

## Yrityksen sivuston uudistaminen HTML5-tekniikoin

Veli-Matti Lehikoinen

Opinnäytetyö

Tietojenkäsittelyn koulutusohjelma

2012



Tietojenkäsittelyn koulutusohjelma

<b>Tekijä tai tekijät</b> Veli-Matti Lehikoinen	<b>Ryhmätunnus tai aloitusvuosi</b> 2008
<b>Raportin nimi</b> Yrityksen sivuston uudistaminen HTML5-tekniikoin	<b>Sivu- ja liitesivumäärä</b> 38 + 8
<b>Opettajat tai ohjaajat</b> Niina Kinnunen	
<p>HTML5:ssä on tapahtunut merkittäviä edistysaskelaite viime vuosina. Tämä tekee HTML5:stä kiinnostavan aiheen sivuston kehittäjän kannalta. Opinnäytetyön tarkoituksena on tutkia ja hyödyntää uusia HTML5-tekniikoita yrityksen sivuston toteuttamisessa. Opinnäytetyön aihe tuli BPM Helsinki nimisen yrityksen tarpeesta uudistaa internet-sivustonsa. Ajatus uusien HTML5-tekniikoiden hyödyntämisestä lähti opinnäytetyön tekijän kiinnostuksesta web-tekniikoita kohtaan.</p> <p>Teoriaosuuden tavoitteena on esitellä kattavasti HTML5:n merkitys nykyisin, unohtamatta sen kehityshistoriaa, sekä tuoda esille sivuston kehittäjän kannalta olennaisia osa-alueita. Empiriassa keskitytään sivuston kannalta tärkeimpien asioiden, kuten sivurakenteen uudistamiseen, sekä uuden HTML5-video tekniikan sisällyttämiseen sivustolle.</p> <p>Opinnäytetyö on rajattu sivuston kannalta tärkeimpiin uudistuksiin. Opinnäytetyön kokonaistavoitteena on rakentaa yrityksen kanssa neuvoteltu sivusto määräajassa, sekä lisätä opinnäytetyön tekijän ymmärrystä HTML5:n merkityksestä nyt ja tulevaisuudessa. Opinnäytetyön keskittyessä HTML5:n tuomiin uusiin tekniikoihin on lukijan hyvä omata perustiedot HTML:n tekniikoista ja ohjelmoinnista.</p> <p>Sivustoprojekti toteutetaan hyödyntäen NetBeans-kehitysympäristöä, sekä erillistä kotisivua. Sivustoprojekti toteutetaan yhdessä yrityksen kanssa syksyn 2012 aikana.</p> <p>Sivusto valmistui sovitussa aikataulussa. Yritys sekä opinnäytetyön tekijä olivat tyytyväisiä saavutettuun lopputulokseen. Lisäksi opinnäytetyön tekijä sai hyvän kokonaiskäsityksen HTML5:n monimuotoisuudesta ja sen mahdollisuuksista tulevaisuudessa.</p>	
<b>Asiasanat</b> HTML, HTML5, CSS, DOM, JavaScript	

Degree programme in Information Technology

<p><b>Authors</b> Veli-Matti Lehtikoinen</p>	<p><b>Group or year of entry</b> 2008</p>
<p><b>The title of thesis</b> Reforming company's website using HTML5 technologies</p>	<p><b>Number of pages and appendices</b> 38 + 8</p>
<p><b>Supervisor(s)</b> Niina Kinnunen</p>	
<p>HTML5 has made significant advances in recent years. This makes HTML5 a topic of interest from the site developer's point of view. The purpose of this study was to explore and take advantage of new HTML5 technologies for the implementation of the company's site. The subject of the thesis came up when BPM Helsinki company expressed a need to reform its website. The idea of using new HTML5 technologies began from the author's interest in web technologies.</p> <p>The aim of the theory is to present a comprehensive role of HTML5 today, not to mention the history of its development as well as to bring out relevant areas from the developer's perspective. The empirical part focuses on the most important issues regarding the reform of the website, such as the restructuring of the page as well as implementing new HTML5-video technology into the site.</p> <p>The thesis is limited to the most important reforms of the site. The overall objective of the thesis is to build the company site within the agreed time, and to increase the author's understanding of HTML5's role now and in the future. The thesis focuses on HTML5 new technologies and the reader should have a basic knowledge of HTML and programming techniques.</p> <p>The site project is implemented using the NetBeans development environment, as well as a separate home page. The site project will be implemented together with the company in autumn 2012.</p> <p>The site was completed on schedule. The company and the author of the thesis were satisfied with the final outcome. Additionally, the author got a good overall understanding of HTML5's diversity and its potential for the future.</p>	
<p><b>Key words</b> HTML, HTML5, CSS, DOM, JavaScript</p>	

# Sisällys

1	Johdanto .....	1
2	HTML5.....	3
2.1	Kehitystyö.....	3
2.2	Nykytila .....	5
2.3	DOM, HTML, JavaScript ja CSS .....	7
2.4	Rakennemuutokset.....	9
2.5	Selaintuki ja eroihin varautuminen .....	11
2.6	Uudet toiminnot.....	13
2.6.1	Semanttiset elementit .....	13
2.6.2	Video median upottaminen HTML- sivuun .....	16
3	BPM Helsinki Oy:n sivustoprojekti.....	20
3.1	Lähtökohta ja projektisuunnitelma .....	20
3.2	Nykyinen sivustorakenne ja uuden rakenteen pohjustus .....	21
3.3	Luonnos uudesta sivustosta .....	23
3.4	Video elementin lisääminen sivustolle.....	27
4	Yhteenveto ja johtopäätökset .....	33
	Liitteet.....	39
	Liite 1. Projektisuunnitelma .....	39
	Liite 2. Termit ja lyhenteet .....	42

# 1 Johdanto

HTML5:n kehitystyö aloitettiin vuonna 2004. HTML5:ssä on tapahtunut merkittäviä edistysaskeleita, erityisesti eri selainten tuen osalta. Tämän lisäksi HTML5 yhtenäistää eri web-tekniikoita, kuten HTML, CSS, DOM ja JavaScript. Tämä tekeekin HTML5:stä yhä kiinnostavamman kohteen sivuston kehittäjän kannalta.

Tämän opinnäytetyön tarkoituksena on tutkia ja hyödyntää uusia HTML5-tekniikoita yrityksen sivuston toteuttamisessa. Opinnäytetyön aihe tuli BPM Helsinki-nimisen yrityksen tarpeesta uudistaa internet-sivustonsa.

Ajatus uusien HTML5-tekniikoiden hyödyntämisestä lähti kiinnostuksestani web-tekniikoita kohtaan. Olin jo aiemmin kuullut puhuttavan HTML5:stä, sekä lukenut siihen liittyvistä tekniikoista W3Schools-sivustolta. En ollut kuitenkaan koskaan kokeillut tekniikoita käytännössä.

Yrityksen nykyinen sivusto on tehty hyödyntäen aiempia HTML-tekniikoita, ja se koostuu kolmesta erillisestä word-dokumentista, jotka on tallennettu Web Page-muotoon. Yrityksen kanssa käytyjen neuvottelujen, sekä omien tutkimusteni perusteella ehdotin sivustolle käytettäväksi uusia tekniikoita, joiden katsoin tuovan uutuuden lisäksi sopivaa lisäarvoa sivustolle. Sivuston lopullinen valmistuminen ja visiointi tulee olemaan pidempiaikaisempi projekti.

Teoriaosuuden tavoitteena on esitellä kattavasti HTML5:n merkitys nykyisin, unohtamatta sen kehityshistoriaa, sekä tuoda esille sivuston kehittäjän kannalta olennaisia osa-alueita. Teoria osuudessa käsitellyt uudet toiminnot on rajattu yrityksen sivuston kannalta olennaisiin. Empirian tavoitteena on luoda uusittu pohja yrityksen sivustolle hyödyntäen uusia HTML5-tekniikoita. Empiriassa keskityn sivuston kannalta tärkeimpiin asioihin, kuten uusitun sivurakenteen uudistamiseen, sekä video-tekniikan sisällyttämiseen sivustolle. Uudistukset luovat samalla hyvän pohjan sivuston tuleville uudistuksille. Opinnäytetyön kokonaistavoitteet ovat, paitsi rakentaa yrityksen kanssa neuvoteltu sivusto määräajassa, myös lisätä opinnäytetyön tekijän ymmärrystä HTML5:n merkityksestä nyt ja tulevaisuudessa.

Sivustoprojekti toteutetaan sivuston tekijän omalla tietokoneella NetBeans-kehitysympäristössä. Lisäksi sivuston tekijällä on käytävissä kotisivu, jota hyödynnetään uuden sivuston kehittämisessä.

Opinnäytetyössä keskitytään HTML5:n tuomiin uusiin tekniikoihin. Ymmärtääkseen lukijan olisi hyvä omata perustiedot HTML:n tekniikoista ja ohjelmoinnista. Omat aiemmat tietotaidot sivustojen toteuttamisesta ovat perusopintojen kautta toteutettuja projekteja, joissa on hyödynnetty vanhempia käytössä olleita HTML- tekniikoita.

## 2 HTML5

Tässä luvussa kerrotaan HTML5:n kehittyminen nykyiseen muotoonsa. Luvussa käydään myös läpi sivuston kehittäjän kannalta keskeisiä asioita, sekä tutustutaan HTML5:n uusiin toimintoihin.

### 2.1 Kehitystyö

HTML5:n kehitys alkoi vuonna 2004, jolloin nk. ”Web 2.0” liike oli nosteessa. Web 2.0 nimike piti sisällään vision, jossa kokonaisia sovelluksia voitaisiin hallinnoida selaimilla. Ajatus oli aikanaan utopinen ja moni, kuten W3C, josta myöhemmin tulisi samaisen teknologian kehittäjä, ei uskonut sen mahdollisuuksiin. W3C piti utopian tavoittelun sijaan tärkeämpänä lähteä kehittämään merkintäkieltä kohti täydellisempää XML-pohjaista muotoa. Moni kehittäjä kuitenkin päätti pitäytyä ”Web 2.0” unelmassaan ja näin syntyi toinen taho, joka kehittäisi tulevaa HTML5:ttä. Tämä uusi taho oli nimeltään WHATWG. (Korpela 2011, 7-8.)

WHATWG jatkoi Web 2.0 kehittämistä uudella nimellä Web Forms 2.0, jonka lisäksi he julkaisivat vuoden 2004 lopussa uuden luonnoksen nimellä Web Applications 1.0. Web Applications 1.0 ajatuksena oli kehittää web-applikaatioita, sekä uutta versiota HTML:stä, jota he leikkimielisesti kutsuivat nimellä HTML5. (Cawrey 2010.)

Vuonna 2006 W3C myönsi ajatuksen maailman muuttamisesta kohti XML-pohjaista muotoa olevan epäonnistunut. Samana vuonna he pyysivät päästä tekemään yhteistyötä WHATWG:n kanssa. Yhteistyön alkaessa WHATWG päätyi nimeämään Web Applications 1.0:n HTML5:ksi, sekä sulauttamaan siihen silloisen Web Forms 2.0:n. (Decentralized Information Group 2006; Cawrey 2010.)

Yhteistyön alkaessa molemmilla oli yhteinen visio webin tulevaisuudesta, joka tultaisiin tuntemaan nimellä HTML5. Yhteistyön tuloksena molemmat tahot tekivät yhteisiä päätöksiä tulevaisuudesta, mutta päättivät samalla pitäytyä omien luonnostensa työstämisessä. Ajatuksena oli koostaa näistä luonnoksista yhteinen HTML5. (Korpela 2011, 18-19.)

Yhteisiä sääntöjä HTML5:n kehittämisessä:

- Uudet toiminnot tulee pohjautua HTML:ään, CSS:ään, DOM:iin ja JavaScriptiin
- Vähentää tarvetta ulkoisille lisäosille (Flash)
- Parempi virheiden käsittely
- Hyödyntää enemmän avoimen standardin mukaista kuvauskieltä, vähentäen skriptikielen tarvetta
- Laitteistoista riippumaton
- Kehitystyö prosessi tulisi olla julkinen

(W3Schools 2012a.)

HTML5:n kehitystyön käydessä kiivaana se alkoi vähitellen löytää tietänsä suuremman yleisön tietoisuuteen. Suuri askel sen olemassaolosta tuli vuonna 2010, jolloin Steve Jobs julkaisi artikkelinsa ”Thoughts on Flash”. Artikkelissa Jobs kehuu HTML5:n ominaisuuksia, sekä siihen liittyviä avoimia kehitysympäristöjä, samalla parjaten suljettuja kehitysympäristöjä, kuten Flash Player. Lisäksi Jobs ilmoittaa Applen luopuvan kokonaan Flash Player:n käytöstä ja keskittyvän HTML5 teknologian kehittämiseen mobiilialustoille. (Jobs 2010.)

Vuonna 2011 yhteiseen visioon tuli kuitenkin särö WHATWG:n ilmoittaessa luopuvansa HTML5 nimen alla tehdystä kehittämyöstä. Syynä oli WHATWG:n ja W3C:n näkemusero, jossa W3C halusi tehdä HTML5:stä tulevaisuuden standardin, kun taas WHATWG katsoi HTML5:n kehityksen olevan vain osa jatkuvaa webin kehitystä. WHATWG päätyikin nimeämään työstämiänsä luonnoksia nimellä ”Living Standard” pitäytyen sillä ajatuksessa, että web- kehitys ei tapahtuisi vain asteittain vaan, että se on alati jatkuvaa kehitystä. Yhteistyö näiden kahden välillä kuitenkin jatkuu nimeämisestä huolimatta. (Elliot 2012.)

Kehitystyö on meneillään tämänkin opinnäytetyön kirjoittamisen hetkellä ja sen lopullista valmistumista ei osaa kukaan tarkoin määritellä. HTML5:n tulevaisuus onkin nykyisellään hieman hämärän peitossa. WHATWG:n halutessa pidättäytyä sen luokittamisesta uudeksi standardiksi on W3C jo luonnostellut HTML5 määrittelyn



ehdotelmaa, jossa HTML5 pyrittäisiin saamaan virallistettua vuonna 2014. (Cotton 2012.)

Vaikkakin HTML5:n standardointityö kehittyi hitaasti, eikä varmuudella voida tietää mitä se tulevaisuudessa pitää sisällään, on sen suuntaa kuitenkin pidettävä oikeana. Erityisen hyvänä asiana on pidettävä selainvalmistajien keskinäistä kilpailua, jossa uusia ominaisuuksia otetaan käyttöön yhä kiihtyvämällä tahdilla. (Niemi 2012.)

