

RESPONSIIVISTEN WWW-SIVUJEN TOTEUTTAMINEN

WordPress-sisällönhallintajärjestelmän
Twenty Thirteen -teeman lapsiteema

Taija Vehmaa

Opinnäytetyö
Toukokuu 2014
Tietojenkäsittelyn
koulutusohjelma
Digitaalinen media

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietojenkäsittelyn koulutusohjelma
Digitaalinen media

VEHMAA, TAIJA:

Responsiivisten WWW-sivujen toteuttaminen
WordPress-sisällönhallintajärjestelmän *Twenty Thirteen* -teeman lapsiteema

Opinnäytetyö 47 sivua, joista liitteitä 3 sivua
Toukokuu 2014

Opinnäytetyön työosuutena toteutettiin responsiiviset WWW-sivut WordPress-sisällönhallintajärjestelmän päälle. Työn toimeksiantajana oli IT-alan yritys Decens Oy. Toimeksiantajan vanhat WWW-sivut sijaitsivat ulkopuolisen toimijan omassa sisällönhallinnassa. Uudet WWW-sivut toteutettiin Decensin omalla palvelimella sijaitsevaan sisällönhallintaan. Sisällönhallintajärjestelmää päätettiin käyttää, koska se helpottaa sivujen päivittämistä. Toimeksiannon tavoitteena olivat responsiiviset sivut, jotka mukautuvat sivuston käyttäjän laitteen mukaisiksi.

Opinnäytetyön raporttiosassa kerrotaan työn toteutuksen eri vaiheista, responsiivisuudesta, avoimen lähdekoodin WordPress-sisällönhallintajärjestelmästä ja WordPressin lapsiteeman käyttämisestä. Raportissa perehdytään responsiivisen suunnittelun tekniikoihin ja toteutukseen sekä siihen, kannattaako yrityksen tehdä responsiivisia sivuja. WWW-sivuilla on käytössä WordPressin responsiivisen *Twenty Thirteen* -teemaan perustuva lapsiteema, jonka toteutuksesta tässä raportissa kerrotaan.

Koska mobiilikäyttäjien määrä kasvaa jatkuvasti, on yrityksen tarkoituksenmukaista rakentaa sivuistaan responsiiviset, jotta sivujen käyttäminen olisi optimaalisempaa. Opinnäytetyön lopputuloksena on toimeksiantajan toiveiden, tarpeiden ja yrityksen yleisen graafisen ilmeen mukaiset sekä käyttäjien eri laitteisiin mukautuvat WWW-sivut.

ABSTRACT

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

VEHMAA, TAIJA:
Implementing a Website with Responsive Layout
Child Theme Using WordPress' Theme Twenty Thirteen

Bachelor's thesis 47 pages, appendices 3 pages
May 2014

The purpose of this thesis was to create a responsive website for a company called Decens Oy. Decens Oy is a Finnish company whose is information technology. Their old website was located in a third-party content management system. The new website was implemented on their own server and content management system. The company wanted to have a responsive website which is located in WordPress content management system.

The theory part of this thesis is about implementation of a responsive website with the WordPress content management system using a child theme. The child theme is based on WordPress' Twenty Thirteen theme which is responsive. WordPress is an Open Source system. Using responsive design is profitable for companies because website visitors use different devices: desktop computers, laptop computers, tablets and smartphones. Responsive design makes it possible for companies to create and maintain only one website suitable for all devices. Maintaining only one website reduces costs and saves time.

The final result of this thesis was a responsive website consistent with the company's graphical outlook.

Key words: responsive web design, wordpress, twenty thirteen, child theme.

SISÄLLYS

1	JOHDANTO	5
2	RESPONSIIVINEN SUUNNITTELU	7
2.1	Responsiivisuuden määrittely	7
3	RESPONSIIVISTEN WWW-SIVUJEN KEHITYSNÄKYMÄT.....	12
3.1	Responsiiviset sivut verrattuna perinteisiin ja erillisiin mobiilisivuihin.....	12
3.2	Käyttäjä- ja laitemäärät	13
3.3	Käyttäjät	13
3.4	Hakukoneet.....	14
3.5	Toimivuus eri selaimissa.....	15
4	RESPONSIIVISTEN WWW-SIVUJEN TOTEUTUS	17
4.1	Joustavien elementtien toteutus.....	17
4.1.1	Joustavat fontit.....	17
4.1.2	Joustava ruudukko	18
4.1.3	Joustavat kuvat	20
4.2	Responsiivisuuden toteutus	20
4.2.1	Mediakyselyt	21
4.2.2	Resoluutiopysäytyspisteet	25
4.2.3	Mobile first.....	26
4.3	WWW-fontit.....	27
5	WORDPRESS-SISÄLLÖNHALLINTAJÄRJESTELMÄ	29
5.1	WordPress sisällönhallintajärjestelmänä.....	29
5.2	Vanhempi- ja lapsiteema.....	30
5.2.1	Lapsiteeman luominen.....	30
6	RESPONSIIVISTEN WWW-SIVUJEN TOTEUTUS WORDPRESSILLÄ	33
6.1	Ulkoasulliset ja toiminnalliset vaatimukset.....	33
6.2	Twenty Thirteen -teeman lapsiteeman muokkaus.....	35
6.2.1	Palvelut-sivu	41
6.2.2	Asiakkaat-sivu	41
7	POHDINTA	45
	LÄHTEET	46
	LIITTEET	48
	Liite 1. Palvelut-sivun luettelon HTML-koodi	48
	Liite 2. Palvelut-sivun luettelon CSS-koodi	49

1 JOHDANTO

Nykyään yhä useampi käyttäjä liikkuu WWW-sivuilla tabletilla tai älypuhelimella. Tämän vuoksi yritysten on muokattava sivuistaan myös pienimmille laitteille sopivat. Yritys voi rakentaa perinteisten WWW-sivujen lisäksi myös tabletille ja älypuhelimelle sopivat mobiilisivut, mutta kaksien sivujen toteutus ja päivittäminen vaativat enemmän työtä. Responsiivinen suunnittelu on tekniikka, jolla yhdet WWW-sivut voidaan toteuttaa sopimaan kaiken kokoisille laitteille.

Tässä opinnäytetyöraportissa kerrotaan responsiivisten sivujen toiminnasta, toteutuksesta ja tekniikoista. Näistä esimerkkinä ovat mediakyselyt (*media query*). Lisäksi tarkastellaan sitä, miksi yrityksen kannattaa panostaa responsiivisiin WWW-sivuihin, ja mikälainen tarve niille on nykyisin. Raportissa kerrotaan myös yrityksen responsiivisten WWW-sivujen tekemisestä WordPress-sisällönhallintajärjestelmään.

Tästä opinnäytetyöstä hyötyvät toimeksiantajan lisäksi ne henkilöt, jotka miettivät, kannattaako responsiivia sivuja tehdä, tai miten ne tehdään. Raportissa kerrotaan myös WordPress-sisällönhallintajärjestelmän lapsiteeman (*child theme*) käytöstä.

Responsiivisuudesta kerrottaessa lähteinä käytetään responsiivisen suunnittelun keksijän, Ethan Marcotten, kirjaa ja responsiivista suunnittelua hyödyntävän Ben Frainin uudempaa HTML5:stä ja CSS3:sta kertovaa WWW-suunnittelun kirjaa. Opinnäytetyössä hyödynnetään W3C:n standardeja. W3C on kansainvälinen WWW-standardien kehittäjä ja ylläpitäjä. Lisäksi käytetään maailman suosituimman hakukoneen Googlen suosituksia responsiivisuudelle.

Opinnäytetyössä esitellään Accenturen, Adoben, Gartnerin, Deloitteen, StatCounterin ja TSN:n tutkimustuloksia. Kaikki tutkimustulokset ovat alle vuoden vanhoja, lukuun ottamatta Accenturen (2012) ja Adoben (2011) tutkimuksia. Muun muassa Taloussanomien kirjoittaa TSN:n, Deloitteen, Gartnerin ja Accenturen tutkimuksista. Accenturen maailmanlaajuinen tutkimus on toteutettu vuoden 2012 alussa. Siihen on vastannut 17 225 henkilöä 13 eri maasta. Suomesta kyselyyn vastasi 1085 henkilöä. TSN:n globaaliin tutkimukseen on haastateltu 38 000 mobiilikäyttäjää 43 eri maassa. Deloitteen kansainvälinen tutkimukseen on haastateltu 40 000 mobiilikäyttäjää 20 eri maassa. Suomesta

tutkimukseen vastasi 1000 kuluttajaa. StatCounter monitoroi käyttäjien internetin selailutietoja. Muun muassa Frain käyttää kirjassaan StatCounterin globaaleja ja reaaliaikaisia tilastotietoja. Gartner on maailman johtavia informaatioteknologian tutkimus- ja tiedotusyrityksiä.

Opinnäytetyön työosuutena toteutettiin IT-alan yrityksen responsiiviset, käytettävyydeltään kohderyhmälle sopivat WWW-sivut. WWW-sivut rakennettiin WordPress-sisällönhallintajärjestelmän päälle. Alustaksi valittiin sisällönhallintajärjestelmä, koska se helpottaa WWW-sivujen päivittämistä. Toimeksiantaja valitsi sisällönhallintajärjestelmäksi WordPressin. Sivusto on tehty WordPress-sisällönhallintajärjestelmän responsiivisen teeman *Twenty Thirteen* pohjalta käyttäen lapsiteemaa.

Opinnäytetyön työosuuden toimeksiantaja on Decens Oy. Decens Oy on 53-henkinen yritys, joka toimii kuudella paikkakunnalla: Tampereella, Jyväskylässä, Kuopiossa, Oulussa, Vaasassa sekä Vantaalla. Sivuston käyttäjiä ovat erikokoiset yritykset, jotka tarvitsevat IT-ammattilaisen palveluita. Palveluvalikoimaan kuuluu laaja kirjo palveluita laitetoimituksista koko IT-ympäristön ulkoistamiseen.

Decensin vanhat sivut sijaitsivat ulkopuolisen toimijan omassa sisällönhallintajärjestelmässä ja palvelimella. Uudet sivut tehtiin Decensin omalle palvelimelle. WWW-sivut päivitettiin uuden graafisen ilmeen mukaisiksi. Lisäksi WWW-sivuille tehtiin muutamia rakennemuutoksia ja sisältöä päivitettiin.

WWW-sivut toteutettiin yhdessä yrityksen toimitusjohtaja Jarno Lahnalahden kanssa. Hän vastasi sivuston sisällöstä ja oli mukana ideoinnissa. Valmistuneen sivuston ylläpito ja päivittäminen on minun vastuullani.

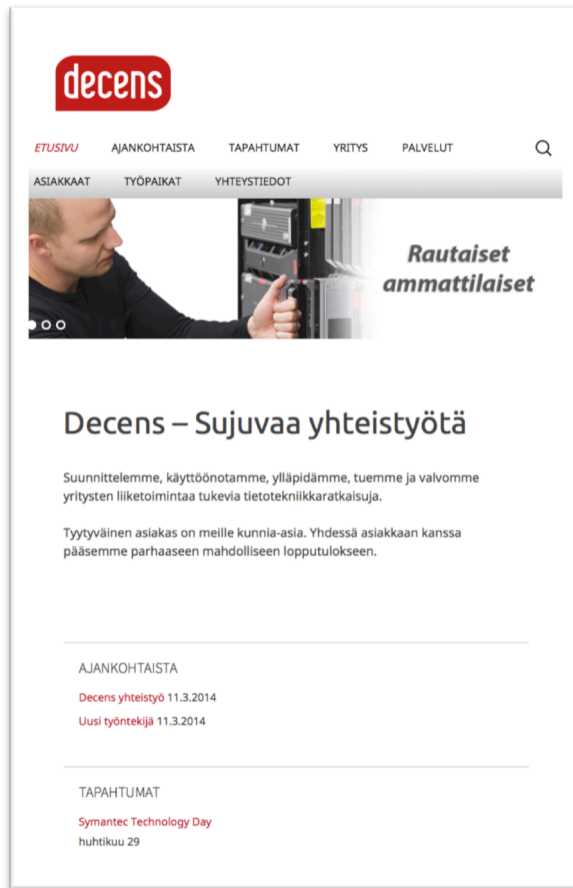
2 RESPONSIIVINEN SUUNNITTELU

2.1 Responsiivisuuden määrittely

Termin responsiivinen suunnittelu (*responsive web design*) keksi Ethan Marcotte. Hänen mukaansa responsiivisuus syntyy joustavasta ruudukkosommitelmasta (*flexible grid layout*), joustavista kuvista ja mediasta sekä mediakyselyistä (*media query*). Responsiivisen suunnittelun tarkoituksena on muuttaa pelkästään sivun ulkoasua laitteen ikkunan koolle sopivaksi. (Frain 2012, 10. Marcotte 2010.) Esimerkkinä tästä on kuva 1, jossa on sivu täydessä leveydessään, kuva 2, jossa sivu sopii tabletille ja kuva 3, jossa sivu sopii puhelimen näytölle. Responsiivisesta tai sen tyyppisestä suunnittelusta voidaan käyttää myös termejä *fluid design*, *cross-device design*, *flexible design*, *liquid design*, *elastic layout*, *rubber layout* ja *adaptive layout* (Frain 2012, 10).

