Angry Birds
7. Torus
8. Audioburst Animation
9. Ball Pool
10. Blob Sallad
11. Bomomo
12. Interactive Polaroid
13. JS Fireworks
14. Spread
15. Evolving Monster
16. CanvasMol
17. Cartoon Builder
18. PhysisSketch
19. Sketchpad
20. 3Bored
21. Chain Reaction
22. Jigsaw Puzzle
23. An Awesome CSS3 Lightbox Gallery With jQuery
24. Geolocation
25. Magnetic
26. 3D Chess
27. Sinuous
28. Rotating Image Slideshow
29. Wolfenstein 3D
30. Command & Conquer - Tiberian Dawn

## 4.1 Testaus I - Cloth

<http://andrew-hoyer.com/andrewhoyer/experiments/cloth/>



### KUVIO 4. Cloth-demo

Demo on nimensä mukaisesti kankaanpala, jota pystyy repimään ja riuhtomaan hiirellä haluamaansa suuntaan. Kangas reagoi hiiren liikkeisiin ja palautuu aina sulavasti takaisin lähtömuotoonsa. Se koostuu piirretyistä viivoista ja pisteistä, jotka voi halutessaan piilottaa tai tuoda näkyviin.


Jokaisen kankaan pisteen voi kiinnittää taustaan pitämällä pohjassa mitä tahansa näppäimistön näppäintä. Kuvakaappauksessa (kuvio 4) kangasta onkin pingotettu eri suuntiin.

#### 4.1.1 Tulos

Taulukosta 1 nähdään kaikkien selainten suoriutuneen samalla tavalla, ja demo toimi moitteettomasti. Ainoastaan itse demon ympärillä sijaitsevat tekstit ja

elementit asemoituivat hieman eri tavalla jokaisessa selaimessa. Tämä johtuu todennäköisesti vain siitä, että selaimet käyttäjät eri taittokomponenttia (**layout engine**), joten HTML-elementit näytetään hieman eri kohdissa.