Lisäksi on hyvä pitää mielessä HTML5:n pitävän mukanaan suurta painolastia vanhoista HTML-versioista, joissa monet piirteet olivat kehnosti määriteltyjä. Myös selainten aiempi toiminta, johon kuului määrittelyjen vastaisia toimintoja, puuttellista dokumentointia, ovat vaikeuttaneet kehitystyötä. (Korpela 2011, 24.)

## **2.2 Nykytila**

Vaikka HTML5 nimitys on nykyisin epämääräisessä asemassa osan sen kehittäjien luopuessa siitä, voidaan se kuitenkin edelleen nähdä yhtenäisenä kokonaisuutena, joka kokoaa uudet web-tekniikat yhden iskusanan alle. Siinä ei ole enää kyse ”Web 2.0” kaltaisista visioista, vaan se sisältää paljon enemmän konkreettista. (Korpela 2011, 13.)

HTML5 niputtaa yhteen asioita, kuten HTML, CSS, DOM ja JavaScript, samalla tuoden HTML:n perusrakenteisiin kauan kaivattuja uudistuksia. Lisäksi se tarjoaa myös uusia ominaisuuksia HTML merkkaukielen ulkopuolelta, kuten esimerkiksi paikkatieto (geolocation).

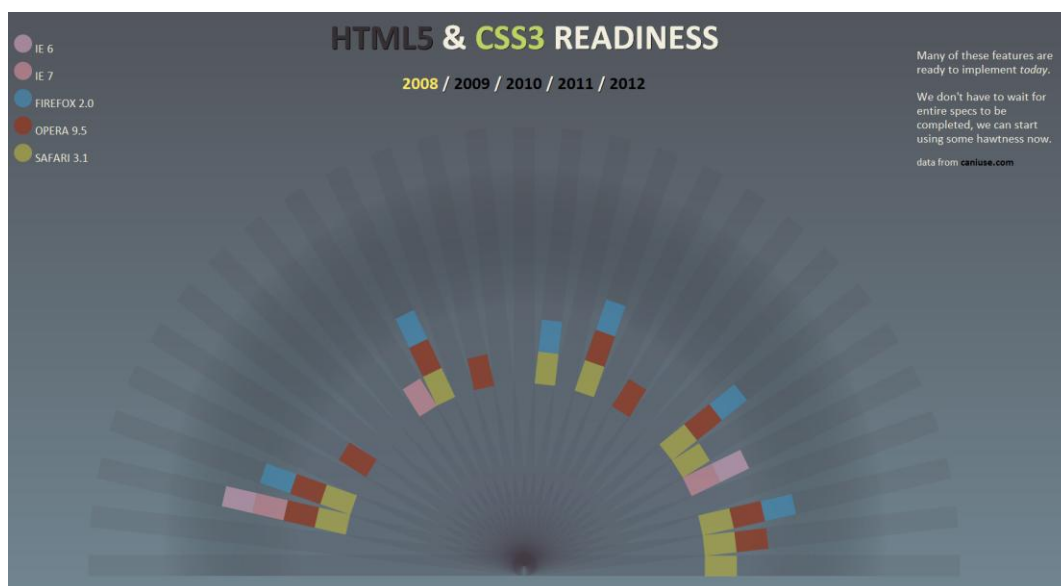
HTML, CSS ja JavaScriptin käyttäminen verkkosivujen rakentamisessa on ollut jo pitkään tuttua puuhaa sivustojen kehittäjille. HTML5 on kuitenkin alusta asti pyrkinyt hyödyntämään näiden välistä toimintaa. Tämä on voitu tehdä pyrkimällä kuvaamaan HTML:ää nykyistä paremmin dokumenttioliomallin (DOM) avulla. (Korpela 2011, Johdanto.)

Vaikka HTML5 tuntuu sisältävän kokonaan uuden tavan ajatella verkkosivujen toteuttamista on se todellisuudessa enemmänkin korillinen uusia tekniikoita, joiden

tarkoitus on hyödyntää mahdollisimman monia kehittäjiä. Uusia tekniikoita onkin tällä hetkellä hyödynnettävissä jo varsin paljon. Esimerkiksi verkkosivujen kehittäjien suosimalla W3School-sivustolta löytyy uusia HTML5- ominaisuuksia nykyisellään jo 30 (Korpela 2011, 13; W3Schools 2012c.)

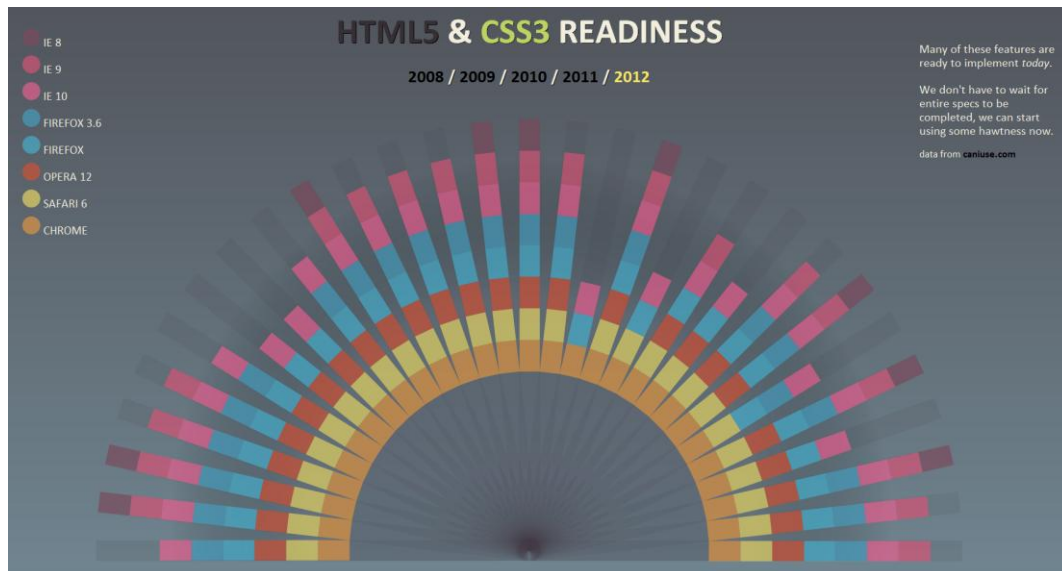
Monet ominaisuuksista ovat virallisesti vasta luonnoksia, mutta niiden käyttö on jatkuvasti lisääntymässä. Tärkeimpinä edistäjinä toimivat kehittäjien lisäksi selainten valmistajat, jotka kilpaa pyrkivät tuomaan uusia ominaisuuksia saataville.

Html5readiness.com sivustolta näkee hyvin HTML5:n ja CSS3:n nopean kehityksen viimeisen neljän vuoden ajalta. Vuonna 2008 HTML5-tuki löytyi vain hyvin harvasta selaimesta (kuvio 1). (Korpela 2011, 13; Irish & Manian 2012.)



Kuvio 1. HTML5 ja CSS3 ominaisuuksien tuki selaimissa vuonna 2008. (Irish & Manian 2012.)

Muutos viimeisen neljän vuoden aikana on ollut suuri. Siinä missä 2008 tuki oli alle 10 %-luokkaa kattavat uudet selaimet jo noin 80% niiden ominaisuuksista (Kuvio 2).



Kuvio 2. HTML5 ja CSS3 ominaisuuksien tuki selaimissa vuonna 2012. (Irish & Manian 2012.)

Niin kauan kuin kyseessä olevat ominaisuudet ovat luonnoksia, eikä kyse ole standardista, on kuitenkin syytä pitää mielessä, että niiden toimivuudesta ei ole takeita. Esimerkinä HTML5:n uusi tuki videoille. Tukea on kehuttu paljon, mutta samalla sen käyttämästä uudesta formaatista ei ole vielä tähänkään päivään mennessä päästy yhteisymmärrykseen, vaan käytettävissä on tälläkin hetkellä 3 eri formaattia, jota eri selaimet tukevat eri tavoin. (Marshall 2011; W3Schools 2012d.)

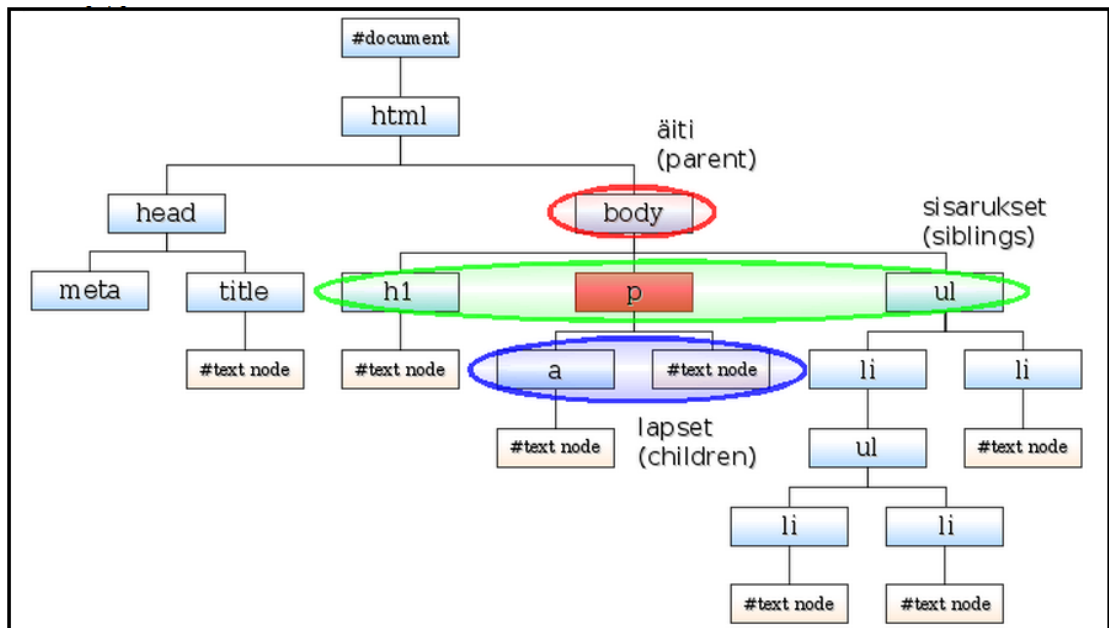
### 2.3 DOM, HTML, JavaScript ja CSS

Sivustojen kehittäjien keskuudessa on jo pitemmän aikaa hyödynnetty eri tekniikoita samalla pyrkien eriyttämään sisältö, ulkoasu ja toiminnallisuus. Näistä saatuja hyötyjä on useita, kuten pienempi tiedostokoko, joka mahdollistaa lyhyemmät latausajat palvelimelta, muutosten tarve vain yhteen kohtaan, avustavien teknologioiden pääsy sisältöön aiempaa helpommin (saavutettavuus), sisältö voidaan helpommin esittää eri päätelaitteilla ja hakukoneiden tehokkuus parantuu.

(Koskinen 2010.)

HTML5 ajatuksena onkin jo lähtökohtaisesti pyritty huomiomaan tekniikoiden toimivuus. Koska sekä CSS että JavaScript tukeutuvat HTML:n elementteihin on HTML5:n määrittely suunniteltu DOM-pohjaisesti. (Korpela 2011, 44.)

DOM luo sivustosta eräänlaisen kartan (mallin), jolloin siinä olevia elementtejä voidaan käsitellä olioina. Malli kuvataan usein puurakenteena, joka sisältää solmuja. Näitä solmuja ovat juuri, äiti, sisarus ja lapsisolmut (kuvio 3). Näin saadaan aikaiseksi rakenne, joka sisältää eri sukulaissuhteita. Jokainen solmu kuvaa oliomallin mukaista luokkaa, jolla on metodeja ja kenttiä. Näitä hyödyntämällä on document-oliosta mahdollista poimia mitä tahansa elementtejä. (Ohjelmointiputka 2008.)



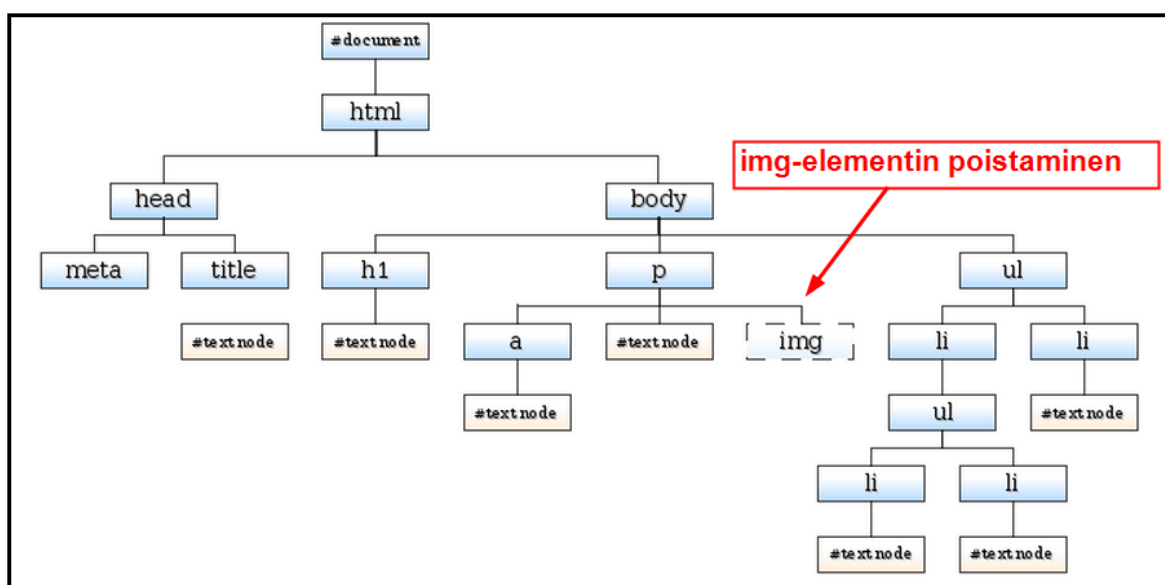
Kuvio 3. DOM puurakenne. (Ohjelmointiputka 2008.)

Mallinnuksen lisäksi DOM tarjoaa myös valmiita helppokäyttöisiä metodeja, joiden avulla on mahdollista valita, muuttaa, luoda tai poistaa elementtejä puurakenteen sisältä. Käyttääkseen näitä metodeita tarvitsee ainoastaan luoda JavaScriptin avulla tarvittava elementti (olio). Tämä onnistuu käyttämällä hyödyksi document-elementin metodeita getElementById ja getElementsByTagName (kuvio 4). getElementById poimii halutun elementin sille annetun id:n avulla ja getElementsByTagName puolestaan hyödyntää elementin tyyppiä. (Ohjelmointiputka 2008.)

```
var listaSolmu = document.getElementById('sisempi');
var kohdat = listaSolmu.getElementsByTagName('li');
```

Kuvio 4. Elementin valitseminen JavaScript:n avulla. (Ohjelmointiputka 2008.)