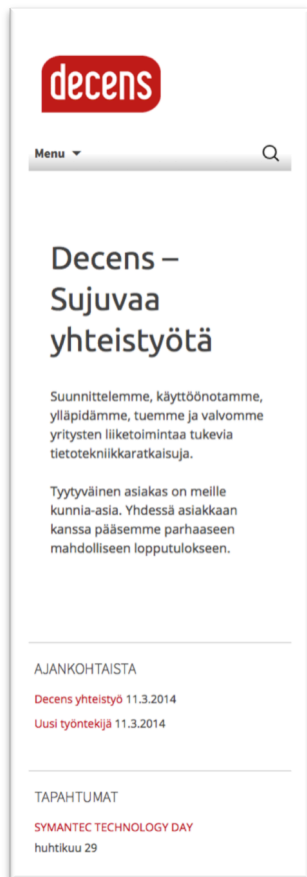


KUVA 1. Responsiivinen WWW-sivu, jonka leveys on 1300 pikseliä



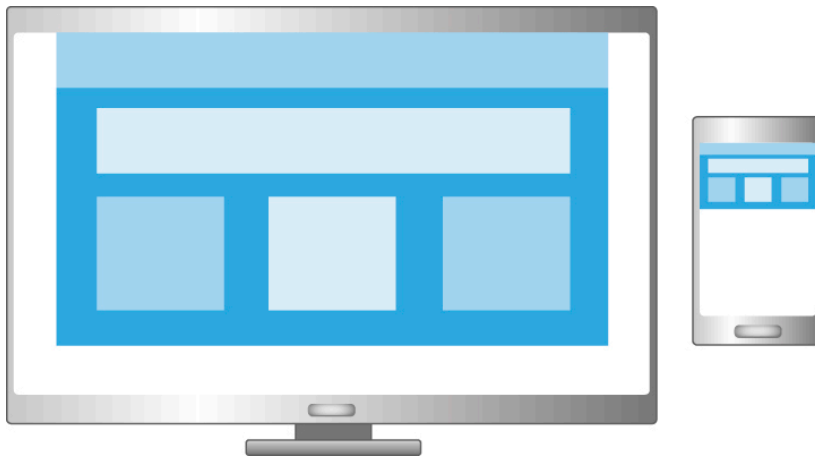
KUVA 2. Responsiivinen sivu, jonka leveys on 700 pikseliä

Responsiivinen sivusto toimii siten, että palvelin, jossa sivusto sijaitsee, lähettää muun muassa HTML- ja CSS-koodit laitteelle, jossa CSS muuttaa sivuston näkymän laitteelle sopivaksi. (Google Developers 2014.) CSS:n responsiivisuuteen vaikuttavat tekijät ovat joustava ruudukkosommittelu ja mediakyselyissä olevat, näytön koon mukaan määräytyvät asetukset. Laitteen tunnistukseen CSS:ssä käytetään mediakyselyjä.

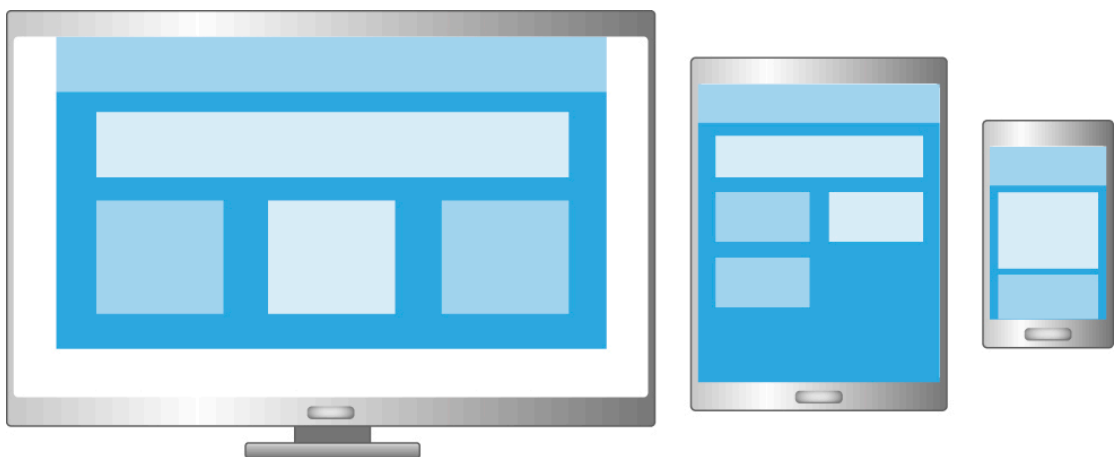


KUVA 3. Responsiivinen WWW-sivu, jonka leveys on 300 pikseliä

Ero responsiivisella ja perinteisellä sivustolla on siinä, että perinteinen WWW-sivu pienentyy tabletin ja puhelimen näytöille suhteellisessa koossa. Tästä on esimerkki kuvassa 4. Sivuston korkeus ja leveys ovat suhteessa samat kuin täysikokoisena. Tällöin pienemmän näytön käyttäjä joutuu lähentämään sivua nähdäkseen esim. siinä olevat tekstit. Sivua joutuu selaamaan myös sivuttaissuunnassa, koska esim. kaikki teksti ei näy kokonaan näytöllä. Responsiivinen WWW-sivu mukautuu käyttäjän laitteen selaimen mukaan. Teksti pysyy laitteesta huolimatta luettavassa koossa ilman, että käyttäjän täytyy lähentää sivua tai liikkua sivulla sivuttaissuunnassa (kuva 5).



KUVA 4. Kuinka perinteinen WWW-sivu käyttäytyy laitteen ikkunan mukaan



KUVA 5. Kuinka responsiivinen WWW-sivu käyttäytyy laitteen ikkunan mukaan

Responsiivisessa suunnittelussa ei huomioida pelkästään selaimen tai laitteen kokoa, vaan sivuston on lisäksi mukauduttava ikkunaan laitteen ollessa vaak- tai pystysuunnassa. Tämän lisäksi suunnittelussa pitää ottaa huomioon tapa, jolla näyttöä käytetään. Älypuhelinta käytetään sormen kosketuksella, kun taas tietokoneen näyttöä ohjataan yleensä hiirellä. On esimerkiksi huomioitava, että sormella kosketettavien painikkeiden, syöttökenttien ja linkkien kosketusalue on riittävän suuri. (Marcotte 2010, 108.)

Laitteita voidaan käyttää myös eri paikoissa ja tilanteissa. On eri asia katsella nettisivua rauhassa pöydän ääressä kuin kiireessä samalla kun kävelee. Tällöin käyttäjän on saatava etsimänsä tieto mahdollisimman helposti. Esimerkiksi yhteystiedot on hyvä laittaa selkeästi saataville. Sivun ei myöskään voi sisältää tiedostokooltaan liian suuria elementtejä, sillä puhelimen internetyhteys on yleensä pöytäkoneita hitaampi. Rauhassa on

helpompi syventyä kaikkeen siihen, mitä sivulta löytyy, kun taas kiireinen käyttäjä on usein vain tarkistaa jonkin asian nopeasti. (Marcotte 2010, 108.)

Kaikki eivät kuitenkaan ole juoksemassa kännykkä kädessä jonnekin samalla internetiä selaillen. Tutkimuksissa on todettu, että monet käyttävät kännykkää nettisivujen selaamiseen langattoman verkon välityksellä myös kotonaan. (Marcotte 2010, 108.) WWW-sivuja ei siis voi tehdä pelkästään sillä olettamuksella, että kännykän käyttäjät vain pikaisesti vilkaisevat niitä.

3 RESPONSIIVISTEN WWW-SIVUJEN KEHITYSNÄKYMÄT

3.1 Responsiiviset sivut verrattuna perinteisiin ja erillisiin mobiilisivuihin

WWW-sivut voidaan rakentaa siten, että ne toteutetaan tietokoneelle (niin sanotut perinteiset sivut) ja mobiililaitteelle erikseen. Tällöin molemmat sivut ovat oman URL:nsä takana. Responsiivisten sivujen rakentaminen, päivittäminen ja hallinta on helpompaa ja nopeampaa kuin sivujen, joissa mobiilisivu on erikseen. Vähäisemmän työmäärän vuoksi sivujen toteuttaminen on halvempaa. (Bing Blogs – Webmaster Blog 2012.)

Hakukoneiden ja käyttäjien on helpompi hallita responsiivisen sivuston linkkejä kuin erillisten mobiili- ja perinteisten WWW-sivujen linkkejä. Hakukoneet hakevat tietoa WWW-palvelimilta URL:n perusteella. Siirtyminen responsiivisiin sivuihin vähentää URL:ien määrää. Tämä vähentää hakukoneiden kaistaliikennettä WWW-palvelimille. Kaistankäyttö vähenee olennaisesti erityisesti suurten sivustojen kohdalla, kun on vähemmän läpikäytäviä URL:ja. (Bing Blogs – Webmaster Blog 2012.)

Erillinen mobiilisivusto kannattaa tehdä silloin, kun halutaan erityinen, toimintoiltaan kännykälle tarkoitettu sivusto. Tällöin on mietittävä, millainen ulkoasu ja millaiset ominaisuudet sivutolla tarvitaan. Tällainen sivusto voi olla sisällöltään ja ulkoasultaan erilainen kuin perinteinen, tietokoneelle toteutettu sivu. Sivustolla voidaan hyödyntää interaktiivisuutta sijainnin, yhteysnopeuden ja laitteen teknisten ominaisuuksien mukaan. Tällainen sivusto voi esimerkiksi laitteen GPS:n avulla paikantaa käyttäjän ja osoittaa lähimmän ostopaikan. Muuten erillistä mobiilisivustoa ei kannata tehdä, sillä sen tekeminen tulee kalliimmaksi perinteisen WWW-sivuston rinnalle, kuin pelkkä responsiivinen sivusto. Jos mobiilisivustolle ei ole erityistä tarvetta, toimii responsiivinen sivusto jokaisella laitteella. (Frain 2012, 10.)

Kun yrityksessä mietitään sitä, tehdäänkö yritykselle perinteiset vai responsiiviset sivut, ja sitä, tarvitaanko erillisiä mobiilisivuja, on yrityksen selvítettävä, käytetäänkö sen sivuja älypuhelimella tai tabletilla, ja tuottaako erillisten mobiilisivujen luominen sille lisäarvoa.

3.2 Käyttäjä- ja laitemäärät

Arki muuttuu yhä mobiilikeskisemmäksi ja älypuhelinta ja tablettia käytetään enemmän WWW-sivujen selaamiseen (TSN Gallup 2013). Tietoliikenteen määrän ennustetaan kasvavan nopeasti (Deloitte 2013). Jotta yritys vastaisi paremmin WWW-sivuilla kävijöiden tarpeisiin, on sivujen käytettävyyttä parannettava soveltumaan pienten päätelaitteiden tarpeisiin. Käyttökokemusta voi parantaa responsiivisilla sivuilla, jotka ovat myös kustannustehokkaat.

Suomessa kannettavien laitteiden käyttö on vähäisempää kuin Euroopassa, mutta älypuhelinten omistaminen ja tablettien käytön lisääntyminen on nopeinta muihin maihin verrattaessa (Deloitte 2013).

Deloitteen (2013) tutkimuksen mukaan Suomessa älypuhelinten omistus on kasvanut 15 % vuodesta 2012. Suomalaiset omistavat tai käyttävät keskimäärin neljää tai viittä kannettavaa laitetta. Vuonna 2013 toteutetun kyselyn mukaan 51 % suomalaisista omistaa älypuhelimien ja 20 % tabletin. Tämä on 42 % enemmän kuin vuonna 2012.

Älypuhelinten myynnin kasvu kiihtyy. Gartnerin (2014) mukaan vuonna 2013 älypuhelimia myytiin 968 miljoonaa, joka on 53,6 % kaikkien myytyjen puhelinten määrästä. Älypuhelimien myynti kasvoi vuonna 2013 42,3 % vuodesta 2012. Vuonna 2013 melkein kolmannes (31 %) älypuhelimista oli Samsungin valmistamia. Android-käyttöjärjestelmä on 78,4 %:ssa älypuhelimista, iOS 15,6 %:ssa ja Microsoft 3,2 %:ssa.

3.3 Käyttäjät

Vuoden 2012 alussa toteutetun Accenturen maailmanlaajuisen tutkimuksen mukaan yli kaksi kolmasosaa (69 %) käyttää internetiä kannettavalla laitteella päivittäin. Älypuhelinta käyttää 61 %, minikannettavaa 37 % ja tablettia 22 %. Ranskan ja Saksan lisäksi Suomessa internetin käyttö on vähäistä suhteessa muihin länsimaihin sekä globaalisti. Suomessa langatonta internetiä käyttää 48 % naisista ja 68 % miehistä. Kannettavista laitteista on nopeasti tulossa ensisijainen internetin käyttöalusta maailmanlaajuisesti. (Accenture 2013.)

Peräti 61 % 16–60-vuotiaista suomalaisista omistaa älypuhelimien ja 16 % tabletin, kertoo tuore *Mobile Life 2013* -tutkimus. Älypuhelimien lisääntymisen myötä myös mobiiliin internetin käyttö kasvaa. Internetin selaaminen nouseekin suurimmaksi syyksi hankkia älypuhelin ajasta ja paikasta riippumattomuuden vuoksi. WWW-sivujen selaaminen on suurimpia syitä myös tabletin hankintaan. (TSN Gallup 2013.)

Miös verkkokauppojen käyttö älypuhelimilla ja tableteilla on lisääntynyt. Vuoden 2011 *Adobe Digital Index* -tutkimuksen mukaan tabletin käyttäjät käyttivät ostokerralla 54 % enemmän ostoksiinsa kuin älypuhelimilla ja 21 % enemmän kuin pöytäkoneen tai kannettavan käyttäjät. (Adobe 2014, 4.) Adobe suosittelee verkkokaupan sovittamista mobiililaitteille, erityisesti tableteille (Adobe 2014, 2).

