

SELAINRIIPPUMATON WEB-TUOTANTO

Lauri Pakkala

Opinnäytetyö
Joulukuu 2012

Mediatekniikan koulutusohjelma
Tekniikka ja liikenne





Tekijä(t) PAKKALA, Lauri	Julkaisun laji Opinnäytetyö	Päivämäärä 12.12.2012
	Sivumäärä 46	Julkaisun kieli Suomi
		Verkojulkaisulupa myönnetty (X)
Työn nimi SELAINRIIPPUMATON WEB-TUOTANTO		
Koulutusohjelma Mediatekniikka		
Työn ohjaaja(t) NIEMI, Kari		
Toimeksiantaja(t) Preoni Oy		
Tiivistelmä <p>Selainriippumaton web-tuotanto tarkoittaa yksittäisten, selaimesta ja laitteesta huolimatta toimivien verkkosivustojen suunnittelua ja toteutusta. Opinnäytetyö tehtiin, koska asiakkaat osaavat yhä useammin vaatia selaimesta ja laitteesta riippumattomia sivustoja.</p> <p>Yhtenä opinnäytetyön tavoitteista oli selvittää, mitä tulee ottaa huomioon siirryttäessä kiinteän rakenteen sivustojen tuotannosta selainriippumattomiin tuotantoihin. Selvitys on tästä syystä suunnattu pääasiassa web-tuotantoihin perehtyneille henkilöille, eikä siinä siten käsitellä projektinhallintaa.</p> <p>Opinnäytetyössä perehdytetään selainriippumattomuuden perusteisiin ja annetaan ohjeet sekä laskukaavat mukautuvien ulkoasujen rakentamiseen. Sen lisäksi opinnäytetyössä käydään läpi tarvekartoitus ja tämän hetken suosituimpien laitteiden teknologiset ominaisuudet sekä niiden vaikutukset tuotannossa.</p> <p>Selainriippumattomaan web-tuotantoon on paljon ohjeistuksia, mutta opinnäytetyössä niistä on koostettu toimeksiantajan käyttämiin web-tuotantomenetelmiin sopiva kokonaisuus, jota voidaan käyttää työntekijöiden perehdytysmateriaalina.</p> <p>Opinnäytetyön tavoite täyttyi, sillä toimenpiteet, joita vaaditaan selainriippumattomaan web-tuotantoon siirtymisessä, selvitettiin onnistuneesti.</p>		
Avainsanat (asiasanat) selainriippumaton web-tuotanto, mediakyselyt, mukautuvat ruudukot, mukautuvat kuvat		
Muut tiedot		



Author(s) PAKKALA, Lauri	Type of publication Bachelor's Thesis	Date 12.12.2012
	Pages 46	Language Finnish
	Confidential ()	Permission for web publication (X) Until
Title RESPONSIVE WEB DESIGN		
Degree Programme Media Engineering		
Tutor(s) NIEMI, Kari		
Assigned by Preoni Oy		
Abstract <p>Responsive web design is a web development method where the usability of the produced site is not dependent on the device or browser. The assignment was given because the growing needs of the clientele of the assigner company were starting to make the adopting of responsive elements necessary for the assigner.</p> <p>The main goal was to find out the most important issues that should be accounted for when developing a responsive site. The thesis is aimed at people who are familiar with contemporary web design and therefore it does not cover the absolute basics, such as project management.</p> <p>The thesis instructs in the basics of responsive web design by notifying what should be done differently in the pertinent parts of the production process and teaches how to calculate proportions. It also covers the ways to analyze the need for a responsive site, the most popular devices and technologies together with their effects on the design process.</p> <p>Even though there are multiple guidance documents on responsive web design, the thesis was specifically created to suit the needs of the company. As such, it can be used as an efficient way for the employees to familiarize themselves with the topic.</p> <p>The thesis was successful, since it covers the subject of adapting web design process to accommodate flexible design methods.</p>		
Keywords responsive web design, media queries, fluid grids, flexible images		
Miscellaneous		

SISÄLTÖ

SANASTO	4
1 TYÖN LÄHTÖKOHDAT	5
1.1 Taustaa ja toimeksiantaja	5
1.2 Tehtävät ja tavoitteet	6
2 SELAINRIIPPUMATTOMAN VERKKOSIVUSTON SUUNNITTELU.....	7
2.1 Selainriippumattomuuden tarpeen kartoitus.....	7
2.1.1 Yleisesti	7
2.1.2 Käyttäjäkunta.....	8
2.1.3 Resurssit.....	9
2.2 Päätelaitteiden tekniset ominaisuudet.....	10
2.2.1 Näyttö	10
2.2.2 Verkko	12
2.3 Selainteknologiat	15
2.3.1 Yleisesti	15
2.3.2 Selainlaajennukset ja HTML5.....	15
2.3.3 Käyttöjärjestelmät	19
2.4 Ulkoasun muodostaminen	20
2.4.1 Avainasioiden sijoittelu.....	20
2.4.2 Mobiililähtöisyys	21
2.4.3 Mukautuva ruudukko	22
2.5 Suhteellisten mittojen käyttäminen	23
2.5.1 Mittatyypit	24
2.5.2 Laskukaavat	27

2.5.3	Kirjasin	28
3	SELAINRIIPPUMATTOMAN VERKKOSIVUN TOTEUTUS	30
3.1	Valmiit pohjat	30
3.1.1	Julkaisujärjestelmät	30
3.1.2	Frameworkit.....	31
3.1.3	Vanhan sivuston päivittäminen	32
3.2	Sisällön selainriippumattomuus	32
3.2.1	Muutospisteet	32
3.2.2	Kuvat	38
3.2.3	Videot.....	39
3.2.4	Interaktiivinen sisältö	40
3.3	Haasteet	40
3.3.1	Kolmannen osapuolen sisältö.....	40
3.3.2	Testaus.....	41
4	POHDINTA JA JOHTOPÄÄTÖKSET	42
	LÄHTEET	44

KUVIOT

KUVIO 1.	Tietokoneilla ja älypuhelimilla selaamisen muutos 2009 – 2012	9
KUVIO 2.	Euroopan yleisimmät älypuhelimien resoluutiot lokakuussa 2012	11
KUVIO 3.	Resoluutioiden muutos Euroopassa 2009 – 2012	12
KUVIO 4.	Selainlaajennusten yleisyyden kehitys syyskuun 2008 ja maaliskuun 2012 välillä.....	17
KUVIO 5.	Selainten arvosana HTML5-yhteensopivuudessa. 500 pisteen selain on täydellisen yhteensopiva. Huomaa, että Internet Explorerin korkein arvosana on saavutettavissa vain Windows 8-käyttöjärjestelmässä	18

KUVIO 6 Esimerkki kiinteästä 12 pylvään 960px leveästä ruudukkomallista	23
KUVIO 7. Esimerkki em-arvojen käyttäytymisestä. Kohdan 2 teksti on käytännössä kooltaan 18px ja kohdan kolme 13,5px	25
KUVIO 8. Em:n kontekstin periytyminen. Koska kohdassa 3 ei ole määritetty fonttikokoa, se perii sen kohdasta 1.	25
KUVIO 9. Prosenttiarvojen periytyminen.....	26
KUVIO 10. Esimerkit tekstirivien pituuksien vaikutuksesta luettavuuteen	29

TAULUKOT

TAULUKKO 1. Matkapuhelinteknologioiden sukupolvet	14
TAULUKKO 2. Suosituimmat verkkoselaimet kävijädatan perusteella	19
TAULUKKO 3. Media query-käskyt	34
TAULUKKO 4. Mediatyypit ja niitä käyttävät laitteet.....	36

SANASTO

HTML	Hypertext Markup Language on merkintäkieli, jolla verkkosivustot rakennetaan.
Koodipohja	Ohjelmointi- tai merkintäkielellä tuotettu yksittäinen kokonaisuus. Yhdellä sivustolla on useampi koodipohja, mikäli se sisältää erillisen rakenteen eri tarkoituksiin eli omat näkymät vaikkapa mobiililaitteille ja pöytätietokoneille.
Layout	Sivun ulkoasu ja sen rakenne. Käsittää elementtien, kuten sivupalkkien paikat ja koot, mutta ei itse sisältöä.
PPI	Pixels per inch. Yhdellä neliötuumalla olevien pikseleiden määrä.
Resoluutio	Näyttötarkkuus. Ilmoitetaan muodossa vaakasuunnan pikselit x pystysuunnan pikselit. Suurempi resoluutio ei välttämättä tarkoita suurempaa näyttöä, koska PPI vaihtelee.
Tagi	HTML-merkintäkieleessä käytettävät merkinnät, jotka määrittävät mitä HTML-käsittelymenetelmää kyseinen kohta käyttää. Käsiteltävä kohta alkaa aloitusmerkinnästä ja päättyy lopetusmerkintään.
Tietokone	Opinnäytetyössä käytetään kuvaamaan perinteistä kannettavaa tai pöytätietokonetta, jossa on näyttö, näppäimistö ja hiiri. Ei käytetä tässä yhteydessä puhuttaessa älypuhelimista tai tablet-tietokoneista.

1 TYÖN LÄHTÖKOHDAT

1.1 Taustaa ja toimeksiantaja

Preoni Oy on jyvaskyläläinen digitoimisto, joka työllistää tällä hetkellä neljä henkilöä. Yrityksen toimisto sijaitsee Jyväskylän keskustassa. Perinteisten mainostoimistopalveluiden lisäksi Preoni tarjoaa myös mobiilimarkkinointituotantoja.

Älypuhelimista alkanut mobiililaitteiden eli tietokoneisiin verrattavien kannettavien laitteiden yleistymisen on ajamassa web-tuotantoa uuteen suuntaan. Aiemmin Internetiä käytettiin lähes yksinomaan kannettavilla tai pöytätietokoneilla, jolloin näyttökokojen erot olivat niin pieniä, ettei niitä tarvinnut ottaa huomioon sivuston tuotannossa. Mobiililaitteiden yleistyttyä erot ovat kuitenkin kasvaneet niin suuriksi, etteivät ääripäiden sivustot ole kelpollisia joko käytettävyydeltään tai visuaaliselta puoleltaan toisen ääripään laitteella tarkasteltaessa.

Selainriippumattomassa web-tuotannossa sivuston sisältö ja käytettävyys pysyvät muuttumattomina päätelaitteesta ja selaimesta riippumatta. Opinnäytetyön aiheeksi se valikoitui, kun Preonilla havaittiin asiakkaiden tarpeiden olevan muuttumassa. Sisällönhallintajärjestelmän päälle rakennettuja, lähinnä suurelta ruudulta katsottaviksi tilattuja sivustoja tarkasteltiin asiakkaiden toimesta myös mobiililaitteilla. Suunnitteluvaiheissa ei kuitenkaan ollut katsottu tarpeelliseksi ottaa huomioon mobiililaitteiden pienempiä näyttökokoja, koska asiakkaat eivät olleet sitä pyytäneet. Työn tilaajat olivat kuitenkin olettaneet, että sivustot toimivat kaikilla päätelaitteilla moitteetta.

Koska asiakkaat eivät välttämättä ymmärrä laitteiden rajoituksia vasta kuin jälkeen päin, on Preoni päättänyt aloittaa siirtymän kohti selainriippumattomuutta. Kun

sivustot suunnittelee ja toteuttaa alusta asti mahdollisimman selainriippumattomiksi, on muutospyyntöihin reagointi tehokkaampaa.

1.2 Tehtävät ja tavoitteet

Opinnäytetyössä pyrittiin luomaan ohjeistus, joka helpottaisi siirtymää kiinteän kokoisten sivustojen tekemisestä selainriippumattomuuteen. Ohjeistuksessa tulisikin siten lähestyä aihetta kiinteiden sivujen tuotantoon perehtyneen henkilön näkökulmasta, mutta sisällyttää myös hyvän web-tuotannon peruseräitä.

Aluksi käsiteltävässä suunnitteluosiossa perehdytään ulkoasun muodostamisen lisäksi tarvekartoitukseen, teknologioihin ja suhteellisten mittojen laskukaavoihin. Tämän jälkeen tulevassa toteutusosiossa tarkastellaan valmiiden runkojen sopivuutta, sisällössä huomioitavia asioita sekä selainriippumattomuuden mahdollisia ongelmakohtia.