CSS ja JavaScript pystyvätkin hyvin hyödyntämään DOM:n tarjoamia metodeja, sekä elementeillä olevia ominaisuuksia. CSS pystyy esimerkiksi hyödyntämään elementtien style-ominaisuutta ja näin vaihtamaan halutun elementin väriä. JavaScript puolestaan mahdollistaa elementtien lisäämisen, muokkaamisen ja poistamisen. Hyödyntämällä elementtien poimimista id:n tai tyyppin avulla voidaan esimerkiksi poistaa haluttu elementti tai sen ominaisuus. Jos halutaan esimerkiksi poistaa sivulta img-elementti, tarvitsee ainoastaan etsiä siihen liittyvä solmu ja käyttää `removeChild(oldChild)`-ominaisuutta (kuvio 5). (Ohjelmointiputka 2008.)



Kuvio 5. Elementin poistaminen JavaScriptin avulla. (Ohjelmointiputka 2008.)

## 2.4 Rakennemuutokset

HTML5 tuo yksinkertaistuksia sen rakenteisiin ja merkkaukseen. Niiden avulla on pyritty paitsi yksinkertaistamaan koodia, myös vähentämään virheiden määrää. (Korpela 2011, 32.)

HTML5:n myötä poistuu käytöstä aiemmin hankalat dokumenttityypin ilmoitukset, joiden tehtävä oli vaikuttaa siihen toimiiko selain standarditilassa vai oikkutilassa. Uusi dokumenttityypin ilmoitus (kuvio 6) käyttääkin jatkossa aina standarditilaa, paitsi jos dokumenttityyppi puuttuu tai on virheellinen. Uuden dokumenttityypin käyttöä on kuitenkin syytä välttää vanhemmissa sivustoissa. Jos esimerkiksi haluaa käyttää uusia ominaisuuksia vanhoihin sivustoihin, on parempi olla lisäämättä tai vaihtamatta

dokumenttityypin ilmoitusta ellei tiedä mitä tekee. Lisäksi on muistettava, että HTML5:n ollessa yhä luonnosasteella siihen liittyvät tarkistimien muotoasetukset eivät välttämättä vastaa uusinta olemassa olevaa versiota. (Korpela 2011, 15; 28-29.)

```
<!DOCTYPE html>
```

Kuvio 6. HTML5:n mukainen dokumenttityypin ilmoitus. (Korpela 2011, 28.)

HTML5:ssä on yksinkertaistettu HTML:n muutosääntöjä sallien kaikki määrittelyt, jotka selaimet yleisesti hyväksyvät. Näin aiemmin ns. ”virheet” muuttuvatkin ”lailliseksi”. Näin on pystytty poistamaan aiemmin turhia määrittelyksiä, kuten style-elementin tyyliohjekielä CSS:ssä ja JavaScriptissa (kuvio 7). (Korpela 2011, 32; Vanhala-Nurmi 2012a.)

CSS	JavaScript
<code>&lt;style&gt;</code>	<code>&lt;script&gt;</code>
....	....
<code>&lt;/style&gt;</code>	<code>&lt;/script&gt;</code>

Kuvio 7. HTML5:n uudet tyylimäärittelyt. (Vanhala-Nurmi 2012a.)

Tämän lisäksi on voitu myös luopua aiemmista mimetyypeistä, joita on käytetty muun muassa merkistökoodauksessa, CSS- ja JavaScript- tiedostojen liittämässä. Näin saatu koodi on huomattavasti aiempaa selkeämpää (kuvio 8). (Korpela 2011, 32; Vanhala-Nurmi HTML5: rakenne ja merkkäminen.)

#### **Merkistökoodaus**

```
<meta charset="windows-1252">
```

#### **CSS tyylikirjaston liittäminen dokumenttiin**

```
<link rel="stylesheet" href="tyyli.css" />
```

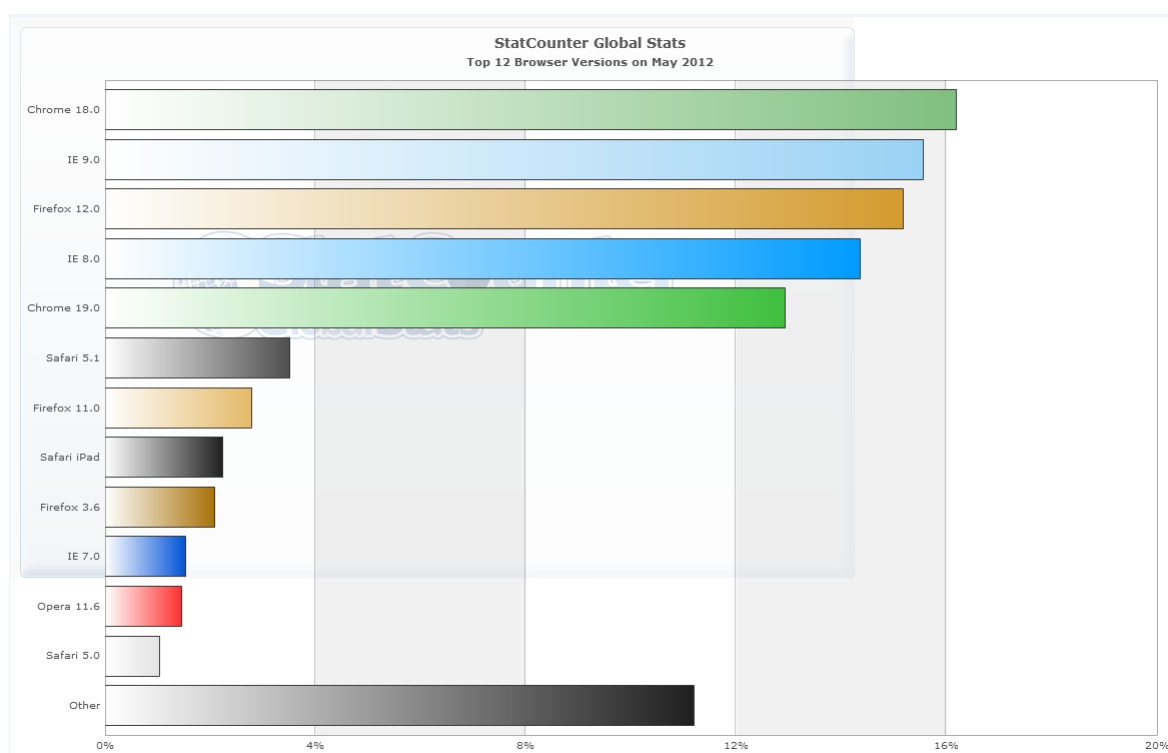
#### **JavaScript tiedoston liittäminen dokumenttiin**

```
<script src="scripti.js" />
```

Kuvio 8. HTML5:n uudet merkistökoodaus ilmoitukset. (Vanhala-Nurmi 2012a.)

## 2.5 Selaintuki ja eroihin varautuminen

Vaikkakin uudet selaimet tukevat uusia HTML5:n ominaisuuksia jo verrattain hyvin on silti huomioitava, että selainten välinen ero on edelleen merkitsevä. Selainten suuren määrän ja vanhempien selainten olemassaolon takia (kuvio 9) onkin syytä pyrkiä turvaamaan sivuston toiminnot mahdollisimman hyvin. HTML5 tarjoaakin useimmissa tapauksissa tavan esittää vaihtoehtoisälttöä selaimille, jotka eivät uutuutta tunnista. Tämän lisäksi on myös mahdollista käyttää JavaScriptia tai sopivaa ohjelmakirjastoa apuna tunnistamaan uusien ominaisuuksien tuki. (Korpela 2011, 34.)



Kuvio 9. Suosituimmat selaimet toukokuussa 2012. (StatCounter GlobalStat 2012.)

Kun käytössä oleva selain ei tue sivustolle asetettua ominaisuutta, voi siihen varautua erilaisin tavoin. Huonoin vaihtoehto on jättää reagoimatta siihen kokonaan. Tällöin sivu saattaa kuitenkin näyttää oudolta tai toimimattomalta. Toinen vaihtoehto on tunnistaa tuki ja tehdä siitä ilmoitus, kuten ”Selaimesi ei tue uutta video ominaisuutta. Ominaisuuden tuki on mukana uudemmissa selaimissa, kuten IE 9”. Kolmas vaihtoehto on tarjota varasisältöä, kuten videon tilalle kuva tai kuvan tilalle teksti. (Korpela 2011, 77)

Varasisällön näyttämiseen onkin olemassa eri tapoja, joista toimivin on aiemmissa HTML- versioissa käytössä ollut object-elementti. Object-elementti on määritetty niin, että selain pyrkii ensisijaisesti näyttämään elementissä olevan tiedon. Jos selain ei tunnista elementtiä se pyrkii seuraavaksi näyttämään elementin sisällön. (kuvio 10).

```
<object data="varoitus.gif" type="image/gif">  
Varoitus!  
</object>
```

Kuvio 10. Varasisältö object-elementissä. (Korpela 2011, 78)

HTML5:n uudet ominaisuudet onkin määritetty samaan tapaan kuin object-elementti. Näin selain pyrkii näyttämään varasisällön aina tarvittaessa (kuvio 11).

```
<canvas id="piirtoalue" width="500" height="300">  
Selaimesi ei tue HTML5:n uutta canvas-elementtiä. Elementin tuki on mukana  
uudemmissa selaimissa, kuten IE 9.  
</canvas>
```

Kuvio 11. Varasisältö canvas-elementissä. (Korpela 2011, 34)

Käyttämällä JavaScriptia pystytään myös tunnistamaan eri selaimet, sekä niiden versiot. Tällöin ei kuitenkaan voida taata tietojen todenperäisyyttä. Tästä syystä onkin parempi pyrkiä tunnistamaan käytettyjen HTML5 piirteiden tuki käyttäen apuna JavaScriptia. Eri HTML5-elementtien tuen tunnistamiseen on lisäksi olemassa vieläkin helpompi tapa, Modernizr- kirjasto. Sivuuun lisätyn tiedoston viittauksen jälkeen Modernizr luo globaalin Modernizr-olion, joka sisältää tietoa HTML5, sekä CSS3 tuesta. Käyttämällä oliota, voidaan tutkia, tukeeko sivusto kyseistä piirrettä (kuvio 12).(Korpela 2011, 76-77.)



```
if(Modernizr.canvas) {  
    // canvas- elementtiä käsittelevä koodi  
}  
else {  
    // koodi selaimelle, jotka eivät tue canvas- elementtiä  
}
```

Kuvio 12. Modernizr-olion käyttö HTML5 ominaisuuden tunnistamiseen. (Korpela 2011, 77.)

## 2.6 Uudet toiminnot

### 2.6.1 Semanttiset elementit

Ymmärtääkseen paremmin HTML5:n uusia semanttisia elementtejä on hyvä selkeyttää semanttinen sanan tarkoitus. Jokainen sivuja toteuttanut on varmasti kohdannut semanttisia elementtejä, esimerkkinä h1 ja p. Niitä käytettäessä voisikin kysyä, miksi ei käytettäisi yksinkertaisesti pelkkiä div tai span merkintöjä. Syy piilee h1 ja p merkintöjen sisältämässä semantiikassa, merkitystä kuvaavassa tavassa. Siinä missä pelkkä div merkintä ei sisällä informaatiota, voidaan h1 nähdä jo valmiiksi otsikkona ja p kappaleena. (Ryan 2012.)

Yksinkertaistetusti semanttinen HTML tarkoittaa HTML:ää, jossa elementeille pyritään luomaan niiden tarkoitusta vastaava tehtävä. Näin voidaan HTML:lle antaa rakenne, jota CSS:n ja JavaScript:n on helpompi käsitellä. (Ryan 2012.)

Sen lisäksi, että semanttiset elementit sisältävät paitsi lisäinformaatiota voidaan niissä nähdä lisäksi seuraavanlaisia etuja: lyhyempi koodi, parempi saavutettavuus (esteettömyys), parempi hakukoneiden näkyvyys, monikäyttöisempi ja kehittäjäystävällisempi koodi. (Ryan 2012.)

HTML5:ssä on otettu käyttöön useita uusia rakenne-elementtejä:

- header – ylätunniste, esimerkiksi otsikko, sisällysluettelo tai logo
- nav – navigointiosa, joka sisältää linkkejä sivuston muihin osiin tai sivuihin
- article – artikkelityyppinen itsenäinen kokonaisuus
- section – dokumentin osa, esimerkiksi kirjan luku

- aside – lisäinfoa asiasta
- footer – alatunniste, esimerkiksi tekijätiedot, yhteystiedot
- figure – voidaan koota yhteen kuvat ja niihin liittyvät kuvatekstit
- hgroup – otsikkoryhmä, kuten otsikko ja sen alaotsikko

(Korpela 2011, 86; Vanhala-Nurmi 2012b.)

Elementtien tarkoitus on korvata aiemmin div- tai span-elementtien avulla toteutettuja rakenteita. Näitä ovat mm. navigaatio ja alatunniste. Uusien elementtien kohdistaminen CSS:ssä toimii samoin kuin esimerkiksi h1 merkintä (kuvio 13). (Korpela 2011, 86-88.)

```
h1 {color: red}
footer {background-color: gray}
```

Kuvio 13. footer elementin CSS-valitsin. (Korpela 2011, 88.)

HTML5 määritelmä lisäksi tarkentaa useiden vanhojen elementtien merkitystä. Näin se pyrkii julistamaan virheellisiksi jotkin vanhoista elementeistä, joita on käytetty lähinnä ulkoasun muotoilun välineinä. Näin pyritään johdonmukaisemmin siihen, että kutakin elementtiä käytettäisiin ainoastaan sille tarkoitettulla tavalla. (Korpela 2011, 127.)

A-elementin määrittelyä on muutettu niin, että sen sisällä saa olla aiempaa kattavammin eri tyylistä sisältöä, kuten kappale, kuva, lista tai taulukko (Kuvio 14). Ainoa sääntö on, että sisältö ei saa olla vuorovaikutteista, kuten linkki. (Korpela 2011, 127.)