3.4 Hakukoneet

Miksi hakukoneoptimointi on niin tärkeää? Vuonna 2013 suoritettiin keskimäärin 5 922 000 000 Google-hakua yhdessä päivässä. Hakujen määrä päivässä on kasvanut vuosittain vuodesta 2008 lähtien noin miljardilla. (Statistic Brain 2014.)

StatCounterin mukaan Googlen hakukoneen käyttö on ollut tasaista läpi vuoden helmikuun 2013 ja helmikuun 2014 välillä. Maailmanlaajuisesti kaikkien laitteiden hakukonekäytöstä 90 % tehdään Googlella. Seuraavaksi yleisin hakukone on Bing, jonka osuus hauista on 3,5 %. Kaikilla laitteilla Suomessa tehtävistä hauista Googlen osuus on 96,5 % ja kännykällä tai tabletilla 99 %. (Stat Counter 2014.)

Googlen hakukoneella ruudulle ilmestyvät ne sivut, joita Google pitää kaikista relevantteimpina. Google määrittelee sivujen relevanttiuden yli 200:n eri asian mukaan. Näistä yhtenä esimerkkinä on *PageRank*. Jokainen linkki, joka tulee sivulle, nostaa sivun *PageRank*-sijoitusta. Sivuston Google-hakusijoitusta voi nostaa parantamalla Googlen mahdollisuuksia hakea (*crawl*) ja indeksoida (*index*) sivustoa. Google-haku huomioi sivuilla näkyvät sanat ja niiden paikat sekä sisältötagit ja attribuutit (esim. *title*-tagit ja *ALT*-attribuutit). Responsiivisilla sivuilla voidaan helpottaa Googlen hakukoneen työtä. (Google 2014.)

Google suosittelee optimointiin responsiivisia sivuja, koska sen on helpompi indeksoida yhden URL:n päässä oleva tieto. Googlebotin tarvitsee läpikäydä sivut vain kertaalleen, kun koko sivusto on olemassa vain yhtenä, responsiivisena kappaleena. Tehokkaammasta läpikäynnistä johtuen Google pystyy indeksoimaan enemmän sivuston sisällöstä ja pitämään tulokset tuoreina. (Google Developers 2014.)

Itsessään HTML5 parantaa hakukoneiden hakua, sillä muun muassa navigaatio on merkitty `<nav>`-tagilla, jolloin haku tunnistaa navigaation helposti (Frain 2012, 24).

Bingin hakukone suosittelee responsiivista sivustoa, koska sille tulee enemmän linkityksiä verrattaessa erillisiin mobiili- ja perinteisiin WWW-sivuihin. Myös hakukoneiden tuottama kaistaliikenne WWW-palvelimille vähentyy. Kaistankäyttö vähenee olennaisesti erityisesti suurten sivustojen kohdalla, sillä läpikäytäviä URL:ja on vähemmän. (Bing Blogs – Webmaster Blog 2012.)

3.5 Toimivuus eri selaimissa

Frainin mukaan responsiivisen suunnittelun menetelmät hyödyttävät kasvavaa internetkäyttäjäkuntaa. Sen menetelmät soveltuvat selkeään, nopeaan ja hallittavimpaan koodikantaan. Käyttäjäkokemus on kuitenkin erilainen eri laitteiden selaimilla. Responsiivinen sivusto ei toimi samalla tavalla kaikilla selaimilla, esimerkiksi Internet Explorer versiolla 8 ja vanhemmilla. Tulee siis miettiä, kuinka moni kävijöistä käyttää Internet Explorerin vanhempia selainversioita. Responsiivinen sivusto ei kaikissa tapauksissa ole paras mahdollinen mobiililaitteille. Jos mobiilikäyttäjälle halutaan tarjota täysin erilainen sisältö ja käyttökokemus, kannattaa silloin tehdä mobiili- ja tavalliset WWW-sivut erikseen. (Frain 2012, 32.)

Internet Explorer -selaimen käytön ennustetaan vähenevän maailmanlaajuisesti. StatCounterin mukaan maailmanlaajuisesti Internet Explorer -selaimen 8.0-version käyttö on laskenut vuoden 2013 helmikuun 9,5 %:sta vuoden 2014 helmikuun 5 %:iin. Version 9.0 käyttö on laskenut samassa ajassa 14,5 %:sta 3 %:iin. 10.0-version käyttö on noussut yhdestä prosentista kolmeen. Siinä ajassa, missä Chrome-selaimen käyttö on noussut vuodessa 32 %:sta 36,5 %:iin, on Internet Explorer -selaimen käyttö laskenut 26 %:sta 17,5 %:iin. Firefox, Safari ja Opera ovat selaimena harvinaisempia kuin Internet Explo-

rer, mutta niiden käyttö on pysynyt melko tasaisena koko vuoden ajan. Kolme neljästä pöytäkoneen selaimesta on jokin muu kuin Internet Explorer. Kun tarkastellaan kaikkia laitteita, ei neljä viidestä käytä Internet Exploreria. Internet Explorer -selaimen 8.0-versio on yleisin selainversio. (Stat Counter 2014.)

Suomessa Internet Explorer on harvinaisempi kuin maailmalla, sillä selaimen käyttö on laskenut 13,5 %:iin. Internet Explorer on myös Suomessa harvinaisempi kuin Firefox, jota käyttää yksi neljästä suomalaisesta. Yleisin selain on Chrome, jota käyttää 34,5 %. Suomessa Internet Explorer -selaimen versiot 10.0 ja 9.0 ovat yleisimmät Internet Explorer -versiot. (Stat Counter 2014.)

StatCounterin mukaan pöytäkoneen käyttäjämäärä on maailmanlaajuisesti laskenut vuodessa yli 10 % 71 %:iin. Kännykän käyttäjien määrä taas on noussut 10 %:lla 23 %:iin. Tablettien käyttäjissä on tapahtunut vajaan 1,5 % kasvu vajaan 6 %:iin. Suomessa on tapahtunut vastaavanlaista kehitystä, mutta muutokset eivät ole olleet yhtä suuria. (Stat Counter 2014.)

Vanhemmat selaimet, kuten IE 6, 7 ja 8, hyppäävät CSS3-koodissa niiden kohtien yli, joita ne eivät pysty prosessoimaan. Tämä mahdollistaa uuden CSS-koodin käytön ilman, että se vaikuttaa millään tavalla sivuston ulkoasuun. (Frain 2012, 25.)

4 RESPONSIIIVISTEN WWW-SIVUJEN TOTEUTUS

4.1 Joustavien elementtien toteutus

CSS:ään tehtävä joustava ruudukkosommittelu on responsiivisten sivujen perusta. Sivusto luodaan ensin joustavaksi, jotta se mukautuisi paremmin responsiiviseksi. Joustavuuteen käytetään suhteellisia arvoja, joiden yksiköt ovat joko em tai %. (Marcotte 2010, 79.) Esimerkiksi sivun palstojen leveys ja kuvien koko määritellään prosenttiluvuin. em:llä voidaan vaikuttaa fontin kokoon. Fontti, joka on 1 em:n koossa, on oletusarvoisesti 16 pikseliä. Jos fontin koko on 2 em, se on 32 px. Oletusarvo määräytyy selaimen oletusfonttikoon mukaan. 1 em:n koko voi muuttua, jos elementille on asetettu CSS:ssä jokin muu kiinteä fonttikoko. (W3Schools 2014a.) Jotta sivustosta saataisiin kunnolla responsiivinen, tulee sen fontit, kuvat ja muu media olla myös suhteellisessa koossa. Joustavan elementin koon laskemiseen käytetään kaavaa $kohdekoko / oletuskoko = tulos$. (Marcotte 2010, 79).

4.1.1 Joustavat fontit

HTML-dokumenttiin luodaan otsikkoteksti, jonka tagin sisällä on myös linkki (esimerkki alla). Fonteista saadaan joustavat, kun käytetään kaavaa $kohde (target) / konteksti (context) = tulos$. Kohteena on koko, jota halutaan käyttää ja konteksti on kohteen sisältävä elementti. Jotta fontista saadaan suhteellinen, on käytettävä pikseleiden sijasta em:iä. Selaimen fontti on oletuskooltaan 16 pikseliä, ja kun otsikkotekstistä halutaan 24-pikselinen, jaetaan kohdekoko (24) oletuskoolta (16). Tulos on 1,5. Eli 24 on puoli-toista kertaa oletusfonttikokoa suurempi. CSS:ssä fontin kokoon (*font-size*) asetetaan 1.5 em (koodiesimerkki 1). (Marcotte 2010, 20.)

```
<h1>Otsikko<a href="#">linkki</a></h1>
```

```
h1 {
    font-size: 1.5em;
}
```

KOODIESIMERKKI 1. Fontin koon asettaminen (Marcotte 2010, 20)

Kun halutaan muuttaa linkin kokoa, pitää ottaa huomioon, että linkki on h1-tagin sisällä eli sen konteksti on eri kuin otsikkotekstillä. Kun kohteesta halutaan 11-pikselinen, jaetaan se otsikkotekstin koolla 24. Kaava on siis $11/24 = 0,4583333333333333$. Marcotten mukaan lukua ei tule pyöristää, sillä selain laskee luvun pikselinä ja muuttaa fontista juuri sen kokoisen kuin olit halunnutkin (koodiesimerkki 2). (Marcotte 2010, 20.)

```
h1 a {
    font-size: 0.4583333333333333em;
}
```

KOODIESIMERKKI 2. Fontin koon asettaminen, kun sillä on konteksti (Marcotte 2010, 20)

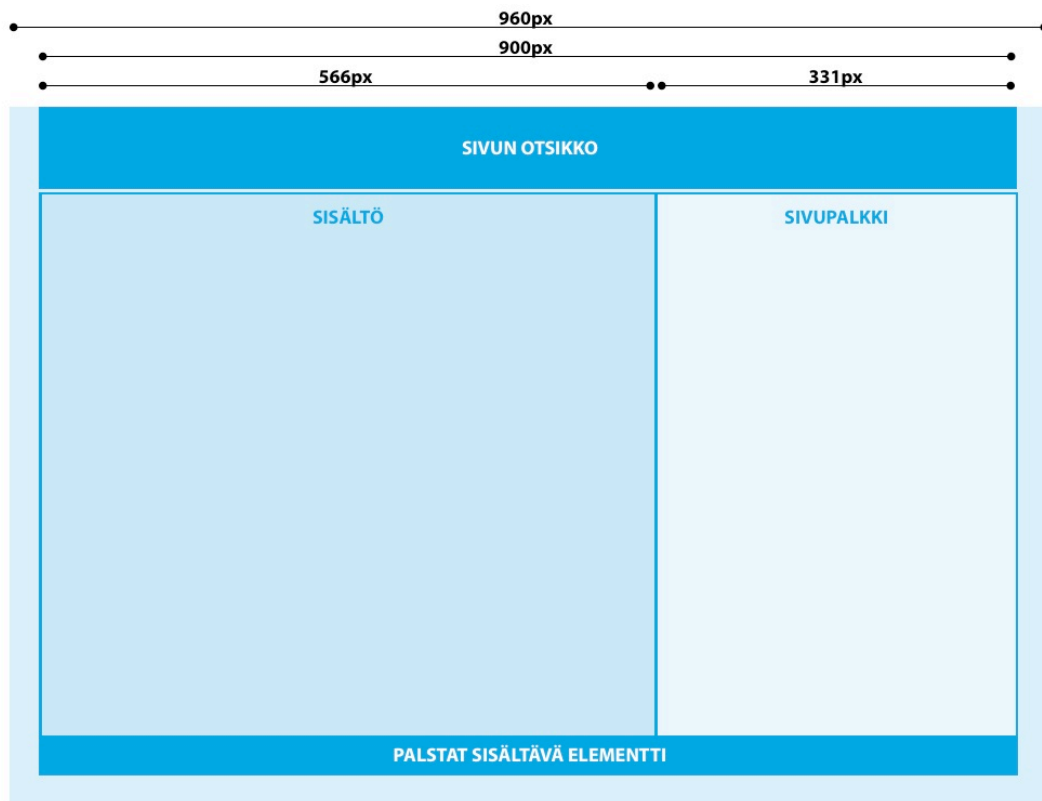
Fonttien koon voi myös asettaa vaihtumaan näytön koon mukaan, jolloin näytön koon tunnistukseen käytetään mediakyselyjä. Mediakyselyiden käytöstä kerrotaan luvussa 4.2.1 mediakyselyt. Kun fontin koko asetetaan vaihtumaan näytön koon mukaan, ei sitä tällöin aseteta joustavaksi. Silloin fontille riittää kiinteä koko tarvittaviin resoluutiopysäytyspisteisiin. Näytön koon pienentyessä, fontin kokoa kannattaa suurentaa, jotta teksti pysyy luettavana. Joustavia fontteja käytettäessä fontin koko muuttuu sivua skaalattaessa jatkuvasti, kun taas näytön koon mukaan muuttuva fontti vain niissä resoluutiopysäytyspisteissä, joihin koon muutos on asetettu.