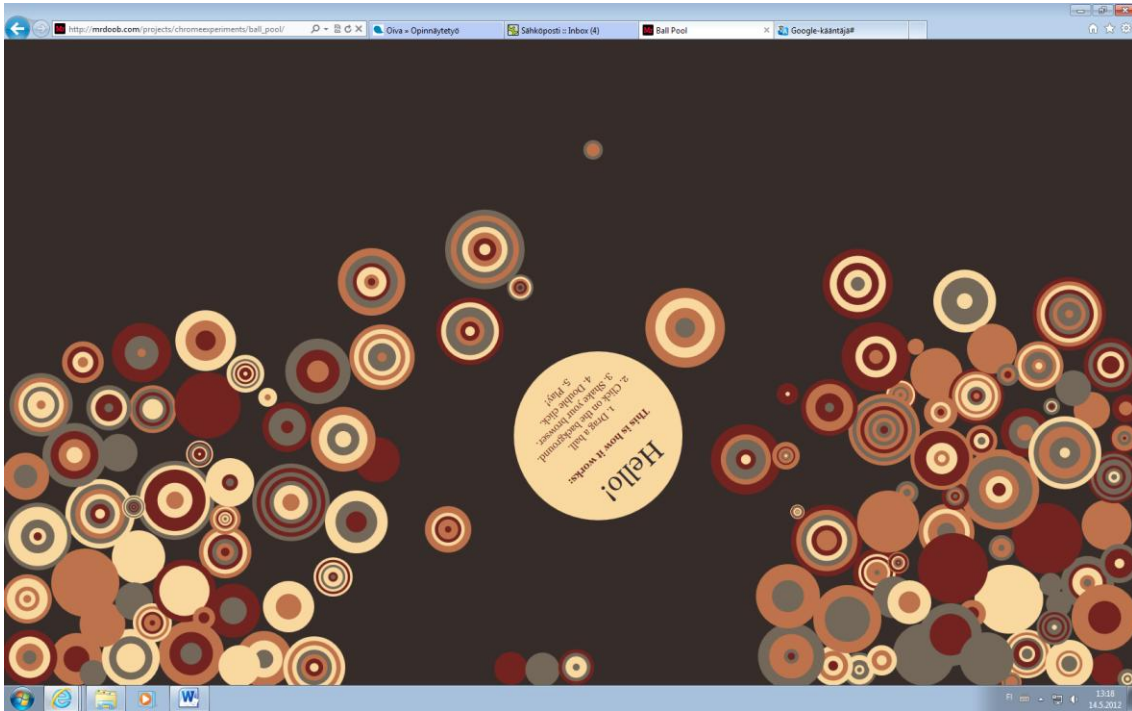
TAULUKKO 1. Cloth-testauksen tulos

Selain	Kommentit
	Toimi moitteettomasti, hyvin pieniä eroja elementtien asemoimisessa.
	Toimi moitteettomasti, hyvin pieniä eroja elementtien asemoimisessa.
	Toimi moitteettomasti, hyvin pieniä eroja elementtien asemoimisessa.
	Toimi moitteettomasti, hyvin pieniä eroja elementtien asemoimisessa.
	Toimi moitteettomasti, hyvin pieniä eroja elementtien asemoimisessa.



## 4.2 Testaus II – Ball Pool

[http://mrdoob.com/projects/chromeexperiments/ball\\_pool/](http://mrdoob.com/projects/chromeexperiments/ball_pool/)



*KUVIO 5. Ball Pool -demo*

Kuviossa 5 näkyvä Ball Pool –demo sisältää useita eri toimintoja. Voit tarttua mihin tahansa selaimessa näkyvään palloon ja liikutella tai heitellä sitä vapaasti ruudulla. Pallot reagoivat toisiinsa ja törmätessä keskenään pallot pomppivat ympäriinsä, ja klikkaamalla hiirellä taustaa luot lisää erikokoisia palloja mutta tuplaklikkaus aloittaa demon alusta pienin graafisin muutoksin. Voit myös ravistella selaintasi, jolloin kaikki ruudulla olevat pallot lähtevät liikkeelle.

### 4.2.1 Tulos

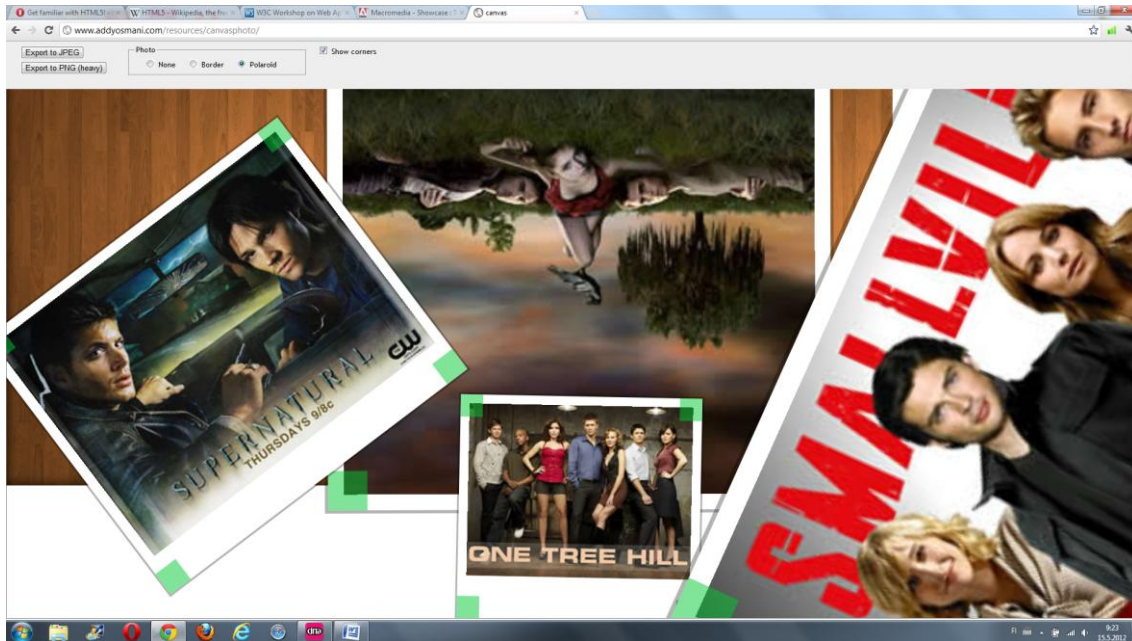
Selainten välillä ei juurikaan eroja suorituksessa, ainoastaan Opera ei osannut liikuttaa palloja selainta liikuttaessa. Muuten täysin identtinen toimivuus selaimessa kuin selaimessa (katso taulukko 2).