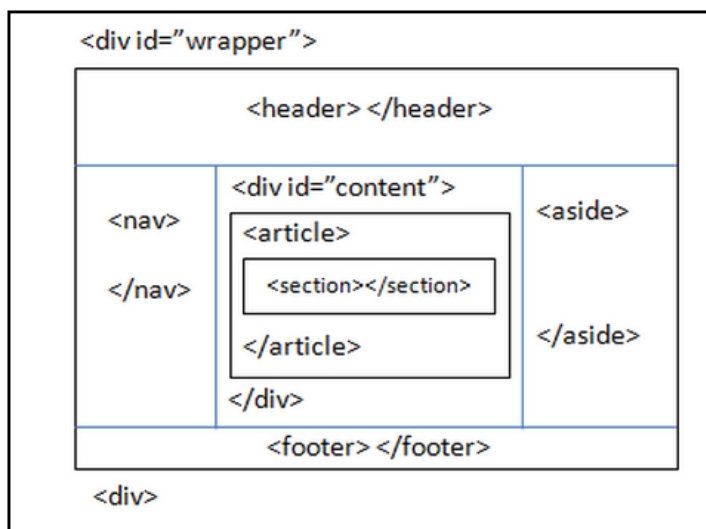
```
<a href="manse-info.html" class="lohko">

<h3>Käy oikeassa mansessa!</h3>
<p>Manchester, Englannin Tampere, on monipuolinen
kulttuurikaupunki, jossa on mm. chinatown.</p>
</a>
```

Kuvio 14. A-elementin laajennettu määrittely. (Korpela 2011, 127.)

Vaikka div-elementtien käytön on tarkoitus vähentyä uusien elementtien tieltä on se edelleen laajasti käytössä. Sitä käytetäänkin kokoamaan sivun osia yhtenäisiksi

elementeiksi, joita voidaan tämän jälkeen muotoilla tyyliasetuksilla tai käsitellä JavaScriptin avulla (kuvio 15). (Korpela 2011, 132.)



Kuvio 15. HTML5- sivu käyttäen apuna div-kokoomaelementtejä. (Vanhala-Nurmi 2012b.)

Uusien elementtien käyttö ei ole vielä kovinkaan yleistä. Niiden oikeanlaisesta käytöstä ei kuitenkaan voida katsoa olevan haittaa vaan lähinnä hyötyä. Tämä johtuu niiden semanttisesta vaikutuksesta, jolloin niitä voidaan käyttää vaikuttamatta mitenkään sivuston ulkoasuun. Niiden yhtenäinen merkkauksetapa helpottaa niin pienempien, kuin suurempien sivustojen JavaScript ja CSS koodin laatimista. Niillä voidaan lisätä monenlaista tukea sivustolle, esimerkiksi korostaa tiettyä osa-aluetta, kuten vaikkapa navigointielementtejä. Toisena esimerkkinä sivuston navigointielementtien poistaminen kokonaan tapauksessa, jolloin sivuston sisältö halutaan tulostaa tulostimella. Myös hakukoneiden kannalta niiden käyttö on toivottavaa, vaikka ei vielä tällä hetkellä kovinkaan tuettua. Esimerkiksi figure-elementin käyttö mahdollistaa hakukoneelle toimivamman tavan etsiä hakuun kohdistuneita kuvia sivustoilta. (Korpela 2011, 88.)

Monet, uusien rakenne-elementtien käyttäjistä haluavat lisätä niihin tyyliasetuksia. Esimerkiksi asettamalla reunaviivoja tai esittämällä niiden sisältöä eri fontilla. Uusien elementtien viittaustapa poikkeaa CSS:n osalta aiemmasta. Haluttaessa määrittää nav-elementin ympärille reunaviiva, ovat aiemmat tavat olleet, joko `.nav` tai `#nav`. Uusien elementtien viittaustapa on kuitenkin muuttunut niin, että niihin viittaamisessa riittää

pelkkä elementin nimi, kuten nav tai footer. Tämä uusi toiminto ei ole tuettu edes uusimmissa selaimissa, vaan uudemmat selaimetkin voivat käsitellä elementit virheellisesti tekstitason elementteinä. Lisäämällä CSS tiedostoon määrittymisen, jossa kaikki elementit muotoillaan lohkoina (block), saadaan selaimet käsittelemään uudet elementit oikein. (Korpela 2011, 89.)

## 2.6.2 Video median upottaminen HTML- sivuun

Kuten Korpela mainitsee (2011, 177), on eri medioiden upottaminen HTML-sivuun yksi HTML5:n tunnetuimmista asioista. Web-sivustoista on tähän asti puuttunut video- ja audio-standardi, jolloin toistamiseen on tarvinnut erillisten liitännäisten, kuten Flash Playerin käyttöä. HTML5:n myötä on tarkoitus standardoida uudet video- ja audio-elementit, jotka mahdollistavat niiden upottamisen sivulle sekä toistamisen ilman liitännäisiä. Lisäksi HTML5 sisältää DOM- metodeita, ominaisuuksia ja tapahtumia näille elementeille, jolloin niitä voidaan muokata käyttäen apuna JavaScriptia. Vaikkakin uudet elementit ovat hyvin tuettu uusimmissa selaimissa, vaatii niiden käyttö tukea eri mediamuodoille. Lisäksi niiden tuki puuttuu kokonaan vanhemmista IE-selaimista. (W3Schools 2012d.)

Tähän asti toteutettu tapa liittää video tai audio sivulle on toteutettu liitännäisen (plugin) avulla. Liitännäisen käytön ongelmana on kuitenkin sen näkymättömyys selaimen kannalta, jolloin selain ei pysty vaikuttamaan esitykseen millään tavoin. HTML5:n uudet video- ja audio-elementit perustuvatkin ajatukseen, että selain pystyy vaikuttamaan niiden esitykseen, koska niiden tehtävän hoitaa selaimen integroitu ohjelma. Uusien elementtien lisääminen sivulle on mahdollista yksinkertaisen koodin avulla. Liittääkseen sivulle videon tarvitsee ainostaan määrittää video-elementin sisälle src-määrite, joka ilmoittaa videotiedoston sijainnin, sekä width-määrite videon koon määrittämiseen. (kuvio 16). (Korpela 2011, 178, 181.)

```
<video src="Video1.mp4" controls="controls" width="400"></video>
```

Kuvio 16. Video-elementin upottaminen sivulle. (Korpela 2011, 178.)

Kuvassa (kuvio 16) voidaan lisäksi nähdä määrite (controls), joka liittää elementteihin sisäänrakennetut ohjaimet. Sekä video, että audio-elementeille on olemassa useita eri määritteitä, joilla voidaan vaikuttaa niiden toimintaan. Video-elementillä on audio-elementtien lisäksi määritteet, kuten (muted), (height), (poster) ja (width). Määritteiden käytöllä voidaan lisäksi varmistaa, että haluttu toiminto toimii myös ilman JavaScriptin käyttöä. (Korpela 2011, 186; 192.)

Audio- ja video-elementtejä kutsutaan HTML5:ssä mediaelementeiksi, joilla tarkoitetaan niiden sisältävän yleisten ominaisuuksien lisäksi HTMLMediaElement-liitännän mukaisia ominaisuuksia. Nämä ns. DOM-ominaisuudet tekevät mediaelementeistä ”natiiveja” olennaisesti HTML5:een kuuluvia elementtejä. (Korpela 2011, 190.)

DOM:n ominaisuudet avaavat videon ja audion käsittelyyn paljon uusia mahdollisuuksia. Video-objektin ominaisuuksia hyödyntämällä on mahdollista esimerkiksi ohjata videon toimintoja erillisillä sivustolle asennettavilla painikkeilla. Painikkeiden toiminnot voidaan tämän jälkeen toteuttaa JavaScriptin avulla. Tämä onnistuu hyödyntämällä video-elementin sisäänrakennettuja (play) ja (pause) funktioita sekä (currentTime) ominaisuutta. Näin voidaan vaikuttaa esimerkiksi painikkeiden sisältöihin, sekä kontrolloida videon toimintaa (kuvio 17). (Microsoft 2012a.)

```
function vidplay() {
    var video = document.getElementById("Videol");
    var button = document.getElementById("play");
    if (video.paused) {
        video.play();
        button.textContent = "||";
    } else {
        video.pause();
        button.textContent = ">";
    }
}

function restart() {
    var video = document.getElementById("Videol");
    video.currentTime = 0;
}
}
```

Kuvio 17. Video-elementin, sekä siihen liitettyjen painikkeiden hallinta JavaScriptin avulla. (Microsoft 2012a.)

Video-elementtien käytössä suurin ongelma liittyy niiden tukeen eri selaimissa. Ongelmana on, että eri selainvalmistajat tukemat eri lailla eri formaatteja, samalla kuitenkin tukematta yksinomaan yhtä tiettyä formaattia (kuvio 18). Lisäksi vanhat IE-selaimet eivät tunnista uusia elementtejä ollenkaan. (W3Schools 2012d.)

Browser	MP4	WebM	Ogg
Internet Explorer 9	YES	NO	NO
Firefox 4.0	NO	YES	YES
Google Chrome 6	YES	YES	YES
Apple Safari 5	YES	NO	NO
Opera 10.6	NO	YES	YES

Kuvio 18. Eri selainvalmistajien tukemat video formaatit. (W3Schools 2012d.)

Käyttämällä yksinkertaisinta mallia videon liittämisessä sivulle antaa se mahdollisuuden käyttää vain yhtä formaattia videon esittämiseen. Tämän src-ominaisuuden sijaan on kuitenkin mahdollista käyttää HTML5:n uutta source-elementtiä. Source-elementti mahdollistaa sopivimman formaatin valinnan useammasta eri vaihtoehdosta. Näin voidaan koodissa antaa useampi vaihtoehto, joista selain pyrkii etsimään sopivimman (kuvio 19). (Microsoft 2012a.)

```

<video width="320" height="240" controls="controls">
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  <source src="movie.webm" type="video/webm">
  <p>Selaimesi ei tue videon toistamista. Video on katsottavissa uudemmilla</p>
  <p>IE, Chrome, Firefox, Opera ja Safari selaimilla tai voit vaihtoehtoisesti</p>
  <p>ladata videon koneellesi<a href="demo.mp4"> tästä</a>.</p>
</video>

```

Kuvio 19. Source-elementin hyödyntäminen video-elementtiä liittämisessä, sekä vanhempien selainten varasisältöön varautuminen. (W3Schools 2012d.)

Toiston epäonnistuessa siirtyy selain automaattisesti näyttämään varasisältöä, joka voi olla ilmoitus tuen puuttumisesta, sekä mahdollisuus ladata tiedosto omalle koneelle katsomista varten (kuvio 19). Varasisältönä voidaan käyttää myös Flash Playerin avulla toistettavaa tiedostoa tai esimerkiksi youtuben tuottamaa upotuskoodia. Näin videon toistaminen on mahdollista myös vanhemmilla IE-selaimilla. (Korpela 2011, 179.)

Flash Playerin käyttö antaakin mahdollisuuden näyttää video myös vanhemmissa selaimissa. Tämä on mahdollista tehdä eri tavoin. Microsoftin sivulla neuvotaan hyödyntämään object- ja embed-elementtejä yhdistettynä Flash Playeriin. Toinen tapa, jota on suositeltu Camen Designin sivulla, on object- ja param-elementtien yhdistäminen Flash Playeriin. Flash Player tarvitsee toimiakseen soittimen, joka voidaan tallentaa valmiiksi omalle palvelimelle tai vaihtoehtoisesti viitata erillisen palvelimen tiedostoon. Camen Designin sivulta löytyy lisäksi muunnin, johon voi muun muassa syöttää valmiiksi tiedostojen osoitteen, valita halutun varasisällön, videon koon jne. Tämän jälkeen valmiin koodin voi kopioida omalle koneelle.  
(Camen Design 2012; Microsoft 2012b.)

### 3 BPM Helsinki Oy:n sivustoprojekti

Tässä luvussa kerrotaan BPM Helsinki Oy:n sivustoprojektista. Sivustoprojektissa lähdetään liikkeelle nykyisestä sivustosta, jonka jälkeen rakennetaan uusittu sivusto hyödyntäen uusia HTML5-tekniikoita.

#### 3.1 Lähtökohta ja projektisuunnitelma

BPM Helsinki Oy:n pääasiallinen toiminta koostuu viisumiasioiden hoitamisesta pääkaupunkiseudulla. Tämän lisäksi he tarjoavat myös rakennuspalveluita, johon kuuluvat mm. lavasteiden ja studioakustiikan rakentamista. Toimintojen laajetessa tulisi myös sivusto saada paremmin tukemaan ja vastaamaan tulevaisuuden suunnitelmia.

Ensimmäisessä neuvottelussa käymme läpi nykyistä sivustoa ja yrityksen ajatuksia uudesta sivustosta. Uuteen sivustoon kaivattiin erityisesti uudenlaista rakennetta. Ilmeni, että sivuston työstäminen tulee olemaan yritykselle jatkuva prosessi, johon se tulee lisäämään uusia ominaisuuksia aina tarpeen vaatiessa. Tulimme siihen tulokseen, että uusi HTML5 olisi omiaan tukemaan tätä tarvetta.

Koska visio ensimmäisestä vaiheesta oli selkeä, päädyttiin suunnitelmaan, jossa rakennan opinnäytetyönä pohjan uudelle sivustolle ja keskityn tärkeimpien asioiden ratkaisemiseen, jotka ovat sivurakenteen uudistaminen, sekä video-tekniikan sisällyttäminen siihen. Neuvottelujen pohjalta itselleni muodostui selkeä käsitys siitä, mitä sisällytän opinnäytetyöhöni. Lähtökohtana on tutkia HTML5-tekniikoita ja pyrkiä hyödyntämään niitä parhaiten soveltuvalla tavalla.

Projektisuunnitelman tavoitteena on rakentaa BPM Helsinki Oy:lle toimiva pohjaratkaisu hyödyntäen HTML5-tekniikoita. Pohjaratkaisussa huomioidaan nykyinen sivusto, sekä yrityksen toiveet. Tekniikoita tehdessä pyritään huomioimaan sekä vanhempien selainten tuki, että varmistamaan sivuston toimintojen hyödyntäminen tulevaisuudessa.



Sivustoprojekti toteutetaan sivuston tekijän omalla tietokoneella NetBeans-kehitysympäristössä. Lisäksi sivuston tekijällä on käytettävissä Nettihotelli.fi kotisivu, joka mahdollistaa sivuston esittämisen aika ajoin yrityksen omistajille.