2 SELAINRIIPPUMATTOMAN VERKKOSIVUSTON SUUNNITTELU

2.1 Selainriippumattomuuden tarpeen kartoitus

2.1.1 Yleisesti

Vaikka selainriippumattomuus on yleistymässä ja sen hyödyt ovat merkittäviä, ei sitä kuitenkaan kannata käyttää kaikissa sivustoissa. Koska kyseessä on suhteellisen uusi käsite, ovat nykyiset mobiililaitteet suunniteltu toimimaan myös suurille näytöille tarkoitettujen sivustojen kanssa sisällyttämällä niihin ominaisuuksia, kuten zoomauksen, jotka mahdollistavat suuren sivuston tarkastelun pieneltä ruudulta. Tämän vuoksi on perusteltua tuottaa sivustoja myös kiinteän kokoisina, jos se katsotaan sisällön kannalta käytännöllisimmäksi ratkaisuksi. (Ewer 2012.)

Selainriippumattomuutta edeltävä ratkaisu sivustojen mobiililaitteyhteensopivuuden parantamisessa oli useamman koodipohjan sivustot, joissa käyttäjät ohjattiin päätelaitekohtaisille sivuille. Selainriippumattomuuteen liittyvän englanninkielisen termin ”responsive web design” luonut Ethan Marcotte (2011, 6) kuitenkin mainitsee, että useamman koodipohjan sivujen linkkien jakaminen voi olla ongelmallista. Jos käyttäjä jakaa mobiilikoodipohjan linkin sosiaalisessa mediassa, päätyvät myös tietokoneilla linkin avanneet mobiililaitteille tarkoitettulle sivulle. Samalla Marcotte epäilee, että mobiililaitteille erikseen tehdyt koodipohjat tulevat olemaan käyttökelvottomia, koska laitteet kehittyvät liian nopeasti ja niiden väliset erot kasvavat.

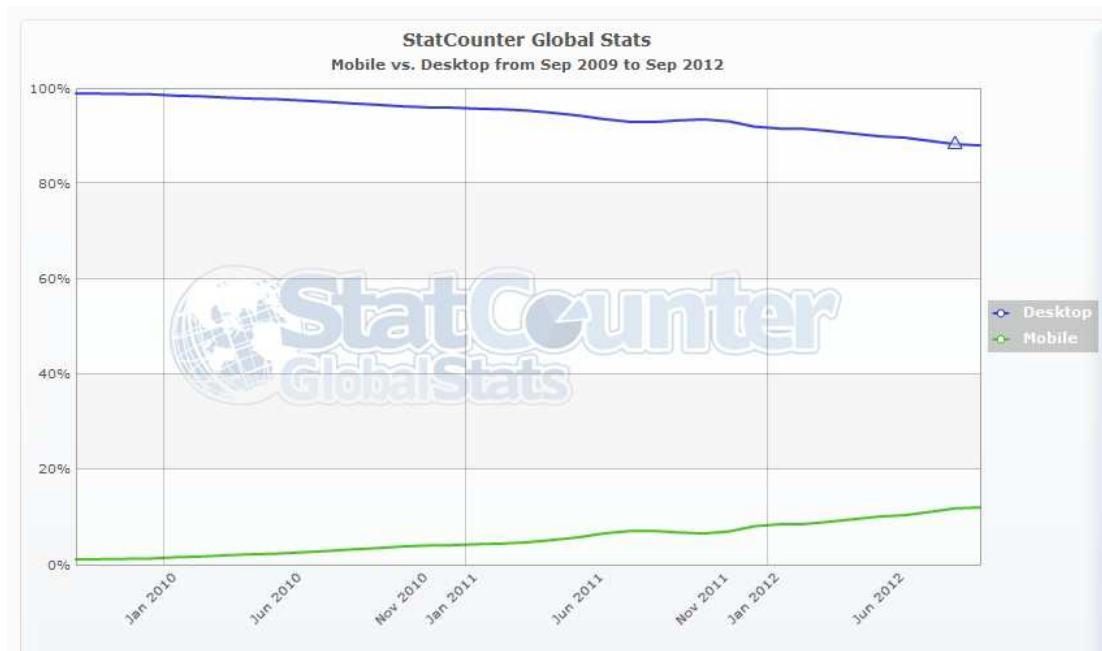
Useampi koodipohja voi kuitenkin olla eduksi, jos uskoo tietokoneilla vierailevien ja mobiililaitteella kävijöiden tarvitsevan sivustoa eri tarkoituksiin. Tietokonetta voitaisiin käyttää yritykseen tutustumiseen eli esittelykuvien, ruokalistojen tai

yrittäjähistorian katsomiseen. Mobiililaitetta käytettäisiin puolestaan asioiden nopeaan tarkistukseen liikkeessä eli aukioloaikojen tai sijaintien tarkastamiseen. (Croft 2010.)

2.1.2 Käyttäjäkunta

Selainriippumattomuuden tarve riippuu kävijöiden käyttämistä päätelaitteista. Mikäli sivuston tilaajalla on jo olemassa oleva sivusto, joka käyttää web-analytiikkaa, on tarpeen selvittäminen helpompaa. Google Analyticsin tyylistä analytiikkajärjestelmistä voi katsoa olennaisia tilastoja, kuten mobiililaitteella käyneiden kävijöiden määrä ja kävijöiden käyttämät resoluutiot.

Tarvetta kartoitettaessa kannattaa ottaa huomioon, että yli kolmen miljoonan sivuston käyttöä seuraava interaktiivinen tilastopalvelu StatCounter (2012) näyttää, että maailmanlaajuisesti älypuhelimella selauksen määrä on lähes kaksinkertaistunut syyskuun 2011 ja syyskuun 2012 välillä 6,74:stä 12,03 prosenttiin. Euroopassa muutos oli samalla aikavälillä suhteellisesti suurempi; 3,3:sta 6,64 prosenttiin. Merkittävää tässä on se, että vaikka älypuhelimilla selauksen kasvu on ollut lähes jatkuvaa StatCounterin näyttämän historian aikana (ks. kuvio 1), se on kiihtynyt viimeisen vuoden aikana.



KUVIO 1. Tietokoneilla ja älypuhelimilla selaamisen muutos 2009 – 2012 (StatCounter 2012)

2.1.3 Resurssit

Selainriippumattoman sivuston tekeminen vaatii enemmän resursseja kuin kiinteän kokoisen, koska molemmat rakennetaan pohjimmiltaan samalla tavalla, mutta selainriippumattomuudessa prosessi on monimutkaisempi. Kiinteitä sivustoja tehtäessä suunnittelu ja toteutus jaetaan erillisiksi kokonaisuuksikseen, koska sivusto on helppo suunnitella niin tarkasti, ettei siihen palaamiseen ole useimmiten tarvetta enää toteutusvaiheessa. Selainriippumattomissa tuotannoissa vaihteleva sivustorakenne tekee suunnittelusta vaikeampaa, jolloin suunnittelun ja toteutuksen kannattaa toimia yhtenä ketteränä kokonaisuutena. (Marcotte 2011, 120-121.)

Selainriippumattomuus on kuitenkin edullisin tapa tehdä sivusto, joka mukautuu päätelaitteen mukaan. Useamman koodipohjan sivustoihin täytyy sekä suunnitella että toteuttaa kunkin koon rakenne erikseen, kun taas selainriippumattomassa tuotannossa toteutuksia tarvitaan vain yksi. (Chanda 2012.)

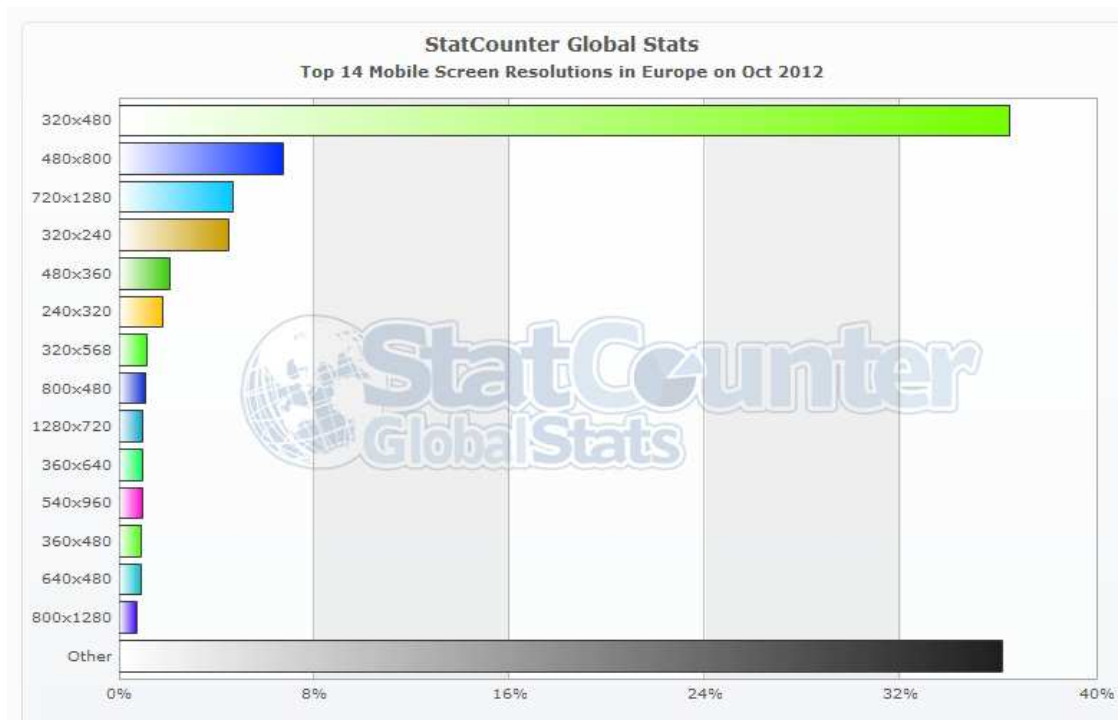
2.2 Päätelaitteiden tekniset ominaisuudet

2.2.1 Näyttö

Näytöt ovat suurin syy selainriippumattomuuden tarpeelle. Vaikka kokoerojen vaikutuksia on pyritty lieventämään zoomaustoiminnolla, zoomattua sivustoa selatessa täytyy käyttää sekä pysty- että vaakavierityspalkkia, mikä on pitkiä artikkeleita lukiessa erittäin vaivalloista. Ihanteellisesti toimiakseen sivuston täytyykin olla käytettävän laitteen ruudulle suunniteltu. (Chanda 2012.)

Älypuhelimet

Älypuhelistenkin väliset erot kuvaruudun koossa ja resoluutiassa ovat mittavia (ks. kuvio 2). Euroopan ylivoimaisesti yleisintä älypuhelimien näyttötarkkuutta 320x480 käytetään useissa vanhemmissa laitteissa, kuten ensimmäisissä iPhoneissa. Tämä ei kuitenkaan tarkoita sitä, että resoluutiot olisivat pelkästään kasvamassa - joissain vasta markkinoille tulevissa halvemmissä älypuhelinmalleissa on näytön leveydellä 240 pikseliä.



KUVIO 2. Euroopan yleisimmät älypuhelinien resoluutiot lokakuussa 2012 (StatCounter 2012)

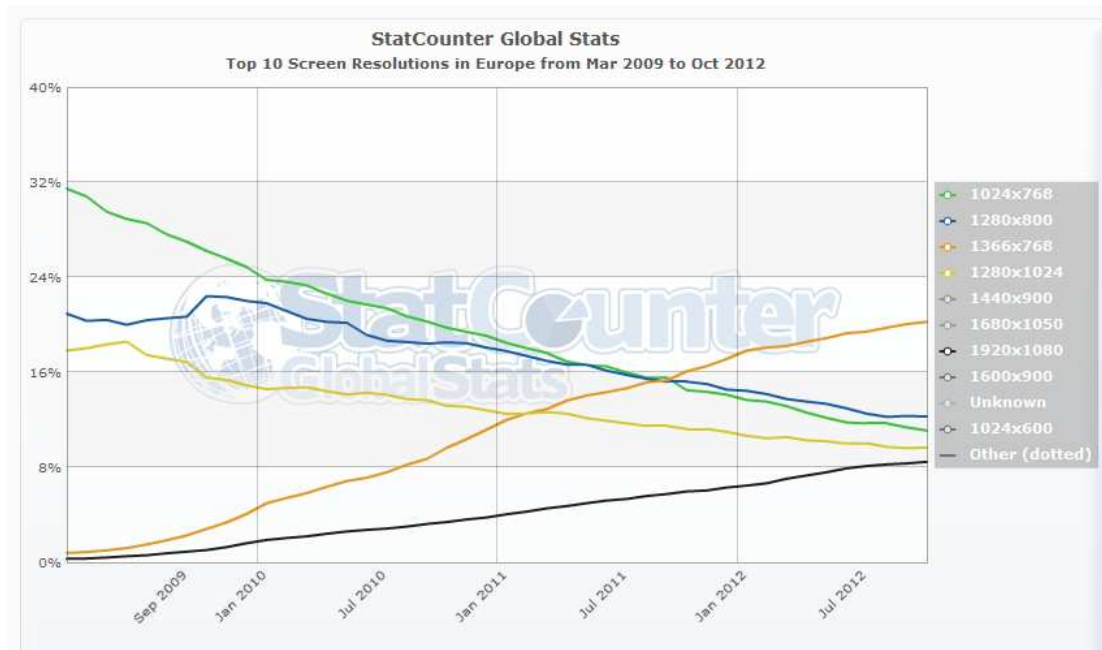
Tietokoneet ja tablet-tietokoneet

Kannettavien tietokoneiden suosio tekee niissä käytettävistä resoluutioista suosituimpia (ks. kuvio 3). Koska tablet-tietokoneissa käytetään useimmiten 1024x768 tai 1280x800 pikselin näyttötarkkuuksia, sopivat kannettaville suunnitellut sivustot myös niillä käytettäviksi. Poikkeamaa tietokoneiden resoluutioihin tuovat etenkin pöytäkoneisiin hankitut 1920 vaakapikselin näytöt.