4.1.2 Joustava ruudukko

Joustava ruudukko (*flexible grid*) tehdään samaan tapaan. Ensimmäinen on tiedettävä, minkä levyinen sivusta halutaan pikselinä. Esimerkiksi sivun leveydeksi halutaan 900 px ja se on suuremman 960 pikseliä leveän elementin sisällä (kuva 6). Suurempi elementti on tällöin konteksti. 900 jaetaan 960:lla jolloin tulos on 0,9375. Tulos on prosenttiluku 93,75, joka merkitään CSS:än *width*-kohtaan (koodiesimerkki 3). (Marcotte 2010, 31.)

```
sisaltavaelementti {
    width: 93,75%;
}
```

KOODIESIMERKKI 3. Elementin leveyden asettaminen (Marcotte 2010, 31)



KUVA 6. Joustava ruudukkosommitelma

Palstojen sisällön ja sivupalkin leveydet lasketaan jakamalla niiden leveydet ne sisältävän elementin leveydellä. $566 / 900 = 0,628888889$ ja $331 / 900 = 0,367777778$ (koodiesimerkki 4). (Marcotte 2010, 32.)

```
.sisalto {
    float: left;
    width: 62.8888889%;
}

.sivupalkki {
    float: right;
    width: 36.7777778%;
}
```

KOODIESIMERKKI 4. Palstojen leveyden asettaminen (Marcotte 2010, 32)

4.1.3 Joustavat kuvat

Kuvasta saa joustavan, kun sen asettaa HTML:ssä `<div>`-elementin sisään ja kuvalle asettaa CSS:ssä maksimileveydeksi 100 % (koodiesimerkki 5). Tällöin kuva on niin suurena sen sisältävän elementin sisällä kuin se voi. Kun sivua suurentaa, elementin koko ja sen sisällä olevan kuvan koko kasvaa. Maksimileveyden asettaminen toimii myös videoille ja interaktiiviselle medialle (*rich media*). (Marcotte 2010, 46.)

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

KOODIESIMERKKI 5. Kuvan asettaminen joustavaksi (Marcotte 2010, 46)

Tällä tavalla säädetyt kuvat eivät kuitenkaan toimi kaikilla selaimilla. Internet Explorer -selaimen versio 6 ja sitä aiemmat versiot eivät tue *max-width*-asetuksen käyttöä. Näitä versioita varten on asetuksen tilalla käytettävä JavaScript-koodia (esimerkiksi *respond.js*). Kuvalle voi myös asettaa pakotetun leveyden (*width: 100%*). Asetus toimii suurilla kuvilla, mutta resoluutioltaan pienempien kuvien laatu kärsii sivun suurentuessa. Jos tätä asetusta halutaan käyttää vain tietyissä kohdissa, valitaan tietyn tyyppiset kuvat tai videot (koodiesimerkki 6). Kun kuvaa skaalataan CSS:llä, saattaa se näyttää joillakin Windows-selaimilla hieman virheelliseltä. (Marcotte 2010, 49.)

```
img.full, object.full, .sisalto img, .sisalto object {
    width: 100%;
}
```

KOODIESIMERKKI 6. Kuvan pakotetun leveyden asettaminen (Marcotte 2010, 49)

4.2 Responsiivisuuden toteutus

Vaikka joustava ruudukkosommitelma tekee sivustosta joustavamman kuin perinteinen, kooltaan kiinteä WWW-sivusto on, ei se kuitenkaan jousta tarpeeksi. Kun joustavaa sivua skaalataan suuremmaksi tai pienemmäksi, sivu alkaa näyttää visuaalisesti huonolta. Esimerkiksi sivua pienennettäessä eivät elementit välttämättä mahdu näkymään

kunnolla, koska niille ei ole riittävästi tilaa. Mediakyselyillä jatketaan sivun parantelua. Mediakyselyt viimeistelevät sivuston sopimaan kaikille laitteille. Näin sivustosta tulee responsiivinen. (Marcotte 2010, 79.)

Mediakyselyillä voidaan rakentaa jokaiselle laitteelle erilainen mukautettu näkymä (kuvat 3, 4 ja 5). Erilaiset näkymät vaikuttavat suuresti sivun käytettävyyteen. Kun mediakyselyt luodaan, vapautetaan sivun ulkoasu osittain joustavan ruudukkosommitelman tarkoista määrittelyistä. (Marcotte 2010, 79.)

Mediakyselyt tehdään siis joustavan ruudukkosommitelman päälle. Jos mediakyselyt tehtäisiin mitoiltaan kiinteän sivun päälle, jouduttaisiin jokainen mediakysely koodaamaan alusta asti uudestaan. Joustava eli kooltaan suhteellinen sivu säilyttää mittasuhteensa resoluution muuttuessa. (Marcotte 2010, 101.)

4.2.1 Mediakyselyt

Mediakyselyt ovat vuoden 2012 puolivälistä asti W3C:n suosittelemia HTML4:n ja CSS2:n tukemia mediariippuvaisia (*media dependent*) tyyliohjeita (*style sheets*). Niitä käytetään CSS-tyylitiedostossa tunnistamaan median tyyppiä (*media type*) ja median toimintoja (*media features*). Median toimintojen tila voidaan tarkistaa yhdellä tai useammalla lausekkeella tai lauseke voidaan jättää kokonaan pois. (W3C 2012.)

Dokumentilla voi esimerkiksi olla kaksi tyylitiedostoa, toinen näytölle ja toinen tulostettavaksi. HTML-tiedoston ylätunnisteeseen (*header*) linkitetään tällöin kaksi CSS-tiedostoa, joissa on määritelty näytölle ja printtiin erilaiset tyylit. W3C:n (2012) mukaan median tunnistuksen avulla voidaan tunnistaa oikea media asettamalla median tyyppiä näyttö (*screen*) tai tuloste (*print*):

```
<link rel="stylesheet" type="text/css" media="screen"
href="sans-serif.css">
<link rel="stylesheet" type="text/css" media="print"
href="serif.css">
```

KOODIESIMERKKI 7. Kahden eri CSS-tyylin asettaminen (W3C 2012)

Mediakyselyitä voidaan käyttää kahdella tavalla: joko HTML-dokumentin `<head>`-osassa tai CSS-tiedostossa. HTML-dokumentin yläosassa käytetään CSS-linkitystä `<link>`-tagilla, josta esimerkki yllä. CSS:ssä kysely aloitetaan toiminnolla (*feature*) `@media`, jonka jälkeen laitetaan median tunnistava koodin osa. Koodiesimerkissä 8 tapa, jolla CSS-tiedossa määritellään, miten tyyliohje asetetaan koskemaan näyttö (*screen*) mediaa. (W3C 2012.)

```
@media screen {
  * { font-family: sans-serif; }
}
```

KOODIESIMERKKI 8. Näyttömedian käyttö CSS:ssä (W3C 2012)

Mediakysely on looginen lauseke, joka on joko tosi (*true*) tai epätosi (*false*). Mediakysely on tosi silloin, kun median tyyppi ja toiminto on sama mediakyselyssä ja käyttäjän laitteessa. (W3C 2012.)

HTML4:n tukemat mediatyypit ovat *all*, *aural*, *braille*, *handheld*, *print*, *projection*, *screen*, *tty* ja *tv*. CSS2 ja CSS3 tukevat näiden lisäksi myös mediatyyppejä *embossed* ja *speech*. *all*-mediatyyppi koskee kaikkia mediatyyppejä. Koska mediatyyppi *all* on mediakyselyssä oletus, ei sitä tarvitse erikseen määrittellä kyselyyn (koodiesimerkki 9). (W3C 2012.)

```
@media all and (min-width:500px) { ... }
```

tarkoittaa samaa kuin

```
@media (min-width:500px) { ... }.
```

KOODIESIMERKKI 9. Kaikkea mediaa koskevan mediatyyppin käyttö (W3C 2012)

Mediakyselyissä käytettävät median toiminnot ovat esimerkiksi *width*, *height* tai *color*. Kyselyllä, johon leveys (*width*) on asetettu, voidaan määrittää esimerkiksi näytön leveys. *color*-toiminnolla kysytään, onko esimerkiksi näyttö värillinen. Kaikki median toiminnot on lueteltu taulukossa 1. Kaikkiin median toimintoihin voi laittaa etuliitteen *min-* tai *max-*, paitsi median toimintoihin *scan* ja *grid*. Median toiminnot tulee laittaa

sulkuihin. Kun kyselyssä on useampi mediatyyppi, tulee niiden väliin pilkku (koodiesimerkki 10). (W3C 2012.)

```
@media screen and (min-width: 400px) and (max-width: 700px) {  
    ...  
}  
  
@media screen and (color), projection and (color) {  
    ...  
}
```

KOODIESIMERKKI 10. Useamman mediatyyppin määrittely (W3C 2012)

TAULUKKO 1. Median toimintojen toiminta (W3C 2012)

Median toiminto	Toiminta
<i>aspect-ratio</i>	Suhteellinen ikkunan koko. Merkitään esim. 16:9 olevan ikkunan kohdalla: <i>aspect-ratio: 16/9</i> . Toiminto on <i>width</i> ja <i>height</i> suhteellisena.
<i>color</i>	Kuinka monta bittiä on värikomponenttia kohden. Arvo on positiivinen luku. Esim. <i>min-color: 16</i> tarkoittaa vähintään 16-bittistä näyttöä. Värittömän näytön arvo on 0.
<i>color-index</i>	Kuinka monta kirjausta (<i>entry</i>) laitteen värihakutaulukossa (<i>color lookup table</i>) on. Arvo on positiivinen luku.
<i>device-aspect-ratio</i>	Sama kuin <i>aspect-ratio</i> , mutta koskee suhteellista renderöitävää pintaa. Toiminto on <i>device-width</i> ja <i>device-height</i> suhteellisena.
<i>device-height</i>	Laitteen rendaavan pinnan korkeus. Jatkuvalle medialle se on näytön korkeus ja sivuja käyttävälle medialle arkin korkeus. Arvo on positiivinen luku.
<i>device-width</i>	Laitteen rendaavan pinnan leveys. Jatkuvalle medialle se on näytön leveys ja sivuja käyttävälle medialle arkin leveys. Arvo on positiivinen luku.
<i>grid</i>	Onko laite ruudukko- vai bittikarttaperustainen. Arvo on joko, -0, 0 tai 1. Luku 1 tarkoittaa ruudukkoperustaista laitetta.
<i>height</i>	Ikkunan korkeus. Arvo on positiivinen numero. Yksikkö voi olla esim. cm, px tai em.
<i>monochrome</i>	Kuinka monta bittiä yksivärisen ruudun puskurissa (<i>frame buffer</i>) on pikselillä. Arvo on positiivinen luku. Monivärisellä laitteella, arvo on 0.
<i>resolution</i>	Näytön tai tulosteen resoluutio. Esim. <i>min-resolution: 300dpi</i> ilmaisee, että tyylitiedostoa voidaan käyttää laitteilla, joiden resoluutio on 300 dpi tai yli.
<i>scan</i>	Käytetään tunnistamaan onko tv lomittamaton eli progressiivinen vai lomitettu eli interpoloitu. Esim. <i>@media tv and (scan: progressive) { ... }</i>
<i>width</i>	Ikkunan leveys. Arvo on positiivinen numero. Yksikkö voi olla esim. cm, px tai em.

Kun halutaan yhdistää useampi toiminto, laitetaan väliin *and*-avainsana (*keyword*). Avainsanoja ovat myös *or*, *not* ja *only*. *or*-avainsanaa käytetään, kun halutaan esimerkiksi määritellä tyypiksi jokin leveys tai jokin toinen leveys. *not*-avainsanaa voidaan käyttää tekemään kyselystä päinvastainen. Kun halutaan esimerkiksi tyylitiedoston vaikuttavan vain johonkin tiettyyn laitteeseen, käytetään *only*-avainsanaa (koodiesimerkki 11). (W3C 2012.)

```
<link rel="stylesheet" media="only screen and (color)"
href="example.css" />
```

KOODIESIMERKKI 11. Only-avainsanan käyttö (W3C 2012)

CSS-tiedostoon voidaan myös ladata toinen CSS-tiedosto käyttämällä *@import*-toimintoa. Tämän yhteydessä voidaan käyttää median tunnistusta. Toisen CSS-tiedoston tuomiseen käytetään HTTP-pyyntöä, joka pidentää sivun latausaikaa. Esimerkki toisen CSS-tiedoston tuomisesta koodiesimerkissä 12. (W3C 2012.)

```
@import url("tyyli.css") screen and {max-width: 360px};
```

KOODIESIMERKKI 12. Toisen CSS-tiedoston lataaminen CSS-tiedostoon (W3C 2012)

Mediakyselyitä tukevat selaimet Firefox versiosta 4.0 ylöspäin, Chrome, Safari versiosta 3.2 ylöspäin, Opera Mobile versiosta 10.0 ylöspäin, Android versiosta 2.1 ylöspäin ja Internet Explorer versiosta 9.0 ylöspäin. JavaScript-koodia voidaan käyttää paikkamaan Internet Explorer -selaimen vanhemmilla versioilla (6, 7 ja 8) puuttuvaa mediatyylien tukea. (Frain 2012, 35.)

4.2.2 Resoluutiopysäytyspisteet

Koska perinteinen tai edes liukuvalla (*fluid, liquid*) suunnittelulla toteutettu sivusto ei taivu pienemmille näytöille, on sivuston ulkoasuun vaikuttaviin tekijöihin puututtava tiukemmin. Sivustosta tulee responsiivinen, kun siihen lisätään mediakyselyt. Mediakyselyitä käytetään tunnistamaan selaimen leveys, jonka jälkeen sivusto muokataan

laitteelle sopivaksi. Sivuston leveyden määrittelyyn suositellaan käytettävän yleisimpiä selainleveyksiä, jotta mahdollisimman monelle laitteelle saataisiin optimaalinen näky-
mä. Mediakyselyissä käytettäviä leveyksiä kutsutaan resoluutiopysäytyspisteiksi (*reso-
lution breakpoint*) (taulukko 2). Pysäytyspisteiden tai mediakyselyiden määrää ei ole
määritelty.