TAULUKKO 2. Ball Pool -testauksen tulos

Selain	Kommentit
	Toimi moitteettomasti.
	Toimi moitteettomasti.
	Pallot ilmestyivät toivotulla tavalla, mutta selainta heiluttaessa ne eivät kuitenkaan reagoineet mitenkään. Välillä pallot katosivat selaimen alalaitaan kun selaimen kokoa muutettiin.
	Toimi moitteettomasti.
	Toimi moitteettomasti.

### 4.3 Testaus III – Canvas Photo

<http://www.addyosmani.com/resources/canvasphoto/>



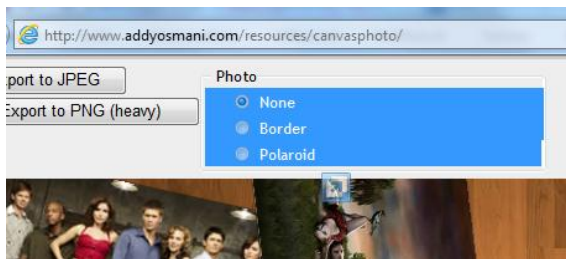
*KUVIO 6. Canvas Photo -demo*

Kolmas demo on canvas-elementin avulla rakennettu valokuvataulu (kuvio 6). Valokuvien asemointia ja kokoa voi vapaasti muokata valokuvan kuvasuhteen kuitenkaan muuttumatta. Valokuville voi halutessaan lisätä valikosta tasaiset valkoiset reunukset tai valkoiset kuvakaappauksessakin näkyvät polaroid-kehukset. Valittavana on myös kulmien näyttäminen, mikä oletettavasti on lisätty helpottamaan kuvien asemointia. Kuvakaappauksen vihreät neliöt ovat juurikin kyseisiä kulmia.

Demo tarjoaa myös mahdollisuuden ottaa luomuksen talteen JPEG- tai PNG-muodossa. Valintaa painamalla aukeaa selaimen uusi välilehti, johon tulee valokuvataulun mukainen kuva. PNG-valinnassa varoitetaan kyseisen toiminnan raskaudesta, mikä pikaisen testailun jälkeen kaatoikin välillä selaimen.





### 4.3.1 Tulos

Kolmannesta demosta selaimet suoriutuivat ulkoisesti varsin hyvin, tosin pientä taittokomponentista johtuvaa eroa asemoinnissa. Internet Explorerin valikko oli alla olevan kuvio 7:n mukaisesti poikkeavin asemoimalla valinnat pystysuunnassa muiden asemoidessa valikon vaakatasoon. IE ei myöskään osannut tuoda missään muodossa tuotosta pois demosta, mistä muut selaimet suoriutuivat moitteettomasti (ks. taulukko 3).



*KUVIO 7. Internet Explorerin virhe*

*TAULUKKO 3. Canvas Photo -testauksen tulos*

Selain	Kommentit
	Toimi moitteettomasti.
	Toimi moitteettomasti.
	Toimi moitteettomasti.
	Toimi moitteettomasti, mutta muista poiketen aukaisi tuodut kuvat uudessa selainikkunassa välilehden sijaan.