PPI

Koska kuvaruudun fyysistä kokoa ei voi kasvattaa määrättömästi, sillä se vaikuttaa etenkin mobiililaitteiden käytännöllisyyteen, pyritään niistä tekemään mahdollisimman tarkkoja. Kahdessa tuumakooltaan yhtäläisessä näytössä voi olla eri resoluutiot, koska näyttöjen PPI-arvo eli neliötuumalla olevien pikseleiden määrä vaihtelee. Ihmissilmän erottelukyky ei kuitenkaan ole rajaton – täydellinen silmä erottaa noin 477 pikseliä tuumalta. Applen Retina-näyttöjen nimi juontuu siinä,

ettei tavallisen ihmisen pitäisi pystyä erottamaan niistä erillisiä pikseleitä 30cm päästä katsottaessa. Näin ollen ihmissilmän erottelukyvyn ylittyessä ei myöskään PPI:n kasvattamisesta ole käytännön hyötyä. (Hruska 2012.)



KUVIO 3. Resoluutioiden muutos Euroopassa 2009 – 2012 (StatCounter 2012)

2.2.2 Verkko

Verkkoteknologioiden huomioon ottaminen on erittäin tärkeää laitteisiin mukautuvaa sivustoa tehdessä, koska hitaasti latautuvan sivuston käytettävyys on heikkoa. Raskas sivusto voi vaikuttaa myös sivuston hyötysuhteeseen, sillä Googlen tekemien testien mukaan puolen sekunnin lisäviive latauksessa pudottaa kävijämäärää 20 prosentilla. (Linden 2006.)

Verkkoteknologiat

Mobiililaitteissa yleisten langattomien verkkoteknologioiden kehitystasoista puhutaan sukupolvina, koska ne ovat tulleet saataville muutamien vuosien välein. Kirjoitushetkellä teollisuusmaissa ollaan siirtymässä kolmannesta sukupolvesta eli 3G:stä neljanteen. Vanhemmat verkkosukupolvet menettävät suosionsa nopeasti: Thomas Gryta (Wall Street Journal 2012) raportoi, ettei 2G-laitteita ole enää kuin noin 12 %:lla yhdysvaltalaisen operaattori AT&T:n asiakkaista ja että yhtiö aikoo pitää toisen sukupolven verkon toiminnassa vain vuoteen 2017 saakka.

Suomessa tämän hetken nopeimpia langattomia verkkoyhteyksiä ovat LTE-verkon teoriassa jopa 100 Mb/s saavuttavat yhteydet. LTE ei neljännen sukupolven verkkona toimi vielä kovinkaan suurella alueella, mutta hallituksen tavoitteena olisi saada se kattamaan 95% väestöstä vuoteen 2016 mennessä. (Laitila 2012.)

Perttu Pitkänen (2012) raportoi, että Suomessa keskimääräinen verkkoyhteyden latausnopeus on 6,9 Mb/s. Tämä tarkoittaa sitä, kiinteät yhteydet ovat niin nopeita, että sivustojen suunnittelussa tarvitsee ottaa huomioon lähinnä vanhempien matkapuhelinsukupolvien hitaat yhteydet (ks. taulukko 1). Jotkin älypuhelimet pystyvät tosin hyödyntämään kiinteitä verkkoyhteyksiä WLAN-verkon kautta, mikäli käyttäjä löytää tukiaseman, johon pääsee yhdistämään.

TAULUKKO 1. Matkapuhelinteknologioiden sukupolvet (RadioShack's Guide to Wireless Telecommunications 2005; Laitila 2012.)

Teknologiasukupolvi	Kehitysvuodet	Latausnopeudet
1G	70- ja 80-luvut	9,6–14,4 kb/s
2G	90-luku–2000	9,6–14,4 kb/s
2.5G	2001–2004	56–144 kb/s
3G	2004–2005	384–3100 kb/s
4G	2006+	3–100 Mb/s

Sreeram Ramachandran (2012) ilmoittamien Googlen tilastojen mukaan keskimääräinen verkkosivu on kooltaan 320 kilotavua. Koska tavu on kooltaan 8 bittiä, on keskimääräisen sivun koko 2560 kilobittiä. Ihanteellisessa tilanteessa sivun lataus 3G-yhteydellä, jonka latausnopeus on 384 kb/s, kestäisi siis yli 6 sekuntia. Tom Ewer (2012) arvioi, että selainriippumattomat sivustot ovat kooltaan jopa suurempia, koska eri ruutukokojen elementit ovat usein piilotettuina, jolloin ne kaikki ladataan kerrallaan selaimeen, vaikkei niitä näytettäisikään. Kun otetaan huomioon aiemmin mainittu latausajan vaikutus kävijämäärään, on siis syytä pitää huolta, ettei sivun koko ainakaan ylitä 320 kilotavua.

2.3 Selainteknologiat

2.3.1 Yleisesti

Selainten tehtävänä on kääntää sivusto koodimuodosta selkeämmäksi visuaaliseksi kokonaisuudeksi. Selaimet eivät kuitenkaan käännä kaikkea sisältöä samannäköisiksi, sillä vaikka World Wide Web Consortium säätelee käännösstandardeja, ovat jotkin kohdat jääneet tulkinnanvaraisiksi. Kaikki selaimet voivat ymmärtää, mihin kohtaan jokin tietty määritelmä vaikuttaa, mutta koska sille ei ole määritetty standardissa oletusarvoa, täytyy selainten tekijöiden päättää itse, mitä arvoa käytetään. Selainten väliset erot voi kiertää tältä osin määrittelemällä kaikki sivustossa käytettävät arvot toteutuksessa. (Dahm 2010.)

HTML-standardi kehittyi niin nopeasti, etteivät selainvalmistajat aina ehdi päivittää tuotteitaan käyttämään uusimpia suosituksia. Useimmat käyttäjät eivät tosin vaivaudu lataamaan selaimen uusinta versiota, vaikka sellainen olisikin tarjolla. Siksi Dahm (2010) kehottaa toteuttamaan sivustot sellaisiksi, että ne toimivat virheettää ainakin selaimen kahdella edellisellä versiolla. (Dahm 2010.)

2.3.2 Selainlaajennukset ja HTML5

Puhtaasti selainpohjainen web-tuotanto ei ole perinteisesti pystynyt tarjoamaan kovinkaan monipuolisia selauselämyksiä, vaikka dynaamisen sisällön tekeminen onkin mahdollista JavaScriptin avulla. Videoiden, äänitiedostojen ja näyttävien pelien lisääminen sivustoille on työlästä, ja selainkohtaisia komponenttikirjaston puutteita vasta paikataan. Siksi mediasisällön näyttämiseen käytetään selainlaajennuksia.

Selainlaajennukset ovat kolmannen osapuolen ohjelmia, jotka tuovat selaimen lisätoiminnallisuuksia, kuten mahdollisuuden PDF-tiedostojen lukemiseen.

Laajennusten heikkoutena on se, että ne eivät ole selainvalmistajien tekemiä, joten

ne täytyy ladata erikseen. Lataaminen ei kuitenkaan onnistu Applen mobiililaitteiden ollessa kyseessä, sillä yhtiön tiukat ohjelmistosäännökset estävät niiden käytön.

Multimedialaajennukset

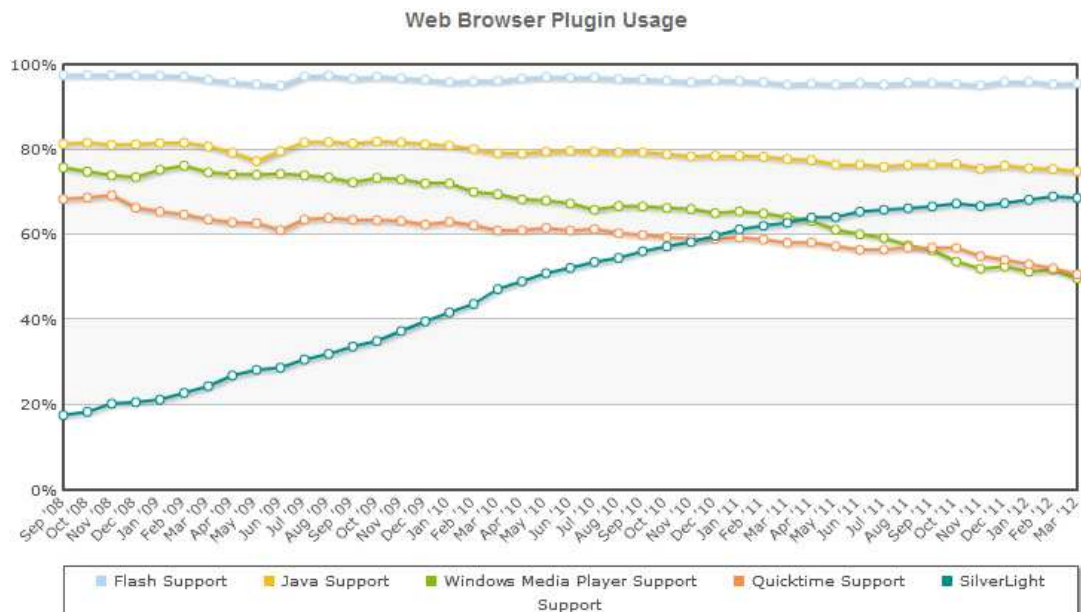
Java on ensimmäisen kerran vuonna 1995 julkaistu kehitysympäristö, ja on vapaana ohjelmistona käytännössä ilmainen. Java-sovelmat ajetaan hiekkalaatikossa, eli suljetussa järjestelmässä, mikä lisää niiden turvallisuutta, mutta rajoittaa sovelmien kommunikaatiota hiekkalaatikon ulkopuolelle ja siten ohjelmistoalustan monikäyttöisyyttä. Javan etuna on myös suuri komponenttivalikoima.

Flash –kehitysympäristön toisto-ohjelma on monien videopalveluiden, kuten YouTube:n oletustoistimena. Myös huomattavan moni selainpeli on ohjelmoitu Flashin ohjelmointikieli ActionScriptillä, mikä on osasyynä siihen, että toisto-ohjelma nauttii suurta suosiota (ks. kuvio 4). Tämän vuoksi Applen päätös estää selainlaajennusten käyttöä tuotteissaan olikin melko riskialtis. Flashin kehittäjä Adobe on sittemmin lopettanut Flashin mobiiliversion tukemisen (Winokur 2011).

Silverlightia voi pitää Microsoftin vastineena Flashille. Laajennus on asennettuna moneen tietokoneeseen, koska se tulee Windows-käyttöjärjestelmien mukana, mutta siltikään tätä kehitysympäristöä ei ainakaan toistaiseksi hyödynnetä kuin harvoissa sivustoissa.

Quicktime on Applen kehittämä mediatoistin, joka tulee Mac-tietokoneiden mukana, mutta on saatavilla myös PC:lle.

Windows Media Player on Microsoftin tekemä videotoisto-ohjelma, joka on pohjimmiltaan työpöytäkäyttöön tarkoitettu, mutta sisältää myös selainlaajennuksen. Laajennuksen Mac-version tukeminen on lopetettu.