TAULUKKO 2. Esimerkkejä resoluutiopysäytyspisteiksi (Marcotte 2010, 114.)

320 pikseliä	Pieniruudullisille laitteille, kuten kännyköille, jotka ovat pysty- suunnassa.
480 pikseliä	Pieniruutuisille laitteille, kuten kännyköille, jotka ovat vaaka- suunnassa.
600 pikseliä	Pienemmille tableteille, joita pidetään pystysuunnassa.
768 pikseliä	10 tuuman tableteille, joita pidetään pystysuunnassa.
1024 pikseliä	Laitteille, kuten tableteille, joita pidetään vaakasuunnassa. Myös tietyille kannettaville, minikannettaville ja pöytäkoneen näytöille.
1200 pikseliä	Laajakuvanäytöille, joita ovat pääasiassa kannettavien ja pöytä- koneiden näytöt.

Responsiivisen sivuston rakentaminen voidaan aloittaa puhelimen näytölle tarkoitettusta
leveydestä esimerkiksi 320 pikselistä tai pöytäkoneen näytölle tarkoitettusta leveydestä
esimerkiksi 1200 pikselistä. Tavasta riippuen merkitään seuraava pysäytyspiste maksi-
mi- tai minimileveyttä käyttäen *max-width*- tai *min-width*-ominaisuudella (*property*).
Kun sivuston rakentaminen aloitetaan pöytäkoneen näytölle tarkoitettusta koosta, määri-
tellään seuraavaksi pienemmälle näytölle tarkoitettu koko käyttäen *max-width*-
ominaisuutta. (Marcotte 2010, 113.)

4.2.3 Mobile first

Responsiivisessa suunnittelussa lähtökohtana on aina resoluutioltaan tietyn levyinen
sivusto. Vaikka WWW-sivujen tekeminen aloitetaan yleensä pöytäkoneen näytölle
tarkoitettusta, eli suurimmasta koosta, Marcotte suosittelee tekemään responsiiviset sivut
ensin puhelimille (*mobile first*). (Marcotte 2010, 122.)

Kun ensisijaisesti pöytäkoneen näytölle tehty responsiivinen sivusto avataan selaimessa, joka ei tue mediakyselyjä tai niitä paikkaamaan asetettua JavaScript-koodia, ei se toimi halutulla tavalla. Kun sivusto toteutetaan ensisijaisesti puhelimelle, näyttää se hyvältä myös suurena. (Marcotte 2010, 122.)

Mobile first toteutetaan niin, että ensin toteutetaan pienin, puhelimelle leveydeltään sopiva sivu, jota lähdetään kasvattamaan mediakyselyillä (koodiesimerkki 13). Oletukseltaan mobiilikokoon tehty sivusto tukee paremmin selaimia kuin sivu, joka on oletukseltaan suuri. Marcotte pitää tätä tapaa kestävimpanä (Marcotte 2010, 122.):

```
/*pienet näytöt*/
@media screen and (min-width: 600px) { ... }
/*pöytäkoneen näytöt*/
@media screen and (min-width: 860px) { ... }
/*yli 900 px näytöt*/
@media screen and (min-width: 1200px) { ... }
```

KOODIESIMERKKI 13. Responsiivisten WWW-sivujen toteutus ensin puhelimille (Marcotte 2010)

4.3 WWW-fontit

WWW-sivujen toteuttamisessa on olennaista miettiä sitä, millä tavoin valitut fontit näkyvät sivuilla. Jotta WWW-sivulla käytetty fontti näkyisi oikein laitteella, täytyy laitteen muistissa olla kyseinen fontti. Fontti on hyvä valita niin kutsuttujen *web safe* -fonttien listalta, jossa olevien fonttien oletetaan löytyvän jokaiselta laitteelta. (font-face.com 2014.) Mobiililaitteissa ei kuitenkaan usein ole yhtä suurta fonttivalikoimaa. Niille ei välttämättä pysty määrittelemään mitään tiettyä fonttia, vaan käytettävissä saattavat olla vain CSS:ssä käytetyt generiset fonttinimet: *sans-serif*, *serif* ja *monospace*. Jos sivustolla halutaan käyttää muuta fonttia, kuin mitä *web safe* -fonttien listalla on tai mitä puhelimessa on, voidaan käyttää WWW-fonttia (*web font*). (Korpela 2012.) Kun sivustolle on määritelty jokin WWW-fontti, eikä käyttäjän laitteessa ole valittua WWW-fonttia, latautuu fontti laitteelle, kun WWW-sivu renderöidään (font-face.com 2014). Se toimii siis lähes jokaisella laitteella ja selaimella. WWW-fontti määritellään CSS:ään *@font-face*-säännöllä (*rule*) (koodiesimerkki 14) (W3Schools 2014b).

```
@font-face {
    font-family: myFirstFont;
    src: url('Sansation_Light.ttf'),
        url('Sansation_Light.eot'); /* IE9 */
}
```

KOODIESIMERKKI 14. WWW-fontin määrittely (W3Schools 2014b)

@font-face-sääntö toimii Internet Explorer 9 -, Firefox-, Opera-, Chrome- ja Safari-selaimissa. Kaikki selaimet, paitsi Internet Explorer, tukevat *.ttf*- tai *.oft*-muotoisia fontteja. Internet Explorer tukee vain *.eot*-muotoa. Internet Explorer -selaimen 8-versio ja sitä aikaisemmat versiot eivät tue *@font-face*-sääntöä. (W3Schools 2014b.)

Web-fontteja käytettäessä vaaditaan kaksi asiaa; fontin nimen määrittely *font-family*-ominaisuuteen sekä fontin lähde-URL. *@font-face*-sääntöön siis määritellään, minkä nimistä fonttia käytetään, ja mistä se löytyy. Web-fontteja käytetään antamalla nimi fontille kirjoittamalla *font-family*-ominaisuuden (*property*) *arvo* (*value*)-kohtaan (W3Schools 2014b.):

```
font-family: myFirstFont;
```

KOODIESIMERKKI 15. Fontin asettaminen

WWW-palvelimella sijaitseva fonttitiedosto linkitetään CSS-tiedostoon. Jos tiedosto sijaitsee paikallisella palvelimella *src: url('Sansation_Light.ttf')*, tai jos se sijaitsee muualla *src: url('http://www.w3schools.com/css/Sansation_Light.ttf')*. Selain lataa fontin sieltä, mihin URL osoittaa. Fonttia voi käyttää CSS-tiedostossa (koodiesimerkki 16) (W3Schools 2014b.)

```
p {font-family: myFirstFont;}
```

KOODIESIMERKKI 16. Fontin asettaminen

5 WORDPRESS-SISÄLLÖNHALLINTAJÄRJESTELMÄ

5.1 WordPress sisällönhallintajärjestelmänä

WordPress on ilmainen avoimen lähdekoodin (*open source*) sisällönhallintajärjestelmä (*CMS = content management system*). Sisällönhallintajärjestelmällä voidaan luoda, päivittää ja muokata WWW-sivuja. WordPress perustuu PHP-koodiin ja tallentaa tiedot MySQL-tietokantaan. Sen voi ladata ilmaiseksi internetistä ja asentaa WWW-palvelimelle. WordPress-sisällönhallintajärjestelmää, joka on alun perin luotu blogien ylläpitoon, voi käyttää blogien ja nettisivujen luontiin ja ylläpitoon. (WordPress 2014.)

Sivustoa voi muokata WordPressin hallintanäkymässä (*dashboard*), johon sivuston luoja tai ylläpitäjä kirjautuu sisään. Hallintanäkymässä on muun muassa mahdollista vaikuttaa sivuston asetuksiin ja käytössä olevaan teemaan (*theme*) sekä siellä voi luoda yksittäisiä sivuja ja sisältöä niille. Sivustolla käytössä oleva teema vaikuttaa siihen, millaisia toiminnallisuuksia, millainen rakenne ja ulkoasu sivustolla on. (WordPress 2014.)

Kun pelkät hallintaliittymän tarjoamat muokkausmahdollisuudet eivät riitä, voi sivustoon vaikuttavia tiedostoja muokata suoraan esimerkiksi FTP-yhteyden avulla. Teeman muokkaamisella sivustolla saadaan aikaan hyvinkin suuria muutoksia. Teemaa suositellaan muokattavaksi niin, että tehdään erillinen lapsiteema (*child theme*), joka kirjoittaa muokattavan teeman päälle. Lapsiteemasta kerrotaan lisää luvussa 5.2 Vanhempi- ja lapsiteema. Käyttäjä voi lisätä sivustolle toiminnallisuutta lataamalla erilaisia lisäosia (*plugin*) WordPressin tietokannasta, asentamalla ne WordPressiin ja aktivoimalla ne hallintanäkymästä. Lisäosia käytetään pääosin hallintanäkymästä, jossa ne voi asettaa, lisäosasta riippuen, muun muassa vimpaimeksi (*widget*). Vimpaimia voi asettaa sivustolla oleville vimpainalueille, joita esimerkiksi *Twenty Thirteen* -teemassa on sivupalkissa ja alatunnisteessa. Lisäosan ja teeman voi myös koodata itse. Vimpainalueiden lisäys on myös mahdollista. (WordPress 2014.)

5.2 Vanhempi- ja lapsiteema

Aloitin sivuston toteutuksen valitsemalla teeman, jota halusin käyttää. Teemaksi valitsin WordPressin oman teeman *Twenty Thirteen*. Valitsin tämän teeman viidestä syystä, koska

- 1) se on responsiivinen,
- 2) se on ilmainen,
- 3) internetistä löytyy paljon tietoa teeman muokkaamiseen,
- 4) teeman ulkoasuun ei tarvinnut tehdä suuria muutoksia, sillä se vastasi suurimmilta osin tekemääni ulkoasusuunnitelmaa ja
- 5) se on WordPressin oma, joten se päivittyy varmasti virheiden löytyessä.

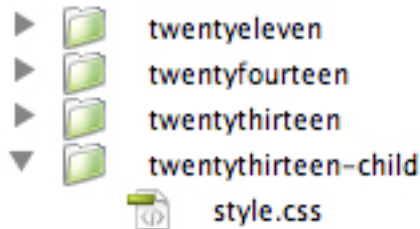
Ennen teeman muokkaamista loin sille lapsiteeman. Kun lapsiteema luodaan, aletaan muokattavaa teemaa kutsua vanhempiteemaksi (*parent theme*). Lapsiteemaa käytetään päällekirjoittamaan ja lisäämään vanhempiteemaan erilaisia toiminnallisuuksia ja muita säätöjä.

WordPress suosittelee lapsiteeman käyttöä, sillä se on turvallisin ja helpoin tapa muokata jo olemassa olevaa teemaa. Kun lisäyksiä ja muutoksia tehdään vain lapsiteemaan, pysyy vanhempiteema koskemattomana. Tällöin voi palata takaisin lähtöpisteeseen poistamalla jotain lapsiteemasta. Lisäksi, kun vanhempiteemaan tulee päivitys, eivät tehdyt muutokset katoa, koska ne on tehty lapsiteemaan. Lapsiteeman avulla on myös nopea tehdä muutoksia, sillä pohjana toimii vanhempiteema. (WordPress Codex 2014.) Sivuston lapsiteemaan tekemistäni muutoksista kerron luvussa 6.2 *Twenty Thirteen* -teeman lapsiteeman muokkaus.

5.2.1 Lapsiteeman luominen

Ensimmäiseksi luodaan lapsiteemalle oma kansio *wp-content/themes/-*hakemistoon. WordPress suosittelee, että teema nimettäisiin nimellä *vanhempiteema-child*. Koska työosuuden sivuston vanhempiteema on *Twenty Thirteen* -teema, on kansion nimi *twentythirteen-child*. (WordPress Codex 2014.)

Lapsiteeman kansioon tehdään CSS-tyylitiedosto (kuva 7), jonka nimeksi tulee *style.css*. Tyylitiedoston alkuun tulee kommentti, joka ilmaisee WordPressille, että kyseessä on lapsiteema (koodiesimerkki 17). Jotta teema toimisi, on tyylitiedostoon pakko täyttää kohdat *Theme Name* ja *Template*. Malli (*template*) on vanhempiteeman kansion nimi. (WordPress Codex 2014.)



KUVA 7. WordPressin hakemiston *wp-content/themes/* sisältö, jossa on vanhempiteema (*twentythirteen*) ja lapsiteema (*twentythirteen-child*) (WordPress Codex 2014)

```

/*
Theme Name: Twenty Fourteen Child
Theme URI: http://example.com/twenty-fourteen-child/
Description: Twenty Fourteen Child Theme
Author: John Doe
Author URI: http://example.com
Template: twentyfourteen
Version: 1.0.0
Tags: light, dark, two-columns, right-sidebar,
      responsive-layout, accessibility-ready
Text Domain: twenty-fourteen-child
*/

@import url("../twentyfourteen/style.css");

```

KOODIESIMERKKI 17. Esimerkki lapsiteeman *style.css* alun komenttiin tulevat tiedot (WordPress Codex 2014)

Tämän koodin jälkeen voi lisätä teeman ulkoasua muokkaavaa koodia. Tiedosto *style.css* päällekirjoittaa vanhempiteeman *style.css*-tiedoston. Lapsiteeman saa aktivoitua WordPressin hallintapaneelista (*dashboard*). Lapsiteema löytyy hallintapaneelissa

navigoimalla *Administration Panels > Appearance > Themes*. Klikkaamalla aktivoi (*Activate*) lapsiteeman kohdalla saadaan aktivoitua se. (WordPress Codex 2014.)