Toimi muuten moitteettomasti, mutta kuvien tuominen ei onnistunut laisinkaan ja valokuvan kehysten valinta-ikkunan asemointi oli muista poiketen listana.

## 4.4 Testaus IV – Hello Racer

<http://helloracer.com/webgl/>

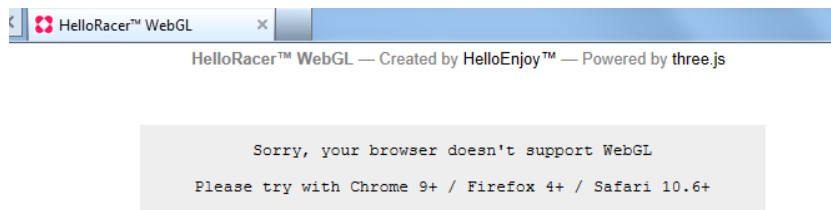


*KUVIO 8. Hello Racer -demo*

Demon ideaana on ohjata näppäimistöllä kolmiulotteista formula-autoa selaimessa (ks kuvio 8). Auton ohjaukseen käytetään joko nuoli- tai WASD-näppäimiä, käsijarru toimii välilyönnillä ja Enter-näppäimellä voi vaihtaa halutessaan kuvakulmaa.






### 4.4.1 Tulos

Selain joko tuki täysin demoa tai ei näyttänyt demoa lainkaan, ja viimeiseen kategoriaan pääsivät IE, Opera sekä Safari (ks taulukko 4). Jos selain ei tukenut demoa niin ruutuun tuli kuvio 9:ssä näkyvä ilmoitus WebGL-tuen puuttumisesta. Pikaisen tutkimuksen jälkeen vaikutti siltä, että Safarin uusimmat versiot eivät toimi lainkaan Windows-käyttöjärjestelmissä. Uusin Windowsille saatava Safari on tässäkin testauksessa käytetty 5.1.7.