KUVIO 4. Selainlaajennusten yleisyyden kehitys syyskuun 2008 ja maaliskuun 2012 välillä (StatOwl 2012)

HTML5

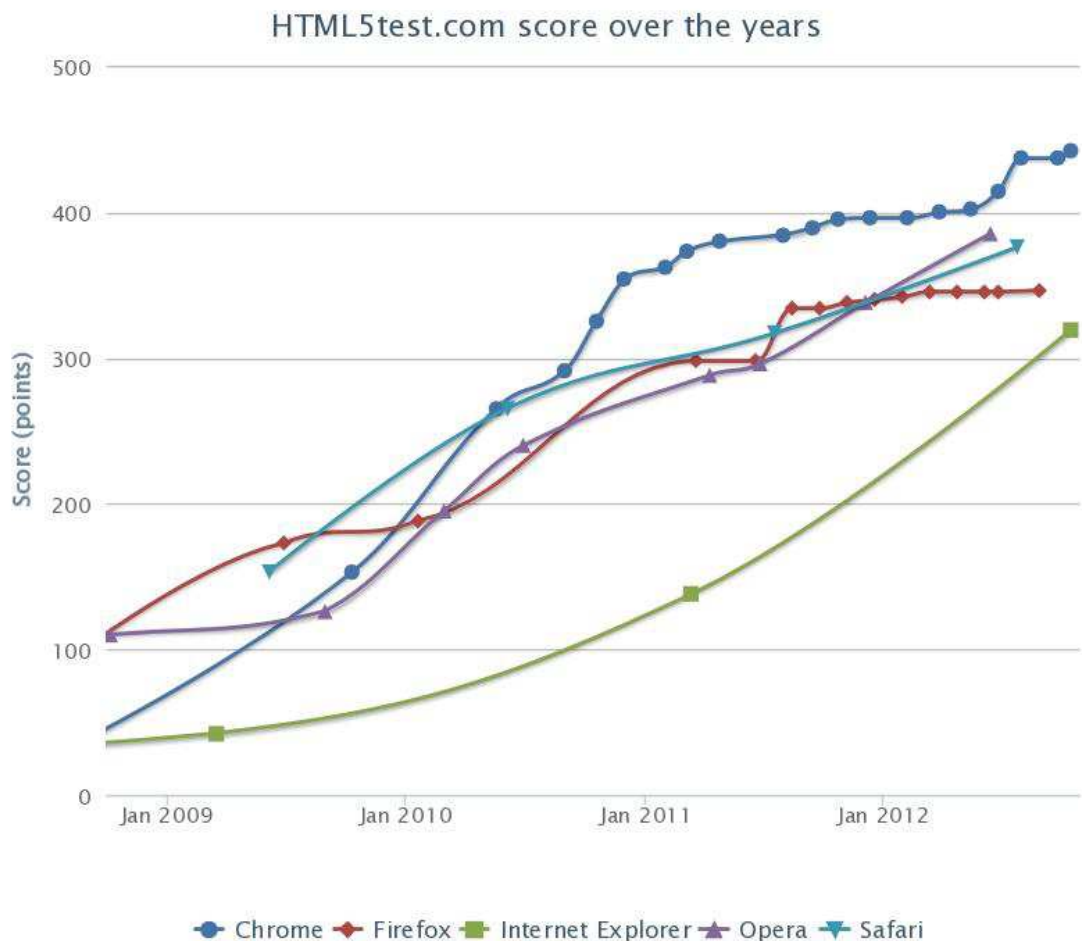
Koska selainten työ on kääntää HTML-koodi ymmärrettävään muotoon, on niissä kaikissa sisäänrakennettu tuki tälle merkintäkielelle. Tämä tarkoittaa sitä, että HTML:n kehittyessä kehittyvät myös selaimet.

HTML5 sopii selainriippumattomuuteen erinomaisesti, koska sen kehityksessä painotettiin laiteriippumattomuutta ja selainlaajennusten tarpeen vähentämistä. Selainriippumattomuuden kannalta tärkeimpinä uusina elementteinä ovat <video> ja <audio>, jotka mahdollistavat video- tai äänitiedoston sisällyttämisen sivustolle ilman selainlaajennuksen tarvetta.

HTML5 on vielä keskeneräinen projekti, joten sitä ei ole vielä määritetty uudeksi HTML-standardiksi. Selainvalmistajat voivat siten parantaa tuotteensa HTML5-yhteensopivuutta haluamassaan tahdissa, jolloin tuen tasossa on paljon vaihtelua (ks. kuvio 5). Myös vakavia puutteita esiintyy: Opera (HTML5 elements, attributes, and

APIs support in Opera Presto 2012) ilmoittaa sivuillaan, että sen uusimman selainversion tuki <video>- ja <audio>-elementeille on vain osittaista. Tilastojen (ks. taulukko 2) mukaan parhaiten HTML5:ttä tukeva Chrome on suosituin verkkoselain, joten merkintäkielen käyttö on perusteltua joidenkin selainten puutteista huolimatta.

HTML5:n tulevaisuus on niin lupaava, että Mozilla on kertonut tuovansa vuonna 2013 markkinoille Firefox-mobiilikäyttöjärjestelmän, jonka on määrä toimia kaikilta osin HTML5-kielellä. (Storås 2012.)



KUVIO 5. Selainten arvosana HTML5-yhteensopivuudessa. 500 pisteen selain on täydellisen yhteensopiva. Huomaa, että Internet Explorerin korkein arvosana on saavutettavissa vain Windows 8-käyttöjärjestelmässä (HTML5Test 2012)

TAULUKKO 2. Suosituimmat verkkoselaimet kävijädatan perusteella (w3schools 2012)

Kuukausi 2012	Internet Explorer	Firefox	Chrome	Safari	Opera
Lokakuu	16,1%	31,8%	44,9%	4,3%	2,0%
Syyskuu	16,4%	32,2%	44,1%	4,2%	2,1%
Elokuu	16,2%	32,8%	43,7%	4,0%	2,2%

2.3.3 Käyttöjärjestelmät

Käyttöjärjestelmät vaikuttavat sivuston toimintaan ennen kaikkea ohjelmistotuen kautta. W3Schoolsin (2012) tilastojen mukaan Flash –selainlaajennusta tukemattomien Applen tuotteiden osuus käyttöjärjestelmistä oli lokakuussa 2012 yhteensä 10,14%. Koska osuus oli vielä kasvussa, täytyy Flashin käyttöä web-tuotannossa harkita tarkkaan.

Ohjelmistotuen lisäksi web-tuotantoon vaikuttavat käyttöjärjestelmien erot kirjasinten näyttämässä. Sen lisäksi, että kaikkien käyttöjärjestelmien mukana ei tule samoja kirjasimia, on Windowsin periaatteena näyttää kaikki kirjaimet vain kokonaisten pikselien tarkkuudella, jotta teksti pysyisi terävänä. Tämän vuoksi pieni teksti voi olla Applen käyttöjärjestelmissä sameampaa ja vaikeampaa lukea. (Atwood 2007.)

2.4 Ulkoasun muodostaminen

2.4.1 Avainasioiden sijoittelu

Selainriippumattomassa web-suunnittelussa sivuston rakennetta suunnitellessa on erittäin tärkeää pitää huolta, että itse sisältö on välittömästi näkyvässä kuvakoosta huolimatta.

Päävalikko

Yksi elementeistä, jonka tulisi olla helposti käytettävissä on päävalikko, koska sen osuus käytettävyydessä on merkittävä. Valikko ei saa kuitenkaan olla liian paljoa esillä, jottei käyttäjien tarvitsisi vierittää ruutua päästäkseen tarkastelemaan sisältöä esteettömästi. Vaikka valikko pysyisi luettavana ja yhdellä rivillä vielä pienimmässäkin kuvakoossa, on silti hyvä käytäntö laittaa sivun toiseen ääripäähän toinen valikko tai painike, joka siirtää takaisin päävalikon kohdalle. Tällöin käyttäjän ei tarvitse selata sisällön läpi päästäkseen käyttämään päävalikkoa.

Mikäli valikko on kuitenkin niin laaja, että sen painikkeet vievät valtaosan ruudun tilasta mobiililaitteella tarkasteltaessa, on suositeltavaa käyttää luovempaa ratkaisua.

Pudotusvalikko vie hyvin vähän tilaa, joten sen sijoittaminen on helppoa. Sen suurin ongelma on kuitenkin siinä, että perinteisesti pudotusvalikoita käytetään lomakkeissa tai käyttötilan vaihtimissa. Niinpä käyttäjällä saattaa olla ongelmia päävalikon tunnistamisessa, etenkin jos sivustolla on vielä pudotusvalikkona toimiva kielenvalitsin tai vastaava. Myös useamman tason valikoiden teko on haasteellista.

Ankkuri alaviitteessä eli footerissa toimii siten, että päävalikko on sivun alalaidassa, jolloin sen viemä tila ei haittaa sivujen tarkastelua. Tällöin sivun ylälaidassa oleva päävalikkopainike vie käyttäjän tähän alaviitteen valikkoon. Tätä menetelmää käytettäessä on kuitenkin pidettävä huolta, että käyttäjä tunnistaa mihin painike on hänet vienyt, jottei hyppäyksestä aiheutuisi hämmennystä.

Valikkopainike toimii muuten alaviitteen ankkurin tavoin, paitsi että siinä valikko on piilotettuna ja yläosan päävalikkopainike avaa tämän piilotetun valikon. Tällöin käyttäjä ei hämmenny siirtymästä, mutta koska tämä metodi on riippuvainen javascriptistä, kannattaa kehittää myös vaihtoehtoinen päävalikko niille joilla se ei toimi.

2.4.2 Mobiililähtöisyys

Mobiililähtöisyydessä eli mobile first –menetelmässä on periaatteena suunnitella sivusto alun perinkin mobiililaitteille sopivaksi. Koska mobiililaitteiden koko rajoittaa näytettävän sisällön määrää, täytyy sitä karsia ja jättää vain sitä, mitä käyttäjän uskotaan tulleen etsimään. Sisällön määrä ei kuitenkaan kasva suuremmille päätelaitteille laajennettaessa, vaan samat periaatteet pätevät kauttaaltaan. Mobiililähtöisyyden puolestapuhujat vetoavat siihen, että jos jotkin kohdat voi pudottaa mobiililaitteelle suunniteltaessa, eivät ne ole tarpeellisia isolle ruudullekaan tehtäessä. (Marcotte 2011. 111-113.)

Brad Frost (2012) uskoo hakutoiminnon olevan olennainen osa mobiilikäyttöliittymää, koska se nopeuttaa sivustolla navigointia. Hakutoiminnon hyöty korostuu etenkin useamman valikkotason sivustoissa, mutta yksinkertaisissa yritysesittelysivustoissa se ei ole viemänsä näyttötilan arvoinen.

Yhtenä mobiililähtöisyyden eduista voisi pitää myös sitä, että aloitettaessa pienimmästä päästä voidaan saada laajempi resoluutioskaala käyttöön. Selainikkunaa ei voi pienentää määrättömästi, mutta käytännössä loputon suurentaminen on mahdollista.

Kosketusnäytöt

Mobiililähtöisyydessä tulisi rakentaa sivusto myös kosketusnäytöllisyyden pohjalta. Hiirellä pystyy tekemään kaiken mitä kosketusnäytölläkin, mutta päinvastainen ei päde.

Internet Explorerin kehitystiimiin kuuluva Jacob Rossi (2012) ohjeistaa, että hover-toiminto eli hiiren osoittimen pitäminen painikkeen päällä ei ole mahdollista kosketusnäytöillä, koska kosketusnäytöt tunnistavat vain painallukset. Siksi hover-toiminnon käyttöä tulisi välttää ja esimerkiksi useamman tason valikkopuiden kunkin tason pitäisi avautua vasta painalluksesta.