Sivustoprojektin suunnitelma ja aikataulu (syksy 2012):

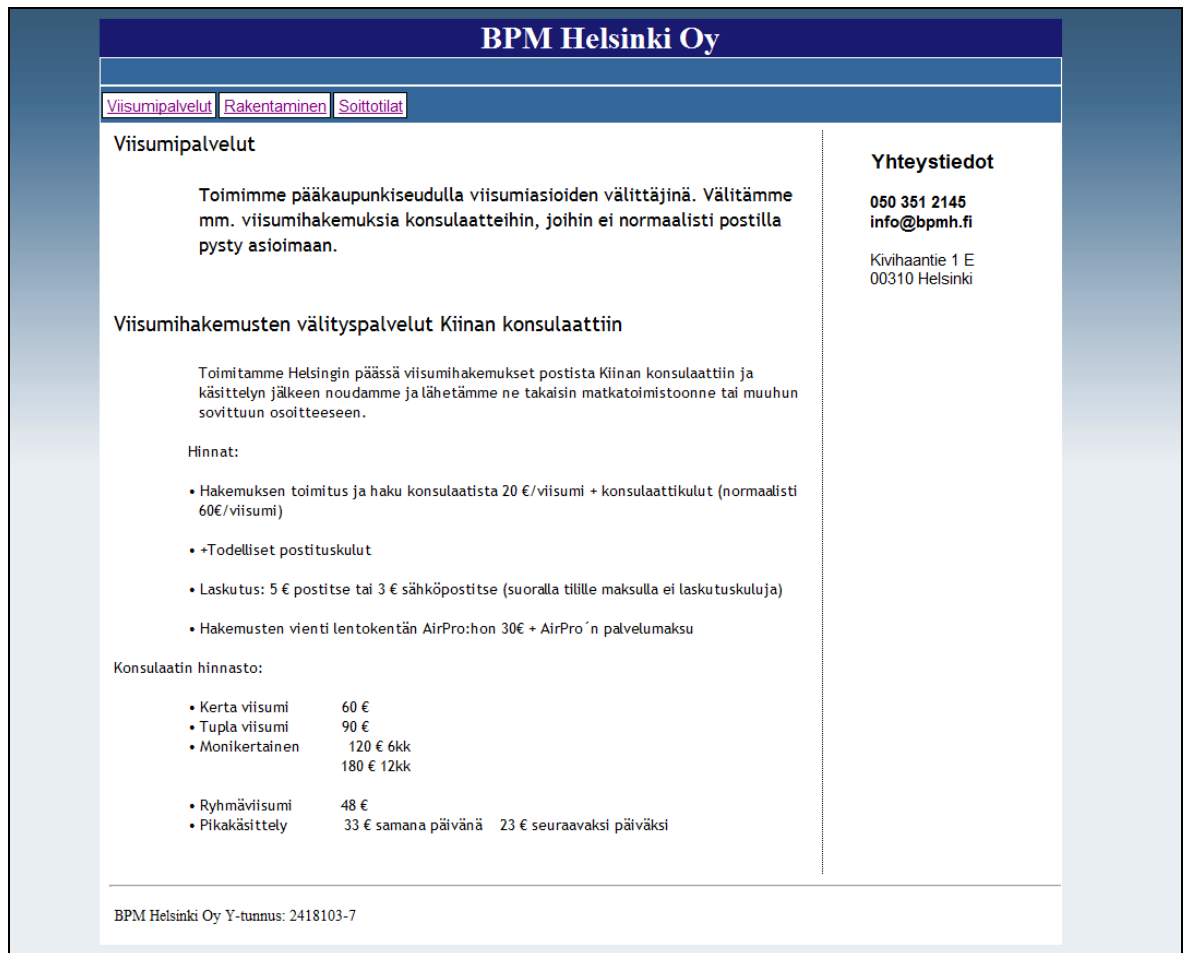
- Tutustuminen HTML5-tekniikoihin, alkuneuvottelut sekä nykyiseen sivustoon tutustuminen (viikot 1-3)
- Sivuston muuntaminen HTML5:n mukaiseksi (viikko 4)
- Etusivun, sekä muiden sivupohjien työstäminen (viikot 5-6)
- HTML5-videotuen lisääminen sivustolle (viikot 7-8)

### **3.2 Nykyinen sivustorakenne ja uuden rakenteen pohjustus**

Aloitin työn tutustumalla nykyiseen sivustoon ja sen rakenteeseen. Ajatuksena oli tutkia nykyisen sivuston rakennetta ja sitä, voisiko se toimia uuden sivuston pohjana.

Tiedossa oli, että sivuston muuntaminen vastaamaan uusia HTML5-määrittäjiä voisi olla haastavaa ja luultavim vaihtoehto näin pienen sivuston kohdalla olisi luultavasti puhtaalta pöydältä aloittaminen.

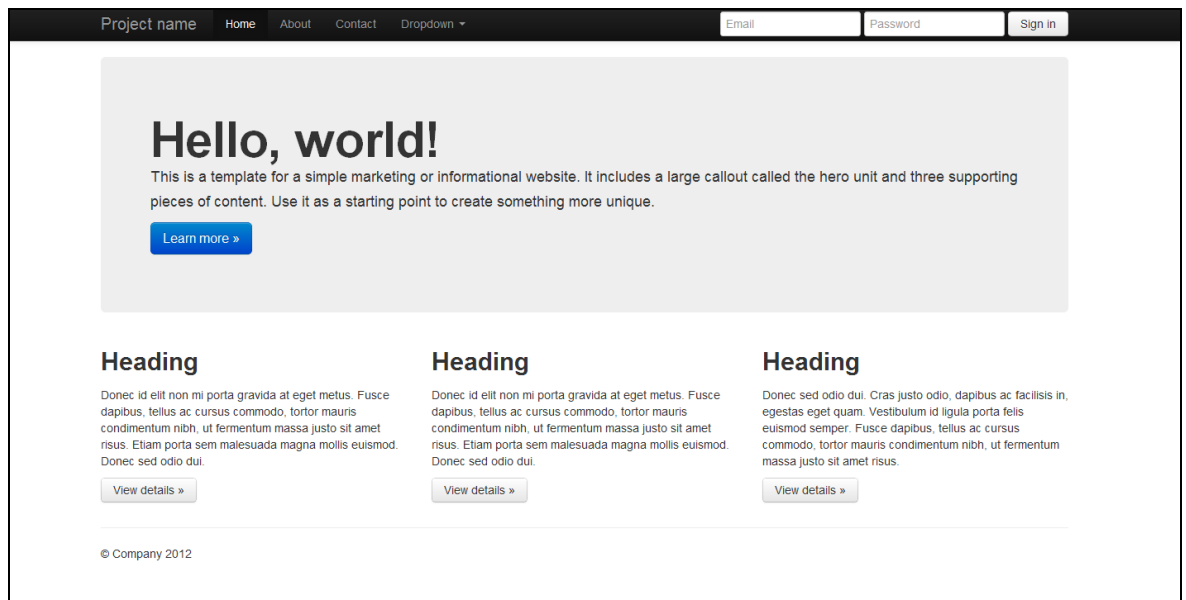
Aiempi sivusto koostui kolmesta erillisestä word-dokumentista, jotka oli tallennettu Web Page-muotoon ja liitetty yhteen PHP:n avulla (kuvio 20). Ulkoasu oli toteutettu käyttäen apuna taulukoita ja niiden sisällä olevia div-elementtejä.



Kuvio 20. BPM Helsinki, nykyinen sivusto.

Vaikkakin sivuston hyödyntämä PHP-tekniikka on tulevaisuutta silmälläpitäen järkevä, on HTML5:n mukaan taulukoiden käyttäminen muotoiluun vanhentunut tapa, jota ei tulisi käyttää. Sivuston muuntaminen uuden HTML5:n vaatimusten mukaiseksi vaatisi kaikkien ulkoasumääritysten uudelleen tekemistä, jolloin helpoin ja nopein vaihtoehto olisi tehdä kokonaan uusi sivusto.

Seuraavaksi tuli päättää sivuston uuden rakenteen toteuttamistapa. Vaihtoehtoina oli toteuttaa rakenne ja sivupohja kokonaan itse tai hyödyntää jo valmiita HTML5:n mukaisia sivupohjia. Päädyin valitsemaan Bootstrap:n tarjoaman ”Basic marketing site”-sivupohjan, joka oli lisäksi yrityksen mielestä toimivan näköinen. Sivupohja tarjosi toimivan rakenteen ja ulkoasun sekä etusivun, että muiden sivujen tekemiselle. (kuvio 21). Tämän lisäksi se on toteutettu HTML5-määrittäjiä silmällä pitäen, jolloin siihen olisi helppo lisätä uudet ominaisuudet.



Kuvio 21. Basic marketing site-sivupohja.

### 3.3 Luonnos uudesta sivustosta

Luonnosta varten oli tärkeä saada valmiiksi tarvittava erillinen etusivu, sekä yksinkertainen mallipohja muita sivuja varten. Bootstrapin sivupohja oli ulkoasullisesti sekä teknisesti hyvin lähellä ajateltua etusivua, jolloin muutosten tekeminen olisi kohtuullisen helppoa.

Sivupohjan hyödyntäessä jo valmiiksi uusia yksinkertaistettuja muutosääntöjä jäi ainoaksi tehtäväksi vaihtaa kielityyppi ja merkistöasetus. Muita tarvittavia muutoksia olivat html5shiv- tuen muuttaminen suoraan palvelimelle, omien CSS ja JavaScript tiedostojen, mukaanlukien modernizr.js:n liittäminen sivulle. Bootstrap sivupohja pitää sisällään valmiita CSS, sekä JavaScript-tiedostoja, mutta liittämällä omat tiedostot näiden tiedostojen perään pystyy helpoiten vaikuttamaan haluttuihin määrittäisiin ilman, että tarvitsee muokata valmiita suurikokoisia tiedostoja (kuvio 22).

```

<!DOCTYPE html>
<html lang="fi">
  <head>
    <meta charset="windows-1252">
    <title>BPM Helsinki</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="Sivustokuvaus">
    <meta name="author" content="">
    <!-- Le styles -->
    <link href="assets/css/bootstrap.css" rel="stylesheet">
    <link href="assets/css/bootstrap-responsive.css" rel="stylesheet">
    <link href="assets/css/bpmh.css" rel="stylesheet">
    <script src="assets/js/modernizr.js"></script>
    <!-- Le HTML5 shim, for IE6-8 support of HTML5 elements -->
    <!--[if lt IE 9]>
      <script src="assets/js/html5shiv.js"></script>
    <![endif]-->
  </head>
  <body>
    ...
    <!-- Le javascript
    ===== -->
    <!-- Placed at the end of the document so the pages load faster -->
    <script src="assets/js/jquery.js"></script>
    ...
    <script src="assets/js/bpmhScripts.js"></script>
  </body>
</html>

```

Kuvio 22. HTML5 Head-osion asetukset ja tarvittavat tiedostomäärittelyt.

Tutkiessani Bootstrap sivupohjaa huomasin siinä käytettävän uusista elementeistä ainoastaan footer-elementtiä. Vaikkakin Bootstrap tarjoaa mahdollisuuden muokata sivun näkymän eri päätelaitteille, en pystynyt löytämään tulostukseen liittyvää tukea. Tämän takia päädyin lisäämään sivustoon nav-elementin. Nav-elementtiä voidaan näin hyödyntää tulevissa projekteissa, esimerkiksi jättämällä sivuston navigoinnin pois tulostuksesta (kuvio 23).

```

<div class="navbar navbar-inverse navbar-fixed-top">
  <div class="navbar-inner">
    <div class="container">
      <a class="btn btn-navbar" data-toggle="collapse" data-target=".nav-collapse">
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </a>
      <div class="nav-collapse collapse">
        <nav>
          <ul class="nav">
            <li class="active"><a href="index.php">Etusivu</a></li>
            <li><a href="viisumipalvelut.php">Viisumipalvelut</a></li>
            <li><a href="rakentaminen.php">Rakentaminen</a></li>
            <li><a href="soittotilat.php">Soittotilat</a></li>
            <li class="dropdown">
              <a href="#" class="dropdown-toggle" data-toggle="dropdown">Lisätiedot<b class="caret"></b></a>
              <ul class="dropdown-menu">
                <li><a href="yhteystiedot.php">Yhteystiedot</a></li>
                <li><a href="yritys.php">Tietoa yrityksestä</a></li>
              </ul>
            </li>
          </ul>
        </nav>
      </div><!--/.nav-collapse -->
    </div>
  </div>
</div>

```

Kuvio 23. Nav-elementin lisääminen ul-listaukseen.

Listauksen ollessa kunnossa oli seuraava vaihe muokata Basic marketing site-sivupohjassa olevia div-elementejä, sekä vaihtaa painikkeiden linkitys sivuston eri sivuille. Lisäämällä etusivulla oleviin div-elementteihin kuvat sekä otsikot, on sivulla kävijän helppo ymmärtää olevansa etusivulla. Koska ainoa tapa siirtyä muille sivuille oli klikata ”Siirry sivulle”-painiketta, päätin lisätä divien ympärille HTML5:n mukaisen uusitun a-elementin. Ympäröimällä div-elementit a-elementeillä (kuvio 24) sain tehtyä myös div-elementtien sisällä olevasta kuvasta, sekä tekstistä linkin, jonka avulla voi siirtyä muille sivuille.

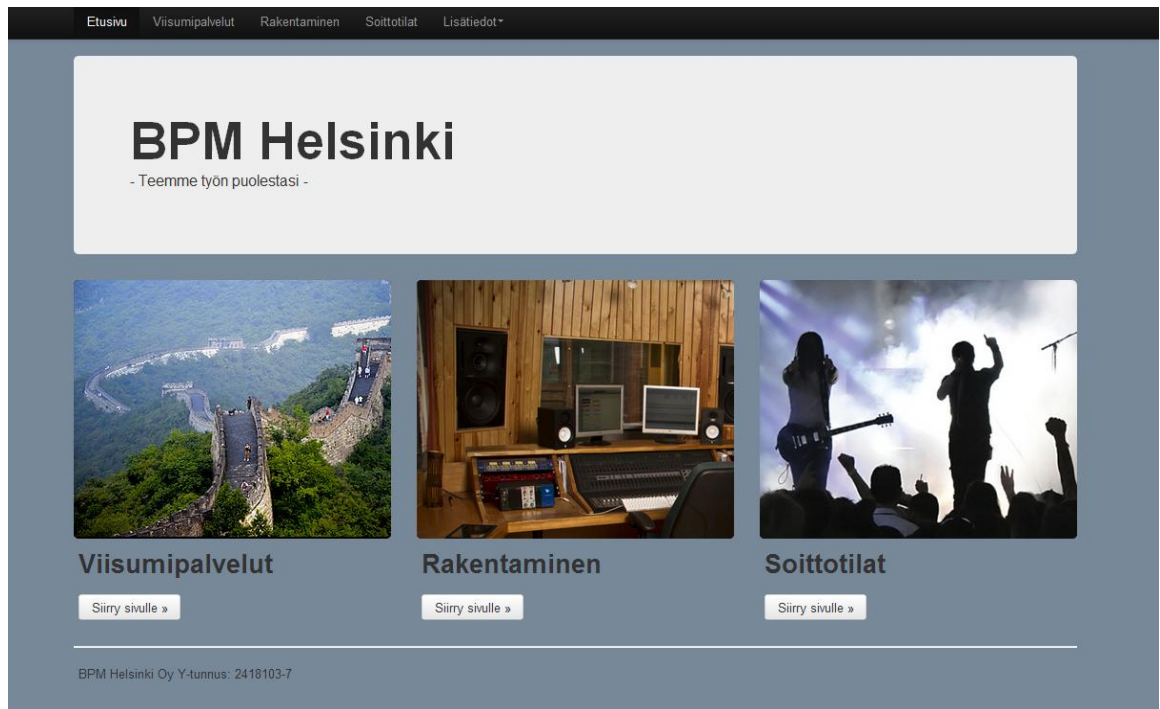
```

<a href="viisumipalvelut.html">
  <div class="span4">
    
    <h2 class="indexA">Viisumipalvelut</h2>
  </div>
</a>

```

Kuvio 24. A-elementti käärittynä div-, img- ja a-elementtien ympärille.