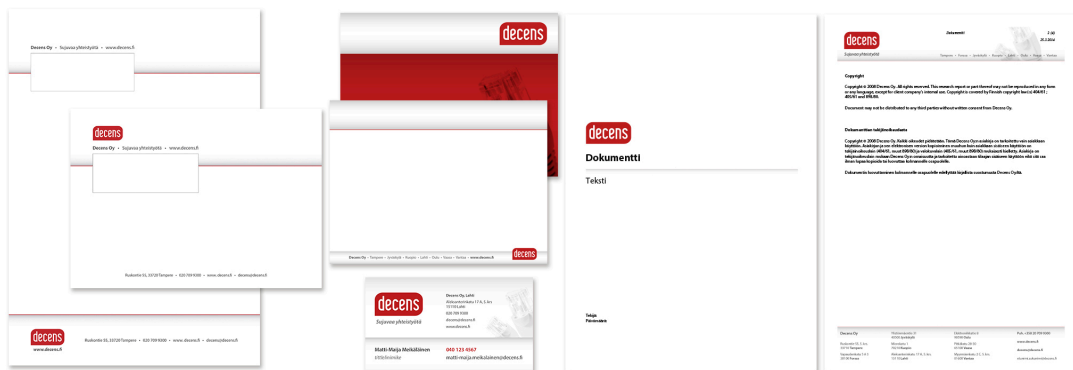
Kun vanhempiteeman mallitiedostoja (*template file*) halutaan muokata, ne kopioidaan lapsiteeman kansioon samalla nimellä. Esimerkiksi, jos halutaan muokata alatunnistetta, kopioidaan *footer.php*-tiedosto vanhempiteeman kansioista lapsiteeman kansioon, jossa muutokset tehdään. Myös omien mallitiedostojen luominen on mahdollista. (WordPress Codex 2014.)

Teeman *functions.php*-tiedosto sisältää sivuston toiminnallisuudet. Sivuston toiminnallisuuksia voi muokata lisäämällä *functions.php*-nimisen tiedoston lapsiteeman kansioon. Siihen kirjoitetaan vain koodi, joka lisätään vanhempiteeman omaan *functions.php*-tiedostoon. Tiedosto ei päällekirjoita vanhempiteeman vastaavaa tiedostoa. Toiminnallisuustiedostot toimivat WordPressissä niin, että ensin luetaan lapsiteeman *functions.php*-tiedosto, ja sen jälkeen vanhempiteeman saman niminen tiedosto. (WordPress Codex 2014.)

6 RESPONSIIVISTEN WWW-SIVUJEN TOTEUTUS WORDPRESSILLÄ

6.1 Ulkoasulliset ja toiminnalliset vaatimukset

Decens Oy:n vanhat sivut eivät olleet enää ajan tasalla, jonka vuoksi sivut haluttiin päivittää. Uudet sivut rakennettiin uuteen sisällönhallintajärjestelmään, joksi toimeksiantaja halusi WordPressin. Sivuihin haluttiin ulkonäöltään Decens Oy:n uuden graafisen ilmeen mukaiset. Opinnäytetyön alkaessa yrityksen materiaaleista uuden ilmeen mukaisia olivat Word-lomakepohja, käyntikortti, sähköpostin allekirjoitus, kirjekuoret sekä PowerPoint-esityspohja (kuva 8).



KUVA 8. Decens Oy:n yritysilme. Kuvassa on kirjekuoret C5- ja C4-koossa, PowerPoint-kalvopohja, käyntikortti ja Word-dokumenttipohja.

Yritysilmeessä olennaisia elementtejä ovat logon lisäksi valkoinen tausta, vaakamuotoinen vaaleanharmaasta valkoiseen kulkeva liukuväri, tummanharmaa teksti, punainen korostettu teksti ja slogan ”*Sujuvaa yhteistyötä*”. Kuvana yritysilmeessä toistuu verkko-kaapeli. Yrityksen virallisena fonttina on käytetty *Myriad Pro* -fonttia, joka on käytössä sivustolla olevissa kuvissa. Muutoin sivustolla on käytetty Googlen WWW-fontteja *Ubuntu* ja *Noto Sans*. *Ubuntu*-fonttia on käytetty otsikoissa ja *Noto Sans* -fonttia leipätekstissä ja listoissa.

Suunnittelemani ulkoasun mukaan uusilla WWW-sivuilla (kuva 10) väreinä on käytetty alla olevassa kuvassa 9 mainittuja värejä, sekä näiden punaisen, harmaan ja sinisen leikattuja sävyjä. Punaisia elementtejä sivuilla ovat logon lisäksi sivuilla olevat linkit, navigaation *hover*- ja *current*-linkit, ja painikkeiden pohjaväri on punainen.

decens

ETUSIVU AJANKOHTAISTA TAPAHTUMAT YRITYS PALVELUT ASIAKKAAT TYÖPAIKAT YHTEYSTIEDOT

Rautaiset ammattilaiset

Decens – Sujuvaa yhteistyötä

Suunnittelemme, käyttöönottamme, ylläpidämme, tuemme ja valvomme yritysten liiketoimintaa tukevia tietotekniikkaratkaisuja.

Tyytyväinen asiakas on meille kunnia-asia. Yhdessä asiakkaan kanssa pääsemme parhaaseen mahdolliseen lopputulokseen.

AJANKOHTAISTA

- Decens yhteistyö 11.3.2014
- Uusi työntekijä 11.3.2014

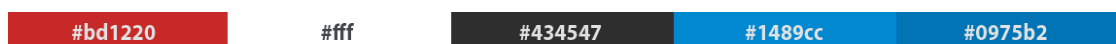
TAPAHTUMAT

- Symantec Technology Day huhtikuu 29
- Tapahtuma 2
toukokuu 25., klo 08:00 - 12:00
- Decens Expo 2014
kesäkuu 6., klo 08:00 - 23:00
- Näytä kaikki tapahtumat

Etätuki

<p>Yhteydenotto</p> <p>Nimi*</p> <input type="text"/> <p>Sähköposti*</p> <input type="text"/> <p>Aihe</p> <input type="text"/> <p>Viesti</p> <input type="text"/> <p>Lähetä</p>	<p>Decens Oy</p> <p>Ruskontie 55, 5. krs 33710 Tampere</p> <p>Puh. +358 20 709 9300 Info: decens@decens.fi</p>	<p>Jyväskylä Ylistönmäentie 31, 40500 Jyväskylä</p> <p>Kuopio Microkatu 1, 70210 Kuopio</p> <p>Oulu Elektronikkatie 8, 90590 Oulu</p> <p>Vaasa Pitkäkatu 28-30, 65100 Vaasa</p> <p>Vantaa Myrmyäentie 2 C, 01600 Vantaa</p>
---	---	---

KUVA 9. Decens Oy:n uudet WWW-sivut



KUVA 10. Sivustolla käytössä olevat päävärit

Responsiivisuuden lisäksi sivustolle haluttiin tapahtumakalenteri, ajankohtaiset asiat, yhteydenottolomake ja Asiakkaat-sivulle toiminnallisuus, joka näyttää lisättekstiä paini-

ketta klikatessa. Ajankohtaisia asioita varten asennettiin lisäosa, jonka avulla tiedote lisätään aika- ja päivämäärätietojen kanssa. Ajankohtaiset asiat näkyvät nyt pienempänä laatikkona etusivun vimpainpalkissa, sekä Ajankohtaista-sivulla. Etusivulla olevassa palkissa näkyy vain tiedotteen otsikko sekä julkaisuaika, ja tekstiä klikkaamalla pääsee lukemaan koko tiedotteen.

Asiakkaat-sivulle asetettiin asiakkaiden logot ja linkki, jota klikkaamalla voi avata kertomukset asiakascaseista. Linkkiä klikkaamalla avautuu lisää tekstiä, jossa kerrotaan palveluista, joita asiakas on käyttänyt sekä asiakkaan kommentit yhteistyöstä Decensin kanssa. Asiakkaat-sivusta kerrotaan lisää luvussa 6.2.2 Asiakkaat-sivu. Yhteydenotto-lomakkeen vaatimuksena oli, että siihen päästään käsiksi mm. etu-, Yhteystiedot-, Työpaikat- ja Palvelut-sivuilla. Yhteydenottolomake on nyt alatunnisteessa ja mainituilla sivuilla myös sivupalkissa.

6.2 Twenty Thirteen -teeman lapsiteeman muokkaus

Aloitin työn WordPress-sisällönhallintajärjestelmän versiolla 3.8, jonka jälkeen ohjelmasta tuli uudet versiot 3.8.1 ja 3.8.2. Päädyin käyttämään WordPressin omaa teemaa *Twenty Thirteen*, joka oli valmiiksi responsiivinen. Koska teema ei ollut täysin sellainen, kuin sivusta haluttiin, tein siihen useita muutoksia ja korjauksia, joista kerrotaan seuraavaksi.

Muokkasin jo luotua lapsiteemaa sijoittamalla logon paikalleen ylätunnisteeseen (*header*) ja tekemällä siitä responsiivisen (koosiesimerkki 18). Jotta sain logosta responsiivisen, säädin jokaiseen pysäytyspisteeseen arvoja, jotta logo kohdistuu oikeaan paikkaan ruudulla. Tein tämän suhteellisin arvoin eli prosenttilukuina.

```
.site-header {
    background-position: -5% 2% !important;
}

@media (max-width: 1069px) {
    .site-header {
        background-position: -1% 2% !important;
    }
}
```

```

}

@media (max-width: 999px) {
    .site-header {
        background-position: 2% 2% !important;
    }
}

@media (max-width: 767px) {
    .site-header {
        background-position: 2% 2% !important;
    }
}

@media (max-width: 643px) {
    .site-header {
        background-position: -2% 5% !important;
    }
}

@media (max-width: 359px) {
    .site-header {
        background-position: 3% 10% !important;
    }
}

```

KOODIESIMERKKI 18. Ylätunnisteen muokkaus

Navigaation alle halusin suuren kuvakarusellin, jossa pyörivät yrityksen mainokset. Erilaisia kuvakaruselleja vertailemalla löysin responsiivisen *Soliloquy*-kuvakarusellin. Tätä lisäosaa on ladattu n. 189 900 kertaa ja se on päivitetty viimeksi 9.4.2014 (WordPress Plugin Directory 2014a). Kuvakaruselli tuli vain etusivulle, jonka vuoksi lisäsin PHP-koodia *page.php*-tiedostoon. *is_front_page()*-koodin tehtävänä on tunnistaa, onko kyseessä etusivu (koodiesimerkki 19). Kun käyttäjä on etusivulla, karusellikoodi ajetaan.

```

<?php if ( is_front_page() ) :?>
    <?php if ( function_exists( 'soliloquy_slider' ) )

```

```

        soliloquy_slider( '33' ); ?>
<?php endif;?>

```

KOODIESIMERKKI 19. Etusivun tunnistus ja kuvakarusellin lisäys

Seuraavaksi jouduin muuttamaan sivupalkkia, sillä halusin sen menevän kuvakarusellin alapuolelle (koodiesimerkki 20). Ensimmäiseen mediakyselyyn, johon leveydeksi on määritelty vähintään 1599 pikseliä, on asetettu 100 % leveä sivupalkki leijumaan (*float*) oikealla. Seuraavassa pysäytyspisteessä sisältösivu on asetettu 60 %:iin ja leijumaan vasemmalla. Sivupalkki on leveydeltään 30 % sivun leveydestä ja se leijuu oikealla. Sivupalkin sisältämä vimpainalue (*widget-area*) leijuu oikealla ja on leveydeltään 100 % sisältävän elementin leveydestä.

```

@media (min-width: 1599px) {
    .site-main .widget-area {
        float: right;
        width: 100%;
    }
}
@media (min-width: 1069px) {
    .site-main .widget-area {
        float: right;
        width: 100%;
    }
    #primary.content-area {
        width: 60%;
        float: left;
    }
    .site-main .sidebar-container {
        position: relative;
        float: right;
        width: 30%;
        height: auto;
        right: 0;
    }
}

```

KOODIESIMERKKI 20. Sivupalkin muokkaus

Sivustolla oli automaattisesti käytössä tavutus. Tavutus ei toiminut oikein. Lisäksi siitä johtuen menivät tekstit sivun kaventuessa epäselviksi ja rumiksi. Tästä syystä poistin tavutuksen sisältösivulta ja sivupalkista (koodiesimerkki 21).

```
.entry-content, .widget {
    -webkit-hyphens: none;
    -moz-hyphens: none;
    -ms-hyphens: none;
    hyphens: none;
    word-wrap: normal;
}
```

KOODIESIMERKKI 21. Tavutuksen poistaminen

Kuvakaruselli on *#slideri*-nimisen `<div>`-elementin sisällä. Koska kuvakaruselli pienee liikaa ja on turha pienemmissä näyttökoissa, se piilotetaan. Piilottaminen tapahtuu mediakyselyssä koodin avulla, joka on koodiesimerkissä 22. Pysäytyspiste on 644 px.

```
#slideri {
    display: none !important;
}
```

KOODIESIMERKKI 22. Kuvakarusellin piilotus leveydeltään alle 644 pikselin selaimessa

Sivupalkissa on käytössä neljä erilaista vimpainta. Niissä näkyvät viimeksi julkaistut ajankohtaiset asiat eli WordPressin näkökulmasta viimeksi julkaistut blogimerkinnät, viimeisimmät tapahtumat, yhteydenottolomake ja kuva, joka toimii painikkeena. Vimpaimien näkyvyyttä eri sivuilla on rajoitettu. Ajankohtaiset asiat ja tapahtumat näkyvät Etu-, Ajankohtaista-, Yritys-, Asiakkaat- ja kaikilla Palvelut-sivuilla. Etusivulla näkyy myös kuvapainike Decensin etätukiyhteyteen. Työpaikat- ja Yhteystiedot-sivuilla näkyy yhteydenottolomake. Vimpainten näkyvyyttä rajoitetaan *Widget Logic* -lisäosalla. Hallinnassa jokaisen vimpaimen perässä on kenttä, johon voi laittaa koodia, jolla voi vaikuttaa vimpaimen näkyvyyteen (kuva 11). Kentässä käytetään Wordpressin omia ehtoja (*conditional tag*) tai PHP-koodia (WordPress Plugin Directory 2014b). *Widget Logic*

-lisäosaa on ladattu n. 756 800 kertaa ja se on päivitetty viimeksi 1.2.2014 (WordPress Plugin Directory 2014b).