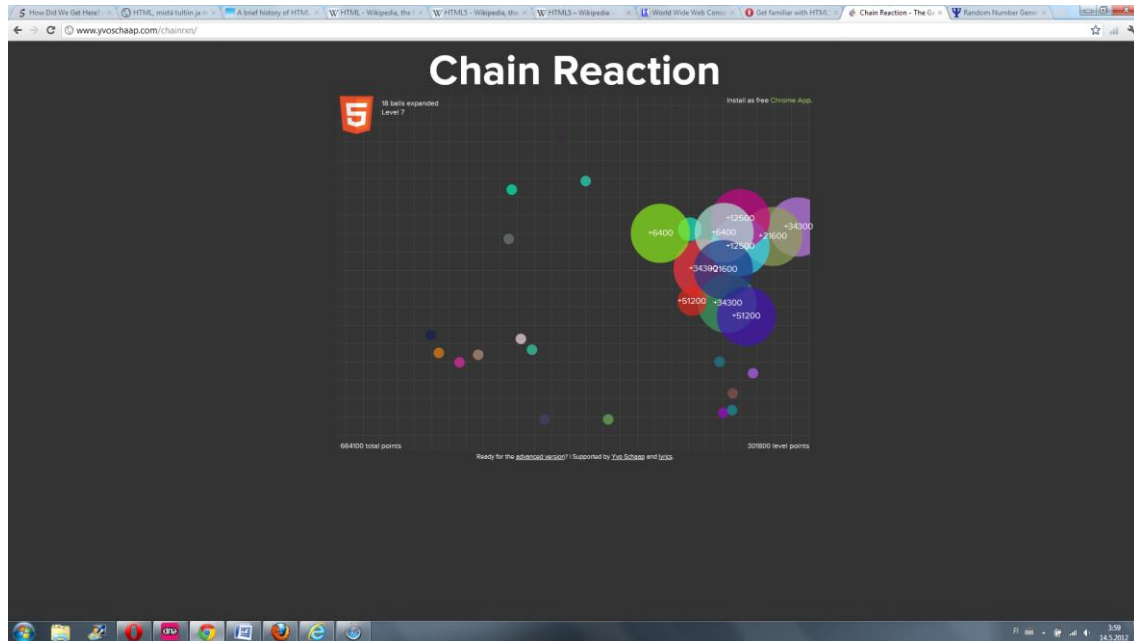
*KUVIO 9. WebGL-ilmoitus*

*TAULUKKO 4. Hell Racer -testauksen tulos*

Selain	Kommentit
	Toimi moitteettomasti.
	Toimi moitteettomasti.
	Ei toiminut lainkaan, ilmoitti ettei selain tue WebGL:ää.
	Ei toiminut lainkaan, ilmoitti ettei selain tue WebGL:ää.
	Ei toiminut lainkaan, ilmoitti ettei selain tue WebGL:ää.

## 4.5 Testaus V – Chain Reaction

<http://www.yvoschaap.com/chainrxn/>



### *KUVIO 10. Chain Reaction -demo*



Viimeisessä demossa on tarkoituksena luoda mahdollisimman suuri ketjureaktio pelissä poukkoilevista palloista. Pelissä laukaistaan hiiren avulla ensimmäinen laajeneminen, ja poukkoilevan pallon osuesa tähän niin se itsekin laajenee aiheuttaen mahdollisimman laajan ketjulaajenemisen kuvio 10:n mukaisesti. Vaatimustaso kasvaa taso tasolta ja samalla pallot lisääntyvät.

### 4.5.1 Tulos

Kaikki selaimet suoriutuivat tästä identtisesti, ei minkäänlaisia eroja havaittavissa (ks. taulukko 5).



TAULUKKO 5. Chain Reaction -testauksen tulos

Selain	Kommentit
	Toimi moitteettomasti.
	Toimi moitteettomasti.
	Toimi moitteettomasti.
	Toimi moitteettomasti.
	Toimi moitteettomasti.

## 4.6 Tulokset

Selainvertailun tuloksena Opera ja Internet Explorer saivat huonoimmat tulokset, koska ne eivät pystyneet näyttämään ollenkaan neljättä demoa sekä molemmilla oli teknisiä vaikeuksia yhdessä muussa demossa. Opera ei osannut suorittaa toista demoa täysin, sillä selainta liikuttaessa se ei reagoinut halutulla tavalla. IE puolestaan ei osannut tuoda kolmannen demon tuotoksia pois demosta. Kummatkin epäonnistuivat täysin neljännen demon näyttämisessä, mutta niin epäonnistui myös Safari.

Ball Pool –demo, eli toinen testaus, jäi mietityttämään Operan suoriutuessa huonosti, sillä muut selaimet suoriutuivat tästä testauksesta vaivatta. Tulokseen voi vaikuttaa henkilökohtaisen tietokoneen vika tai jokin huomaamatta jäänyt asetus, mutta se ei ole tässä tapauksessa relevanttia.

Tuloksissa itselleni oli yllättävintä Operan huono sijoitus, sillä olen pitänyt sitä aina edistyksellisenä selaimena. Näyttävästi se ei kuitenkaan ole kiirehtinyt HTML5:den ominaisuuksien tukemista versioinnissaan. Yllätyksenä ei tullut Internet Explorerin huono menestys, vaikka salaa toivoinkin etteivät ennakkoluuloni toteutuisikaan. Toisin kävi, ja IE:n maine huonoja HTML5-selaimena jatkaa kasvamistaan. Omasta mielestäni Microsoftin kaltainen suuri ohjelmistotalo olisi jopa velvollinen olemaan ensimmäisten joukossa uusia ominaisuuksia tuettassa, mutta eihän se näin mene.