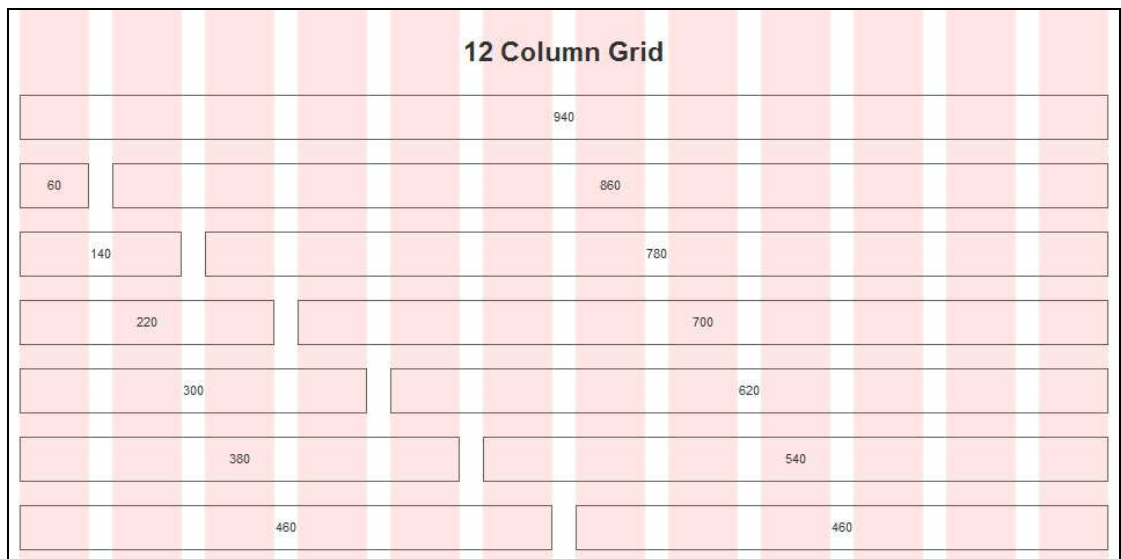
Tekstikenttiä tehdessä kannattaa hyödyntää HTML5:n *input type* –vaihtoehtoja, joiden avulla voidaan esimerkiksi sähköpostikenttään syöttämistä varten määrittää siihen sopiva kosketusnäppäimistö, jossa on vain sähköpostissa tarvittavat merkit. Koska *input type*jä on myös päivämäärille ja vastaaville, helpottaa niiden käyttö lomakkeiden täyttöä huomattavasti. (Rossi 2012.)

Rossin (2012) mukaan Windows 8:aa varten tehdyt tutkimukset osoittivat, että ihmisen sormi on keskimäärin 11mm paksu. Näin ollen kuvaruudun ollessa pienimmillään täytyy painikkeiden olla korkeudeltaan ja leveydeltään vähintään 40px. Lisäksi niiden välissä tulisi olla vähintään 10px levyiset raot. Näin ollen esimerkiksi iPhone 3:n 320px leveälle ruudulle mahtuisi kuusi vähimmäiskokoista painiketta.

2.4.3 Mukautuva ruudukko

Toteutustapaa, jossa sivuston kaikki elementit muuttuvat kuvaruudun kokoa muuttaessa siten, että kokonaisuus pysyy suhteiltaan samanlaisena, kutsutaan mukautuvaksi ruudukoksi.

Vaikka termi viittaa vain suhteellisissa mitoissa olevaan layoutiin, mukautuva ruudukko eli fluid grid noudattaa usein tavanomaista ruudukkomallia eli se koostuu useista samanleveyisistä pylväistä, joiden välissä on aina yhtä leveät raot (ks. kuvio 6). Koska elementit rakennetaan pylväiden tasalle, on niiden leveyksien määrittely helpompaa, mikä tekee lopputuloksesta selkeän.



KUVIO 6 Esimerkki kiinteästä 12 pylvään 960px leveästä ruudukkomallista (Smith 2012)

Mukautuvan ruudukon pohjalle tehtävissä sivustoissa ei kuitenkaan käytetä kiinteitä mittoja, kuten pikseleitä, koska selainriippumattomuudessa elementtien koon tulisi muuttua kuvakoon mukana. Itse ruudukko voi toki olla pikselimitoissa, koska sitä käytetään vain ulkoasun ensimmäisen vaiheen työstämiseen, jolloin kiinteät mitat voidaan muuttaa suhteellisiksi laskennallisesti. Tällöin ulkoasu pitää tämän ensimmäisen vaiheen mittasuhteet kaikilla kuvakooilla.

2.5 Suhteellisten mittojen käyttäminen

Mukautuvien elementtien mitat määrittyvät siten, että ne tarkistavat ylemmän tason elementin eli kontekstin tilan ja muuttavat kokonsa suhteellisten arvojen mukaisiksi. Ylimmän tason elementti eli juuri määrittää kokonsa selainikkunan perusteella, joten ikkunan koon muuttaminen aiheuttaa ketjureaktion, jonka yksikin kiinteä arvo katkaisee. Siksi suhteellisten mittojen käytössä tulee olla yhtenäinen. Tämä kuitenkin mahdollistaa myös kiinteiksi tarkoitettujen sivustojen tuottamisen vaivattomasti mukautuviksi muutettaviksi, sillä kiinteäksi säädetty juurielementti

varmistaa, että sivuston leveys pysyy samana, vaikka kaikki muut elementit olisivatkin mukautuvia.

2.5.1 Mittatyypit

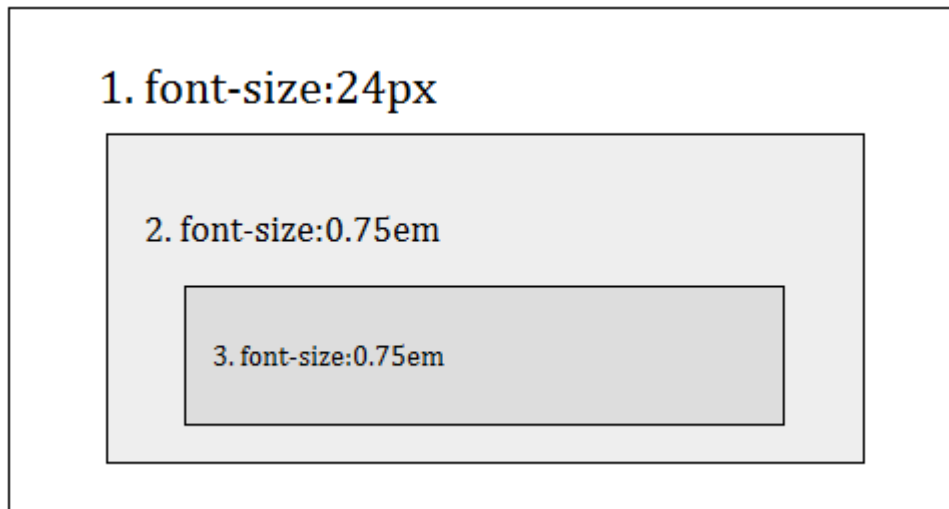
Vaikka selainriippumattomuus on melko uusi käsite, on siinä käytettäviä suhteellisia mittayksiköitä käytetty parantamaan kiinteiden sivustojen selainyhteensopivuutta jo pitkään. Käytettävyysasiantuntija Richard Rutter (2004) suositteli kiertämään niiden avulla muun muassa Internet Explorerin taipumuksen lukita kirjasinkoot yhteen kokoon.

Em

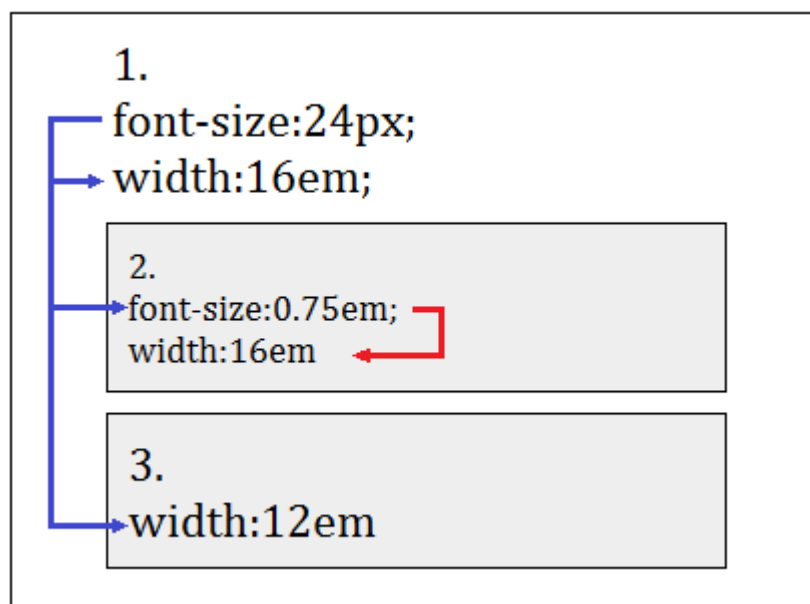
Em on mittayksikkö, joka suhteuttaa arvonsa ylemmän tason arvon eli kontekstin kertoimena. Koska em on alun perin tekstinkäsittelyyn tarkoitettu, se tarkkailee ylemmiltä tasoilta vain kirjasinten kokoja. Jos kahdella elementillä on samansuuruinen em-arvo, mutta eri fonttikoko ylemmällä tasolla, ne antavat eri lopputuloksen (ks. kuvio 7).

Yksi em vastaa suurin piirtein yhtä isolla kirjoitettua M-kirjainta. Em:n nimi juontuukin m-kirjaimen englanninkielisestä ääntämyksestä. Em suhteuttaa itsensä lähimpään fonttimääritykseen, joten jos elementin sisällä on määritetty fontti, suhteuttavat elementin muut osiot em-arvonsa tähän kirjasinkokoon (ks. kuvio 8).

Em:n heikkoutena on sen riippuvuus kontekstista, koska joissain tapauksissa voi olla vaikeaa päätellä mihin tutkittavana oleva elementti on yhteydessä. Myös kertoimien laskeminen hidastaa työtahtia pikseliarvojen käyttöön verrattuna, mutta harjaantumisen myötä aikaero on mitätön.



KUVIO 7. Esimerkki em-arvojen käyttäytymisestä. Kohdan 2 teksti on käytännössä kooltaan 18px ja kohdan kolme 13,5px

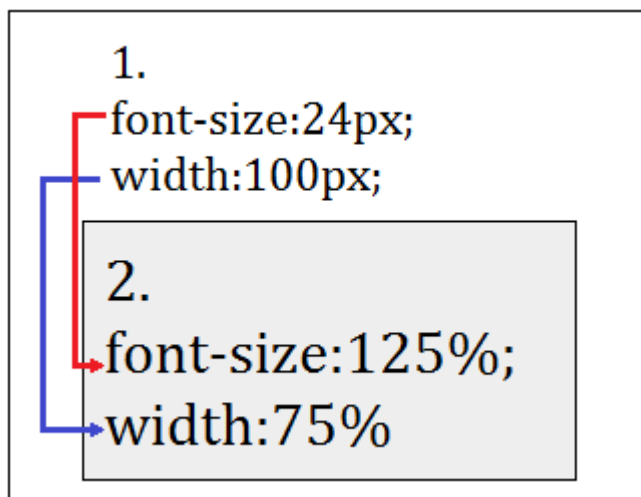


KUVIO 8. Em:n kontekstin periytyminen. Koska kohdassa 3 ei ole määritetty fonttikokoa, se perii sen kohdasta 1.

Rem toimii kuten em, mutta siinä on aina kontekstina sivuston juuressa määritetty arvo. Tällöin ei tarvitse pohtia suhdelukua laskiessa, minkä elementin alle se kuuluu. Rem on tosin vasta varsin vaihtelevasti tuetun CSS3:n (Can I use rem units?, 2012)

tuoma ominaisuus. Rem-arvojen käyttöä tuleekin vielä harkita, sillä ne eivät hyödytä itse loppukäyttäjää millään lailla em-arvoihin verrattuna, mutta voivat aiheuttaa yhteensopivuusongelmia.

Prosenttiarvot ovat helposti ymmärrettävissä, sillä niiden matemaattinen perusta kuuluu yleistietoon. Elementtien koon määrittelyssä tarvitsee vain muistaa, että alemman elementin näkökulmasta yläelementti on aina kooltaan 100% x 100%. Kontekstina prosenttiarvot käyttävät lähintä ylemmää sopivan tyyppin arvoa (ks. kuvio 9). Niinpä *margin*- ja *padding*-määrittelyksetkin toimivat ylemmän elementin mittojen mukaan.



KUVIO 9. Prosenttiarvojen periytyminen

Avainsanat eli keywordit ovat tekstin muotoiluun sopivia suhteellisia arvoja. Ne ovat selkokielisessä muodossa, mikä tarkoittaa sitä, että arvo *medium* vastaa käyttäjän oletusfonttikokoa ja astetta suurempi arvo *large* on 1,2 kertaa tätä suurempi (Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification, 2011). Koska mahdollisia arvoja on vain kahdeksan kappaletta ja ne toimivat vain kiinteinä

kertoimina, on avainsanojen käyttö liian rajoittunutta suuremman mittakaavan tuotantoihin.

2.5.2 Laskukaavat

Suhteellisten mittayksiköiden ansiosta selainriippumattomuuden laskukaavat eivät ole monimutkaisia, joten ne voi opetella helposti ulkoa.