Viimeisimmät artikkelit: Ajankohtaista

Otsikko:
Ajankohtaista

Näytettävien artikkelien lukumäärä: 3

Näytä artikkelin päivämäärä?

Widget logic:
!is_page('yhteydenotto') && !is_page('tyopaikat') &&
!is_page('tampere') &&

[Poista](#) | [Sulje](#) Tallenna

KUVA 11. Esimerkki *Widget Logic* -lisäosan käytöstä (WordPress Plugin Directory 2014b)

Sivustolle haluttiin myös tapahtumakalenteri. Tapahtumakalenteri on tehty *The Events Calendar* -lisäosaa käyttäen. Lisäosaa on ladattu n. 820 400 kertaa ja se on päivitetty viimeksi 1.4.2014 (WordPress Plugin Directory 2014c). Kalenteri näkyy myös Tapahtumat-sivulla. Kalenterin ulkoasua on muutettu sen oletuksesta (koodiesimerkki 23). Kalenteri asetui Tapahtumat-sivulla liian ylös, joten sille on asetettu ylös pehmustetta (*padding*) 45 px. Otsikkotekstille on asetettu 40 px alamarginaalia. Kalenterin taulukonäkymän rivien korkeutta on säädetty pienemmäksi 60 px:iin.

```
#tribe-events-content {
    padding-top: 45px;
}

.tribe-bar-is-disabled .tribe-events-page-title {
    margin-bottom: 40px;
}
```

```
#tribe-events-content .tribe-events-calendar td {
    height: 60px;
}
```

KOODIESIMERKKI 23. Tribe Events Calendar -lisäosan ulkoasun muokkaus

Yhteydenottolomake on toteutettu *Contact Form 7* -lisäosalla. Lisäosaa on ladattu n. 16 443 200 kertaa ja se on päivitetty viimeksi 26.2.2014 (WordPress Plugin Directory n.d.d). Yhteydenottolomake näkyy jokaisen sivun alatunnisteessa (*footer*) sekä Työpai-
kat- ja Yhteystiedot-sivuilla sivupalkissa. Yhteydenottolomake lähettää tiedot hallinnas-
sa asetettuun sähköpostiosoitteeseen.

Sivustolla on käytössä Googlen fontteja. *Google Fonts* on Googlen kehittämä interaktii-
vinen ilmaisten fonttien kirjasto. Fontteja voidaan käyttää esimerkiksi WWW-sivuilla ja
ohjelmissa. (Google Fonts 2014a.) Fonttia voi käyttää WWW-sivulla lisäämällä HTML-
dokumentin ylätunnisteeseen (*header*) koodiesimerkkissä 24 oleva koodi ja osoittamalla
siihen CSS-tiedostossa koodiesimerkkissä 25:ssä olevalla koodilla (Google Fonts 2014b).

```
<link href='http://fonts.googleapis.com/css?family=Fonttti'  
rel='stylesheet' type='text/css'>
```

KOODIESIMERKKI 24. Google-fontin käyttö HTML:tiedostossa (Google Fonts 2014b)

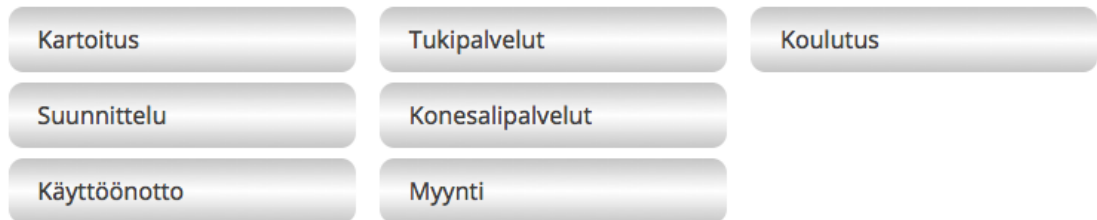
```
font-family: 'Fonttti';
```

KOODIESIMERKKI 25. Google-fonttiin osoittaminen CSS-tiedostossa (Google Fonts 2014b)

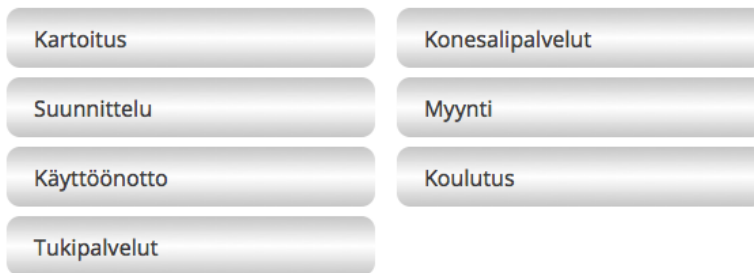
Olen asentanut Google-fonttien käyttöä varten lisäosan nimeltään *WP Google Fonts*.
Lisäosaa on ladattu n. 605 600 kertaa ja se on päivitetty viimeksi 2.11.2013 (WordPress
Plugin Directory 2014e). Lisäosa tekee tarvittavat koodin muutokset, ja fonttia pystyy
vaihtamaan hallinnassa valitsemalla luettelosta haluamansa fontin.

6.2.1 Palvelut-sivu

Palvelut sivulla on listattuna kaikki palvelut (kuva 12). Luettelo toimii responsiivisesti sivukoon mukaan. Leveällä näytöllä lista on kolmipalstainen, tablettikoossa kaksipalstainen (kuva 13) ja puhelimella yksipalstainen. Listan sisältävä elementin koko on maksimissaan 960 px. Teksti on asetettu vasempaan reunaan, alue on keskitetty ja välilyönti on 4 %. Sisemmässä elementissä (*.palvelulista*) palstan leveydeksi on asetettu 11,5 em. Listasta (*ul*) on poistettu kaikki ylimääräiset automaattiset listamuotoilut. Listan ensimmäisestä elementistä (*.palvelulista ul li:first-child*) on poistettu joidenkin selaimien lisäämä yläpuolinen tyhjä tila. Listan elementeille on asetettu pehmusteet ja marginaalit. Kulmat on pyöristetty ja väriksi on asetettu liukuväri. Linkin väri on tummanharmaa. Palvelut-sivun luettelossa käytetyt HTML- ja CSS-koodit ovat liitteissä 1 ja 2.



KUVA 12. Palvelut Palvelut-sivulta kolmipalstaisena



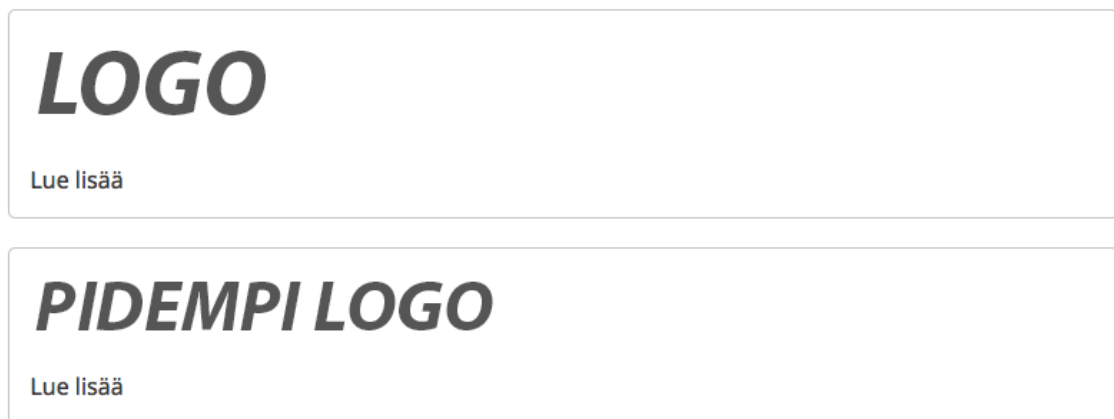
KUVA 13. Palvelut Palvelut-sivulta kaksipalstaisena

6.2.2 Asiakkaat-sivu

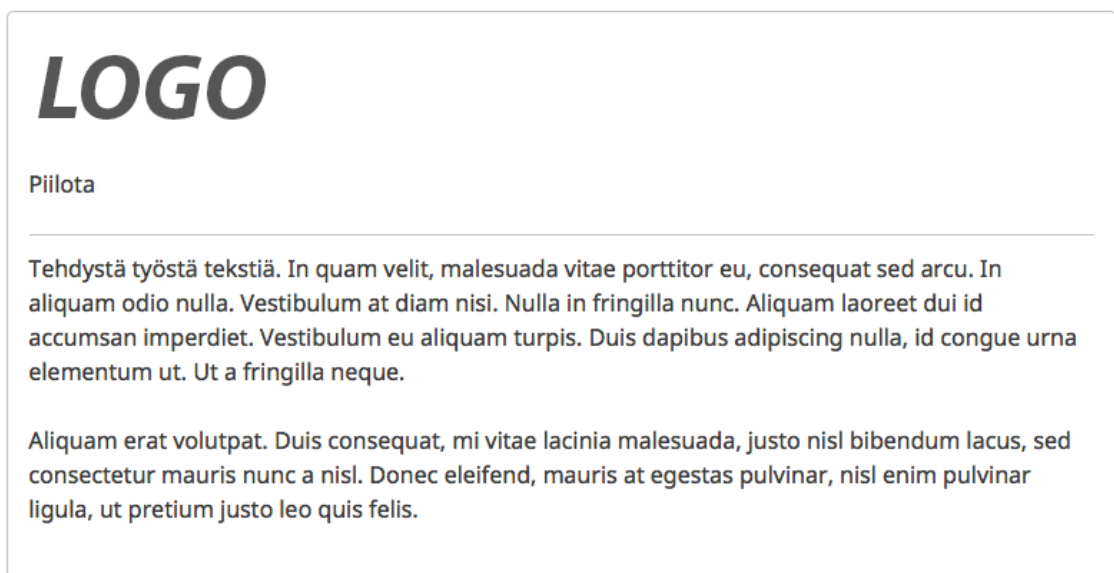
Asiakkaat-sivulla on luettelo Decensin asiakkaista. Jokainen asiakas esitellään omassa laatikossaan, jossa on asiakasyrityksen logo ja *Lue lisää* -linkki (kuva 14). Linkkiä klikkaamalla sen alle ilmestyy teksti, jossa kerrotaan kyseiselle asiakkaalle tehdystä

työstä (kuva 15). Myös linkin teksti vaihtuu *Piilota*-tekstiksi, jota klikkaamalla tekstin voi piilottaa.

Tekstin piilottamiseen ja näyttämiseen käytetään lisäosaa nimeltä *WP-ShowHide*, jota on ladattu n. 12 180 kertaa ja se on päivitetty viimeksi 10.4.2013 (WordPress Plugin Directory 2014f). Valitsin tämän lisäosan, koska se toimii hallinnan sivunluonnissa työn vaatimalla tavalla. Käyttö vaatii koodin tuntemusta ja asettamani *<div>*-elementit vaikeuttavat lisäosan käyttöä.