Alla olevasta taulukko 6:sta näkee myös sen, että selaimista parhaiten suoriutuivat Firefox ja Chrome. Molemmat läpäisivät kaikki testaukset puhtaasi, ja olivat myös ainoat, jotka läpäisivät WebGL:ään tukeutuvan nelostestin. Parhaiten suoriutuneet selaimet eivät tulleet itselle yllätyksenä. Vuosien saatossa on huomannut kyseisten selainten olevan koko ajan kehityksen harjalla, ja usein nopeasti tuomassa päivityksiinsä uusia ominaisuuksia. Tämän varjopuolena on alati tulevat versiopäivitykset ja useiden päivitysten takia myös tietoturva-aukkoja syntyy tasaiseen tahtiin. Aukkojen paikkaamiseen tarvitaan

taas uusia päivityksiä, mikä lisää päivitystahtia entisestään. Tämä ei kuitenkaan ole itseä koskaan haitannut, koska selaimet osaavat suorittaa sen lähes itsenäisesti.

*TAULUKKO 6. Yhteenveto demo-testauksien tuloksista*

	I	II	III	IV	V
	✓	✓	✓	✓	✓
	✓	✓	✓	✓	✓
	✓		✓		✓
	✓	✓	✓		✓
	✓	✓			✓

## 5 POHDINTA

HTML5 ei aluksi kuulostanut vaikealta aiheelta, mutta tarkempi tarkastelu paljasti karun totuuden. Työn vaativuutta lisäsi oma osaamattomuus ja tietämättömyys HTML:stä, ja ohjelmointikokemuksen puute. Työn aiheen rajauksen avulla pääsin eroon lähes kaikesta ohjelmointiin liittyvästä, mutta pelkkä koodin tulkitseminen tuntui välillä hankalalta.

Aikataulu oli alusta asti ankara ja osoittautui varsin pian epärealistiseksi. Opinnäytetyö olisi ollut mahdollista tehdä paljon lyhyemmässä ajassa kuin nyt, mutta vastaan tuli oma huolettomuus ja laiskuus. Pääosa kirjoitustyöstä on tehty opinnäytetyöprosessin loppuvaiheessa, ja kirjoittaminen on tapahtunut rennoissa useita tunteja kestävässä kiireisissä pyrähdyksissä.

Opinnäytetyön aikana opin paljon lisää HTML-kielestä, ja en koskaan voinut kuvitellakaan kuinka monimutkainen syntymähistoria sillä on. Vaikka lähteitä historiasta oli paljon, niin se ei helpottanut kirjoittamisprosessia lainkaan. Jokainen lähde oli kirjoitettu eri näkökulmasta ja usein kävinkin niin, ettei lähteessä ollutkaan mitään uutta tietoa opinnäytetyöhön liittyen. Vuosilukujen kanssa oli myös ongelmia, sillä asioiden tapahtuma-ajankohdat riippuvat niin kirjoittajan näkökulmasta. Toinen laskee WWW:n syntyneen Tim Berners-Leen vuonna 1989 kirjoittamasta muistiosta, kun toinen laskee saman asian syntyneen ensimmäisen virallisen määrittelyn myötä vuonna 1993.

Yleiskäsitykseni HTML:stä ja sen kanssa toimivista teknologioista on ainakin kehittynyt huimasti, ehkä jopa pelottavan huimasti. Työtä läpikädessäni huomasin aina kun olin oppinut jonkin uuden sanan merkityksen, sillä asiayhteydet saattoivat poukkoilla kappaleenkin sisällä. Työn jälkeen tuntuu kuin pitäisi kirjoittaa koko työ alusta. Tällä kertaa ainakin olisi pohjatietoa, eikä kaikkea tarvitsisi katsoa Wikipediasta.

Selainvertailun tulokset olivat osittain yllättäviä. Pidin Operaa vartenotettavana vastustajana omille suosikeilleni Chromelle ja Firefoxille, mutta tulosten perusteella Opera jäikin kuitenkin Internet Explorerin kanssa viimeiselle sijalle. Olin lukenut aikaisemmin jotain WebGL:stä ja siihen liittyvistä selainongelmista, joten siltä osin selainten toimimattomuus ei yllättänyt tuloksissa. Kummankin selaimet puutteet tuntuivat varsin mitättömiltä, sillä muista testidemoissa oli lähes vastaavia ominaisuuksia ja niistä ne suoriutuivat helposti. Ehkä tässä tullaan siihen, että HTML5 on vielä luonnosasteella ja selainkehittäjät saavat itse päättää kuinka nopeasti ottavat uusia ominaisuuksia mukaan selaimiinsa.