Em ja rem

Kontekstikertoimen laskeminen on laskukaavoista tärkein, sillä sitä käytetään em-, rem- ja prosenttiarvojen laskemisessa.

$$\text{kohde} / \text{konteksti} = \text{em-arvo}$$

Näin ollen pienennettäessä peritty, oletusarvoltaan 16px kokoinen kirjaisin 14px kokoiseksi, laskutoimitus on tällainen:

$$14 / 16 = 0.875(\text{em})$$

Prosentti

Prosentteja laskiessa täytyy vielä muistaa kertoa tulos sadalla, jotta pilkku siirtyisi oikeaan paikkaan.

$$\text{kohde} / \text{konteksti} * 100 = \text{prosenttiarvo}$$

Jos siis oletetaan, että ylemmän elementin ollessa 800px levyinen alempi elementti on leveydeltään 600px, lasketaan alemman elementin suhteellinen mitta seuraavasti:

$$600 / 800 = 0.75$$

$$0.75 * 100 = 75(\%)$$

Painikkeet

Luvussa 2.4.2 mainitun painikkeiden vähimmäisleveyden vuoksi on hyvä tietää tapa laskea kullekin resoluutiolle mahtuvien vierekkäisten painikkeiden enimmäismäärä. Tätä voi hyödyntää etenkin päävalikkoa suunnitellessa. Painikkeiden tulee olla vähintään 40px leveitä ja niiden molemmilla puolilla täytyisi olla 10px raot. Näin ollen ruudulle mahtuvat painikkeet lasketaan seuraavasti:

$$\text{Vaakaresoluutio} / (40 + [2 \cdot 5]) = \text{painikkeiden määrä}$$

2.5.3 Kirjasin

Web-tuotannoissa kirjasinten koko on usein määritelty pikseliarvoina, koska niitä käytetään yleisesti tekstinkäsittelyssä, minkä vuoksi haluttu fonttikoko on helppo arvioida ja syöttää tarkasti. Mukautuvissa sivuissa pyritään kuitenkin toimimaan mahdollisimman paljon käyttäjän ehdoilla, jolloin kiinteä tekstikoko on huono käytäntö. Selain voi olla määritetty käyttämään suurempaa fonttia heikon näön vuoksi tai laite voi olla säädetty muuttamaan tekstin kokoa käytettävyyden parantamiseksi. Tällöin sen ulkoasu saattaa vaihdella selainten välillä. Tästä syystä sivuston juureen määritettävän fonttikoon tulisi olla prosentuaalinen arvo, typografiaan perehtyneen Richard Rutterin (2007) mukaan ihanteellisin olisi 100%.

On tärkeää huomata, että myös tekstin rivikorkeus eli line height määrittyy lähimmän kirjasimen koon mukaan, joten suhteellista rivikorkeutta laskiessa täytyy kontekstina käyttää kyseisen kohdan fonttikokoa. Pikselimäärittelyyn verrattuna tämä suhteutus helpottaa rivikorkeuksien säätöä sivustoilla, joilla niiden tulisi olla aina samassa suhteessa fonttiin nähden. Pikseleinähän ne olisi tarvinnut määrittää jokaiselle fontille erikseen.

Muuttuvaa ruutukokoa varten täytyy myös ottaa huomioon tekstirivien mitat. Kun kappaleen leveyden määrittää samoilla suhteellisilla mittayksiköillä, pysyy rivin kirjainmäärä luettavuuden kannalta samoissa lukemissa. Hyväksi havaittu rivin pituus on 66 kirjainta, eli kirjasimesta riippuen noin 30em (ks. kuvio 10). Tällöin silmien ei

tarvitse tehdä liian pitkiä sivuittaisliikkeitä, eikä teksti ole myöskään rivien lyhyiden vuoksi katkonaisen oloista (Sarmiento 2012.)

<p>Liian pitkä</p> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut tortor erat, pellentesque quis blandit vitae, congue vel magna. Mauris aliquam tristique nisl, eu condimentum enim vestibulum id. Aenean fringilla, nisl sed molestie aliquet, felis tortor vulputate urna, ac condimentum eros leo eget magna. Phasellus vel enim nisi, sit amet pulvinar mi. Donec suscipit pharetra pharetra. Duis ipsum massa, placerat ut condimentum suscipit, malesuada ut ligula. Nullam vel nulla orci. Donec dapibus aliquam pretium. Nullam eu nulla id orci eleifend tristique. Sed est lacus, hendrerit id sollicitudin et, rutrum non ante. Vestibulum feugiat consectetur dictum. Ut rutrum venenatis nisl, in molestie urna cursus eget. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Sed consequat euismod neque in sollicitudin.</p> <p>Nulla et augue vitae purus scelerisque iaculis in eget nulla. Donec congue bibendum lectus non lobortis. Suspendisse turpis urna, adipiscing venenatis rhoncus non, congue quis dui. Morbi a ante orci. Fusce quis velit lacus, at vestibulum urna. Curabitur sed felis eros. Mauris ultricies, est eu dignissim sollicitudin, mauris lacus venenatis nibh, mollis tristique sem erat quis leo. Vestibulum sed metus sem, pretium sagittis lectus. Curabitur venenatis, purus vel adipiscing vulputate, enim magna vulputate nibh, eu ornare magna nibh nec ante.</p>	<p>30em</p> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut tortor erat, pellentesque quis blandit vitae, congue vel magna. Mauris aliquam tristique nisl, eu condimentum enim vestibulum id. Aenean fringilla, nisl sed molestie aliquet, felis tortor vulputate urna, ac condimentum eros leo eget magna. Phasellus vel enim nisi, sit amet pulvinar mi. Donec suscipit pharetra pharetra. Duis ipsum massa, placerat ut condimentum suscipit, malesuada ut ligula. Nullam vel nulla orci. Donec dapibus aliquam pretium. Nullam eu nulla id orci eleifend tristique. Sed est lacus, hendrerit id sollicitudin et, rutrum non ante. Vestibulum feugiat consectetur dictum. Ut rutrum venenatis nisl, in molestie urna cursus eget. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Sed consequat euismod neque in sollicitudin.</p> <p>Nulla et augue vitae purus scelerisque iaculis in eget nulla. Donec congue bibendum lectus non lobortis. Suspendisse turpis urna, adipiscing venenatis rhoncus non, congue quis dui. Morbi a ante orci. Fusce quis velit lacus, at vestibulum urna. Curabitur sed felis eros. Mauris ultricies, est eu dignissim sollicitudin, mauris lacus venenatis nibh, mollis tristique sem erat quis leo. Vestibulum sed metus sem, pretium sagittis lectus. Curabitur venenatis, purus vel adipiscing vulputate, enim magna vulputate nibh, eu ornare magna nibh nec ante.</p>	<p>Liian kapea</p> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut tortor erat, pellentesque quis blandit vitae, congue vel magna. Mauris aliquam tristique nisl, eu condimentum enim vestibulum id. Aenean fringilla, nisl sed molestie aliquet, felis tortor vulputate urna, ac condimentum eros leo eget magna. Phasellus vel enim nisi, sit amet pulvinar mi. Donec suscipit pharetra pharetra. Duis ipsum massa, placerat ut condimentum suscipit, malesuada ut ligula. Nullam vel nulla orci. Donec dapibus aliquam pretium. Nullam eu nulla id orci eleifend tristique. Sed est lacus, hendrerit id sollicitudin et, rutrum non ante. Vestibulum feugiat consectetur dictum. Ut rutrum venenatis nisl, in molestie urna cursus eget. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Sed consequat euismod neque in sollicitudin.</p>
---	--	--

KUVIO 10. Esimerkit tekstirivien pituuksien vaikutuksesta luettavuuteen

3 SELAINRIIPPUMATTOMAN VERKKOSIVUN TOTEUTUS

3.1 Valmiit pohjat

Web-tuotannot noudattavat niin tarkasti samaa kaavaa, että niiden tekemisessä on yleensä järkevintä käyttää mahdollisimman paljon valmiita osia. Kiinteille sivuille suunnitellut osat eivät kuitenkaan välttämättä sovellu selainriippumattomiin tuotantoihin, mikä saattaa haitata niiden monipuolisuutta.

3.1.1 Julkaisujärjestelmät

Julkaisujärjestelmät ovat valmiita sivustorunkoja, joiden päälle voi rakentaa helposti ylläpidettäviä, mutta silti monipuolisia sivustoja. Tässä käsiteltävät, yhteisökehityspäätelä toimivat järjestelmät muuttuvat kohtuullisen nopeasti teknologioiden mukana, koska niiden käyttäjäkunta – ja sitä kautta kehittäjien määrä – on niin suurta. Tällöin koko järjestelmiä muokataan heidän omiin, uudistuneisiin tarpeisiinsa sopiviksi.

Moduulit eli julkaisujärjestelmiin lisätoiminnallisuuksia tuovat osiot ovat kuitenkin pienempien tiimien tai jopa yksilöiden kehityksen varassa. Jos kehittäjä ei katso omalta osaltaan tarpeelliseksi muuttaa moduulia selainriippumattomiin sivustoihin sopivaksi tai jos kehitys on lopetettu kokonaan, voi sivustolle suunnitellun ominaisuuden toteutus kariutua. Siksi onkin hyvä pohtia soveltuvatko aiemmin hyväksi havaitut moduulit selainriippumattomaan toteutukseen lainkaan.

Drupalin vakaat versiot eivät ole vielä HTML5:ttä varten optimoituja ja niiden oletusteemat ovat kiinteinä mittayksiköinä toteutettuja, joten puhtaalla asennuksella selainriippumattoman tuotannon tekeminen on työlästä. Vuoden 2013 puolivälissä julkaistavaksi suunniteltu Drupal 8 tulee kuitenkin olemaan HTML5-pohjainen ja sen teemoista tehdään selainriippumattomia jo alun perin (Drupal 8 Updates and How to

Help, 2012). Sitä odotellessa Drupal-tuotannoissa voi käyttää pienempien tiimien mukautuvia teemoja ja moduuleita. Tarjolla on myös varsin laajoja kaupallisia lisäyksiä, joilla selainriippumattoman sivuston räätälöiminen onnistuu jopa pelkkää graafista käyttöliittymää käyttämällä.

Wordpress on pyrkinyt kohti selainriippumattomuutta tekemällä ensin hallintasivuistaan mukautuvia ja sittemmin uudistanut myös vakioiteemansa täysin mukautuvaa ruudukkoa käyttäväksi. Wordpressin vakioiteema on erittäin suosittu, joten muutos lisää selainriippumattomien sivustojen määrää huomattavasti. (Gilbertson 2012.)

Joomlan yhteisössä selainriippumattomuudesta on tullut trendi vasta vuoden 2012 viimeisellä kolmanneksella, joten mukautuvien lisäosien puutteessa sen päälle kannattaa koota vain kohtuullisen yksinkertaisia sivustoja. Mukautuvaa ruudukkoa käyttäviä teemoja on kuitenkin jo riittämiin. (Responsive Web Design and Joomla 2012.)

3.1.2 Frameworkit

Frameworkit ovat julkaisujärjestelmien tavoin sivustojen tuotantoon sopivia valmiita pohjia. Niissä ei kuitenkaan ole julkaisujärjestelmien käyttämiä sisällönhallintatoimintoja, vaan ne koostuvat esimerkinomaisista tiedostoista. Tämän järjestelmäpohjan puutteen tuoman yksinkertaisuuden vuoksi jaossa on erittäin monia selainriippumattomia frameworkkeja.

Sisällönhallinnan puutteen vuoksi frameworkit sopivat etenkin sivustoille, joiden ylläpidon helppoudesta ei tarvitse huolehtia. Tähän tarkoitukseen frameworkien joukosta löytyy todennäköisesti juuri sellainen, jossa on kaikki toivotut ominaisuudet ja jonka pohjalta sivuston tuotanto onnistuu varsin nopeasti.

3.1.3 Vanhan sivuston päivittäminen

Selainriippumattomuuden tullessa yhä tunnetummaksi suuren yleisön keskuudessa voidaan melko uusiakin sivustoja joutua päivittämään toimimaan sen periaatteilla. Jos sivusto on vain muutaman kuukauden ikäinen, sen suunnitteleminen täysin uudelta pohjalta voi vaikuttaa järjettömältä ajatukselta.

Yksinkertaisten sivustojen muuttaminen selainriippumattomiksi onkin helppo prosessi, kun tarvitsee vain muuttaa mittayksiköt suhteellisiksi ja vaihtaa sivupalkkien paikkaa muutospisteiden perusteella. Sivustot, jotka sisältävät gallerioita tai moduuleita ovat kuitenkin varsin haastavia muutettavia. James Youngin (2012) tekemän kyselyn mukaan useimmat selainriippumattomia sivustoja tekevät pitävät tällaisissa tilanteissa tehokkaimpana ratkaisuna tehdä sivusto kokonaan alusta.