Lähetin kuvan valmiista etusivusta (kuvio 25) yritykseen, sekä maininnan, että heidän hyväksyessä etusivun, hyödyntäisin pohjaa myös muiden sivujen osalta. Etusivuun oltiin tyytyväisiä, ja siirryin työstämään sivupohjaa muita sivuja varten.



Kuvio 25. Luonnos BPM Helsinki-sivuston etusivusta.

Etusivun muuntamisessa muiden sivujen pohjaksi tein siitä kopion ja muokkasin tarpeellisia osia. Ensin poistin keskiosan linkkiosuuden kuvineen ja lisäsin tilalle Bootstrapin valmiit muotoillut div-elementit. Koska aiemmassa sivustossa olleet tiedot olivat omia erillisiä kokonaisuuksiaan, päädyin lisäämään article-elementin (kuvio 26).

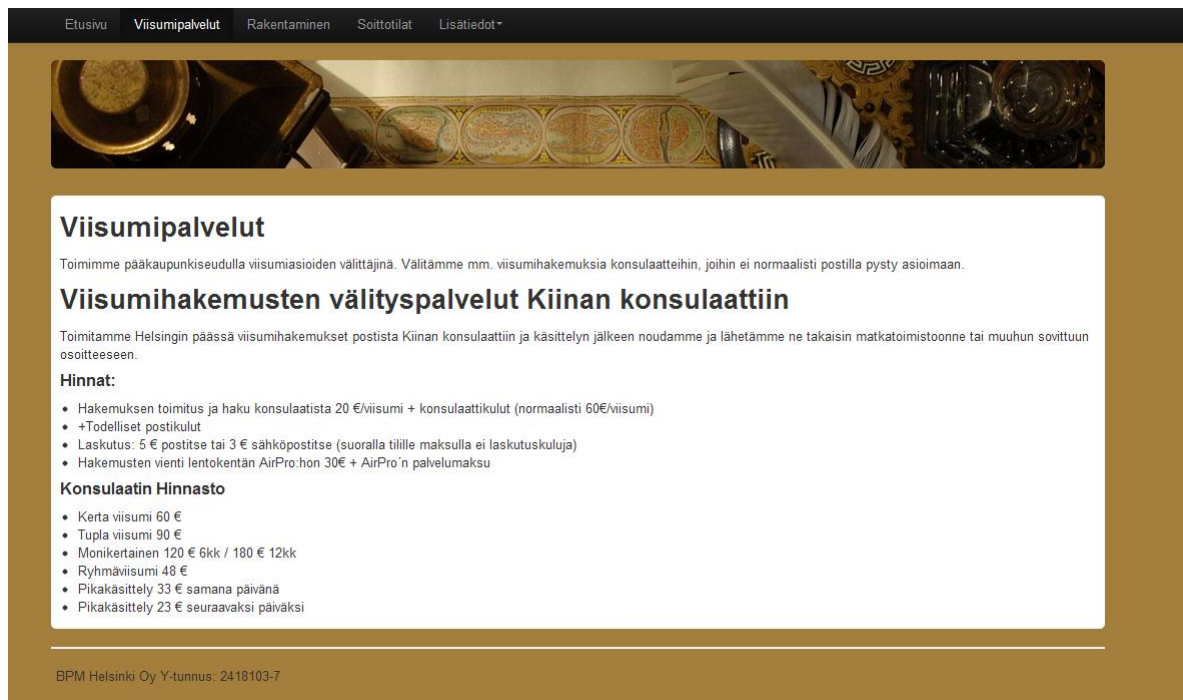
```

<div class="container">
  <div class="row-fluid">
    <div class="span12">
      <article>
        <h2>Viisumipalvelut</h2>
        <p>
          Toimimme pääkaupunkiseudulla viisumiasioiden välittäjinä. Välitämme
          mm. viisumihakemuksia konsulaatteihin, joihin ei normaalisti postilla
          pysty asioimaan.
        </p>
        ..
      </article>
    </div>
  </div>
</div>

```

Kuvio 26. Article-elementin lisääminen sivulle.

Näin sain pienin muutoksin valmiiksi pohjan, jota voitaisiin käyttää muille sivuille. Sivupohjaan voi myös lisätä div-elementtejä tarpeen mukaan (kuvio 27).



Kuvio 27. Luonnos sivupohjasta loppuile sivuille.

### 3.4 Video elementin lisääminen sivustolle

Yrityksen uuden sivuston toivottiin hyödyntävän paremmin videoita sekä kuvia. Niiden tarkoituksena on paitsi parantaa sivuston yleisilmettä myös auttaa yritystä markkinoimaan tuotteitaan tehokkaammin. Tutkittuani HTML5:n tarjoamaa natiivia video-tukea, päätin hyödyntää sivustolla videota, jonka kontrollointi on mahdollista erillisten painikkeiden avulla. Painikkeiden tehtävä on tehdä videosta helppokäyttöisempi, sekä mahdollistaa parempi visuaalinen ilme myöhemmässä vaiheessa. Myöhemmässä vaiheessa painikkeet voisivat esittää esimerkiksi eri soittimia (kuvio 28). Ajatuksena on, että videon toistaminen toimisi mahdollisimman monissa eri ympäristöissä ja selaimissa. Koska varasisältöön varautuminen voidaan tehdä eri tavoin, päädyin ratkaisuun, jossa toistamisen epäonnistuttua olisi videon paikalla kuva. Lisäksi tapauksessa, jolloin erillisiä painikkeita ei voitaisi tukea, niitä ei myöskään näytettäisi.



Kuvio 28. Video-painikkeiden hahmottelua

Päätin lähteä liikkeelle video-tiedostojen muuntamisesta. Käytössäni oli video, joka oli mp4-muodossa. Mp4-tiedoston kattaessa vain osan selainten tukemista formaateista täytyi se lisäksi muuntaa ogg- ja webm-muotoon. Käytin muuntamiseen apuna Aura Video Converter- ohjelmistoa. Mp4, ogg ja webm-formaatit pystyisivät toimimaan natiivissa muodossa, jolloin erillisten painikkeiden pitäisi toimia. Lisäksi mp4-tiedosto tulisi toimimaan varasisältönä, jota toistettaisiin Flash Playerin avulla. Siinä tapauksessa, että mikään yllämainituista ei toimisi, näytettäisiin videon tilalla kuva, sekä linkki, josta video tiedosto olisi ladattavissa.

Seuraavaksi päätin keskittyä videon onnistuneeseen liittämiseen sivustolle. Käyttämässäni (Korpela 2011) kirjassa ei ole Flash Playerin avulla toimivaa varasisältöä esitelty, joten päädyin tutkimaan asiaa internetistä. W3school.com sivuston esimerkeissä varasisältönä toimii ainoastaa muistutus tuen puuttumisesta. Ensimmäisten etsimisten jälkeen päädyin valitsemaan Microsoftin kehittäjille tarkoitettulta MSDN Library sivuston suosittamaa valmista koodia, joka tukisi kaikkia video formaatteja, sekä lisäksi toistaisi swf-tiedostoa varasisältönä. Adding an HTML5 video control to your webpage- sivulta löydetty koodi näytti lupaavalta, mutta osoittautui puutteelliseksi, sekä lisäksi sisälsi kirjoitus virheitä. Kirjoitusvirheiden korjaamisen jälkeen selain näytti videon onnistuneesti kaikilla selaimilla. Ongelmana oli kuitenkin vanhempien selainten Flash-tuki, joka ei mahdollistanut videolle minkäänlaista kontrollointi mahdollisuutta. Käytettäessä vanhempia selaimia video käynnistyi automaattisesti sivulle mentäessä, eikä sitä ollut mahdollista pysäyttää tai kontrolloida muutoin.

Lähdin tutkimaan asiaa internetistä ja sain selville, että vanhemmat IE-selaimet tarvitsevat erillisen video toistamisen tuen esimerkiksi Flowplayerin. Flowplayerin sivuilta on ladattavissa valmis soitin, joka voidaan lisätä tarvittavalle palvelimelle. Tämän jälkeen tarvitsee opiskella tarvittavat dokumentaatiot, joista selviää mitä lisäyksiä tarvitsee tehdä head- osioon, ja mitä body- osioon. Koska tämä vaikutti turhankin vaikeaselkoiselta tavalta jatkoin etsimistä. Lopulta löysin Video for Everybody sivustolta kääntäjän, joka muuntaa valmiin koodin HTML ja Flash tuella (kuvio 29).



```

<!-- "Video For Everybody" http://camendesign.com/code/video_for_everybody -->
<video controls="controls" poster="pictures/soittotilatEsittely.jpg" width="640" height="360">
  <source src="videos/soittotilatEsittely.mp4" type="video/mp4" />
  <source src="videos/soittotilatEsittely.webm" type="video/webm" />
  <source src="videos/soittotilatEsittely.ogv" type="video/ogg" />
  <object type="application/x-shockwave-flash" data="player.swf"
    width="640" height="360">
    <param name="movie" value="player.swf" />
    <param name="allowFullScreen" value="true" />
    <param name="wmode" value="transparent" />
    <param name="flashVars" value="config={'playlist':['pictures%2FsoittotilatEsittely.jpg',
      {'url':'videos%2FsoittotilatEsittely.mp4','autoPlay':false}]}"/>
    
  </object>
</video>
<p>
  <strong>Lataa video:</strong> <a href="videos/soittotilatEsittely.mp4">MP4 muodossa</a>
  <a href="videos/soittotilatEsittely.ogv">Ogg muodossa</a> | <a href="videos/soittotilatEsittely.webm">WebM muodossa</a>
</p>

```

Kuvio 29. Videon liittäminen soittotilat sivulle Video for Everybody generator sivua hyödyntäen. (Camen Design 2012.)

Kääntäjän tekemä koodi osoittautui toimivaksi ja sen ainoa heikkous oli sen käyttämä soitin, joka toimii erillisestä internet osoitteesta. Tämä aiheuttaisi, joskin harvinaisissa tapauksissa, videon toistamisen epäonnistumisen. Päädyin käyttämään Video for Everybodyn sivuston ehdottamaa JW Player-tiedostoa, jonka voi ladata omalle palvelimelle. Ainoaksi tehtäväksi koodin osalta jäi tarvittavien osien suomentaminen, sekä soittimen viittauksen muuttaminen (kuva 333).

Videon toimiessa tarkoituksen mukaisesti on seuraava vaihe lisätä sivulle painikkeet, sekä niiden toimintaan tarvittava JavaScript tiedosto. Painikkeina toimivat HTML:n normaalit painikkeet, joiden ulkoasua on paranneltu Bootstrapin avulla. Lisäksi painikkeille on määritetty onclick-toiminnot, jotka toteuttavat halutut JavaScript funktiot (Kuvio 30).

```

<button onclick="vidplay()" class="btn btn-success" id="play">Toista video</button>
<button class="btn btn-inverse" id="restart" onclick="restart();">Aloita alusta</button>

```

Kuvio 30. HTML-painikkeiden lisääminen sivulle.

”Toista video”-painike on yhteydessä vidplay()-funktioon, joka ohjaa videon käynnistä ja pysäytä toimintoja. Funktion toiminto perustuu yksinkertaiseen if-else rakenteeseen, jonka avulla tarkistetaan videon sen hetkinen tila. Funktion alussa luodaan tarvittavat video ja painike elementit, jotka saadaan valituiksi käyttämällä apuna sivun document-olion metodia getElementById.

Kun tarvittavat elementit on luotu voidaan videon elementin DOM-ominaisuuksia hyödyntämällä tutkia onko video pysäytetty. Jos video on pysäytetty, kutsutaan videon `play()` (toista) funktiota videon käynnistämiseen. Tämän jälkeen hyödynnetään painikkeen `textContent`-ominaisuutta, jonka avulla voidaan sisällyttää sivulla olevan painikkeen tekstisisällöksi ”Pysäytä video”. Siinä tapauksessa, että video ei ollut pysäytettynä kutsutaan videon `pause()` (pysäytä) funktiota, jolloin video saadaan pysäytetyksi. Lisäksi painikkeen sisällöksi muutetaan ”Toista video” (kuvio 31).

```
function vidplay() {
    var video = document.getElementById("video1soittotilat");
    var button = document.getElementById("play");
    if (video.paused) {
        video.play();
        button.textContent = "Pysäytä video";
    } else {
        video.pause();
        button.textContent = "Toista video";
    }
}
```

Kuvio 31. Videon toistaminen JavaScript funktion avulla.

Toisena painikkeena toimii ”Aloita alusta”-painike, joka käynnistää `restart()`-funktion. `Restart()`-funktio hyödyntää samaa `getElementById`-metodia, sekä lisäksi videon `currentTime`-funktioita. Asettamalla `currentTime` arvoksi 0, siirtyy videon toisto alkuun (kuvio 32).

```
function restart() {
    var video = document.getElementById("video1soittotilat");
    video.currentTime = 0;
}
```

Kuvio 32. Videon toistamisen aloittaminen alusta JavaScript funktion avulla.

Videon toimiessa ulkoisten painikkeiden avustamina on viimeinen tehtävä niiden hallitseminen eri tilanteissa. Tilanteessa, jolloin Moderniz tunnistaa HTML5:n mukaisen videon tuen, poistetaan käytöstä videolle määritetyt erilliset kontrollit.

Vastaavasti tilanteessa, jossa varasisältönä toimiva Flash-soitin otetaan käyttöön, poistetaan tarpeettomat painikkeet näkyvistä (kuvio 33).



Kuvio 33. Video painikkeiden ja kontrollien toiminta eri tilanteissa.