Parhaiten suoriutuivat siis Googlen Chrome ja Mozillan Firefox, sillä ne suoriutuivat kaikista demoista ilman mitään ongelmia. Mitä olen opinnäytetyön ohessa käyttänyt Chromea erilaisten HTML5-demojen tutkimiseen, niin en ole vielä törmännyt vielä tilanteeseen jossa Chrome ei olisi pystynyt näyttämään tai suorittamaan jotain demoa tai sivustoa. Voinkin näin ollen sanoa, että jos haluaa testata HTML5:den kyvykkyyttä esimerkiksi Flashin kaatajana, niin kannattaa valita selaimeksi Chrome tai Firefox.

Opinnäytetyöni aihetta olisi mielenkiintoista jatkaa pidemmälle, ja luoda erillisiä omia HTML5-sivustoja uusien ominaisuuksien testaamiseksi. Tätä vaatisi jo laajempaa kokemusta ohjelmoinnista ja rutkasti enemmän aikaa kuin työhön on varattu, mutta eihän kukaan estä tekemästä tuota omalla vapaa-ajalla.

## LÄHTEET

Asleson, R. & Schutta, N. 2007. Ajax – Tehokas hallinta. Suom. M. Kamppila. Jyväskylä: Gummerus Kirjapaino Oy.

CERN 2008. How the web began. Hakupäivä 13.5.2012, <http://public.web.cern.ch/public/en/about/WebStory-en.html>.

COM271 2012. HTML for Structure. Hakupäivä 12.5.2012, [http://www.com.uri.edu/com271/course/notes/1\\_intro\\_to\\_html.htm](http://www.com.uri.edu/com271/course/notes/1_intro_to_html.htm).

Fabritius, J. 2010. HTML, mistä tultiin ja mihin ollaan menossa? Hakupäivä 13.5.2012, [gfx.fi/2010/03/html-mista-tultiin-ja-mihin-ollaan-menossa/](http://gfx.fi/2010/03/html-mista-tultiin-ja-mihin-ollaan-menossa/).

Francis, M. 2008. 2: The history of the Internet and the web, and evolution of web standards. Hakupäivä 13.5.2012, <http://dev.opera.com/articles/view/2-the-history-of-the-internet-and-the-w/>

Garrett, J. 2005. Ajax: A New Approach to Web Applications. Hakupäivä 14.5.2012, <http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications>.

Gay, J. 2012. The History of Flash. Hakupäivä 15.5.2012, [http://www.adobe.com/macromedia/events/john\\_gay/page04.html](http://www.adobe.com/macromedia/events/john_gay/page04.html).

Hawkes, R. 20xx. Foundation HTML5 Canvas: For Games and Entertainment.

HTML5 demos 2012. Geolocation. Hakupäivä 15.5.2012, <http://html5demos.com/geo>.

Kontio, M., Vierimaa, K. & Niskanen P. 2001. WWW-ohjelmointi: Trainer Kit. Jyväskylä: Gummerus Kirjapaino Oy.

Korpela, J. 2011. HTML5: uudet ominaisuudet. Porvoo: Bookwell Oy.

Kyrnin, J. 2012. What is Ajax?: Building Web Applications Just Got More Fun. Hakupäivä 15.5.2012, <http://webdesign.about.com/od/ajax/a/aa101705.htm>.

Laakso, H. 2011. Adobe lopettaa Flashin mobiilista - keskittyy html5:een. Hakupäivä 15.5.2012, [www.tietoviikko.fi/kaikki\\_uutiset/adobe+lopettaa+flashin+mobiilista++keskittyy+html5een/a719758](http://www.tietoviikko.fi/kaikki_uutiset/adobe+lopettaa+flashin+mobiilista++keskittyy+html5een/a719758).