Alusta aloitettaessa voidaan valita myös moduulit, jotka sopivat parhaiten selainriippumattomuuteen ja tarvittaessa karsia kaikki ylimääräiset osiot jo suunnitteluvaiheessa.

3.2 Sisällön selainriippumattomuus

Hyvästä suunnittelusta huolimatta on joitain kohtia, joiden käsittely on tehokkainta vasta tuotantovaiheessa. Muuttuvien osien määrästä johtuen näiden kohtien päättely etukäteen on haastavaa, mutta käytännön testien avulla määrittely onnistuu nopeasti.

3.2.1 Muutospisteet

Vaikka suhteellisilla mitoilla tehdyt elementit voivat muuttaa kokoaan käytännössä määrättömästi, niiden käytännöllisyys laskee tietyn rajan yli mentäessä. Tästä syystä

layoutin asettelua täytyy vaihdella ruudun koon muuttuessa, jotta layoutista saataisiin näyttävän näköinen kaiken kokoisilla kuvaruuduilla.

Muutospisteet eli break pointit ovat kohtia, joissa ulkonäköä muutetaan tyylitiedostossa. Niiden avulla voidaan muun muassa siirtää sivupalkit pääsisällön alapuolelle mobiililaitteita varten ja muuttaa päävalikko pudotusvalikoksi.

Muutospisteiden kohdista ei ole kuitenkaan selkeitä sääntöjä.

Ethan Marcotte (2011. 102 - 105) suositteli kirjassaan, että muutospisteet tulisi määritellä kohdelaitteiden ruutujen koon mukaan, jotta layout pysyisi aina juuri kyseiselle kuvaruudulle suunniteltuna. Tätä menetelmää käytettäessä onkin tärkeää tuntea yleisimmät näyttöteknologiat.

Koska laitteiden kirjo on kuitenkin kasvanut Marcotten kirjan kirjoitushetkestä, on suositeltavaa toteuttaa muutospisteet siten, että selvittää kuvaruutua skaalaamalla resoluution, jossa senhetkinen layout ei enää toimi. Sitten kyseiseen kohtaan laitetaan muutospiste, määritellään ulkoasu jälleen sopivaksi ja jatketaan skaalausta seuraavaan ongelmakohtaan saakka. Tällä menetelmällä pyritään pitämään luettavuus mahdollisimman hyvänä myös odottamattomilta resoluutioilta tarkasteltaessa. On kuitenkin hyvä tapa tarkistaa layoutin toimivuus suosituimpien resoluutioiden kohdalla ja tarvittaessa optimoida sitä uusilla muutospisteillä. (Rupert 2012.)

Media Queries

Muutospisteiden määrittämiseen käytetään media query-komentoja eli mediakyselyitä. Niiden avulla voidaan selvittää useita laitteiston ominaisuuksia, joiden perusteella layoutiin voi tehdä muutoksia (ks. taulukko 3).

TAULUKKO 3. Media query-käskyt (Marcotte 2011 s.76-78)

Tunnus	Määritelmä	Voi olla min- tai max-arvo
width	Piirtoalueen eli useimmiten selainikkunan leveys.	Kyllä
height	Piirtoalueen korkeus.	Kyllä
device-width	Laitteen kuvaruudun leveys. Selainikkunan koon muuttamisella ei vaikutusta.	Kyllä
device-height	Laitteen kuvaruudun korkeus.	Kyllä
orientation	Kuvaruudun suunta. Hyväksyy arvot portrait eli pystytaso ja landscape eli vaakataso	Ei
aspect-ratio	Piirtoalueen kuvasuhde eli sen leveyden suhde korkeuteen. Merkitään muodossa leveys/korkeus	Kyllä
device-aspect-ratio	Sama kuin aspect-ratio, mutta laitteen kuvaruudusta.	Kyllä
color	Bittien määrä värikomponenttia kohden.	Kyllä
color-index	Laitteen väripaletin värien määrä. Palettia käyttämättömillä arvo on 0.	Kyllä
Taulukko 3 jatkuu seuraavalla sivulla		

Tunnus	Määritelmä	Voi olla min- tai max-arvo
monochrome	Bittien määrä mustavalkolaitteen pikselä kohden.	Kyllä
resolution	Laitteen pikselitiheys. Käytettävä min- tai max-arvoa, jos aiotaan saada tiheys muiltakin kuin neliön muotoisia pikseleitä käyttäviltä laitteilta.	Kyllä
scan	TV-laitteiden skannausprosessi. Joko progressive tai scan .	Ei
grid	Käyttääkö laite ruudukkopohjaista piirtoa. Kiinteää tekstikokoa käyttävät puhelimet ovat tällaisia laitteita.	Ei

Ennen mediakyselyä voidaan määrittää mediatyyppi, joiden avulla laitteet ilmoittavat mihin kategoriaan ne kuuluvat (ks. taulukko 4). Tavallisen sivuston tuotannossa tärkeimmät kategoriat ovat *screen*, *all* ja mahdollisesti *print*. Printin avulla voi piilottaa tulosteen kannalta epäolennaiset elementit, kuten päävalikon. Matkapuhelimille tarkoitettu *handheld*-kategoria ei viittaa moderneihin älypuhelimiin, koska handheld-tyypin sivustot oltiin suunniteltu vanhemmille puhelinmalleille, jolloin ne eivät hyödyntäneet älypuhelimien tehoja. Älypuhelinvalmistajat laittoivat tuotteensa ilmoittamaan tyyppikseen *screen*, koska perinteiset sivustot sopivat niille paremmin. Mikäli mediatyyppiä ei määritellä, sen oletetaan olevan *all*.

TAULUKKO 4. Mediatyypit ja niitä käyttävät laitteet. (Media types, 2011)

Mediatyyppi	Kuvaus
all	Kaikki laitteet.
braille	Pistekirjoituspalautetta antavat kosketusnäyttölaitteet.
embossed	Pistekirjoitusta tulostavat laitteet.
handheld	Älypuhelimia yksinkertaisemmat matkapuhelimet.
print	Tulostus ja sen esikatselu.
projection	Projektorit ja piirtoheittimelle tulostettavat kalvot.
screen	Tietokoneiden ruudut. Myös tabletit ja älypuhelimet käyttävät tätä.
speech	Puhetta syntesisoivat laitteet. Aiemmin käytetty termiä 'aural'.
tty	Tasalevyistä kirjasinruudukkoa käyttävät laitteet, kuten terminaalit ja kaukokirjoittimet.
tv	Television ruudulle piirtävät laitteet.

Media queryt voidaan laittaa CSS-tyylitiedostoihin, jossa niiden sisältämät kohdat ajetaan, mikäli kyselyn ehdot täyttyvät. Internet Explorer 9:n edeltäjät eivät tue media queryja suoraan, mutta ongelman voi ratkaista yleisesti jaossa olevilla JavaScript-tiedostoilla. Jos oletetaan, että pienin vaakaresoluutio, jolla sivuston

täytyy toimia on 320px ja halutaan määrittää sivuston rungon fonttikooksi 0.9em, on tyylitiedoston sisäinen kysely tällainen:

```
@media screen and (max-width:320px ) {
    body {font-size:0.9em;}
}
```

Mobiililähtöistä menetelmää käytettäessä max-width –määritelmiä käytetään kaikissa muutosasteissa viimeisintä eli suurinta lukuun ottamatta. Siihen tulisi käyttää min-width –määritelmää, joka vaikuttaa kaikkiin vaakaresoluutioihin kyseisestä arvosta ylöspäin.

Koska määrittelemätön mediatyyppi on oletetaan automaattisesti tyypiksi all, ei kaikkiin laitteisiin vaikuttaviin käskyihin tarvitse määrittää mediatyyppiä lainkaan. Jos siis halutaan, että kuvaruudun ollessa pystyasennossa teksti on valkoista laitteesta riippumatta, voi sen määritellä näin:

```
@media (orientation:portrait ) {
    body{color:#fff;}
}
```

Mediakyselyn voi myös suorittaa laittamalla sivun head-osioon menevällä käskyllä sen käyttämään eri tyylitiedostoja mediatyypistä riippuen. Jos halutaan sivun lataavan eri tyylitiedostot mediatyypeille screen ja print, käy se näin:

```
<link rel="stylesheet" type="text/css" media="screen"
href="screen_tyylitiedosto.css">

<link rel="stylesheet" type="text/css" media="print"
href="print_tyylitiedosto.css">
```

3.2.2 Kuvat

Ennen kuin lisää sivustolleen valmiita kuvia, on suositeltavaa pohtia, voisiko kyseisen kohdan toteuttaa suhteellisilla mittayksiköillä tehdyllä css-spritellä. Niiden käyttö nopeuttaa sivuston toimintaa, koska kuvien lataukselle ei ole tarvetta. Suhteellisten mittayksiköiden ansiosta ne myös mukautuvat ruudun kokoon.

Mukautuvassa layoutissa myös sivustolla olevien kuvien täytyy pystyä muuttamaan kokojaan. Tämä on kuitenkin helppo toteuttaa, koska kuville on oma tunnisteensa css-kielessä. Koko sivuston laajuinen määritelmä mukautuvan kokoisille kuville olisi seuraavanlainen:

```
img {max-width:100%}
```

On kuitenkin tärkeää muistaa, että kuvat, joissa on max-width –määritelmä eivät koskaan kasva alkuperäistä kuvakokoaan suuremmiksi. Useimmissa sisältökuvissa tämä on toivottavaa, mutta se saattaa olla ongelmallista venytettäväksi tarkoitetuissa layoutin kuvissa. Ongelman voi kuitenkin korjata yksinkertaisesti jättämällä ehtomääritelmän max kokonaan pois , jolloin käsky on:

```
img {width:100%}
```

Mikäli kumpikaan vaihtoehto ei sovi kuvan tarkoitukseen, on kuvan ylempään elementtiin mahdollista laittaa määritelmä, joka leikkaa kuvasta ylimääräisen osion pois, mutta pitää kuvan alkuperäisessä koossaan.

```
.esimerkkiluokka{ overflow:hidden }
```

```
.esimerkkiluokka img {
```

```
display:block;
```

```
max-width:auto; }
```

Kuvat, jotka sisältävät paljon tietoa on hyvä tehdä useammalle koolle, koska pienennys haittaa luettavuutta huomattavasti. Tällaisia kuvia ovat esimerkiksi graafit

tai mainostekstit. Yksi tapa kuvan vaihdon toteuttamiseen on laittaa kuvat CSS-tyylitiedostossa background-kuvina ja vaihtaa kuvaa aina muutos pisteiden kohdalla.

3.2.3 Videot

Videoiden koon pystyy rajaamaan samalla tavoin kuin kuvienkin, mutta overflow-leikkauksen käyttö tekee videon seuraamisesta vaikeaa. Videoiden tunnuksina toimivat *video* HTML5-videoissa ja *object* selainlaajennusvideotoistimissa. Molemmat voidaan siis määrittää kerrallaan seuraavasti:

```
video, object {max-width:100%}
```

Molempia tunnuksia tarvitaan selainriippumatonta sivustoa tehdessä, sillä kumpaakaan toistotapaa ei tueta jokaisessa laitteessa. HTML5-videon lisäyksessä tämä ollaan kuitenkin otettu hyvin huomioon, ja sen tagien sisälle laitettut asiat ajetaan, mikäli HTML5-videota ei voida näyttää. Näin ollen HTML5-tagien sisälle laitettu selainlaajennusvideo näkyy vain, jos HTML5:ttä ei tueta. Videon *source*-tageilla voidaan laittaa soitin käyttämään useampia eri videoformaatteja, jotta kaikkien selainten käyttämät formaatit ovat varmasti tuettuina. Typistetty videonlisäys näyttäisi tältä:

```
<video controls>
<source src="videoformaatti.mp4" type="video/mp4">
<source src="toinenvideoformaatti.avi" type="video/avi">

  <object type...>
    <param name...>
    <!--[if IE]><param name...><![endif]-->
    <p>Selaimesi ei tue HTML5-videoita.</p>
  </object>

</video>
```

3.2.4 Interaktiivinen sisältö

Interaktiivisessa sisällössä, kuten selaimessa pelattavissa peleissä on laitteiden erot otettava huomioon jo niiden kehitysvaiheessa. Jos tätä sisältöä on tarkoitus pystyä käyttämään laitteesta riippumatta, voidaan siihen olla tehty resoluutiota tarkkaileva kysely, joka muuttaa itse sisältöobjektia sen mukaisesti. Vaihtoehtoisesti interaktiivisesta sisällöstä ollaan voitu tehdä kokonaan eri versiot eri laitteille. Näin ollen niiden selainriippumattomuudesta huolehtiminen ei ole sivuston tuottajien vastuulla muuten kuin objektien lisäysten suhteen.