Lisäämällä sivun HTML-koodiin painikkeiden alapuolelle Modernizrin mukaisen JavaScript kyselyn, voidaan video-tuki tunnistaa. Tuen löytyessä JavaScript suorittaa `removeVideoControls()`-funktio, joka poistaa erilliset kontrollit. Tuen puuttuessa suoritetaan vastaavasti `removeVideoButtons()`-funktio, joka poistaa erilliset painikkeet (kuvio 34).

```
<script>
  if (Modernizr.video)
    removeVideoControls ();
  else
    removeVideoButtons ();
</script>
```

Kuvio 34. HTML5-videotuen tunnistaminen Modernizrin avulla.

Video kontrollien poistamiseen käytettävä `removeVideoControls()`-funktio luo ensin tarvittavan video elementin. Tämän jälkeen voidaan videosta poistaa sen kontrolli-ominaisuus video elementin `removeAttribute`-funktion avulla (Kuva 35).

```
function removeVideoControls() {  
    var video = document.getElementById("video1Soittotilat");  
    video.removeAttribute('controls');  
}
```

Kuvio 35. Videon kontrolli ominaisuuden poistaminen JavaScriptin avulla.

Video-kontrollien tapauksessa poistettavia ominaisuuksia oli ainoastaan yksi. Video-painikkeiden osalta poistettavia elementtejä on kaksi, sekä tulevaisuudessa varmasti useampia. Jotta useampien elementtien poistaminen olisi tulevaisuudessa helpompaa, hyödynsin video-painikkeiden poistamisessa silmukkaa, sekä DOMin puurakennetta.

Luomalla for-silmukan `removeVideoButtons()`-funktion sisälle pystyn poistamaan tarvittavat kaksi painiketta. Videon ollessa sisällytetty oman div-elementin sisälle pystyn valitsemaan tarvittavan divin. Tämän tarvitsee ainoastaan kohdentaa tarvittava painike div-elementin sisältä, joka onnistuu käyttämällä apuna `getElementsByTagName`-metodia. `getElementsByTagName`-metodi valitsee halutun elementin sen tyyppin perusteella, joka on tässä tapauksessa painike. Lisäämällä loppuun määrite 0, pystyn valitsemaan ensimmäisen painikkeen, joka löytyy. Painikkeen saa lopuksi poistettua hyödyntämällä valitun div-elementin `removeChild`-metodia (kuvio 36).

```
function removeVideoButtons() {  
    for (i=0; i<2; i++)  
    {  
        var videoDiv=document.getElementById('divVideo1Soittotilat');  
        var videoDivButton=videoDiv.getElementsByTagName('button')[0];  
        videoDiv.removeChild(videoDivButton);  
    }  
}
```

Kuvio 36. Video-painikkeiden poistaminen JavaScript funktion avulla.

## 4 Yhteenveto ja johtopäätökset

Opinnäytetyöni aihe tuli BPM Helsinki Oy:n tarpeesta uudistaa internet-sivusto. Yrityksen sivuston uudistaminen on pitkäkestoinen projekti, jossa opinnäytetyön tavoitteeksi otettiin toimivan pohjaratkaisun tekeminen sivustolle hyödyntäen uusia HTML5-tekniikoita. HTML5-tekniikoiden voitiin katsoa hyödyntävän sivustoa tulevaisuudessa. Toimivassa pohjaratkaisussa olivat etusijalla sivupohjien, kuten etusivun, sekä videon lisääminen sivustolle.

Opinnäytetyön tietoperustassa käydään läpi paitsi HTML5:n uusia ominaisuuksia, myös syvennyttään HTML5-käsitteeseen. Tietoperustassa on pyritty huomioimaan sivuston kehittäjän näkökulma uusia sivuja kehittäessä.

Empriisissä osuudessa keskitytään sivustoprojektin toteuttamiseen. Sivustoprojektiin kuuluu nykyisen sivuston hyödyntämisen arviointi, uusien sivupohjien rakentaminen vastaamaan HTML5 mukaisia vaatimuksia. Lopuksi sivustolle lisätään HTML5:n mukainen video.

Sivustoprojektia tehtäessä oli ensimmäinen asia kerätä riittävä määrä tietoa ja ymmärrystä HTML5:stä. Koska HTML5 ei ole vielä standardi on siihen liittyvää kirjallista materiaalia rajatusti saatavilla. Käyttämäni J. Korpelan kirja antoi hyvää yleistietoa aiheesta, erityisesti liittyen vanhojen selainten tukeen. Kirjan lisäksi tiedon kerääminen koostui internet-lähteistä. Kerätty tietoperusta toimi hyvänä pohjana sivustoprojektissa. Ainoan poikkeuksen tekee sivustolle lisätty video, joka ei toiminut odotetulla tavalla, vaan vaati lisätutkimuksia eri lähteistä.

Empiirisen osan alku sujui hyvin. Asiat, kuten vanhan sivuston korvaaminen uudella, sekä Bootstrap-sivupohjan hyödyntäminen olivat toimivia ratkaisuja. Videon lisääminen oli empirian osalta vaativin osuus. Videon lisääminen vaati paitsi JavaScriptin opettelua myös DOM:n ymmärtämistä. Lisäksi videota piti testata eri selaimilla.

HTML5:n historian sekä käsitteen kokonaisuuden ymmärtäminen vei opinnäytetyöstä paljon aikaa, mutta samalla selkiytti paitsi HTML5:ttä myös asioita, joihin olin aiemmin törmännyt sivustoja kehittäessäni, mutta joiden perimmäiä syitä en ymmärtänyt.

Kaiken kaikkiaan HTML5 tarjosi mielenkiintoisen näkökulman sen tarjoamista mahdollisuuksista sekä haasteista. Itse olen tyytyväinen HTML5:n perusajatuksista, jossa yhdistetään voimavaroja, tässä tapauksessa eri web-tekniikoita. Oman lisänsä tuovat uudet toiminnot, kuten natiivi video-tuki sekä semanttiset elementit. Haasteita HTML5:llä on yhä selainten tarjoamassa tuessa, oli kyseessä video-formaatin muoto tai uusien toimintojen tuki. Hyvänä esimerkkinä toimii HTML5:n uusi video-tuki. On hyvä asia, ettei tarvita erillistä Flash-soitinta, mutta huono asia, että selainten valmistajat eivät ole päässeet yhteisymmärrykseen yhtenäisestä formaatista. Videon tarvitsemat 3 eri formaattia vievät paljon tilaa palvelimelta. Lisäksi videon muuntaminen on työläs operaatio.

Sivuston tukiessa uusia HTML5-tekniikoita ovat jatkokehitysmahdollisuudet lähes rajattomat. HTML5-sisältää paljon uusia tekniikoita, joihin en ehtinyt tarkemmin tutustua. Lisäksi nykyinen video-tuki sisältää paljon erilaisia toimintoja, joilla videon näytävyyttä voidaan parantaa.

Sivustoprojektin toteuttaminen yrityksen omistajien kanssa sujui aikataulun mukaisesti, ja sivustoprojektin tuloksiin oltiin tyytyväisiä. Erityisesti erillinen etusivu sai kiitosta. Olin myös itse tyytyväinen saavutettuun tavoitteeseen ja opittuihin asioihin. Sivustoprojektin myötä opin ymmärtämään HTML5:n monimuotoisuuden lisäksi sen mahdollisuuksia tulevaisuudessa.

## Lähteet

Camden Design 2012. Video for Everybody!. 2010. Luettavissa:

[http://camendesign.com/code/video\\_for\\_everybody](http://camendesign.com/code/video_for_everybody). Luettu: 02.11.2012.

Cawrey, D. 2010. HTML5 Standards Author Ian Hickson. Luettavissa:

<http://www.thechromesource.com/interview-html5-standards-author-ian-hickson>.

Luettu: 3.10.2012.

Cotton, P. 2012. Getting HTML5 to Recommendation in 2014. Luettavissa:

<http://lists.w3.org/Archives/Public/public-html/2012Sep/0243.html>. Luettu:

4.10.2012.

Decentralized Information Group 2006. Reinventing HTML. Luettavissa:

<http://dig.csail.mit.edu/breadcrumbs/node/166>. Luettu: 3.10.2012.

Elliot, I. 2012. Which HTML5? – WHATWG and W3C Split. Luettavissa:

<http://www.i-programmer.info/news/191-htmlcss/4531-which-html5-whatwg-and-w3c-split.html>. Luettu: 4.10.2012.

Fiegerman, S. 2012. Google Chrome Just Passed Internet Explorer To Become The World's Most Popular Web Browser. Luettavissa:

<http://www.businessinsider.com/google-overtakes-internet-explorer-as-most-popular-browser-2012-5>. Luettu: 7.10.2012.

HTML5-KIRJA 2012. Twitter Bootstrap. Luettavissa:

<http://html5kirja.fi/2012/08/22/twitter-bootstrap>. Luettu: 09.11.2012.

Irish, P. & Manian, D 2012. HTML5 & CSS3 Readiness. Luettavissa:

<http://html5readiness.com/#ray-12>. Luettu: 5.10.2012.

Jobs, S. 2010. Thoughts on flash. Luettavissa:

<http://www.apple.com/hotnews/thoughts-on-flash>. Luettu: 3.10.2012.

Koskinen, D. 2010. HTML, CSS ja Javascript. Luettavissa:  
<http://danielkoskinen.com/johdanto-html-css-ja-javascript>. Luettu: 16.10.2012.

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

Martin, K. 2011. W3Fools – a foolish attack on W3School.com. Luettavissa:  
<http://aspektas.com/blog/w3fools-a-foolish-attack-on-w3schoolscom>. Luettu:  
5.10.2012.

Marshall, G. 2011. HTML5: what is it?. Luettavissa:  
<http://www.techradar.com/news/internet/web/html5-what-is-it-1047393>. Luettu:  
5.10.2012

Microsoft 2012a. Using JavaScript to control the HTML5 video player. Luettavissa:  
[http://msdn.microsoft.com/en-us/library/hh924823\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/hh924823(v=vs.85).aspx). Luettu:  
31.10.2012.

Microsoft 2012b. Adding an HTML5 video control to your webpage. Luettavissa:  
[http://msdn.microsoft.com/en-us/library/hh924820\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/hh924820(v=vs.85).aspx). Luettu:  
31.10.2012.

Modernizr 2012. Documentation. Luettavissa:  
<http://modernizr.com/docs/#installing>. Luettu: 18.10.2012.

Niemi, P. 2012. HTML5 ja JavaScript – kahvipöydässä kuhisee. Luettavissa:  
<http://tieturi.wordpress.com/2012/08/16/html5-ja-javascript-kahvipoydassa-kuhisee>.  
Luettu: 5.10.2012.

Ohjelmointiputka 2007. JavaScript-perusopas: Osa 1 – Perusteet. Luettavissa:  
[http://www.ohjelmointiputka.net/oppaat/opas.php?tunnus=js\\_01](http://www.ohjelmointiputka.net/oppaat/opas.php?tunnus=js_01). Luettu: 5.10.2012.



Ohjelmointiputka 2008. Document Object Model veppikoodaajan näkökulmasta. Luettavissa: <http://www.ohjelmointiputka.net/oppaat/opas.php?tunnus=dom>. Luettu: 16.10.2012.

Ohjelmointiputka 2012. Tervetuloa ohjelmointiputkaan. Luettavissa: <http://www.ohjelmointiputka.net>. Luettu: 5.10.2012.

Orgera, S 2012. What Is A Web Browser?. Luettavissa: <http://browsers.about.com/od/howbrowserswork/a/whatisabrowser.htm>. Luettu: 6.10.2012.

Ryan, T. 2012. Using Semantic HTML. Luettavissa: <http://www.adobe.com/devnet/html5/articles/semantic-markup.html>. Luettu: 17.10.2012.

StatCounter GlobalStat 2012. Top 12 Browser Versions on May 2012. Luettavissa: [http://gs.statcounter.com/#browser\\_version-ww-monthly-201205-201205-bar](http://gs.statcounter.com/#browser_version-ww-monthly-201205-201205-bar). Luettu: 17.10.2012.

Stucken, A. 2010. What is Flash. Luettavissa: <http://www.bbc.co.uk/webwise/guides/about-flash>. Luettu: 31.10.2012.

Vanhala-Nurmi, V. 2012a. Rakenne ja merkkäminen. Luettavissa: [http://myy.helia.fi/~vanvu/www/html5\\_rakenne.htm](http://myy.helia.fi/~vanvu/www/html5_rakenne.htm). Luettu: 17.10.2012.

Vanhala-Nurmi, V. 2012b. Uudet rakenne elementit. Luettavissa: [http://myy.helia.fi/~vanvu/www/html5\\_uudetom.htm](http://myy.helia.fi/~vanvu/www/html5_uudetom.htm). Luettu: 18.10.2012.

WHATWG 2012. FAQ. Luettavissa: [http://wiki.whatwg.org/wiki/FAQ#What\\_is\\_the\\_WHATWG.3F](http://wiki.whatwg.org/wiki/FAQ#What_is_the_WHATWG.3F). Luettu: 2.10.2012.

W3C 2012a. Facts about W3C. Luettavissa:

<http://www.w3.org/Consortium/facts#history>. Luettu: 2.10.2012.

W3C 2012b. W3C Mission. Luettavissa:

<http://www.w3.org/Consortium/mission.html>. Luettu: 2.10.2012.

W3C 2012c. Standards FAQ. Luettavissa: <http://www.w3.org/standards/faq#std>.

Luettu: 16.10.2012.

W3Schools 2012a. HTML5 DOM Introduction. Luettavissa:

[http://www.w3schools.com/html/dom/dom\\_intro.asp](http://www.w3schools.com/html/dom/dom_intro.asp). Luettu: 5.10.2012.

W3Schools 2012b. HTML5 Introduction. Luettavissa:

[http://www.w3schools.com/html/html5\\_intro.asp](http://www.w3schools.com/html/html5_intro.asp). Luettu: 5.10.2012.

W3Schools 2012c. HTML5 New Elements. Luettavissa:

[http://www.w3schools.com/html/html5\\_new\\_elements.asp](http://www.w3schools.com/html/html5_new_elements.asp). Luettu: 5.10.2012.

W3Schools 2012d. HTML5 Video. Luettavissa:

[http://www.w3schools.com/html/html5\\_video.asp](http://www.w3schools.com/html/html5_video.asp). Luettu: 7.10. 2012.

Youtube 2009. Introduction to the Document Object Model. Luettavissa:

<http://www.youtube.com/watch?v=-0ZcldkGlt8>. Luettu: 16.10.2012.

# Liitteet

## Liite 1. Projektisuunnitelma

### Tehtävä ja taustat

Helsinkiläinen yritys BPM Helsinki haluaa uudistaa internet sivustonsa. Sivuston uudistamisen lähtökohtana on uudistaa sivustoa niin, että se vastaa paremmin heidän tarpeitaan. Sivuston uudistamisen toteuttaa Haaga-Helian opiskelija opinnäytetyönään.