Lawson, B. & Sharp, R. 2011. Introducing HTML5. Berkeley: New Riders.

Lie, H. & Bos, B. 1999. Chapter 20: The CSS saga. Hakupäivä 14.5.2012, <http://www.w3.org/Style/LieBos2e/history/>.

Living Internet 2012. World Wide Web Consortium (W3C). Hakupäivä 13.5.2012, [www.livinginternet.com/w/wi\\_w3c.htm](http://www.livinginternet.com/w/wi_w3c.htm).

Mills, C. 2011. Get familiar with HTML5! Hakupäivä 13.5.2012, [dev.opera.com/articles/view/get-familiar-with-html5/](http://dev.opera.com/articles/view/get-familiar-with-html5/).

Newbury, S. 2012. A brief history of HTML and the World Wide Web. Hakupäivä 12.5.2012, <http://oirsworld.com/2012/04/a-brief-history-of-html-and-the-world-wide-web/>.

Paananen, P. 2008. Flash-julkaisijan opas. Jyväskylä: WSOY.

Peltomäki, J. 2000. JavaScript. Jyväskylä: Docendo Finland Oy.

Pilgrim, M. 2012. Dive into HTML5. Hakupäivä 13.5.2012, <http://diveintohtml5.info/>.

Raggett, D., Lam, J., Alexander, I. & Kmiec, M. 1998. Raggett on HTML 4. Hakupäivä 21.3.2012, [Http://www.w3.org/People/Raggett/book4/ch02.html](http://www.w3.org/People/Raggett/book4/ch02.html).

Raymond, S. 2006. Ajax on Rails. California: O'Reilly Media.

Sani, I. 2010. HTML5:stä rakentuu webin tulevaisuus. MikroPC (8) ,32–35.

Uskali, T. 2011. HTML5 - identtinen käyttökokemus kaikilla selaimilla on unohdettava! Hakupäivä 16.5.2012, [www.meteoriitti.com/fi-FI/tiedotteet/ajankohtaista/html5-identtinen-kayttokokemus-kaikilla-selaimilla-on-unohdettava](http://www.meteoriitti.com/fi-FI/tiedotteet/ajankohtaista/html5-identtinen-kayttokokemus-kaikilla-selaimilla-on-unohdettava).

W3C 2012. What is HyperText. Hakupäivä 7.5.2012, <http://www.w3.org/WhatIs.html>.

W3Schools 2012a. JavaScript Introduction. Hakupäivä 7.5.2012, [http://www.w3schools.com/js/js\\_intro.asp](http://www.w3schools.com/js/js_intro.asp).

W3Schools 2012b. HTML Introduction. Hakupäivä 7.5.2012, [http://www.w3schools.com/html/html\\_intro.asp](http://www.w3schools.com/html/html_intro.asp).

W3Schools 2012c. CSS Introduction. Hakupäivä 14.5.2012, [http://www.w3schools.com/css/css\\_intro.asp](http://www.w3schools.com/css/css_intro.asp).

W3Schools 2012d. HTML <!DOCTYPE> declaration. Hakupäivä 15.5.2012, [http://www.w3schools.com/tags/tag\\_doctype.asp](http://www.w3schools.com/tags/tag_doctype.asp).

Wikipedia 2012a. HTML. Hakupäivä 7.5.2012, <http://fi.wikipedia.org/wiki/HTML>.

Wikipedia 2012b. Adobe Flash. Hakupäivä 15.5.2012, [http://en.wikipedia.org/wiki/Adobe\\_Flash](http://en.wikipedia.org/wiki/Adobe_Flash).

Wikipedia 2012c. Cascading Style Sheets. Hakupäivä 15.5.2012, [http://en.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](http://en.wikipedia.org/wiki/Cascading_Style_Sheets).

Wikipedia 2012d. Ajax (programming). Hakupäivä 15.5.2012, [http://en.wikipedia.org/wiki/Ajax\\_\(programming\)](http://en.wikipedia.org/wiki/Ajax_(programming)).