HTML5:n lisäystagin *canvas* ja selainlaajennusten tagin *object* lisäämisessä pätevät samat toimenpiteet kuin video-objektienkin tapauksessa. Selainlaajennuksen tagit lisätään siis HTML5-tagien väliin, jolloin selainlaajennusobjekti ajetaan vain silloin, jos HTML5:ttä ei tueta.

3.3 Haasteet

3.3.1 Kolmannen osapuolen sisältö

Web-tuotannossa käytetään vielä oletusarvoisesti kiinteitä arvoja, joten ei kannata odottaa kolmannen osapuolen toimittavan sisältöä suoraan selainriippumattomaksi tehtynä.

Koska Facebook on ainakin vielä toistaiseksi erillisten mobiili- ja tietokonesivustojen varaan rakennettu, ovat sen tykkäyspainikkeet kiinteän kokoisia ja erillisille koodipohjille optimoituja. Niiden muuttaminen kooltaan mukautuvaksi edellyttää elementin *fb_root* poistamista tai piilottamista ja muiden elementtien muuttamista prosenttikokoon.

Kolmannen osapuolen mainospalvelimilta sisältönsä lataaviin osioihin voi olla vaikea tehdä kaikkia osapuolia tyydyttävää ratkaisua. Jotkin mainoksista saattavat olla liian epätarkkoja pienemmille resoluutioille tai niiden sijoittelu on ongelmallista. Mainoksia ei kuitenkaan voi piilottaa, sillä mainostajat maksavat sivujen latauskerroista, joten tällöin mainostaja maksaisi mainoksesta, jota käyttäjät eivät näe. Ongelmaa voi yrittää ratkaista olemalla yhteydessä mainospalvelimien ylläpitoon, mutta mikäli se ei onnistu, täytyy toteutuksessa tustustua mainospalvelimen käyttämään koodiin ja tehdä sen perusteella jonkinlainen hätäratkaisu. (Flaherty 2012.)

3.3.2 Testaus

Testaus on kaikkia selainriippumattomia web-tuotantoja koskeva ongelma, koska sivuston on tarkoitus toimia useammalla laitteella. Selainikkunan kokoa muuttamalla saa hyvin selvitettyä layoutin toimintaa eri resoluutioilla, etenkin, jos käyttää jotain testaukseen suunniteltua eri laitteiden resoluutioita simuloivaa selainliitännäistä.

Ainoa tapa testata sivuston käytännön toimivuutta on käyttää mahdollisimman montaa laitetta. Olisi suositeltavaa, jos testaus suoritettaisiin kaikilla suosituimpien käyttöliittymien laitteilla, jotta mahdollisesti huomaamatta jääneet erot saataisiin esiin. Useammalle samaa käyttöliittymää käyttävälle laitteille ei ole sinänsä tarvetta, koska resoluutioiden testaaminen onnistuu muutenkin.

Mikäli laitteiden hankkimiseen ei ole mahdollisuutta, tarjolla on myös laitteita emuloivia eli matkivia ohjelmistoja, joilla saa kokeiltua sivuston toimivuutta. Tietokoneiden ja emuloitavien laitteiden erot ovat kuitenkin niin suuria, että ohjelmistot eivät pysty matkimaan laitteiden erikoisominaisuuksia, kuten kallistuksentunnistimia tai näytön pikselitiheyksiä.

4 POHDINTA JA JOHTOPÄÄTÖKSET

Aihevalinnan aikaan en ollut vielä perehtynyt selainriippumattomaan web-suunnitteluun, koska uskoin, että kyseistä menetelmää käyttämällä tuotantoon tulee enemmän ongelmia kuin hyötyjä. Kun sivustojen tilaajat alkoivat kuitenkin yhä useammin pohtia palautteessaan sivuston käytettävyyttä tablet-tietokoneilla, oli aiheeseen tutustuminen tarpeen.

Aluksi tehtävänä oli erään sivuston selainriippumattomaksi muuttaminen, mutta sittemmin ohjauksen tuloksena tehtäväksi tuli ohjeistus, jota voisi käyttää selainriippumattomuuden perusteisiin perehdyttämiseen. Ohjeistusta pidettiin yksittäisen sivuston päivittämistä hyödyllisempänä, sillä tällöin dokumenttia pystyisi käyttämään myös tulevilla projekteilla.

Aihepiirin rajausta ei ollut sinällään hankalaa, koska web-tuotannossa huomioitavien asioiden listaus oli aikaisemman työkokemuksen perusteella helppoa. Ohjeistukseen sisällytettävien osioiden valinta oli kuitenkin haasteellista. Suuremmat koodikokonaisuudet oli jätettävä pois, koska tavoitteena ei ollut tehdä valmiita toteutuksia, vaan ohjeistaa selainriippumattomuuden periaatteisiin. Tämän vuoksi myös JavaScriptiä vaativat osiot on käsitelty lyhyesti.

Selainriippumattomuus on suhteellisen uusi käsite, joten sen kehitys tulee todennäköisesti olemaan nopeaa vielä jatkossakin. Selainriippumattomuuden kannalta ihanteellisimman tilanteen eli kaikkien laitteiden täyden HTML5- ja CSS3-standardien tuen toteutumista voi pitää epärealistisena, koska Internet Explorerin vanhempien version suosion perusteella suurimmalla osalla ihmisistä ei ole tarvetta pitää laitteitaan ajan tasalla.

Mitä enemmän selainriippumattomia tuotantoja tekee, sitä todennäköisemmin aiheesta tulee kuitenkin nousemaan uusia ongelmia, joiden dokumentoiminen on tarpeen. Standardien kehittyessä näihin ongelmiin saatetaan saada sitten kehittyneempiä ratkaisuja.

Lähteiden löytäminen oli myös hankalaa, sillä selainriippumattomuuteen kriittisesti suhtautuvia lähteitä oli vaikea löytää ja löydettyjenkin luotettavuus oli kyseenalaista. Kritiikki kohdistui käsiteltävästä aiheesta riippumatta useimmiten selainriippumattomaan kokonaisuutena yksittäisten menetelmien, kuten mukautuvien ruudukoiden sijaan. Tämä teki lähteistä hyödyttömiä käsiteltävien osioiden suhteen. Kyseenalaistaminen oli siis enimmäkseen oman kokemuksen ja kokeilujen varassa. Tämä on kuitenkin perusteltua, sillä päätelmät perustuivat kokemukseeni toimeksiantajan web-tuotantomenetelmistä, jolloin opinnäytetyön tavoite täyttyi.

LÄHTEET

Atwood, J. 2007. Font Rendering: Respecting The Pixel Grid. Viitattu 11.10.2012. <http://www.codinghorror.com/blog/2007/06/font-rendering-respecting-the-pixel-grid.html>

Can I use rem units? 2012. Viitattu 10.11.2012. <http://caniuse.com/rem>

Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification. 2011. Viitattu 19.11.2012. <http://www.w3.org/TR/CSS21/>, Fonts

Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification. 2011. Viitattu 20.11.2012. <http://www.w3.org/TR/CSS21/>, Media types

Chanda, H. 2012. What is Responsive web design – Everything a business ever needs to know. Viitattu 3.10.2012. <http://bizdharma.com/blog/what-is-responsive-web-design-business-perspective/>

Croft, J. 2010. On “responsive web design” and the mobile context. Viitattu 1.10.2012. <http://jeffcroft.com/blog/2010/aug/06/responsive-web-design-and-mobile-context/>

Dahm, T. 2010. Browser Compatibility Tutorial. Viitattu 12.10.2012. <http://www.netmechanic.com/products/Browser-Tutorial.shtml>

Drupal 8 Updates and How to Help. 2012. Viitattu 12.11.2012. <http://drupal.org/community-initiatives/drupal-core>

Ewer, T. 2012. 5 Reasons Why Responsive Design Is Not Worth It. Viitattu 1.10.2012. <https://managewp.com/5-reasons-why-responsive-design-is-not-worth-it>

Flaherty, R. 2012. Responsive Ads in the Real World: Ad Server Implementation. 27.11.2012. <http://www.ravelrumba.com/blog/responsive-ads-real-world-ad-server-implementation/>

Frost, B. 2012. Anatomy of a Mobile-First Responsive Web Design. Viitattu 10.11.2012. <http://bradfrostweb.com/blog/mobile/anatomy-of-a-mobile-first-responsive-web-design/>

Gilbertson, S. 2012. WordPress Embraces Responsive Design With New ‘Twenty Twelve’ Theme. Viitattu 20.11.2012. <http://www.webmonkey.com/2012/08/wordpress-launches-new-default-theme-twenty-twelve/>

Gryta, T. 2012. AT&T to Leave 2G Behind. The Wall Street Journal 3.8.2012. Viitattu 6.10.2012.

http://online.wsj.com/article_email/SB10000872396390443687504577567313211264588-lMyQiAxMTAyMDAwMzAwODM3Wj.html

Hruska, J. 2012. LG's new 440 PPI display is way too much of a good thing. Viitattu 9.10.2012. <http://www.extremetech.com/computing/130051-lgs-new-440-ppi-display-is-way-too-much-of-a-good-thing>

HTML5 elements, attributes, and APIs support in Opera Presto 2.5. 2012. Viitattu 13.10.2012. <http://www.opera.com/docs/specs/presto25/html5/>

The HTML5 test – how well does your browser support HTML5? 2012. Viitattu 14.10.2012. <http://html5test.com/results>, other browsers

Laitila, T. 2012. Testissä: joko 4G-yhteys korvaa kiinteän liittymän? Viitattu 8.11.2012. http://fin.afterdawn.com/uutiset/artikkeli.cfm/2012/05/27/testissa_joko_4g-yhteys_korvaa_kiinteaa_liittymaa

Linden, G. 2006. Marissa Mayer at Web 2.0. Viitattu 7.10.2012. <http://glinden.blogspot.fi/2006/11/marissa-mayer-at-web-20.html>

Marcotte, E. 2011. Responsive web design. New York: A Book Apart.

Pitkänen, P. 2012. Suomen nettinopeudet nousivat maailman kärkikymmenikköön. Digitoday 10.8.2012. Viitattu 8.10.2012. <http://www.digitoday.fi/data/2012/08/10/suomen-nettinopeudet-nousivat-maailman-karkikymmenikkoon/201235359/66>

RadioShack's Guide to Wireless Telecommunications. 2005. Viitattu 7.10.2012. http://support.radioshack.com/support_tutorials/telephone/cell-03.htm

Ramachandran, S. 2010. Web metrics: Size and number of resources. Viitattu 6.10.2012. <https://developers.google.com/speed/articles/web-metrics?hl=fi-FI>

Responsive Web Design and Joomla. 2012. Viitattu 20.11.2012. <http://www.ijoomla.com/blog/responsive-web-design-and-joomla/>

Rossi, J. 2012. Guidelines for Building Touch-friendly Sites. Viitattu 10.11.2012. <http://blogs.msdn.com/b/ie/archive/2012/04/20/guidelines-for-building-touch-friendly-sites.aspx>

Rupert, D. 2012. Five steps to gettin' flexy in responsive web design. Viitattu 21.11.2011. <http://www.netmagazine.com/features/five-steps-gettin-flexy-responsive-web-design>

Rutter, R. 2004. How to size text using ems. Viitattu 19.11.2012. <http://clagnut.com/blog/348/>

Sarmiento, J. Responsive Web Design, most complete guide. Viitattu 20.11.2011. <http://www.webdesignshock.com/responsive-web-design/>

Smith, N. 2012. 960 Grid System. Viitattu 12.11.2012. <http://960.gs/demo.html>

StatCounter Global Stats. 2012. Viitattu 1.10.2012. <http://gs.statcounter.com/>

Storås, N. 2012. Saisiko olla Firefox-kännykkä? Ensi vuonna niitä saa. MikroPC 4.7.2012. Viitattu 10.11.2012.

http://www.mikropc.net/kaikki_uutiset/saisiko+olla+firefoxkannykka+ensi+vuonna+niita+saa/a820563

Web Browser Plugin Market Share. 2012. Viitattu 13.10.2012.

http://www.statowl.com/plugin_overview.php, timeframe muutettu

Winokur, D. 2011. Flash to Focus on PC Browsing and Mobile Apps; Adobe to More Aggressively Contribute to HTML5. Viitattu 14.10.2012.

<http://blogs.adobe.com/conversations/2011/11/flash-focus.html>