Uudistuksen tekniseksi lähtökohdaksi otettiin uusi HTML5-teknologia, joka sisältää jo nykyisellään paljon toimintoja ja sitä voidaan hyödyntää myös tulevaisuudessa.

Sivuston lopullinen valmistumis-ajankohta ei ole vielä tiedossa, koska yritys haluaa vielä kartoittaa asiaa, ja kerätä tarvittavia materiaaleja, kuten kuvia ja tekstejä. Lisäksi sivustoon käytettävä teknologia on sivuston tekijälle uutta. Sivuston osalta päädyttiin projektiin, jossa ensi sijainen tavoite tulee olemaan sivuston perusrakenteen sekä uuden video-tuen sisällyttäminen sivustoon.

Ajatus uusien HTML5-tekniikoiden hyödyntämisestä lähti omasta kiinnostuksestani web-tekniikoita kohtaan. Olin jo aiemmin kuullut puhuttavan HTML5:stä sekä lukenut siihen liittyvistä tekniikoista W3Schools-sivustolta. En ollut kuitenkaan koskaan kokeillut tekniikoita käytännössä.

Projektin aikatauluksi otettiin syksy 2012. Projektin alkuvaiheessa käydään läpi ajatuksia ja ehdotuksia puhelimitse sekä skypeen välityksellä. Tarvittaessa käytetään myös apuna sähköpostia, sekä pidetään yhteisiä neuvotteluja.

Alkuvaiheen neuvottelujen jälkeen sivuston tekijä käy läpi nykyistä sivustoa, jolloin voidaan päättää sivuston mahdollinen hyödyntäminen uutta sivustoa varten. Seuraava kokonaisuus on uuden sivurakenteen luominen. Yrityksellä on tällä hetkellä toiveena

saada uuteen sivustoon erillinen etusivu, josta olisi pääsy muihin sivuihin. Viimeinen vaihe on video-tekniikan sisällyttäminen sivustoon, sekä projektin päättäminen.

## **Tavoitteet**

Kokonaistavoite on rakentaa yrityksen kanssa neuvoteltu sivusto määräajassa käyttäen apuna uusimpia HTML5-tekniikoita. Sivuston tulisi, paitsi vastata yrityksen tarpeita, myös toimia yleisimmissä selaimissa.

Lisäksi sivuston tekijän tavoitteena on omaksua kattavasti uusi HTML5-teknologia, sekä ymmärtää sen merkitys ja mahdollisuudet myös tulevaisuudessa.

## **Ympäristö**

Sivustoprojekti toteutetaan sivuston tekijän omalla tietokoneella NetBeans-kehitysympäristössä. Lisäksi sivuston tekijällä on käytettävissä Nettihotelli.fi kotisivu, joka mahdollistaa sivuston esittämisen aika ajoin yrityksen omistajille.

Sivuston kehittäjällä on käytössään eri selaimia, kuten Mozilla, IE:n eri versioita ja Chrome. Tämän lisäksi sivuston toimivuutta voidaan todentaa kotisivun, sekä yrityksen omistajien omien selainten avulla.

## **Kehityssuunnitelma**

Ensimmäisessä vaiheessa käydään läpi sivuston rakennetta, kuten erillistä etusivua, sekä tutkitaan nykyistä sivustoa. Sivuston kehittäjän tärkein tehtävä on tässä vaiheessa kartoittaa uusia HTML5-teknologioita.

Toinen vaihe koostuu nykyisen sivuston muuntamisesta vastaamaan uutta HTML5-teknologiaa ja rakentaa sivuston perusta.

Kolmannessa vaiheessa keskitytään uuden etusivun, sekä muiden sivujen pohjien työstämiseen. Tässä työvaiheessa pyritään hyödyntämään sivuston tekijän kotisivua, jotta yrityksen haluama ulkoasu vastaa heidän toiveitaan.

Neljännessä vaiheessa pyritään keskittymään natiivin video-tuen lisäämiseen sivustolle. Tässä vaiheessa pyritään hyödyntämään sivuston tekijän kotisivua, jotta voidaan varmistaa eri selainten toimivuus.

Viimeisessä viidennessä vaiheessa kerään yhteen oppimiani asioita. Pyrin listaamaan onnistumiset ja epäonnistumiset. Lisäksi yrityksen kanssa pidetään viimeiset neuvottelut, joissa käydään läpi saavuttiko sivusto tavoitteensa yrityksen kannalta.

Lisäksi työn jokaisessa konkreettisessa vaiheessa pyritään testaamaan työn tulosta eri selaimilla.

## **Tarkennettu suunnitelma**

Sivustoprojektin suunnitelma ja aikataulu (syksy 2012):

- Tutustuminen HTML5-tekniikoihin, alkuneuvottelut sekä nykyiseen sivustoon tutustuminen (viikot 1-3)
- Sivuston muuntaminen HTML5:n mukaiseksi (viikko 4)
- Etusivun, sekä muiden sivupohjien työstäminen (viikot 5-6)
- HTML5-videotuen lisääminen sivustolle (viikot 7-8)

## Liite 2. Termit ja lyhenteet

HTML5 Uusin versio viimeisimmästä kehitteillä olevasta HTML- kuvauskielestä.

W3C W3C on kansainvälinen yhteisö, jonka tavoite on tehdä internetistä tasapuolisesti saavutettava ja toimiva kokonaisuus riippumatta käyttäjän kulttuurista, resursseista tai sijainnista. Sen tärkein tehtävä on kehittää kansainvälisiä internet standardeja, joiden avulla voidaan hyödyntää internetin mahdollisuuksia parhaalla mahdollisella tavalla. (W3C 2012b.)

W3C:n kehitti Tim Berners-Lee vuonna 1994. Se kehitettiin Massachusettsin teknillinen korkeakoulun (MIT) ja Euroopan ydinfysiikan tutkimusjärjestön (CERN) yhteistyönä ja sen tukijoina toimivat Yhdysvaltain puolustusministeriö (DARPA) , sekä Euroopan komissio. (W3C 2012a.)

### WHATWG

Web Hypertext Application Technology Working Group (WHATWG) ylläpitää ja kehittää HTML- teknologioita, sekä internet sovellusten ohjelmointirajapintoja. Se perustettiin 2004, jolloin yksittäiset henkilöt olivat huolissaan W3C:n suuntauksesta keskittyä XHTML:ään, jättäen HTML:n kehityksen vähäisemmälle huomiolle. HTML5:n kehittämisessä WHATWG on hyödyntänyt teknologioita, kuten Web Forms 2.0 ja Web Controls 1.0. Molemmat tekniikat ovat WHATWG:n kehittämiä, joskin Web Controls 1.0 on päätetty toistaiseksi hylätä. (W3C 2012c.)

Web 2.0 Nimitys visiolle uudesta web- kehityksestä, jossa kokonaisia sovelluksia voitaisiin hallinnoida selaimilla

### Web Applications 1.0

WHATWGN:n kehittämä luonnos, jonka pyrkimys oli kehittää HTML-kieltä, sekä web applikaatioita.

## Web-sovellus

Web- käyttöliittymällä toimiva sovellus

HTML Hypertekstin merkintäkieli, jolla kuvataan hyperlinkkejä sisältävää tekstiä.

CSS Cascading Style Sheet- tyyliohje on yksinkertainen mekanismi, jolla voidaan liittää HTML- dokumenttien elementteihin eri ominaisuuksia. HTML5 mahdollistaa CSS- tiedoston liittämisen dokumenttiin aikaisempaa helpommalla tavalla. W3C:n uusin luonnos CSS:stä on nimeltään CSS3.

## Flash Player

Web- selaimen asennettava liitännäinen, joka mahdollistaa video- ja animaatio toiston selaimessa. Tutkimusten mukaan Flash löytyy asennettuna n. 99% selaimista. (Stucken 2010.)

## Ohjelmointiputka

Suomalainen ohjelmointiin liittyvä sivusto, joka tarjoaa oppaita, koodivinkkejä sekä keskustelualueen. (Ohjelmointiputka 2012.)

## W3Schools.com

W3Schools.com sivusto tarjoaa laajan tarjonnan tutoriaaleja eri web- tekniikoista. Sivusto on hyvin suosittu web- kehittäjien keskuudessa ja se esiintyykin useinmiten ensimmäisinä hakukoneiden tuloksissa. Vaikkakin sivusto on hyvin suosittu on sitä myös syytetty epätarkoista tulkinnoista. Syytösten takana on joukko kehittäjiä, jotka ovat listanneet ongelmia sivutolle nimeltä W3Fools.com. Lisäksi W3Fools.com sivusto pitää sivuillaan yhä listaa virheistä, joita W3Schools.com sivusto on päivittänyt virheiden korjaamiseksi. (Martin 2011.)

DOM DOMin avulla pystytään kuvaamaan dokumenttia rakenteisena oliona, jolla on ominaisuuksia. DOM on itsessään käsitteellinen, ja usein se kuvataan puurakennekuvana dokumentista. (Korpela 2011, 42.)

DOMin ominaisuuksien avulla pystytään muuttamaan dynaamisesti HTML- elementtien ominaisuuksia. Turvallisuussyistä selaimiin ja sen asetuksiin voidaan kuitenkin määrittää rajoituksia, jolloin elementit voivat olla esimerkiksi vain lukukelpoisia. (Korpela 2011, 44.)

JavaScript JavaScript on oliopohjainen ohjelmointikieli, jonka avulla voidaan lisätä dynaamisia toimintoja www- sivuille. (Ohjelmointiputka 2007.)

JavaScript näyttää olevan tulevaisuudessa tärkein selainten ohjelmointikieli. Tätä tukee se, että sitä tukevat lähes kaikki selaimet. Sen tärkein tehtävä on mahdollistaa selain suorittamaan haluttuja toimintoja. Ohjelmointikielenä se on nopeasti opittava, vaikkakin sen kokonaisuuden omaksuminen voi viedä runsaasti aikaa. (Korpela 2011, 47.)

JavaScriptin voi liittää HTML- dokumenttiin eri tavoin. Suositeltavinta on kuitenkin säilyttää skriptit erillisessä tiedostossa ja viitata tiedostoon käyttäen src-määrittettä. Näin meneteltynä dokumenttiin ei esimerkiksi tarvitse lisätä erikoismerkkejä skriptiä lisättäessä. (Korpela 2011, 47.)

JavaScript onkin oikein käytettynä hyvinkin tehokasta. Sudentuoppa on kuitenkin JavaScriptin helppo syntaksi, joka antaa helpon mahdollisuuden kopioida valmista koodia saaden nopeasti aikaiseksi kohtuullisen toimivia sovelluksia. Tämä aiheuttaa helposti tilanteen, jossa virheellinen koodi toistuu useissa sovelluksissa. Nämä virheet voivat aiheuttaa ongelmia, kun koodia ajetaan uudessa kontekstissa. (Niemi 2012.)

## Verkkoselain

Verkkoselain on ohjelmisto, joka vastaanottaa koodia web- serveriltä pääosin käyttäjän antaman web- osoitteen avulla. Koodin voi olla



muodoltaan erilaista, mutta yleisin muoto on HTML. Tämän jälkeen selain kääntää koodin ja näyttää sen selaimen ikkunassa käyttäjälle selkeämmässä muodossa. (Orgera 2012.)

Tämän hetken käytetyimmät verkkoselaimet ovat IE, Firefox, Chrome, Safari ja Opera. Verkkoselainten välillä on ollut aina kilpailua käyttäjien suosiosta. Viime vuosina markkinaosuuksia on rohmunnut erityisesti Chrome, joka on tämän hetken suosituin selain yli 30% osuudella. (Fiegerman 2012.)

HTML5:n kehityksessä on tehty yhteistyötä eri selainten valmistajien kanssa. Selainten valmistajat ovat olleet mukana päättämässä mitä piirteitä tullaan toteuttamaan. Näin on saatu lisävarmuutta valittujen ominaisuuksien tuelle tulevaisuudessa. (Korpela 2011, 14.)

#### Web standard

W3C julkaisee eri web-teknologioihin liittyviä dokumentteja. Dokumenttien lähtökohtana on käydä läpi prosessi, jonka lopputulemana on standardoitu dokumentti. Dokumenttien julkaisemisen ja prosessin eri vaiheiden tarkoitus on pyrkiä saamaan mahdollisimman oikeudenmukainen ja yksimielinen päätös dokumentista. Prosessin lopussa W3C julkaisee suosituksia, joita voidaan pitää web-standardeina. (W3C 2012c.)

#### Modernizr

Modernizr on JavaScript kirjasto, jonka avulla voidaan tunnistaa sekä HTML5:n, että CSS3:n uusia piirteitä. Kirjasto voidaan ottaa käyttöön asentamalla modernizr.js tiedosto palvelimelle, jossa sivusto sijaitsee. Tämän jälkeen riittää, että tiedostoon viitataan sivulla. (Korpela 2011, 76)

Modernizr tiedosto tulisi sijoittaa head-osioon. Saadakseen sen toimimaan parhaalla mahdollisella tavalla se tulisi lisäksi sijoittaa CSS-tiedostojen jälkeen. (Modernizr 2012.)

## Aura Video Converter

Ohjelmisto, joka mahdollistaa eri video-formaattien kääntämisen.

## Bootstrap

Yleiskäyttöinen työkaluvalikoima web-sivujen käyttöliittymän toteuttamiseen. Tarjoaa valmiita rakennuspalikoita, esimerkiksi navigointiin tarvittavia komponentteja, sekä valmiita sivupohjia. Bootstrapin käyttöönottoon riittää tarvittavan sivupohjan tallentaminen koneelle. Sivupohjan mukana tulee tarvittavat CSS, ja JavaScript-tiedostot. Bootstrap hyödyntää jo valmiiksi joitain HTML5:n mukaisia tekniikoita ja samalla mahdollistaa niiden käytön myös tulevaisuudessa. (HTML5-KIRJA 2012.)

## Plug-in

Laajennus, jonka avulla voidaan parantaa alkuperäisen ohjelman toimintoja.

## Flow Player

Suoratoistoa hyödyntävä soitin, jonka avulla on mahdollista toistaa mm. swf- tiedostoja.

## JW Player

Suoratoistoa hyödyntävä soitin, jonka avulla on mahdollista toistaa mm. swf-tiedostoja.

## html5shiv

JavaScript- tiedosto, joka mahdollistaa HTML5:n uusien elementtien toimivuuden vanhemmissa IE- selaimissa. Koostuu yksinkertaisesta silmukasta, joka suorittaa createElement- toiminnon määrätyleisille elementeille. Tiedostoon viittaminen voidaan tehdä käyttäen apuna internetissä sijaitsevaa tiedostoa, mutta varmin tapa on ladata tiedosto valmiiksi palvelimelle, jossa sivusto sijaitsee. (Korpela 2011, 80-81)