KUVA 14. Asiakkaat-sivulla olevan *WP-ShowHide*-lisäosan toiminta



KUVA 15. Asiakkaat-sivulla olevan *WP-ShowHide*-lisäosan toiminta

Kun uuden asiakkaan tiedot lisätään sivulle, on se laitettava *<div>*-elementin sisään. Tälle *<div>*-elementille on asetettava *class="laatikko"*-luokitus, joka asettaa elementille alamarginaalin, pehmusteen ja harmaan pyöreäkulmaisen reunaviivan. Kaikkien

asiakkaiden tietojen tulee olla `<div id="asiakkaat">` -elementin sisällä. `#asiakkaat` määrittelee 10 px marginaalin asiakaslaatikoiden ympärille. Esillä olevan piilotetun tekstin yllä on 10 px pehmuste ja 1 px harmaa viiva. Koodiesimerkissä 26 on HTML-koodi, jolla laatikot on määritelty ja koodiesimerkissä 27 CSS-koodi, jolla laatikoiden ulkoasu on määritelty.

```
<div id="asiakkaat">
  <div class="laatikko">
    <a href="http://polku.fi/kuva.gif">
      </a>

    [showhide type="1" more_text="Lue lisää"
    less_text="Piilota" hidden="yes"]

      <div class="piilotettu">
        Piilotettu teksti
      </div>

    [/showhide]

  </div>
</div>
```

KOODIESIMERKKI 26. HTML-koodi, joka määrittelee Asiakkaat-sivun laatikot

```
#asiakkaat {
    margin: 10px;
}
.laatikko {
    margin-bottom: 20px;
    padding: 15px;
    border: 1px solid #c6c6c6;
    border-radius: 5px;
}
```

```
.piilotettu {  
    padding-top: 10px;  
    border-top: 1px solid #c6c6c6;  
}
```

KOODIESIMERKKI 27. CSS-koodi, joka määrittelee Asiakkaat-sivun laatikoiden ulkoasun

7 POHDINTA

Eri lähteet osoittivat, että mobiililaitteiden, eli tablettien ja älypuhelinien, käyttö kasvaa jatkuvasti. Perinteiset WWW-sivut eivät toimi parhaimmalla mahdollisella tavalla näillä laitteilla. WWW-sivustojen käyttäjäystävällisyyttä voi parantaa responsiivisella suunnittelulla. Responsiivisessa suunnittelussa käytetään joustavaa sommittelua ja CSS:n mediakyselyjä. Mediakyselyt eivät kuitenkaan toimi vanhemmilla WWW-selaimilla. Tutkimuksista kuitenkin selviää, että vanhempien selainten käyttö vähenee jatkuvasti. Mediakyselyjä tukemattomille selaimille mediakyselyjä paikkaamaan voidaan käyttää JavaScript-koodia. Sivusto voidaan toteuttaa myös *mobile first* eli responsiivisen sivuston toteutuksen lähtökohtana on pienelle resoluutiolle sopiva sivusto, josta mediakyselyillä lisätään pysäytyspisteitä suuremmille resoluutioille. Tällainen sivusto näyttää myös mediakyselyjä ja JavaScriptiä tukemattomalla selaimella hyvältä, vaikka resoluutio olisi suuri. Responsiiviset sivut, varsinkin *mobile first* -tavalla toteutettuna, joustavat hyvin eri selaimiin.

Työn käytännön toteutusosan tavoitteena oli Decens Oy:n WWW-sivujen uudistus. Uusista sivuista tehtiin responsiiviset ja ne tehtiin yrityksen omalla palvelimella sijaitsevaan WordPress-sisällönhallintajärjestelmään. Jotta tähän tulokseen päästiin, täytyi minun opetella responsiivisuutta ja WordPressin käyttöä. Minulla ei ollut näistä aiempaa kokemusta. Opettelin myös tekemään lapsiteeman, jotta toimeksiantajan toiveet ja suunnittelemani ulkoasu saatiin toteutettua. Lisäksi tulen käyttämään oppimiani asioita työssäni jatkossa. Vastaan jatkossa näiden sivujen päivittämisestä. Jatkossa on lisäksi mahdollista, että toimeksiantajayritykseni voi laajentaa palveluvalikoimaansa siten, että asiakkaille tarjotaan mahdollisuutta ostaa yritykseltä WWW-sivutilaa, sisällönhallintaan toteutetut WWW-sivut ja niiden päivittäminen.

Uudet responsiiviset sivut vastaavat nyt paremmin käyttäjien laitteille. Niiden päivittäminen on helppoa, sillä ne on rakennettu sisällönhallintaan. Ne ovat yritysilmmeen mukaiset ja sisältävät toimeksiantajan määrittelemät toiminnallisuudet, joita ovat tapahtumakalenteri, ajankohtaiset asiat, yhteydenottolomake ja kuvakaruselli. Sivuilta puuttuu vielä tekstisisältö. Kun teksti saadaan paikoilleen, sivut ovat valmiit ja ne julkaistaan.

LÄHTEET

Accenture. 2013. Accenturen tutkimus: suomalaisten langattoman Internetin käyttö johtavia länsimaita vähäisempää. Julkaistu 26.7.2013.

Tutkimus on toteutettu tammi-maaliskuussa 2012. Luettu 10.3.2014.

<http://www.accenture.com/fi-en/company/newsroom-finland/Pages/accenture-study-finnish-mobile-internet-use-leading-western.aspx>

Adobe. 2014. The impact of tablet visitors on retail websites. PDF. Luettu 12.4.2014.

http://success.adobe.com/assets/en/downloads/whitepaper/18011_201205_Digital_Index_Tablet_Report.pdf

Bing Blogs – Webmaster Blog. 2012. Building Websites Optimized for All Platforms (desktop, mobile, etc.). Julkaistu 7.3.2012. Luettu 19.3.2014.

http://www.bing.com/blogs/site_blogs/b/webmaster/archive/2012/03/07/building-websites-optimized-for-all-platforms-desktop-mobile-etc.aspx

Deloitte. 2013. Suuret ikäluokat älypuhelinien ja tablettien suuri kohderyhmä – palveluita käytetään kuitenkin rajallisesti. Päivitetty 20.11.2013. Luettu 10.3.2014.

http://www.deloitte.com/view/fi_FI/fi/ajankohtaista/lehdistotiedotteet/d7d1df192c472410VgnVCM3000003456f70aRCRD.htm

font-face.com. 2014. What is @font-face {}. Luettu 12.4.2014.

<http://www.font-face.com/>

Frain B. 2012. Responsive Web Design with HTML5 and CSS3. Learn responsive design using HTML5 and CSS3 to adapt websites to any browser or screen size. Birmingham: Packt Publishing.

Gartner. 2014. Gartner Says Annual Smartphone Sales Surpassed Sales of Feature Phones for the First Time in 2013. Julkaistu 13.2.2014. Luettu 10.3.2014.

<http://www.gartner.com/newsroom/id/2665715>

Google. 2014. How Google Search Works. Luettu 19.3.2014.

<https://support.google.com/webmasters/answer/70897?hl=en>

Google Developers. 2014. Building Smartphone-Optimized Websites.

Päivitetty 11.2.2014. Luettu 12.4.2014.

<https://developers.google.com/webmasters/smartphone-sites/details>

Google Fonts. 2014a. About Google Fonts. Luettu 19.3.2014.

<http://www.google.com/fonts#AboutPlace:about>

Google Fonts. 2014b. Google Fonts. Luettu 19.3.2014.

<http://www.google.com/fonts/>

Korpela J. 2012. HTML5-kirja. Responsiivinen suunnittelu. Julkaistu 2.8.2012. Luettu 12.4.2014.

<http://html5kirja.fi/2012/08/02/responsiivinen-suunnittelu/>

Marcotte E. 2010. Responsive web design. New York: A Book Apart.

Stat Counter. n.d. Global Stats. Luettu 28.3.2014.
<http://gs.statcounter.com/>

Statistic Brain. 2014. Google Annual Search Statistics. Julkaistu 1.1.2014.
Luettu 28.3.2014.
<http://www.statisticbrain.com/google-searches/>

TSN Gallup. 2013. Mobile Life: Arki muuttuu yhä mobiilikeskemmäksi.
Julkaistu 8.5.2013. Luettu 10.3.2014.
<http://www.tns-gallup.fi/uutiset.php?aid=14935&k=14320>

W3C. 2012. Media Queries. Julkaistu 19.6.2012. Luettu 24.3.2014.
<http://www.w3.org/TR/css3-mediaqueries/>

W3Schools. 2014a. CSS Units. Luettu 2.4.2014.
http://www.w3schools.com/cssref/css_units.asp

W3Schools. 2014b. CSS3 @font-face Rule. Luettu 8.3.2014.
http://www.w3schools.com/cssref/css3_pr_font-face_rule.asp

WordPress. 2014. About WordPress. Luettu 22.3.2014.
<https://wordpress.org/about/>

WordPress Codex. 2014. Child Themes. Luettu 24.3.2014.
https://codex.wordpress.org/Child_Themes

WordPress Plugin Directory. 2014a. Responsive WordPress Slider – Soliloquy Lite.
Luettu 10.4.2014.
<http://wordpress.org/plugins/soliloquy-lite/>

WordPress Plugin Directory. 2014b. Widget Logic. Luettu 10.4.2014
<http://wordpress.org/plugins/widget-logic/>

WordPress Plugin Directory. 2014c. The Events Calendar. Luettu 10.4.2014
<http://wordpress.org/plugins/the-events-calendar/>

WordPress Plugin Directory. 2014d. Contact Form 7. Luettu 10.4.2014
<http://wordpress.org/plugins/contact-form-7/>

WordPress Plugin Directory. 2014e. WP Google Fonts. Luettu 10.4.2014
<http://wordpress.org/plugins/wp-google-fonts/>

WordPress Plugin Directory. 2014f. WP-ShowHide. Luettu 10.4.2014
<http://wordpress.org/plugins/wp-showhide/>

LIITTEET

Liite 1. Palvelut-sivun luettelon HTML-koodi

```
<div id="palvelut">
  <div class="palvelulista">
    <ul>
      <li><a href="/wp/kartoitus"
        class="palvelu">Kartoitus</a>
      </li>
      <li><a href="/wp/suunnittelu"
        class="palvelu">Suunnittelu</a>
      </li>
      <li><a href="/wp/kayttoonotto"
        class="palvelu">Käyttöönotto</a>
      </li>
      <li><a href="/wp/tukipalvelut"
        class="palvelu">Tukipalvelut</a>
      </li>
      <li><a href="/wp/konesalipalvelut"
        class="palvelu">Konesalipalvelut</a>
      </li>
      <li><a href="/wp/laite-ja-ohjelmistomyynti"
        class="palvelu">Myynti</a>
      </li>
      <li><a href="/wp/koulutus"
        class="palvelu">Koulutus</a>
      </li>
    </ul>
  </div>
</div>
```


Liite 2. Palvelut-sivun luettelon CSS-koodi

1 (2)

```
#palvelut {
    max-width: 960px;
    text-align: left;
    margin: 0px auto;
    padding: 4%;
}

.palvelulista {
    -moz-column-width: 11.5em; /* Firefox */
    -webkit-column-width: 11.5em; /* webkit, Safari, Chrome */
    column-width: 11.5em;
}

.palvelulista ul {
    margin: 0;
    padding: 0;
    list-style-type: none;
}

/* correct webkit/chrome uneven margin on the first column*/
.palvelulista ul li:first-child {
    margin-top: 0px;
}

.palvelulista ul li {
    margin-top: 7px;
    margin-bottom: 7px;
    padding: 10px;
    padding-left: 20px;
    border-radius: 10px;
    background: #c6c6c6; /* Old browsers */
    /* IE9 SVG, needs conditional override of
    'filter' to 'none' */
    background: url(data:image/svg+xml;base64,PD94bWwgdm
    Vyc2lvdj0iMS4wIiA/Pgo8c3ZnIHhtbG5zPSJodHRwOi8vd3d3Lncz
```

```

Lm9yZy8yMDAwL3N2ZyIgd2lkdGg9IjEwMCUiIGhlaWdodD0iMTAwJS
Igdmlld0JveD0iMCAwIDEgMSIgcHJlc2VydmVbc3Bly3RSYXRpbz0i
bm9uZSI+CiAgPGxpbnVhckdyYWRpZW50IGlkPSJncmFkLXVjZ2ctZ2
VuZXJhdGVkIiBncmFkaWVudFVuaXRzPSJlc2VyU3BhY2VPblVzZSIg
eDE9IjAlIiB5MT0iMCUiIHgyPSIwJSIgeTI9IjEwMCUiPgogICAgPH
N0b3Agb2Zmc2V0PSIwJSIgc3RvcC1jb2xvcj0iI2M2YzZjNiIgc3Rv
cC1vcGFjaXR5PSIxIi8+CiAgICA8c3RvcCBvZmZzZXQ9IjUwJSIgc3
RvcC1jb2xvcj0iI2ZmZmZmZiIgc3RvcC1vcGFjaXR5PSIxIi8+CiAg
ICA8c3RvcCBvZmZzZXQ9IjEwMCUiIHNoY29sb3I9IiInNmM2Yz
YiIHNoY29sb3BhY2l0eT0iMSIvPgogIDwvbGluZWFyR3JhZGlbnQ+
CiAgPHJlY3QgeD0iMCIgeT0iMCIgd2lkdGg9IjEiIGhlaWdodD0iMS
IgzmlsbD0idXJsKCNncmFkLXVjZ2ctZ2VuZXJhdGVkKSIGLz4KPC9z
dmc+);
background: -moz-linear-gradient(top, #c6c6c6 0%,
#ffffff 50%, #c6c6c6 100%); /* FF3.6+ */
background: -webkit-gradient(linear, left top, left
bottom, color-stop(0%,#c6c6c6), color-
stop(50%,#ffffff), color-stop(100%,#c6c6c6));
/* Chrome,Safari4+ */
background: -webkit-linear-gradient(top, #c6c6c6
0%,#ffffff 50%,#c6c6c6 100%);
/* Chrome10+,Safari5.1+ */
background: -o-linear-gradient(top, #c6c6c6
0%,#ffffff 50%,#c6c6c6 100%); /* Opera 11.10+ */
background: -ms-linear-gradient(top, #c6c6c6
0%,#ffffff 50%,#c6c6c6 100%); /* IE10+ */
background: linear-gradient(to bottom, #c6c6c6
0%,#ffffff 50%,#c6c6c6 100%); /* W3C */
filter: progid:DXImageTransform.Microsoft.gradient(
startColorstr='#c6c6c6', endCo
lorstr='#c6c6c6',GradientType=0 ); /* IE6-8 */
}

.palvelulista ul li a {
    color: #3f3e3e;
}

```